

Emergence, Complexity and Computation ECC

Ivan Zelinka
Massimo Brescia
Dalya Baron *Editors*

Intelligent Astrophysics

 Springer

Emergence, Complexity and Computation

Volume 39

Series Editors

Ivan Zelinka, Technical University of Ostrava, Ostrava, Czech Republic

Andrew Adamatzky, University of the West of England, Bristol, UK

Guanrong Chen, City University of Hong Kong, Hong Kong, China

Editorial Board

Ajith Abraham, MirLabs, USA

Ana Lucia, Universidade Federal do Rio Grande do Sul, Porto Alegre, Rio Grande do Sul, Brazil

Juan C. Burguillo, University of Vigo, Spain

Sergej Čelikovský, Academy of Sciences of the Czech Republic, Czech Republic

Mohammed Chadli, University of Jules Verne, France

Emilio Corchado, University of Salamanca, Spain

Donald Davendra, Technical University of Ostrava, Czech Republic

Andrew Ilachinski, Center for Naval Analyses, USA

Jouni Lampinen, University of Vaasa, Finland


Martin Middendorf, University of Leipzig, Germany

Edward Ott, University of Maryland, USA

Linqiang Pan, Huazhong University of Science and Technology, Wuhan, China

Gheorghe Păun, Romanian Academy, Bucharest, Romania

Hendrik Richter, HTWK Leipzig University of Applied Sciences, Germany

Juan A. Rodriguez-Aguilar , IIIA-CSIC, Spain

Otto Rössler, Institute of Physical and Theoretical Chemistry, Tübingen, Germany

Vaclav Snasel, Technical University of Ostrava, Czech Republic

Ivo Vondrák, Technical University of Ostrava, Czech Republic

Hector Zenil, Karolinska Institute, Sweden

The Emergence, Complexity and Computation (ECC) series publishes new developments, advancements and selected topics in the fields of complexity, computation and emergence. The series focuses on all aspects of reality-based computation approaches from an interdisciplinary point of view especially from applied sciences, biology, physics, or chemistry. It presents new ideas and interdisciplinary insight on the mutual intersection of subareas of computation, complexity and emergence and its impact and limits to any computing based on physical limits (thermodynamic and quantum limits, Bremermann's limit, Seth Lloyd limits...) as well as algorithmic limits (Gödel's proof and its impact on calculation, algorithmic complexity, the Chaitin's Omega number and Kolmogorov complexity, non-traditional calculations like Turing machine process and its consequences,...) and limitations arising in artificial intelligence. The topics are (but not limited to) membrane computing, DNA computing, immune computing, quantum computing, swarm computing, analogic computing, chaos computing and computing on the edge of chaos, computational aspects of dynamics of complex systems (systems with self-organization, multiagent systems, cellular automata, artificial life,...), emergence of complex systems and its computational aspects, and agent based computation. The main aim of this series is to discuss the above mentioned topics from an interdisciplinary point of view and present new ideas coming from mutual intersection of classical as well as modern methods of computation. Within the scope of the series are monographs, lecture notes, selected contributions from specialized conferences and workshops, special contribution from international experts.

Indexed by zbMATH.

More information about this series at <http://www.springer.com/series/10624>

Ivan Zelinka · Massimo Brescia ·
Dalya Baron
Editors

Intelligent Astrophysics

 Springer

Editors

Ivan Zelinka
Faculty of Electrical Engineering
and Computer Science
VŠB-TU Ostrava
Ostrava, Czech Republic

Massimo Brescia
INAF, Astronomical Observatory
of Capodimonte
Napoli, Italy

Dalya Baron
School of Physics and Astronomy
Tel Aviv University
Tel Aviv, Israel

ISSN 2194-7287

ISSN 2194-7295 (electronic)

Emergence, Complexity and Computation

ISBN 978-3-030-65866-3

ISBN 978-3-030-65867-0 (eBook)

<https://doi.org/10.1007/978-3-030-65867-0>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

During the development of humankind and the knowledge of our world, a large amount of information was produced. This information of a scientific as well as a technical nature then again contributed to the further development of our civilization. The amount of data that man has produced during scientific development increases exponentially with time. In the last five years, humankind has produced more information and data than in all its previous existential development. With increasing data, there is a need to have methods that can effectively process them and present the data to humans. Processing here means cleaning from noise, searching for useful information, visualization and the like. In the last hundred years, with the development of computers and information science, efficient algorithms have begun to emerge, which today belong to the so-called artificial intelligence and which are able to perform just such tasks. The first has been used classical algorithms for filtering, compression, dimension reduction, and many other tasks. Later, another algorithms enrich this class, which today belong to machine learning, which is basically part of artificial intelligence, and with the help of these methods and algorithms, we can process very large data in real time, which is important for science as such. The same goes for technology and society. Areas, where these methods are most needed, include physics and astrophysics. In physics, let us mention the CERN accelerator in Geneva, Switzerland, where experiments produce an enormous amount of data in fractions of a second. All this data must be processed precisely. Another area is in astrophysics, which, thanks to high robotics and automation, has become an area that is literally flooding us with by data. Today, it is a common fact and the fact that robotic telescopes spew up petabytes of data in one night. If we realize that at the time of Johannes Kepler's discovery, about 400 Kb of data was enough to discover the famous Kepler laws, it is clear that many discoveries can be hidden in these petabytes, which can literally flow between our fingers. Therefore, especially in astrophysics, all-important methods in the field of machine learning are very effectively applied, most notably neural networks, various filtering algorithms and recently it has been shown that evolutionary algorithms are also a very capable tool for processing such data and their possible modeling.

At first glance, the title “Intelligent Astrophysics” may sound strange, because the “intelligent” term could be wrongly misrepresented, implying as the wanting to consider “unintelligent” all previous research in Astrophysics. However, this is not the case. Intelligence is the ability to adapt to the surrounding environment and conditions. In this sense, the rise of Big Data paradigm and the advent of multi-messenger astrophysics pose Astronomy & Astrophysics (A&A) in a new perspective, as becoming de facto a Big Data Science, thus requiring more efficient solutions. Traditional research in the A&A field is now objectively considered unable to allow scientists to fully exploit all information inherited within observed data in a reasonable time. In particular, astronomical data are represented in a multi-D parameter space, where hidden correlations, anomalies, and peculiarities are very complex to find and to visualize. Since two decades, Astroinformatics, or “intelligent Astrophysics”, as being the virtuous synergy between A&A and Data Science, gained always increasing popularity and interest, enclosing as main features the multi-disciplinary approach (statistics, machine learning, data mining informatics, image analysis, visualization, and astrophysics as well), parameter space exploration and optimization, anomaly detection as well as serendipity. It pushed academic and educational institutions to open and activate dedicated positions and courses and its bibliometric parameters are exponentially increasing.

The present book discusses the application of these methods to astrophysical data from different perspectives. In this publication, the reader will encounter interesting chapters that discuss data processing and pulsars, the complexity and information content of our universe, the use of tessellation in astronomy, characterization and classification of astronomical phenomena, identification of extragalactic objects, classification of pulsars and many other interesting chapters. The authors of these chapters are experts in their field and have been carefully selected to create this book so that we can present to the community a representative publication that shows a unique fusion of artificial intelligence and astrophysics. Of course, this book is not a complete cookbook covering the whole topic in the broadest sense, but rather an inspiring book. The aim of which is to motivate and inspire readers to their own experiments in the intersection of these two disciplines. We hope that this book will be exciting and beneficial for readers and that this book will enrich the issue.

Ostrava, Czech Republic
Napoli, Italy
Tel Aviv, Israel
September 2020

Editors
Ivan Zelinka
Massimo Brescia
Dalya Baron

Contents

Artificial Intelligence in Astrophysics	1
Ivan Zelinka, Thanh Cong Truong, Diep Quoc Bao, Lumir Kojecky, and Eslam Amer	
The Complexity and Information Content of Simulated Universes	29
Franco Vazza	
The Voronoi Tessellation Method in Astronomy	57
Iryna Vavilova, Andrii Elyiv, Daria Dobrycheva, and Olga Melnyk	
Statistical Characterization and Classification of Astronomical Transients with Machine Learning in the era of the Vera C. Rubin Observatory	81
Marco Vicedomini, Massimo Brescia, Stefano Cavuoti, Giuseppe Riccio, and Giuseppe Longo	
Application of Machine and Deep Learning Methods to the Analysis of IACTs Data	115
Alessandro Bruno, Antonio Pagliaro, and Valentina La Parola	
Intelligent Photometric Identification of Extragalactic Objects from AllWISE×Pan-STARRS DR1 Data	137
Vladislav Khramtsov, Volodymyr Akhmetov, Peter Fedorov, Sergii Khlamov, Artem Dmytrenko, and Anna Velichko	
Ensemble Classifiers for Pulsar Detection	153
Jakub Holewik and Gerald Schaefer	
Periodic Astrometric Signal Recovery Through Convolutional Autoencoders	167
Michele Delli Veneri, Louis Desdoigts, Morgan A. Schmitz, Alberto Krone-Martins, Emille E. O. Ishida, Peter Tuthill, Rafael S. de Souza, Richard Scalzo, Massimo Brescia, Giuseppe Longo, and Antonio Picariello	

Comparison of Outlier Detection Methods on Astronomical Image Data	197
Lars Doorenbos, Stefano Cavuoti, Massimo Brescia, Antonio D’Isanto, and Giuseppe Longo	
Anomaly Detection in Astrophysics: A Comparison Between Unsupervised Deep and Machine Learning on KiDS Data	225
Maurizio D’Addona, Giuseppe Riccio, Stefano Cavuoti, Crescenzo Tortora, and Massimo Brescia	
Rejection Criteria Based on Outliers in the KiDS Photometric Redshifts and PDF Distributions Derived by Machine Learning	245
Valeria Amaro, Stefano Cavuoti, Massimo Brescia, Giuseppe Riccio, Crescenzo Tortora, Maurizio D’Addona, Michele Delli Veneri, Nicola R. Napolitano, Mario Radovich, and Giuseppe Longo	
Large Astronomical Time Series Pre-processing for Classification Using Artificial Neural Networks	265
David Andrešič, Petr Šaloun, and Bronislava Pečíková	
Frontiers	295

Artificial Intelligence in Astrophysics



Ivan Zelinka, Thanh Cong Truong, Diep Quoc Bao, Lumir Kojecky,
and Eslam Amer

Abstract Artificial intelligence and its subparts (like evolutionary algorithms, machine learning, ...) are search methods that can be used for solving optimization problems. A particular class of algorithms like bioinspired one mimic working principles from natural evolution (or swarm intelligence) by employing a population-based (swarm) approach, labelling each individual of the population with a fitness and including elements of random, albeit the random is directed through a selection process. In the contemporary astrophysical literature is a lot of research papers, that work with various machine learning methods, while evolutionary/swarm algorithms are almost neglected. In this chapter, we review the basic principles of evolutionary algorithms and discuss their purpose, structure and behaviour. In doing so, it is particularly shown how the fundamental understanding of natural evolutionary processes has cleared the ground for the origin of evolutionary algorithms. Major implementation variants and they are structural as well as functional elements are discussed. We also give a brief overview of usability areas of the algorithm and end with some general remarks of the limits of computing, including demonstration of its use on astrophysical data processing at the end.

1 Introduction

Machine learning (ML) and related techniques like evolutionary algorithms (EA), artificial neural networks (ANN) are a part of artificial intelligence (AI) and are powerful tools that can be used to solve various very complex engineering problems. Generally speaking, those techniques can be divided into two main categories, as depicted in Fig. 3. In principle, ML solve selected problems in the same way as a human, which in general can be used successfully to a large set of engineering

I. Zelinka (✉) · T. Cong Truong · D. Quoc Bao · L. Kojecky · E. Amer
Department of Computing Science, Faculty of Electrical Engineering and Computing Science,
Technical University of Ostrava, Tr. 17. Listopadu 15, Ostrava, Czech Republic
e-mail: ivan.zelinka@vsb.cz

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021
I. Zelinka et al. (eds.), *Intelligent Astrophysics*, Emergence, Complexity
and Computation 39, https://doi.org/10.1007/978-3-030-65867-0_1

problems like the design of different devices and complex systems identification, control and modelling, etc.

To understand the application capability of ML and its topic in general, it is essential to remember some basic facts and some selected algorithms as well as its computational capabilities. This chapter is organized in 3 parts. The first one is focused on a brief overview of AI. It is followed by EAs subsection because while ANN as a part of ML is very popular in the astronomical community, EAs surprisingly, despite its wide applicability, are still overlooked. The last, 3rd part, discuss limits of any kind of computations and simulations. This part is very important because it gives us information about our computer limits. The chapter is then closed by the example of EA use on mathematical classification synthesis, applied on astrophysical data.

2 Artificial Intelligence and Machine Learning

2.1 *Brief Overview of Artificial Intelligence*

Defining AI can take two approaches. First, it is a science that strives to discover the nature of intelligence and develop smart machines in which scientists apply information, logic, self-learning, and determination to make machines becoming intelligently. To put it simply, humans create machines with intelligence. This intelligence can think, learn, decide, and work while trying to solve a problem as human intellect.

On the other hand, scientists define AI as a science that researches and develops methods for resolving complex problems that impossible to be resolved without adopting intelligence. For example, scientists build an AI system for real-time analysis and decision making based on enormous amounts of data. In recent years, AI has resulted in advances in many scientific and technological fields such as computerized robots, image and recognition, natural language processes, expert systems and other majors.

In general, there are two AI classification systems: the first classification is based on their capacity to mimic human characteristics, the technology they use to do this, their real-world applications, and the theory of mind. Using these characteristics for reference, all artificial intelligence systems fall into one of three types: narrow AI, general AI and strong AI.

The latter one is depended on how a machine compares to humans in terms of versatility and performance. According to this classification, there are four types of AI: reactive machines, limited memory machines, the theory of mind, and self-aware AI (Fig. 1).

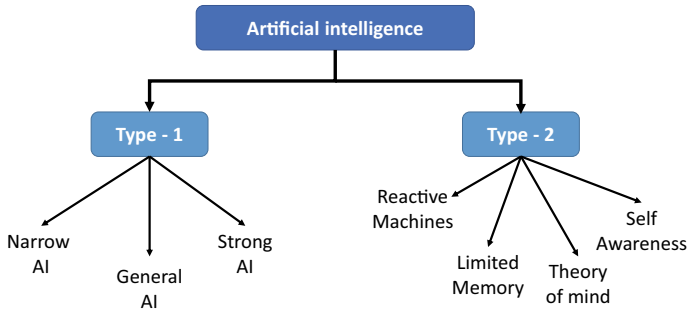


Fig. 1 Type of AI

2.2 Learning Algorithms

AI is a branch of computer science that seeks to produce a new type of intelligent automaton that responds like human intelligence. To achieve this goal, and machines need to learn. To be more precise, we need to train the computer by using the learning algorithms. Generally, learning algorithms help to enhance performance in accomplishing a task through learning and training from experience. There are currently three major types of learning algorithms which we use to train machines:

- **Supervised learning:** This type requires a training process with a large and representative set of data that has been previously labelled. In other words, the data is labelled, and the program learns to predict the output from the input data. These learning algorithms are frequently used as a classification mechanism or a regression mechanism.
- **Unsupervised learning:** In contrast to supervised learning, unsupervised learning algorithms use unlabeled training datasets. More precise, the data is unlabeled, and the program learns to recognize the inherent structure in the input data. These approaches are often used to cluster data, reduce dimensionality, or estimate density.
- **Reinforcement learning:** Reinforcement learning is a type of learning algorithm that learns the best actions based on rewards or punishment. Reinforcement learning is useful for situations where data is limited or not given.

2.3 Machine Learning: An Overview

Machine learning (ML) is a branch of AI that aims to empower systems by utilizing data to learn and improve without being explicitly programmed. ML has strong ties to mathematical techniques that enable a process of extracting information, discovering patterns, and drawing conclusions from data. The outputs of ML algorithms are represented in terms of probabilities and confidence intervals. Manual analysis of

this huge amount of data becomes a difficult task, so ML algorithms are employed to automate the learning process. An ML method often comprises the following step:

- Data preparation
- Choosing algorithm
- Prediction

2.4 Machine Learning Process

The process begins with data collection, cleaning, transform; this phase is often known as data processing. After that, the data then split into a group: one for training and the other for testing. After the model has completed the training phase with the training data, it now needs to be tested with the remaining data not being used. This process can be repeated, and in each iteration, the performance parameters can be tuned.

Through analyzing data, the ML instructs an algorithm of how to learn. In fact, the algorithms can learn through a process of mapping of input to output, detection of patterns or by reward. When the algorithm processes more data, it becomes more intelligent.

In other words, by observing the data, the ML algorithms learning process can be done. Accordingly, the algorithm can enhance its predictive performance by exposed to more observations (Fig. 2).

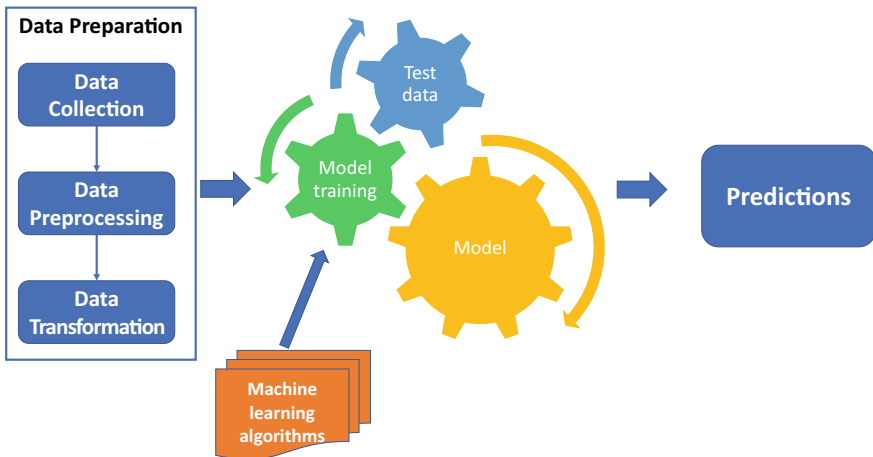


Fig. 2 Machine learning process

2.5 Machine Learning Algorithms

There are different types of the ML algorithm, but they can generally be classified into three main categories: supervised learning, unsupervised learning, and reinforcement learning. The Fig. 3 illustrates types of ML algorithms and their applications.

Some of the commonly ML algorithms:

- **Linear Regression:** This kind of algorithm is often utilized to estimate real values based on continuous variables. Linear Regression is often split into two types: Simple Linear Regression, which is characterized by one independent variable and Multiple Linear Regression, which is characterized by multiple independent variables
- **Logistic Regression:** It is a classification algorithm that is often utilized to estimate discrete values (binary values) based on a given set of independent variables.
- **Decision Tree:** This supervised learning algorithm is used to solve both regression and classification problems. Furthermore, it works for both categorical and continuous dependent variables. This algorithm tries to solve the problem, by using tree representation in which each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label.
- **Support Vector Machine (SVM):** This algorithm is often used for classification. An SVM uses support vectors to define a decision boundary. Classifications are made by comparing unlabeled points to that decision boundary.
- **Naive Bayes:** This is a family of simple probabilistic algorithms based on applying Bayes' Theorem with strong (naive) independence assumptions between the features.
- **k-Nearest Neighbors (kNN):** It is an SL algorithm for identifying an unknown data point based on K nearest neighbour labelled data. The nearness of points is typically determined by using distance algorithms, for instance, Euclidean distance.
- **K-Means:** This unsupervised algorithm often utilizes to solve the clustering problem. K-Means groups unlabeled data into K clusters based on the cluster centres.

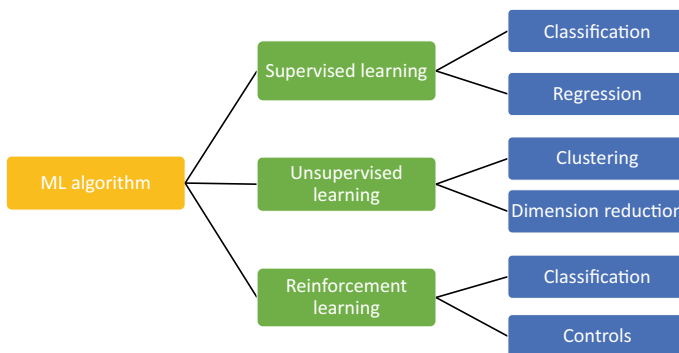


Fig. 3 Types of ML algorithm

- Dimensionality reduction algorithms: These algorithms are used to reduce the number of random variables under consideration by obtaining a set of a principal variable. Some common methods: Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Generalized Discriminant Analysis (GDA).
- Ensemble Algorithms: These are models composed of multiple models that are independently trained and whose predictions are combined in some way to make the overall prediction. Some popular ensemble models: AdaBoost, Random forest, Gradient Boosting.

2.6 Evaluation Metrics

Evaluation metrics measure the performance of a model. An essential aspect of evaluation metrics is their capability to distinguish among model results.

In term of classification problems, the Confusion matrix is one of the most common and most straightforward metrics used for deciding the correctness and accuracy of the model. It is frequently used for the classification issue where the output can be of two or more types of classes. In fact, many performance metrics are calculated basing on the confusion matrix and the values inside it, as shown in Table 1.

Terms associated with this confusion matrix:

- True Positive (TP): The sample is positive, and the model predicts it positive.
- True Negative (TN): The sample is negative, and the model predicts it negative.
- False Positive (FP): The sample is positive, and the model predicts it positive.
- False Negative (FN): The sample is negative, and the model predicts it negative.

The metrics derive from the confusion matrix are describes as following:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{FN + TP} \quad (3)$$

Table 1 Confusion matrix

		Actual values	
		Positive (1)	Negative (0)
Predicted values	Positive (1)	True Positive (TP)	False Positive (FP)
	Negative (0)	False Negative (FN)	True Negative (TN)

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

The accuracy is the rate of precisely predicted news to all of the samples. Precision metric measures the number of correct positive predictions. Recall value shows the ratio of the number of correct positive predictions made out of all positive predictions that could have been made. F1-score is the harmonic mean value of the recall value and precision.

3 EA and Swarm Overview

Optimization is present in almost all areas of life, from finance to medicine and engineering. Many real-world problems require finding an optimal solution such as finding the shortest trajectory for the Travelling Salesman Problem (TSP) or finding suitable control parameters to minimize the system's energy consumption, as well as finding optimal global values for functions. For simple modelled problems, we can apply classical mathematical means to find optimal solutions to the problem. However, with complex and constantly changing models, the classical mathematic solver becomes impossible. It challenges researchers to find the other way to deal with these practical issues. So, the computational optimization was proposed in which the Evolutionary Algorithm and Swarm Intelligence were representatives [31].

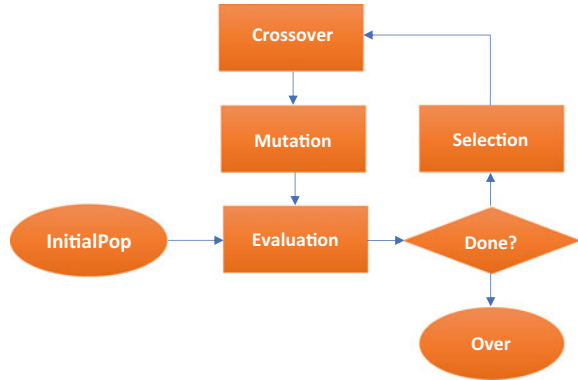
3.1 Evolutionary Algorithm

The Evolutionary Algorithm is a common name for optimization algorithms that have mechanisms essentially based on the evolutionary theory of Charles Darwin. From the individuals in the initial population representing the candidate solution of the problem, through generations of survival, crossover, mutation, and heredity, the individuals evolved, changing their genomes to adapt to the environment, helping the population get closer and closer to the optimal solution of the problem [12].

Although implemented in many different forms, these algorithms have a common characteristic of the evolution of organisms, and can be summarized in the following processes:

- Initialization: This process aims to initialize an initial population of individuals that represents a candidate solution to the given problem in the form of the computer code.
- Selection: The main objective of this process is to select one or several individuals to be activated, ie individuals will evolve or participate in the evolution of other individuals.
- Crossover: A combination of selected individuals in a given rule to create new ones with the heredity characteristics of the parents—selected individuals. Crossover is

Fig. 4 The evolutionary algorithm flowchart



one of two important points of the evolution of creatures as well as evolutionary algorithms. It combines and maintains the good characteristics of organisms from generation to generation.

- **Mutation:** The mutation is the important rest component, indispensable in the evolution of organisms in nature, as well as in evolutionary algorithms. This process creates “new characteristics” that have not existed in the previous individuals in the whole population. Depending on each algorithm, the mutation process will take place in different ways.
- **Evaluation:** Over the long evolutionary period, under the influence of harsh environments, weak individuals are eliminated, and healthy individuals continue their survival and growth. In the evolutionary algorithm, this is done by assessing individuals in the population by the given fitness function. Individuals created after the selection, crossover, and mutation processes will be evaluated. Inappropriate individuals that have bad fitness values will be removed from the population and replaced by better individuals.

These processes are repeated until the algorithm finds an optimization value that meets the requirements of the practice, as depicted in Fig. 4.

3.2 *Swarm Intelligence*

The term “Swarm Intelligence” was first introduced by Beni and Wang in the context of cellular robotics system [7]. However, much more sooner, the idea of swarm robots has been introduced (1964) by Stanislaw Lem in his novel *Invictible* [25, 44]. The swarm intelligence algorithm, different from the evolutionary algorithm, is inspired by the cooperation-competition behaviours of intelligent creatures to solve their problems such as foraging, attacking enemies, or defending the nest. These behaviours reflect the action between individuals in a population, or between individuals and the environment, taking place within the same generation. In other words,

the mechanism of swarm intelligence algorithm is the interaction between individuals with each other or the environment according to a given rule to search the optimal solution to the problem.

However, there are many similarities between the evolutionary algorithm and swarm intelligence algorithm. They are all population-based algorithms, meaning that one population or some sub-populations need to be initialized at the beginning of algorithm; they select individuals in the population to be activated individuals; they use the fitness function as a basis for selecting and evaluating individuals; they create new ones based on previously selected individuals—but the mechanisms for reproducing new individuals are completely different.

To date, many swarm intelligence algorithms have been proposed in the literature and successfully applied in practice, including function optimization problems, finding optimal routes, scheduling, structural optimization, and image analysis [20]. Examples of swarm intelligence algorithms are: Ant Colony Optimization [18], Particle Swarm Optimization [28], Artificial Bee Colony [27], Bacterial Foraging [39], Firefly Algorithm [49], Bat Algorithm [48], Self-Organizing Migrating Algorithm [50], and Whale Optimization Algorithm [38].

3.3 Examples

This subsection introduces the Genetic algorithm and the Self-organizing migrating algorithm—SOMA, which represents two class of the evolutionary algorithm and the swarm intelligence, and briefly analyze to point out their characteristics.

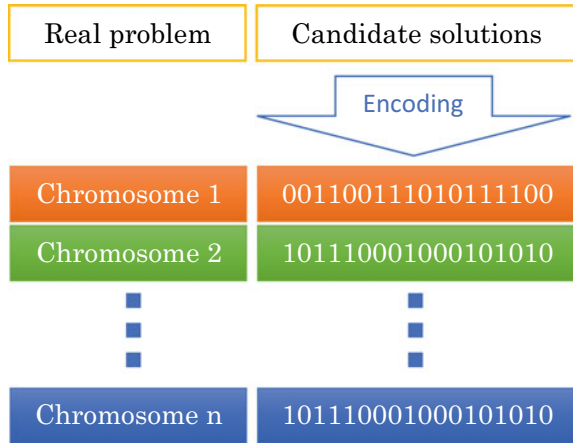
3.3.1 Genetic Algorithm

The term “Evolution strategies” was introduced by I. Rechenberg in the 1960s and developed further in the 1970s and later [41], including ideas about evolutionary computing, and genetic algorithm, derived from evolutionary computing, introduced by John Holland in years later [22].

It has to be said, that origin of the evolutionary algorithms comes from A. M. Turing which defined whole terminology and processes as they are used in EAs up to now. One of the first to transform Turing’s ideas into a real computer numerical experiments were Barricelli (1954) [6]. Results were published in the journal “*Methodos*” with the title “*Esempi Numerici di processi di evoluzione*” and consequently repeated and improved in 1962 [5] when ECT numerical experiment with 500 of 8 bits strings had been successfully done. Later then has been written papers like [41] or [22].

The genetic algorithm works based on Darwin’s theory of evolution and Mendelian theory of heritage [35] that has been developed in the same time as the Darwinian theory of evolution, By coding the candidate solutions of the problem into chromosomes, similar to the genome of organisms. Under the influence of natural selection,

Fig. 5 A population is encoded and initialized in the genetic algorithm



here is the fitness function, these chromosomes evolved, getting closer and closer to the optimal solution of the problem. The following describes the processes of the genetic algorithm:

- **Initialization:** A population is randomly generated, containing many chromosomes that are candidate solutions to the problem. These chromosomes can be a binary string, integer, or real number depending on the problem. This process is also known as the encoding of chromosomes. Figure 5 illustrates the process of initializing an initial population and encodes the solution of the problem into chromosomes.
- **Evaluate:** Chromosomes are evaluated by a given fitness function. The fitness function is the problem to be solved, which represents the influence of the natural environment on creatures.
- **Selection:** Selection is the process of selecting two or more individuals as a parent, according to their fitness values, in principle that individuals with better fitness values will have a better chance to select, in order to inherit good characteristics for the next generations.
- **Crossover:** The segments of chromosomes are exchanged between parents to create offspring. Figure 6 depicts the crossover between chromosomes 1 and 2, in which a part of the father's chromosome combines with a part of the mother's chromosome to form offspring 1 and 2.
- **Mutation:** Crossover and mutation are two very important processes of the genetic algorithm, symbolizing the evolution process of organisms in nature. Figure 7 depicts the mutation taking place at three different positions on chromosomes 1 and 2, whereby the value of 0 will be changed to 1 and vice versa, creating new features that have never appeared before in the genome of the population.

After mutation, the offspring will be re-evaluated with fitness function, and they will replace parents if they have better fitness value. Furthermore, these processes are repeated until the algorithm reaches the given stop conditions.

Fig. 6 The crossover process between chromosomes 1 and 2 to reproduce new offsprings

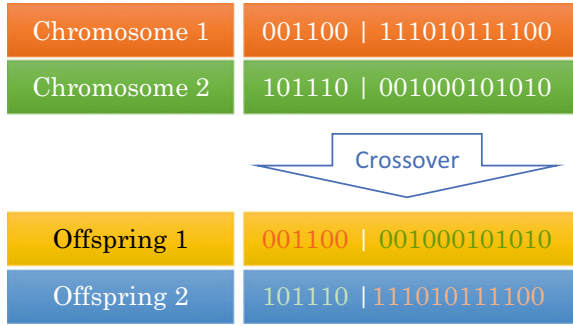
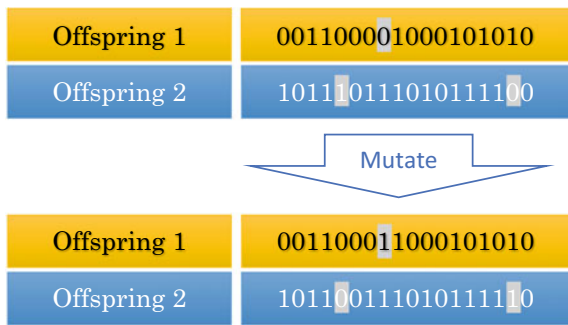


Fig. 7 The mutation takes place in the chromosomes of offspring



Obviously, this is the simplest form of the genetic algorithm. Over the decades of development, genetic algorithm operators have been proposed and continually improved (refer to [37, 42] for more details), as well as widely applied in almost areas of life, such as chaos-DNA [43], power generation [3], path planning of mobile robots [46], and energy recovery [32].

3.3.2 Self-organizing Migrating Algorithm—SOMA

First introduced in 1999, SOMA is a swarm intelligence algorithm, based on sharing knowledge of individuals in a population [51]. Accordingly, first, a population is initialized, consisting of individuals that are candidate solutions to the problem. The population is then evaluated by the fitness function and the best individual is chosen to be the leader, the remaining individuals will jump step-by-step towards this leader, called the migration process. Better positions will be found during their movement and will replace the initial position.

In the SOMA algorithm, individuals share their position. The fitness value of the position of individuals determines the distribution of the whole population. In other words, populations tend to move toward the best individuals, similar to birds or ants in nature that will congregate to where food is most abundant after some members found it.

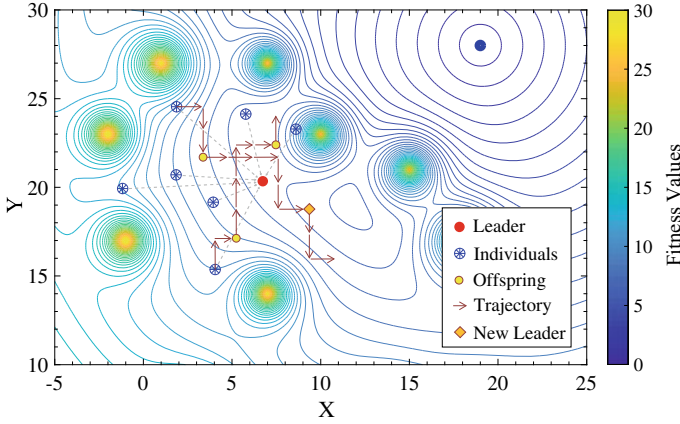


Fig. 8 The principle of SOMA algorithm

The SOMA algorithm contains all the characteristics of swarm intelligence, that is: initializing an initial population, selecting action individuals, sharing knowledge among individuals (cooperation), recreating new information, and eliminating bad individuals (competition). These processes are repeated, named migration loops until the stop conditions are met, see [51, 53] for more details.

Figure 8 describes the operating principle of the SOMA algorithm in geometrically. Accordingly, the problem to be solved is described as a contour map, and the blue points represent the solutions of the problem. These individuals share their positions to find the best individual in the population, as well as cooperation with other individuals to find a better position during their migration. Mathematically the movement of individuals is calculated through Eq. 5.

$$P_{os}^{new} = P_{current} + (P_{leader} - P_{current}) n Step PRTVector_j \quad (5)$$

where:

- P_{os}^{new} : the offspring position in new migration loop,
- $P_{current}$: the offspring position in current migration loop,
- P_{leader} : the leader position in current migration loop,
- $PRTVector_j$: the perturbatively factor, get a value of 0 or 1,
- $Step$: the step of each move,
- n : the number of jumping step, from 1 to N_{jump} .

This is the canonical version of the SOMA algorithm. During its 21-year development, SOMA has had many improved versions such as SOMA T3A [17], hybrid SOMA [15], and the modified Nelder-Mead SOMA [1], as well as being widely applied in many areas of engineering such as reliability-redundancy optimization of systems [19], control a semi-batch chemical reactor [13], and design of electromagnetic components [26].

For more informations, we recommend to read for example [23, 52, 54, 56] or [55].

4 Limits to Computation

Unfortunately, many people believe that everything can be computed if we have a sufficiently powerful computer and elegant algorithm. The goal of this part is to show that some problems cannot be solved algorithmically due to their nature. Popularly speaking, there is not, has not been and will not be enough time for their solution.

Part of these restrictions is also physical limits that follow from the material nature of the universe, which restricts the output of every computer and algorithm by its space-time and quantum-mechanical properties. These limits, of course, are based on the contemporary state of our knowledge in physical sciences, which means that they might be re-evaluated in the case of new experimentally confirmed theories (strings, etc.). At this moment, however, this is only a speculation, and we must adhere to the generally accepted and confirmed facts from which these limits follow.

4.1 Searched Space and Its Complexity

The complexity of the optimization problems can be demonstrated by many examples. Let us follow examples from [36]. A typical representative is the so-called SAT problem (boolean satisfiability problem). This is a problem from the field of logic that is represented by a complex logical function with a great number of logical variables. Relation (6) is an example from [36].

$$F(x) = (x_{17} \vee \bar{x}_{37} \vee x_{73}) \wedge (\bar{x}_{11} \vee \bar{x}_{56}) \wedge \cdots \wedge (x_2 \vee x_{43} \vee \bar{x}_{77} \vee \bar{x}_{89} \vee \bar{x}_{97}), \quad (6)$$

That contains 100 variables, and the objective is to find such values of individual arguments of this function for which the resulting value of relation (6) is TRUE. At first sight, this problem looks very trivial; nevertheless, it is a problem that cannot be solved by classical methods. If we take into account that the expression contains 100 unknown variables that can assume two values (0, 1), then the number of all possible combinations is 2^{100} , which is approximately 10^{30} . In order to get a better impression on the monstrous size of this number, it is sufficient to imagine how long it would take to evaluate all the combinations if 10^{13} of these combinations are evaluated within one second (which is, of course, impossible on a single processor). The correct answer is 10^9 years. This essentially means that the solution to this problem would take approximately the time of the existence of the universe.

Another complication related to this problem is the fact that function (6), as defined, does not make it possible to evaluate the quality of the current solution. This

is a substantial drawback, mainly if the evolutionary techniques are used because there is no possibility how to determine whether the qualities of two subsequently found solutions are close or not. As will be shown further, when using the evolutionary algorithms, it is of vital importance that the information on the quality of the solution is available for the determination in which “direction” the optimum solution lays. This is not possible in the case of the SAT problem, because the function only returns TRUE or FALSE, i.e. “good” or “bad”. It does not return how good or bad a given solution is.

The SAT problem is more or less a scholastic problem. As a more practical problem from real life, one can use the well-known travelling salesman problem. This is a problem, in which a travelling salesman must visit a set of N cities in the shortest possible time or with the smallest fuel consumption or, as the case may be, fulfil other criteria. The travelling salesman problem can be visualized by means of graphs, as demonstrated in Figs. 9, 10 and 11.

The condition is that each route must start and end in the same city and each city should be visited only once. This is, therefore, a purely practical problem. The trajectory of the travelling agent represents a sequence of dots, such as, for example, “2 – 3 – ... – 7 – 26 ...”. The number of all possible combinations is $n!$. In the case of a symmetrical problem of a travelling salesman (the distance from city A to B is the same as from city B to A), $2n$ routes repeats. In this case, the final number of all possible combinations is $(n - 1)!/2$. However, this number is still large. As shown in Fig. 13, the number of all possible combinations very quickly grows with the number of cities. Already for $n > 6$, there are more combinations in the travelling agent problem than in the SAT problem. Figure 13 shows the growth of the number of solutions to the SAT problem in comparison with the growth of the complexity of the travelling salesman problem.

Let us look further. The travelling salesman problem has 181,440 possible solutions for 10 cities. There are 10^{16} possible solutions for 20 cities and 10^{62} for 50 cities. If 60 cities are used, then there is 10^{79} of possible solutions. This number is equal to the estimated number of protons in our universe, i.e. if one proton is used as memory to store one possible solution, then all protons in universe can store only TSP with size 60 cities. No more. It is worth mentioning that there are approximately 10^{21} litres of water on our Planet [36]. It is a trivial task to calculate how many globes could be covered with this volume of water had we used a reservoir with a volume of 10^{62} litres water. It is therefore evident that even from such a trivial example as the optimum distribution of parcels, a problem may arise, whose optimum solution is not known. It is worth mentioning that at the present time, there are particular types of evolutionary algorithms (ACO—Ant Colony Optimization) that manage up to 10,000 cities satisfactorily. We leave it to the kind reader to calculate what is the number of combinations (hint: 2.846259680910^{35659}).

The third and last sample problem is the arbitrary artificial testing function, that is used as a testing function for various types of evolutionary techniques; (for another example, see [36]). This function is strongly nonlinear, and it is complicated. Although the function in this example is artificial, one can encounter even “wilder” functions that represent real physical problems. This type of function looks innocent;

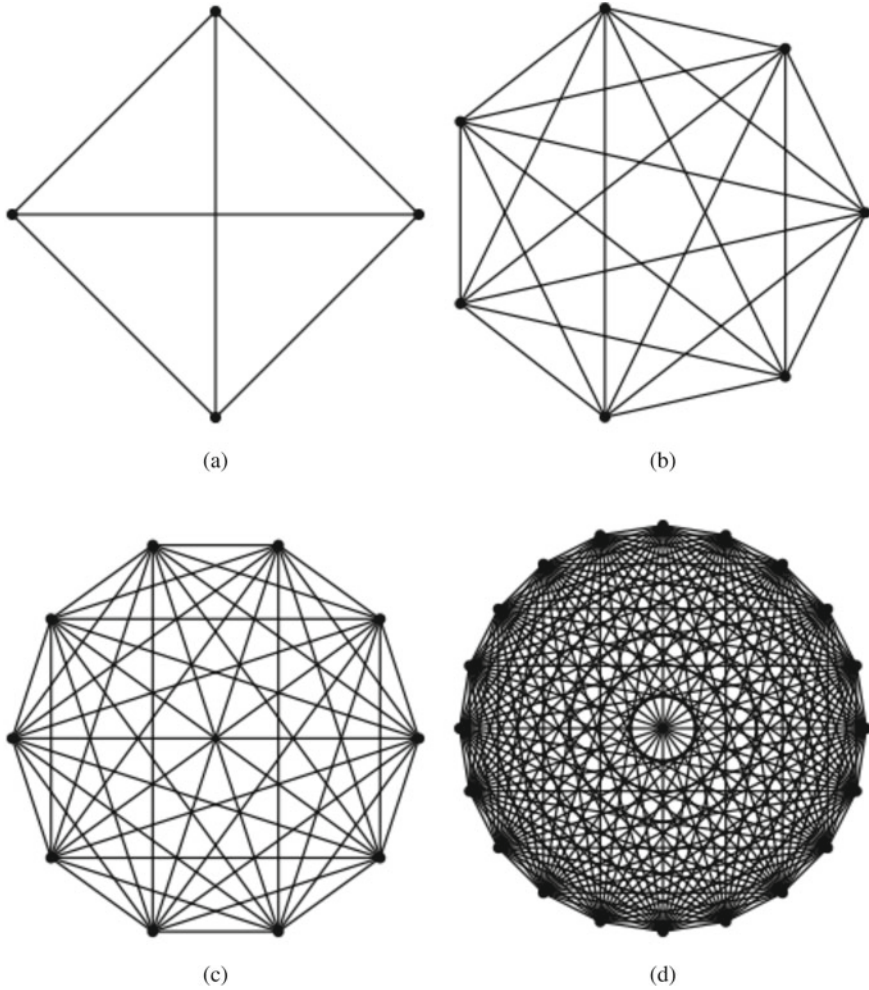


Fig. 9 Connections in the travelling salesman problem that form $n!$ possible trajectories (see Fig. 12). We indicate the number of cities/number of connections between the cities **a** 4/6, **b** 7/21, **c** 10/45, **d** 20/190

however, it is the contrary in this case. It is necessary to realize that everything is running in computers, thus also the optimization of such a function is digitized. If this would not be so, then it would be necessary to calculate the value of the function in an infinite amount of points. Due to digitization, this infinity reduces to a set of values of the function, whose cardinality is finite, even though it is still immense. Let us assume that the computational accuracy of the computer used is 6 decimals. In this case, every variable in a given function assumes real values. Due to digitization, the infinity mentioned above reduces to a set of possible solutions, the cardinality

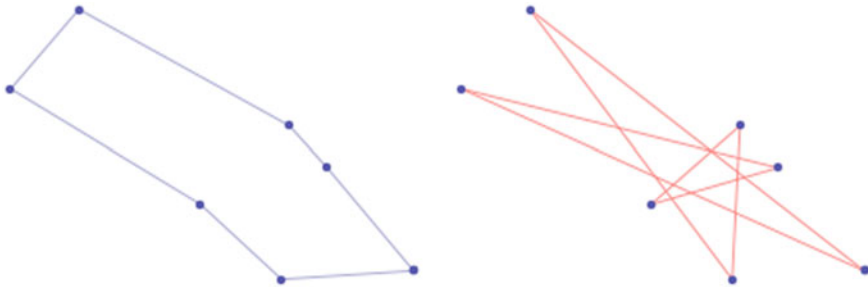


Fig. 10 Travelling salesman visiting seven cities: the best route is on the left and the worst route is on the right

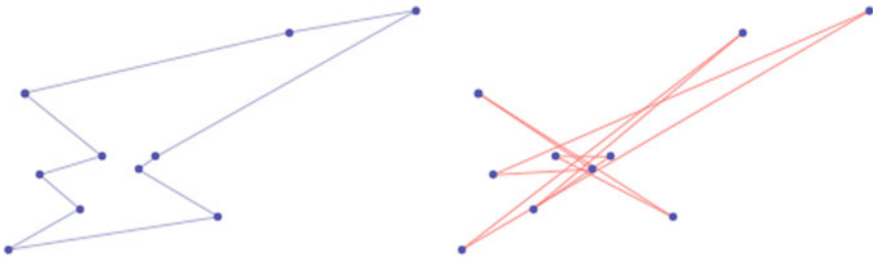


Fig. 11 Travelling salesman visiting ten cities: the best route is on the left and the worst route is on the right

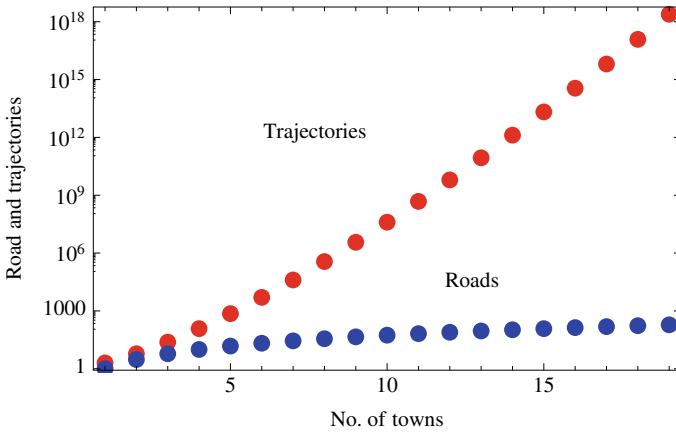


Fig. 12 Visualization of the travelling salesman complexity The difference is illustrated between the number of roads (blue dots) and possible trajectories (red dots)

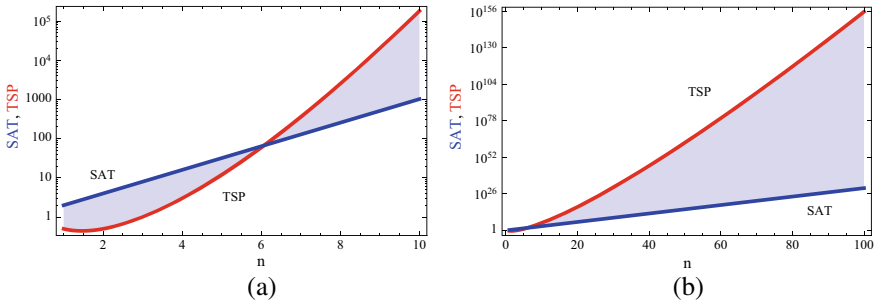


Fig. 13 Growth of the problem complexity for SAT (blue curve) and travelling salesman (red curve). Starting with seven cities (or variables in SAT), the travelling salesman problem is more time consuming

of which is still immense. Let us assume that variable in a given arbitrary complex function may assume up to 10^7 different values. In general terms, this function will assume 10^{7^n} values (n is a number of variables here). This number is many times greater than the number of solutions for the travelling salesman problem for $n \leq 10^7$. For $n = 50$, there are 10^{350} solutions. It is necessary to realize that the accuracy of present computers is much higher and the problem, therefore, generates a gigantic number of possible solutions.

Let us mention that the complexity of problems is not measured in theoretical informatics by the time demand factor (even though it is so de facto in the result), but primarily by the complexity or dependence of the capacity of the algorithm on the growing number of input data. As was already mentioned, there are problems whose complexity grows nonlinearly with the growing input (for example, the travelling salesman problem, see Fig. 13). We then speak about algorithms with polynomial, exponential, etc., complexity. The examples of the complexity of problems are in Tables 2, 3 and 4. Table 2 gives the number of possible solutions for n input parameters. If testing one solution takes the predefined time, the time demand factor for searching all possible solutions is in Table 3. If faster computers are used, the gross estimation of the acceleration of computation is in Table 4. It is obvious from these tables that there are many problems that no computer can help to solve.

For comparison: The number of protons in the visible Universe has approximately 79 digits. The number of microseconds from the “big bang” has 24 digits.

4.2 Physical Limits of Computation

As was already mentioned, there are limits restricting the output of any computer that follows from the quantum-mechanical nature of mass. These limits restrict both the output of the computer and its memory. It is obvious from these restrictions that there are many problems that no computer can help to solve.

Table 2 Estimation of the values of some functions

n	10	50	100	300	1,000
Function					
Polynomial					
5n	50	250	500	1,500	5,000
$n \log_2 n$	33	282	665	2,469	9,966
n^2	100	2,500	10,000	90,000	1 million (7 digits)
n^3	1,000	125,000	1 million (7 digits)	27 million (8 digits)	1 billion (10 digits)
Exponential					
2^n	1,024	16 digit number	31 digit number	91 digit number	302 digit number
$n!$	3.6 million (7 digits)	65 digit number	161 digit number	623 digit number	Giant number
n^n	10 billion (11 digits)	85 digit number	201 digit number	744 digit number	Giant number

Table 3 Estimation of the time of $f(n)$ operations if 1 operation takes 1 μ s

n	10	20	50	100	300
Function					
Polynomial					
n^2	1/10,000 s	1/2,500 s	1/400 s	1/100 s	9/100 s
n^5	1/10 s	3.2 s	5.2 s	2.8 h	28.1 days
Exponential					
2^n	1/1,000 s	1 s	35.7 years	400 trillion centuries	75 digit # of centuries
n^n	2.8 days	3.3 trillion years	70 digit # of centuries	185 digit # of centuries	728 digit # of centuries

Table 4 Estimation of the time of $f(n)$ operations if 1 operation takes 1 μ s

Maximum dimension of the input manageable in a reasonable time			
Function	Current computers	100 times faster computers	1,000 times faster computers
n	N_1	100 N_1	1,000 N_1
n^2	N_2	10 N_2	31.6 N_2
2^n	N_3	$N_3 + 6.64$	$N_3 + 9.97$
$n!$	N_4	$N_4 + 1$	$N_4 + 2$

Basic restriction in this direction is the so-called Bremermann's limit [8], according to which it is not possible to process more than 10^{51} bites per second in every kilogram of matter. In the original work of this author [8], the value of 2×10^{47} bites per second in one gram of matter is indicated. At first sight, this limit does not look frightening, but only until we take "elementary" real examples for comparison. Let us consider chess-mate for illustration. For this game, the estimated number of combinations is 10^{120} . As another example, let us consider the lattice of cellular automata [24] of 100×100 cells that can only assume black and white values that represents $2^{10,000}$, which is approximate $10^{3,000}$ combinations—images. The current TV sets with an LCD monitor have approximately $1,300 \times 700$ pixels, which can assume various colours and degrees of brightness. It is clear that the number of combinations is much higher on an LCD monitor.

This limit can be derived in the following relatively simple manner: For making it possible to measure, process and transfer information, it is **necessary** to store it on some physical carrier. This information may be electromagnetic radiation, paper tape, laser beam, etc., therefore, always something material. Information alone, i.e., without a physical carrier, cannot exist. Because elementary particles and their energy states can also be used as a carrier of information, it is obvious that the limit of how much information the matter can carry follows from the restriction that was discovered at this physical level.

In order to make it possible to measure this information, it must be modulated on the corresponding carrier to resolve the individual carrier's states that represent the value of the information. Von Neumann [47] called the resolvable states "markers". The lowest resolvable energy states are the quantum states of matter, whose resolvability from the bottom is limited by Heisenberg's uncertainty relation. When deriving the already mentioned limits, it does not matter whether mass or energy types of carriers are considered. Both types are physically interchangeable. Therefore, if quantum states are considered as the smallest resolvable energy states, which will be considered as bits in this case, then the "energy-bit" resolution is given by Heisenberg's uncertainty relation. Generally, one can say that according to the Heisenberg principle of uncertainty, it is possible to always identify the final number of states. Because nobody can say which state will be observed, probability has to be used. It is common to say that variable X will have n different values with probability p_1, p_2, \dots, p_n . Based on information theory is clear that we can get

$$H(p_1, p_2, \dots, p_n) = - \sum_{i=1}^n p_i \log_2 p_i \quad (7)$$

bits of information. This function has one global extreme only if it hold $p_1 = p_2 = \dots = p_n = 1/n$ true. Then

$$H(1/n, \dots, 1/n) = - \sum_{i=1}^n (1/n) \log_2 (1/n) = n (1/n) \log_2 n = \log_2 n. \quad (8)$$

Such marker can carry maximally $\log_2 n$ bits of information. Based on quantum nature of our world it is clear that there is no better marker than marker represented by n states (i.e. energy levels) of selected quantum system. All levels have to be in interval $[0, E_{\max}]$ where E_{\max} is maximum of energy. If one can measure energy with precision ΔE , then in the marker, can be distinguished maximally $n + 1 = (E_{\max}/\Delta E) + 1$ energy levels. When one marker with $n + 1$ energy levels will be taken into consideration, then by this marker can be represented maximally $\log_2 (n + 1)$ of bits. On the contrary, when two markers will be used with energy levels in $[0, 1/2E_{\max}]$ it can represent $2 \log_2 (n/2 + 1) = \log_2 (n/2 + 1)^2$ bits whereas $n + 1 \ll (n/2 + 1)^2 = (n^2/4) + n + 1$ and so on. Based on this, it is clear that for representation of the maximal information carried by marker is optimal, when n different markers with energy levels in $[0, \Delta E]$ is used, i.e. with two energy levels which represents 0 and 1. In total it is possible to represent maximally $n \log_2 (n/n + 1) = n \log_2 (n/n + 1) = n \log_2 2$ i.e. n bits of information because clearly $\log_2 2 = 1$ hold.

Carrier with mass m is according to Einstein equation equal to $E_{\max} = mc^2$. It is obvious that in such a carrier it is possible to maximally have

$$n = \frac{E_{\max}}{\Delta E} = \frac{mc^2}{\Delta E} \quad (9)$$

bits of information. To calculate the exact amount of information stored by (9), then we need to use the Heisenberg principle of uncertainty.

$$\Delta E \Delta t \geq \frac{\hbar}{2} \quad (10)$$

In which $\hbar = h/2\pi$ (h is Planck constant, \hbar is Dirac constant). If in (10) the equality is taken into consideration, then one obtained for the upper estimation

$$n = \frac{mc^2}{\frac{\hbar}{2\Delta t}} = 4\pi \frac{mc^2}{h} \Delta t \quad (11)$$

During time interval Δt it is possible to process maximally $4\pi \frac{mc^2}{h} \Delta t$ bits of information. When $\Delta t = 1$ s one can get the maximal number of bits which can be processed or stored in mass per 1 s. For $m = 1$ kg this number (lets call it BL) is

$$BL = 4\pi \frac{c^2}{h} \quad (12)$$

where $[BL] = 1 \text{ kg}^{-1} \text{ s}^{-1}$,

In this moment it is only a matter of simple calculation to get the exact numerical value of BL, lets: speed of light and Planck constant $h = 6, 62607 \cdot 10^{-34}$ J.s. Finally we get

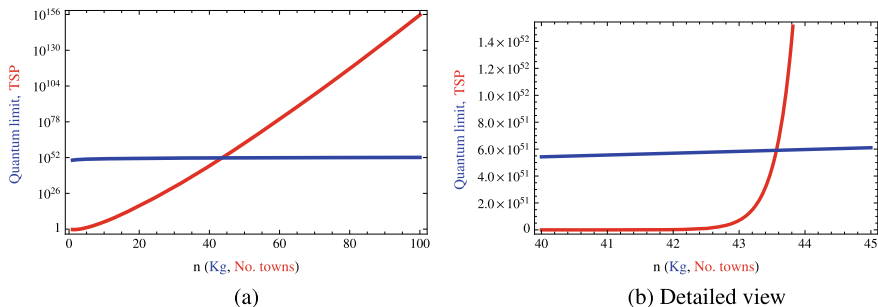


Fig. 14 Simultaneously plotted dependence of the number of possible solutions of the travelling salesman problem on the number of cities n (red) and the number of bits processed in a computer of mass m (blue). Let us add for more attentive readers that there is a logarithmic scale in the left figure, while a “normal” in the right figure. This is the reason why the plots appear considerably different in both figures

$$BL \approx 1,7045 \cdot 10^{51} \text{ kg}^{-1} \cdot \text{s}^{-1}. \tag{13}$$

This number, which we call BL here, is the so-called Bremermann limit. It is the definite limit which gives the maximal number of bits which can be processed or/and stored by an arbitrary matter. In the original paper Bremermann suggested 10^{47} which is caused by the use of nonstandard units (cm instead of m and grams instead of kg, as already mentioned before).

Based on this, it is visible that in our universe, the computational power is limited by matter, and basically there is no computer (existing or theoretical) which would be able to solve arbitrary problems.

If the mass of the Earth (5.9742×10^{24} kg) is taken into account, then a computer of such a mass might store (and subsequently also process) approximately 10^{76} bits every second. During the life of Earth (10^9 years), a computer of its mass might process maximally 10^{92} bits. If the output of a fictive computer is plotted against its mass, it is obvious (Fig. 14) that its “computational capacity” is exceeded already during the solution of the travelling salesman problem for a small number of cities/computer mass.

It is clearly obvious from Fig. 14 that the break between the number of cities and the computer mass occurs somewhere between 43 and 44. Perhaps it is not necessary to mention that the output of our computer is illustrated in bits, which is a little bit misleading, because one bit is not sufficient for storing information on one possible solution of the travelling salesman problem. Had this been taken into consideration during the computation, then the result would have been different, nevertheless approximately the same as for the order of magnitude.

If we take into account the ACO (Ant Colony Optimization) algorithm that satisfactorily solves the travelling salesman problem up to approximately 10,000 cities, then we would need a computer of the mass of 10^{35608} kg for storing and processing the information on all possible trajectories. In other words, 10^{35566} computers of the

mass of the Earth, should the computation be finished during the life of the universe (10^{17} s). In a similar way [34], we would derive the shortest possible time during which it is possible to process the stored information. This value is $t = 10^{-12}$ s; the current computers work in a region of 10^{-9} s.

In the publication [34], these considerations have been worked out in more details and applied to the transfer of information through an information channel (computation can also be considered as a transfer of information through a special channel). Beside other things, it was found that if a certain mass (or energy) of the transfer medium is reached, further information cannot be transferred through the channel, because the channel collapses into an astrophysical object called a black hole. According to [34], the transfer of information is efficient (optimum, maximally usable), if the information channel is on the brink of collapsing into a black hole.

Independently of whether these calculations are accurate or only approximate, it is obvious that physical limits restrict the possibilities of any computer and also of the mathematical computational methods.

5 Stellar Data Classification

Swarm intelligence, used to solve astro-physical problems was used for example in the paper [29, 30] or [16]. Lets discuss here one example.

One of many classification problems in astrophysics that can be taken into consideration is the so-called Be stars [40, 57]. Be stars are hot, rapidly rotating B-type stars with equatorial gaseous disk producing prominent emission lines H_α in their spectrum [45]. Be stars show a number of different shapes of the emission lines, as we can see in Fig. 15. These variations reflect the underlying physical properties of a star. Research in this section is focused on the evolutionary-automated model synthesis of Be stars based on typical shapes of their emission lines.

There has not been done much work on the classification of Be stars. The only application found [14] is focused on a broader category of variable stars including pulsating Be stars. However, the method is not suitable for our goals, as it is applied to the whole spectrum where the local differences in the shapes of Be lines are lost. Another approach is in [11] where the to zoom at the small part of a spectrum with the Be line is applied, a specific set of features characterizing and discriminating the shapes of Be lines.

The data of Be stars spectra come from the archive of the Astronomical Institute of the Academy of Sciences of the Czech Republic in Ondřejov. The aim was to find a suitable model of Be star spectra and mainly check whether more intensive experiments shall be done in this way.

The archive contains a huge amount of stellar data [9, 10] and therefore was selected a dataset containing 1564 sample spectra. This dataset was manually divided into four classes (177, 172, 1159 and 56) of samples. Each sample is given by 1863 equidistant points that represent the dependency of light intensity on light wavelength. In Fig. 15 there is a visualized characteristic part of spectra typical for each class.

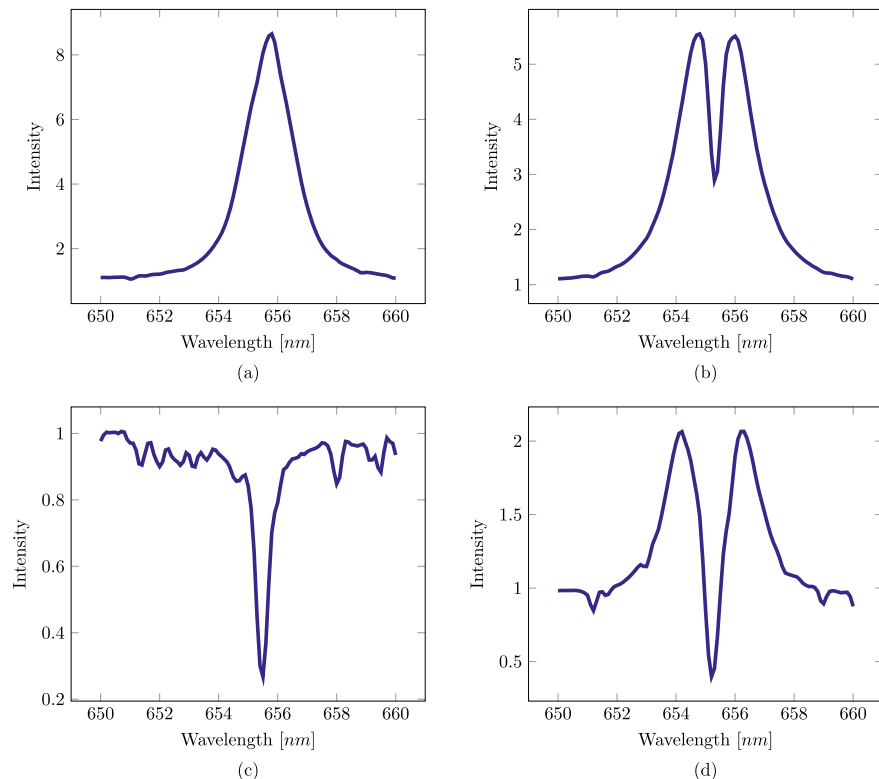


Fig. 15 Examples of selected data samples typical for each of the four classes. In class 1 **a** there is a pure emission. Class 2 **b** contains a small absorption part. In class 3 **c** there is a pure absorption part. Class 4 **d** consists of a larger absorption part

6 Data Preprocessing

From visual observation, it is obvious that each spectrum has a peak of its characteristic part (either emission or absorption) on a different wavelength. To achieve better results in further approximation and classification we identified the peak location x according to (14) and (15) and shifted it to the same location for all spectra. If there are identified more than one peak in a spectrum (for example class 2 and class 4), as x is chosen an average of all peak locations.

$$f(x - 2) < f(x - 1) < f(x) > f(x + 1) > f(x + 2) \tag{14}$$

$$f(x - 2) > f(x - 1) > f(x) < f(x + 1) < f(x + 2) \tag{15}$$

Although all spectra are given by 1863 equidistant points, the characteristic part is located only in the middle 100-point interval. The remaining part can be considered as

noise. Taking into account, only the characteristic part reduces the AP approximation time more than 18 times and can provide better approximation results by focusing only on the important data. In our previous research [21] on approximation Quintic and Sextic functions, AP provided best approximation results on interval $[-1, 1]$ —the characteristic was also normalized to interval $[-1, 1]$. For the same reason were all spectra intensities shifted to have a non-characteristic part around zero.

7 Classification Method

The first basic step of classification of the stellar spectra starts with an approximation of sample spectra with the most suitable mathematical formula. To save the time of approximation all spectra and to reduce the spectral noise, for each class, we created 20 normalized spectra as subclass representatives. The normalization was done by taking 5% of the particular class spectra and calculating average intensity for each wavelength as is visualized in Fig. 16.

The normalized spectra were approximated by AP that synthesized the most suitable mathematical formula—totally was synthesized 80 formulas. These formulas were one by one compared to the original spectra regarding cost function. As a result was chosen the class whose subclass-representative AP formula comparison resulted in the lowest cost value. The comparison is illustrated in Fig. 17.

For more detailed information we recommend papers [29, 30].

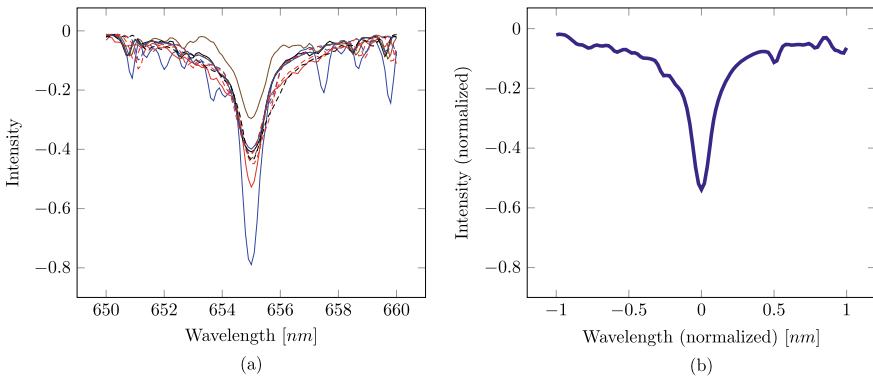


Fig. 16 Process of the spectra normalization. In **a** there are 5% of the particular class spectra; **b** shows the normalized output

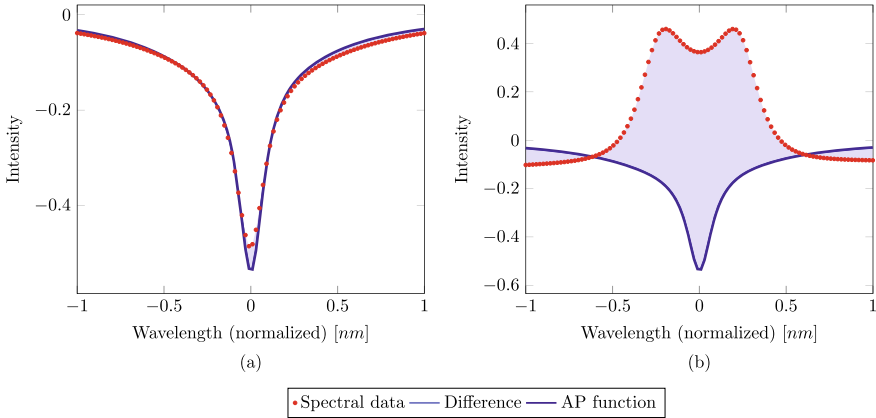


Fig. 17 Process of comparison of the original spectra with AP function. The difference in **a** is much lower than the difference in **b**—spectrum is classified as class 3

8 Conclusion

Artificial intelligence and some of its important subparts are intensively used in nowadays in the astrophysical data processing. We have been focused in this chapter on a very brief overview of its structure and mainly focused on evolutionary algorithms and swarm intelligence, with one application.

Together with modern algorithms, its limitations are there. It is important to realize its existence in order to avoid solving problems that simply cannot be solved ever.

There are many publications on the limits of computational technologies based on quantum physics. However, these publications are relatively very demanding on the knowledge from the field of quantum mechanics and mathematics. For extending the information indicated in this chapter, we recommend the already mentioned publications [2, 8, 33]. The substantial part of limits imposed by mass on processing and storing data is described in the first part [8]. The explanation is so understandable that even a reader at a high school level will understand it. In the paper [33], the relation between transfer channels and black holes is discussed. You can also read in [4] on the representation of individuals, basic concepts of ETV and the properties of the test functions. Of course, there are other monographs and Internet sources providing this information, but we consider publications mentioned above as sufficiently representative.

Acknowledgements The following grants are acknowledged for the financial support provided for this research: Grant of SGS No. SP2020/78, VSB-Technical University of Ostrava.

References

1. Agrawal, S., Singh, D.: Modified Nelder-Mead self organizing migrating algorithm for function optimization and its application. *Appl. Soft Comput.* **51**, 341–350 (2017)
2. Ashby, W.R.: Some consequences of Bremermann's limit for information-processing systems. *Cybern. Probl. Bionics*, 76 (1968)
3. Askarzadeh, A.: A memory-based genetic algorithm for optimization of power generation in a microgrid. *IEEE Trans. Sustain. Energy* **9**(3), 1081–1089 (2017)
4. Bäck, T., Fogel, D.B., Michalewicz, Z.: *Handbook of evolutionary computation*. Release **97**(1), B1 (1997)
5. Barricelli, N.A.: Numerical testing of evolution theories. part i: the-oretical introduction and basic tests. *Acta Biotheoreiica (Parts I/II)* **16** (1962)
6. Barricelli, N.A., et al.: Esempi numerici di processi di evoluzione. *Methodos* **6**(21–22), 45–68 (1954)
7. Beni, G., Wang, J.: Swarm intelligence in cellular robotic systems, In: *Proceedings of NATO Advanced Workshop on Robots and Biological Systems*, Tuscany, Italy, 26–30 June. NATO, NY (1989)
8. Bremermann, H.J.: Optimization through evolution and recombination. *Self-Organ. Syst.* **93**, 106 (1962)
9. Bromová, P., Barina, D., Škoda, P., Vážný, J., Zendulka, J.: Classification of spectra of emission-line stars using feature extraction based on wavelet transform. In: *Proceedings of 23rd Annual Astronomical Data Analysis Software and Systems (ADASS) Conference*, pp. 1–9999 (2013)
10. Bromová, P., Škoda, P., Vážný, J.: Classification of spectra of emission line stars using machine learning techniques. *Int. J. Autom. Comput.* **11**(3), 265–273 (2014)
11. Bromová, P., Škoda, P., Zendulka, J.: Wavelet based feature extraction for clustering of be stars. In: *Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems*, pp. 467–474. Springer (2013)
12. Câmara, D.: 1 - evolution and evolutionary algorithms. In: Câmara, D. (ed.) *Bio-inspired Networking*, pp. 1 – 30. Elsevier, Amsterdam (2015)
13. David, N., Lubomír, M.: Self-organizing migrating algorithm used to control a semi-batch chemical reactor. In: *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, pp. 1266–1269. IEEE (2013)
14. Debosscher, J.: Automated classification of variable stars: application to the ogle and corot databases (2009)
15. Deep, K., et al.: A new hybrid self organizing migrating genetic algorithm for function optimization. In: *2007 IEEE Congress on Evolutionary Computation*, pp. 2796–2803. IEEE (2007)
16. Dhiman, G., Kumar, V.: Astrophysics inspired multi-objective approach for automatic clustering and feature selection in real-life environment. *Mod. Phys. Lett. B* **32**(31), 1850385 (2018)
17. Diep, Q.B.: Self-organizing migrating algorithm team to team adaptive–soma t3a. In: *2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1182–1187. IEEE (2019)
18. Dorigo, M., Di Caro, G.: Ant colony optimization: a new meta-heuristic. In: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. 2, pp. 1470–1477. IEEE (1999)
19. dos Santos Coelho, L.: Self-organizing migrating strategies applied to reliability-redundancy optimization of systems. *IEEE Trans. Reliab.* **58**(3), 501–510 (2009)
20. Engelbrecht, A.P.: *Computational Intelligence: An Introduction*. Wiley, Hoboken (2007)
21. Gajdoš, P., Zelinka, I.: On the influence of different number generators on results of the symbolic regression. *Soft. Comput.* **18**(4), 641–650 (2014)
22. Holland, J.H.: Genetic algorithms and the optimal allocation of trials. *SIAM J. Comput.* **2**(2), 88–105 (1973)
23. Z. I. Analytical programming - an overview, Accessed 28 June 2020
24. Iachinski, A.: *Cellular Automata: A Discrete Universe*. World Scientific Publishing Company, Singapore (2001)

25. Invictible - an overview, Accessed 28 June 2020
26. Kadlec, P., Raida, Z.: Multi-objective self-organizing migrating algorithm applied to the design of electromagnetic components. *IEEE Antennas Propag. Mag.* **55**(6), 50–68 (2013)
27. Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. *Appl. Math. Comput.* **214**(1), 108–132 (2009)
28. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4, pp. 1942–1948. IEEE (1995)
29. Kojecky, L., Zelinka, I., Prasad, A., Vantuch, T., Tomaszek, L.: Investigation on unconventional synthesis of astroinformatic data classifier powered by irregular dynamics. *IEEE Intell. Syst.* **33**(4), 63–77 (2018)
30. Kojecky, L., Zelinka, I., Šaloun, P.: Evolutionary synthesis of automatic classification on astroinformatic big data. *J. Parallel Distrib. Comput.*, accepted, in print (2016)
31. Koziel, S., Yang, X.-S.: *Computational Optimization, Methods and Algorithms*, vol. 356. Springer, Berlin (2011)
32. Lee, S.: Multi-parameter optimization of cold energy recovery in cascade rankine cycle for lng regasification using genetic algorithm. *Energy* **118**, 776–782 (2017)
33. Lloyd, S.: Ultimate physical limits to computation (1999). [arXiv:quant-ph/9908043](https://arxiv.org/abs/quant-ph/9908043)
34. Lloyd, S., Giovannetti, V., Maccone, L.: Physical limits to communication. *Phys. Rev. Lett.* **93**(10), 100501 (2004)
35. Mendel, G.: Attempts “u over plant hybrid negotiations of the natural research association in br ü nn, vol. iv for the year (1865). *Abhand-lungen* **3**, 47 (1866)
36. Michalewicz, Z., Fogel, D.B.: *How to Solve It: Modern Heuristics*. Springer Science & Business Media (2013)
37. Mirjalili, S.: Genetic algorithm. *Evolutionary Algorithms and Neural Networks*, pp. 43–55. Springer, Berlin (2019)
38. Mirjalili, S., Lewis, A.: The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67 (2016)
39. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst. Mag.* **22**(3), 52–67 (2002)
40. Porter, J.M., Rivinius, T.: Classical be stars. *Publ. Astron. Soc. Pac.* **115**(812), 1153 (2003)
41. Rechenberg, I.: Evolution strategy: optimization of technical systems by means of biological evolution. *Fromman-Holzboog, Stuttgart* **104**, 15–16 (1973)
42. Shoba, S., Rajavel, R.: A new genetic algorithm based fusion scheme in monaural casa system to improve the performance of the speech. *J. Ambient Intell. Humaniz. Comput.* **11**(1), 433–446 (2020)
43. Suri, S., Vijay, R.: A bi-objective genetic algorithm optimization of chaos-dna based hybrid approach. *J. Intell. Syst.* **28**(2), 333–346 (2019)
44. Swirski, P.: Of games with the universe: preconceptions of science in stanislaw lem’s” the invincible”. *Contemp. Lit.* **35**(2), 324–342 (1994)
45. Thizy, O.: Classical be stars high resolution spectroscopy. In: *Society for Astronomical Sciences Annual Symposium*, vol. 27, p. 49 (2008)
46. Tuncer, A., Yildirim, M.: Dynamic path planning of mobile robots with improved genetic algorithm. *Comput. Electr. Eng.* **38**(6), 1564–1572 (2012)
47. Von Neumann, J., Kurzweil, R.: *The Computer and the Brain*. Yale University Press, London (2012)
48. Yang, X.-S.: A new metaheuristic bat-inspired algorithm. In: *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, pp. 65–74. Springer (2010)
49. Yang, X.-S., et al.: Firefly algorithm. *Nature-Inspired Metaheuristic Algorithms*, vol. 20, pp. 79–90 (2008)
50. Zelinka, I.: SOMA — Self-Organizing Migrating Algorithm, pp. 167–217. Springer, Berlin (2004)
51. Zelinka, I.: Soma–self-organizing migrating algorithm. *New Optimization Techniques in Engineering*, pp. 167–217. Springer, Berlin (2004)
52. Zelinka, I.: A survey on evolutionary algorithms dynamics and its complexity-mutual relations, past, present and future. *Swarm Evol. Comput.* **25**, 2–14 (2015)

53. Zelinka, I., Bukacek, M.: Soma swarm algorithm in computer games. In: International Conference on Artificial Intelligence and Soft Computing, pp. 395–406. Springer (2016)
54. Zelinka, I., Celikovský, S., Richter, H., Chen, G.: Evolutionary Algorithms and Chaotic Systems, vol. 267. Springer, Berlin (2010)
55. Zelinka, I., Davendra, D.D., Šenkeřík, R., Jašek, R., Oplatková, Z.: Analytical programming—a novel approach for evolutionary synthesis of symbolic structures. In: Evolutionary Algorithms, Eisuke Kita, IntechOpen. InTech (2011). <https://doi.org/10.5772/16166>; Available from: <https://www.intechopen.com/books/evolutionary-algorithms/analytical-programming-a-novel-approach-for-evolutionary-synthesis-of-symbolic-structures>
56. Zelinka, I., Oplatková, Z., Nolle, L.: Analytic programming—symbolic regression by means of arbitrary evolutionary algorithms. *Int. J. Simul. Syst. Sci. Technol.* **6**(9), 44–56 (2005)
57. Zickgraf, F.-J.: Kinematical structure of the circumstellar environments of galactic b [e]-type stars. *Astron. Astrophys.* **408**(1), 257–285 (2003)

The Complexity and Information Content of Simulated Universes



Franco Vazza

Abstract The emergence of a complex, large-scale organisation of cosmic matter into the Cosmic Web well exemplifies of how complexity can be produced by simple initial conditions and simple physical laws. Connecting the stunning variety of multi-messenger observations to the complex interplay of fundamental physical processes is an open challenge for Big Data in astrophysics. In this contribution, I discuss a few relevant applications of Information Theory to the task of objectively measuring the complexity of modern numerical simulations of the Universe. When applied to cosmological simulations, the metric of complexity makes it possible to measure the total information necessary to model the cosmic web. It also allow us to monitor which physical processes are mostly responsible for the emergence of complex dynamical behaviour across cosmic epochs and environments, and possibly to improve mesh refinement strategies in the future.

1 Introduction

“I think the next century will be the century of complexity.” Stephen Hawking, Complexity Digest 2001/10, 5 March 2001

“Don’t go on multiplying the mysteries,’ Unwin said. ‘They should be kept simple. Bear in mind Poe’s purloined letter, bear in mind Zangwill’s locked room.’ ‘Or made complex,’ replied Dunraven. ‘Bear in mind the universe.” (Jorge Luis Borges, The Aleph and Other Stories)

The description of physical processes in Nature often calls for the concept of “complexity”, as the reason why achieving a satisfactory and quantitative description

F. Vazza (✉)

Dipartimento di Fisica e Astronomia, Università di Bologna, Via Gobetti 92/3,
40121 Bologna, Italy

Hamburger Sternwarte, Gojenbergsweg 112, 21029 Hamburg, Germany

Istituto di Radio Astronomia, INAF, Via Gobetti 101, 40121 Bologna, Italy
e-mail: franco.vazza2@unibo.it

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021
I. Zelinka et al. (eds.), *Intelligent Astrophysics*, Emergence, Complexity
and Computation 39, https://doi.org/10.1007/978-3-030-65867-0_2

of a particular phenomenon is challenging, or just impossible. Astrophysics and Cosmology make no exception. Complexity is generally regarded as a difficulty inherent to the many degrees of freedom present in a system, or to the difficulty to compute its evolution, e.g. by direct integration of differential equations.

In the present epoch of “Big Data”, driven by ever-growing multi-wavelength observing facilities, a continuous struggle for astrophysicists is the one of connecting the stunning variety of observations to the complex dynamics behind their origin, with the final goal of establishing the underlying physical processes and initial conditions.

This challenge requires the development of new analysis techniques, derived from fields even outside of standard astrophysics, and that can be scaled up to increasingly larger datasets.

The potential of Machine Learning (ML), a branch of Artificial Intelligence, is now being fully explored in astronomy (for a review see [41, 50]). In particular, a very successful approach within ML is the Deep Learning, which is optimal for tasks related to computer vision, text analysis, fragmentation, speech recognition (e.g. [43, 46, 67]). Deep Learning has become popular in astronomy thanks to the availability of computing power to cope with multi-layered neural networks, and to the existence of large enough data sets to perform the training. For recent applications in astrophysics, see for example [1, 31, 48, 51], just to cite a few among many recent applications. The main drawback of the Machine Learning approach is often represented by the lack of flexibility of a trained model, i.e. networks must be designed and trained on a very specific kind of data-sets, and generally loose applicability when turned onto slightly different kind of data. The training phase itself often poses challenges as it requires large datasets with pre-classified (labeled) images, which are not simple to produce even in modern surveys (e.g. [3, 49]).

For the above reasons, it is of outstanding importance for astrophysics also to explore radically different approaches, capable to identify information-rich pattern in real or simulated data sets, without available pre-labeled training sets.

Information Theory (IT) is a powerful and multidisciplinary field of investigation, which enables a mathematical representation of the conditions and parameters affecting the processing and the transmission of information across physical systems (e.g. [32]). According to IT, all physical systems—the entire Universe included—can be regarded as an information-processing device, which computes its evolution based on a software made of physical laws. Thanks to IT, the complexity of a process becomes a rigorous concept, which can be measured and compared, also between different fields of research (e.g. [58]). In IT, not all systems whose evolution is complicated to compute or to predict are truly *complex* in a physical sense. For example, a purely random process does not allow a precise prediction of its future state, yet its future evolution be trivial to compute in a statistical sense. On the other hand, a truly complex phenomenon demands a significant amount of information in order to predict its future evolution, even in a statistical sense.

Our representation of the Universe, based on the ever-growing collection of multi-wavelength telescope observations gives us the image of an arguably very complex hierarchy of processes, spanning an outstanding range of spatial and temporal scales, leaving us with plenty of unanswered questions.

How and when did the cosmic structure come into shape? How did galaxies and the matter connected to them form and shape the Universe we can observe with telescopes? Which processes are fundamental to explain the observed richness of cosmic structures, and which ones can be neglected to the first degree of approximation?

Getting quantitative answers to the above questions is a challenge in which analytical, semi-analytical and numerical methods are struggling since almost a century. Decades of research suggests that large-scale cosmic structures emerged from a hierarchy of interconnected processes, in which several mechanisms (e.g. the expansion of the space-time, gravity, hydrodynamics, radiative and chemical gas processes, etc.) have coupled in a non linear way. Cosmic matter self-organised across an enormous range of scales, transitioning from the smoothest and simplest possible initial condition (a nearly scale-invariant background of matter fluctuations, $\delta\rho/\rho \leq 10^{-5}$, embedded in an expanding space-time, where ρ is the matter density) into a spectacular hierarchy of clustered sources, with a final density contrast of $\delta\rho/\rho \geq 10^4 - 10^5$ (e.g. [24, 37, 53, 69, 85]).

The paradigm of structure formation perfectly fits into the standard definition of how a complex system arises in Nature¹: complexity is often found to emerge from a minimal set of (seemingly simple) initial conditions and physical laws. Moreover, the observable clustering properties of the Universe cannot be predicted just based on its main build blocks alone (e.g. galaxies or dark matter halos), but emerged from the interplay between many components and many scales of interaction.

Numerical simulations are thus the perfect tool to study how a large number of discrete elements can produce complex collective behaviours through their network of interactions.

Which aspects of cosmic structure formation can benefit from complexity analysis?

In the digital representation of our Universe allowed by modern supercomputers, the emergence of complex dynamics out of simple initial conditions is made manifest by the fact that a single random string of a few digits, combined with a source code that can be stored in a few 10^2 Mb (linked to more external numerical libraries and compilers) can produce extremely rich and structured systems, which require tens or hundreds of Terabytes of disk space to be stored. To give a few reference numbers, the widely used Smoother-Particle-Hydrodynamics (SPH) cosmological code GADGET-2 (<http://www.mpa-garching.mpg.de/gadget/>) has a compressed size of ~ 200 kb, but the largest Magneticum simulations (magneticum.org) need ~ 20 Tb to store each of snapshots. The moving-mesh code AREPO (<http://arepo-code.org>) has a compressed size of ~ 12 Mb, but the final raw dataset produced by the Illustris-1 simulation is of ~ 200 Tb. The latest version of the adaptive mesh refinement code ENZO used in this work (<https://code.google.com/p/enzo/>) has a

¹See <https://complexityexplained.github.io> for a recent public repository of resources and visualization tools to explore complexity in physics.

compressed size of ~ 2.1 Mb, but the outputs of the latest Renaissance runs (<https://rensimlab.github.io>) need ~ 100 Tb of disk space.

Growing almost at the same pace of telescope surveys, cosmological simulations continue to produce larger and larger simulations, with the ambitious goal of representing a big fraction of the observable Universe with a high enough resolution to properly resolve galaxy formation (e.g. [86] for a recent review). The largest cosmological simulations have indeed stepped into the regime of evolving $O(10^{11})$ resolution elements, often storing 3-dimensional properties of gas and dark matter dynamics, chemical composition, star-forming properties and magnetic fields attached to each of them. Data mining in such colossal datasets is a challenge, for which standard analysis methods are continuously being deployed and optimised. Also, the preliminary choice of which dynamical scales and volumes are essential to simulate is often a non-trivial one before starting every extensive simulation campaign.

In this respect, the development of efficient and objective tools to measure the emergence of complexity in any numerical model enables simulators to assess which spatial scales are responsible for complex phenomena observed by real telescopes. This in turn allows simulators to deduce which are the relevant scales for minimal working representation of the cosmic dynamics, which is also crucial to match the extensive sampling of cosmic volumes and redshift space that future multi-band surveys of the sky will deliver (e.g. from Euclid to the Square Kilometer Array, e.g. [26]).

There have been valuable attempts to define and study complexity in several physics topics: climate data analysis (e.g. [34]), cellular automata [88], limnology [29], epidemiology [33], compact stars [18], perturbation reconstruction and non-linear signal analysis [25], and many more. However, the application of Information Theory to the structure formation paradigm has just begun [73, 74].

As I will discuss in this contribution (Sect. 3.1), complexity analysis can also be applied at run-time, and be instrumented to cosmological codes, to enable them to refine numerical simulations on the fly, wherever complex dynamical patterns are formed.

Moreover, accretion phenomena responsible for the powering of diffuse non-thermal radio emissions appears to be associated with the growth of complexity in cosmic structures, which makes complexity an interesting additional tool to focus on a key scientific driver of existing and future radio surveys [10, 72, 84]. The fact indeed that non-thermal phenomena and complex evolutionary patterns are closely associated (see, e.g. Sect. 3.2) means that any progress in the understanding of how complexity has emerged in the Universe will concern the same environment that the largest astronomical enterprise of this decade, i.e. the Square Kilometre Array [38, 81, 82], will be mostly devoted to investigate.

2 Information and Complexity: An Overview

In this section, I give a compact overview of the key concepts from Information Theory, whose origin is commonly fixed to the seminal work by [65] and [66], which were concerned on signal processing in communications. The following methods have been applied in [73] and later on in [74] to cosmological simulations. For more general details on the methods, I refer interested readers to the excellent review by [58].

2.1 Shannon's Information Entropy

Information Theory posits that the information content of the outcome a probabilistic process, x (with probability $P(x)$) can be defined as

$$\log_2 \frac{1}{P(x)} = -\log_2[P(x)], \quad (1)$$

measured in *bits* [66]. The latter is known as *information entropy*, and it measures the degree of randomness contained in the process. Therefore, crucial for any attempt to quantify information and complexity is the consideration that any physical phenomenon can be regarded as an information processing device, whose evolution produces a sequence of outputs (e.g. energy states), which can be analysed through symbolic analysis.

The latter approach also implies that a process with many different possible outcomes has high entropy, and that this measure is a proper quantification of how much choice is involved in the realisation of a specific event (i.e. a specific sequence of symbols). Following from this basis, the complexity of a system equals to the amount of information needed to fully describe it. The strict connection between how unlikely is for a specific sequence of events/symbols to occur and the amount of information necessary to describe such sequence is key to any modern description of complexity.

2.2 The Algorithmic Complexity

The minimal information needed to perfectly (i.e. without any loss of information) describe a phenomenon or system is measured through the *algorithmic complexity* (e.g. [15, 39]). In numerical simulations, this is roughly connected to the disk memory necessary to store every single digit produced by the simulation itself, or by the entire source code used for the simulation as well as its initial conditions. Such

representation of complexity introduces some practical problems, which are best explained by thinking to it as to a compression problem² a simple periodic object is characterised by a very little algorithmic complexity as it can be very significantly compressed because the source code to generate a periodic system can be very short (e.g. a cosinusoidal function). However, the algorithmic complexity for a purely random sequence of data can be enormous (i.e. of the order of the sequence itself), due to the lack of internal structure and to the impossibility of further compressing it.

Therefore, this definition of complexity does not entirely capture the basic physical intuition of natural, or artificial phenomena: for example, an arbitrarily long sequence of rand digits has a higher Kolomogorov complexity than the sequence of velocity fluctuations in a turbulent fluid, of the sequence of orbits of planets in the Solar system, or of a symphony. Our physical intuition regards instead all of the above as more “information rich” phenomena than any purely random sequence of numbers. For this reason, alternative approaches to the measure of complexity in natural systems have been developed.

2.3 *The Statistical Complexity*

More relevant from the physical perspective is the quantification of how much information is needed to statistically describe the evolution of a system: this is given by the *statistical complexity* (e.g. [2]). The statistical complexity quantifies the similarity between different realisations of the same process (e.g. starting from different randomly drawn initial conditions) as well as how likely it is that different outputs are drawn from the same process. It also measures the amount of information needed to produce a sequence of symbols which is statistically similar to the original sequence of symbols under study.

At the practical level, the statistical complexity, C_μ , is measured by partitioning the internal states of a system into N_{bin} discrete levels (e.g. internal energy levels), followed by calculating the conditional probability, $P(E_2|E_1)$, that elements in the system evolve from level E_1 into level E_2 going from epoch t to epoch $t + \Delta t$.

The evolution of each element in the simulation is tracked over time, searching for patterns. If an element always gives the same output, its evolution is simple to prescribe, and the statistical complexity is overall small. On the other hand, elements of the system which require a large amount of information in order to prescribe their evolution, are complex.

The probability distribution function $P(E_2|E_1)$ of possible transitions between the states E_2 and E_1 can be traced a-posteriori in the data stream, by building a matrix of all recorded transitions across the simulation’s elements. From the entire matrix of transition probabilities, it is thus possible to calculate the invariant probability

²<http://www.ics.uci.edu/~dan/pubs/DataCompression.html>.

distribution $P(E)$, over the entire sequence of causal states in the system's history, as well as its associated information content as the Shannon entropy of all transitions:

$$C_\mu = - \sum_{E_i} P_i(E_2|E_1) \log_2 P_i(E_2|E_1) [\text{bits}]. \quad (2)$$

where the summation is performed over all computing elements in the simulation at a given time. Each single computing element thus has a statistical complexity given by $C_{\mu,i} = -\log_2 P_i(E_2|E_1)$.

In general, there is no unique way of partitioning the internal energy levels of a specific simulation. The exact partitioning strategy of the system is the result of a compromise between the need of keeping the computing resources under control (as the computation of the statistical complexity scales as $\propto N_{\text{bin}}^2$ (where N_{bin} is the number of energy bins), and the need of resolving all relevant energy transitions between close timesteps.

2.4 The Block Entropy and the Entropy Gain

The probability of observing a specific sequence of symbols, responsible for the generation of a string of length L , is

$$H(L) = - \sum_{x^L \in X^L} P(x^L) \log P(x^L) [\text{bits}] \quad (3)$$

where X^L contains all possible sequences of symbols with length L in the datastream, and it is called *block entropy* (e.g. [45]).

The block entropy is a monotonically increasing function of L (e.g. [17, 27]), and the increase of $H(L)$ is called *entropy gain*:

$$h_\mu(L) = H(L) - H(L - 1). \quad (4)$$

The entropy gain is a good proxy for the intrinsic randomness in a sequence of symbols, because it measures the information-carrying capacity of a string of L symbols, and it also quantifies the internal level of correlation in an evolutionary sequence. In the limit of large L , such metric converges to $H(L)/L$:

$$h_\mu = \lim_{L \rightarrow \infty} h_\mu(L) = \lim_{L \rightarrow \infty} H(L)/L, \quad (5)$$

which is usually called *source entropy rate*.

2.5 The Excess Entropy and the Efficiency of Prediction

The information due to correlation over larger blocks (i.e. due to the intrinsic redundancy of the source of symbols) is the *excess entropy*:

$$E = \sum_{L=1 \rightarrow \infty} [h_{\mu}(L) - h_{\mu}]. \quad (6)$$

The excess entropy can be interpreted as the apparent memory of structure in a source of L symbols (e.g. [64]). E can be further simplified into a finite partial-sum for a length L :

$$E(L) = H(L) - L \cdot h_{\mu}(L). \quad (7)$$

Systems with a large dynamical range allow the observer to describe them on a variety of scales. The *efficiency of prediction*, e , quantifies the scale at which making future predictions of the system gives the best *emergent* and information-rich view [64]:

$$e = \frac{E}{C_{\mu}}, \quad (8)$$

i.e. the ratio between the excess entropy and the statistical complexity. The spatial scale at which e is maximum allows the observer to make the most informative predictions about the future of a system. Indeed, while the excess entropy E uses the past evolution of the system to predict its future evolution, the statistical complexity C_{μ} gives the amount of information necessary to statistically reproduce its behaviour. Therefore, $e = E/C_{\mu}$ can be regarded as a proxy for “how much can we predict” compared to “how much difficult it is for us to predict” about the evolution of a system [58].

3 Results

In the following Sections I will give an overview of the main results concerning the study of the complexity of large-scale structures in the cosmic web using numerical grid simulations and various proxies for complexity, extending first results presented in [73, 74]. Section 3.1 focuses on the analysis at high-resolution of a massive galaxy cluster while Sect. 3.2 presents a more global study of the cosmic web. All numerical simulations used in this work are Eulerian (grid) simulations produced using the cosmological code *ENZO* [14]. *ENZO* is a highly parallel code for cosmological (magneto)hydro-dynamics, which uses a particle-mesh N-body method (PM) to fol-

low the dynamics of the Dark Matter (DM) and a variety of magneto-hydrodynamical (MHD) solvers to evolve the gas component on a support uniform or adaptive grid [14].

3.1 How Complex Is the Formation of a Galaxy Cluster?

Sitting at the top of the mass distribution of cosmic structure, galaxy clusters are key astrophysical objects which form and evolve over cosmological timescales ($\sim 1 - 10$ Gyr). They behave under many respects as “closed boxes”, whose properties are strongly linked to cosmology (e.g. [40]). Given their large volume and overdensity, they usually represent the first detectable signpost of the cosmic web at most wavelengths (e.g. [59, 72]) as a well as the perfect plasma laboratories in the Universe [11].

The simulation presented in this work includes the effect of magnetic fields, radiative cooling of gas and energy feedback from active galactic nuclei in a standard Λ CDM cosmological setup. Adaptive mesh refinement (AMR) was used to selectively increase the dynamical resolution in the formation region of galaxy clusters, up to $\Delta x_{\max} = 32$ kpc (comoving), which is mandatory to start resolving turbulence and magnetic field amplification in galaxy clusters.

I re-simulated the objects studied in here in two different flavors: (a) with a non-radiative setup, only including gravity, hydrodynamics and magnetic fields, or (b) with a radiative setup, including equilibrium gas cooling (assuming a primordial chemical composition) and thermal/magnetic feedback from active galactic nuclei (see Sect. 3.1.2). The magnetic field has been initialised to $B_0 = 10^{-10}$ G everywhere in the box at the start of the simulation. More details can be found in [47, 73, 87].

All physical fields of the simulations were outputted on disk every $\Delta t \sim 3.11 \cdot 10^6$ yr and at the maximum available resolution ($\Delta x_{\max} = 32$ kpc).

3.1.1 The Morphology of Complexity

Figure 1 shows the spatial distribution of thermal, magnetic and kinetic energy for a thin slice crossing the centre of one simulated galaxy cluster (with a final virial mass $M_{100} \sim 3 \cdot 10^{14} M_{\odot}$) at $z = 0.01$.

The thermal and the kinetic energy fields (and to a smaller extent, also the magnetic energy) closely follow the roughly spherically symmetric distribution of gas density, which reaches $\approx 5 \cdot 10^{-3}$ part/cm³ in the cluster core and $\sim 10^{-5}$ part/cm³ in the cluster periphery, which are typical values for galaxy clusters.

The thin slice shown here allows to see smaller scale perturbations associated to the various mechanisms responsible for the mass growth of the cluster (which is still ongoing at $z = 0$).

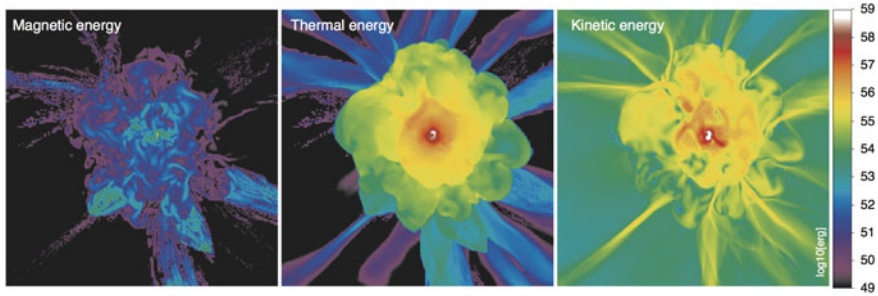


Fig. 1 Magnetic, thermal and kinetic energy for a slice through the centre of a $\sim 3.0 \cdot 10^{14} M_{\odot}$ cluster at $z = 0.01$. The panel is $6 \times 6 \text{ Mpc}^2$ across

Several filaments are connected to the cluster periphery, as well as sharp boundaries, and they penetrate the quasi-spherical envelope of strong shocks (i.e. Mach number $M \geq 10 - 100$) at which the infall gas kinetic energy gets thermalized (e.g. [54, 60]). In such peripheral regions, the flow is predominantly supersonic, with accretion velocities exceeding the local sound speed.

Within the denser cluster atmosphere, the budget of kinetic energy gets smaller than the thermal energy, and velocity fluctuations are due to residual subsonic motions, mostly of turbulent origin (e.g. [20, 35, 77]). The magnetic energy is subdominant everywhere in the cluster volume, and barely reaches a few percent of the thermal/kinetic energy, remaining of order $\leq 10^{-4}$ of the thermal energy in most of the volume.³

The spatial distribution of statistical complexity for the same cluster is given in Fig. 2. In order to compute C_{μ} , I employed $N_{\text{bin}} = 200$ equally spaced logarithmic energy bins, and compared the outputs of two snapshots separated by one root grid time step ($\Delta t \sim 3.11 \cdot 10^6 \text{ yr}$).

The complexity distribution captured by the statistical complexity filtering shows a wide distribution of scales that highlights both large scale complexity pattern associated to several filamentary accretions in the cluster periphery, as well as small-scale fluctuations in the innermost cluster regions. Patches of significant complexity are found in association with large physical jumps of the energy fields, mostly concentrated in narrow zones near shocks (as it can be independently measured with shock finder methods). The thermal energy, on average, requires $\sim 5 - 10$ times more information (reflected in a higher C_{μ}), because the jump of thermal energy in strong shocks is much larger than that of kinetic energy. Although non-radiative numerical shocks obey “simple” Rankine–Hugoniot jump conditions, at every time step only a small fraction of the cells within a specific energy bin (i.e. environment) is modified by

³It shall be remarked that the magnetic field is by far the energy field which is most sensitive to changes in the spatial resolution, because of the strong dependence of the dynamo amplification with the numerical Reynolds number (see discussion in [22]). Due to the limited spatial resolution probed in this simulation, the simulated magnetic field is a factor ~ 10 smaller than suggested by radio observations (e.g. [8]).

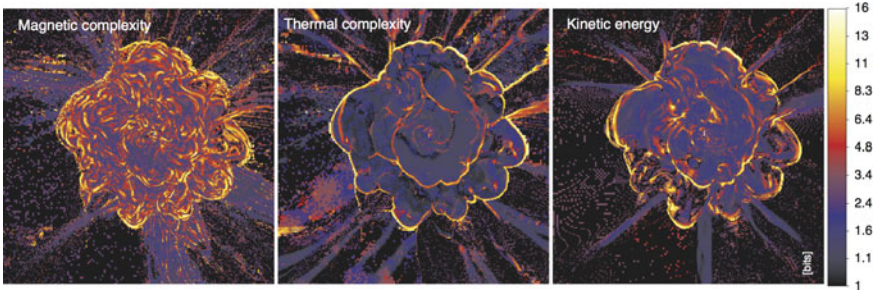


Fig. 2 Magnetic, thermal and kinetic complexity for the same cluster and selection of Fig. 1

shocks. To predict whether or not a specific cell is going to be affected by shocks (or more in general how its energy should evolve) additional information is required to solve the local Riemann problem (on average, of the order of ~ 10 bits/cell here).

Compared to the thermal energy, the kinetic energy E_K shows more complex fluctuations away from shocks, as well as closer to the cluster centre. The ICM is known to host volume filling subsonic turbulence at all epochs, as result of gravity-driven random motions (e.g. [77]); thus most of such fluctuations are due to turbulence on the short timescale which separates the two time steps used for this analysis. Finally, the magnetic energy is sub-dominant compared to the thermal/kinetic energy of the ICM. However, this means that one needs more information in order to predict the evolution of magnetic fluctuations. The ultimate driver of its evolution is indeed turbulence, through small-scale dynamo amplification (e.g. [21, 89]): this makes the evolution of the magnetic typically more complex at all scales and distances from the cluster centre. Very similar results were found for all simulated galaxy clusters using *ENZO*, provided that small differences in the dynamical state and in the shock history of each object are reflected in the final distribution of complexity (e.g. [73, 74]).

Why is complexity useful to simulate the formation of galaxy clusters

If complexity can be measured at run-time while the simulation is running, this information can allow the simulation code to identify exactly where and when complex evolutionary patterns are emerging in the domain. Coupled with adaptive mesh refinement, this approach can selectively increase the local force and spatial resolution at run-time. Traditionally, this task is performed by fixing a-priori some relevant threshold values for the combination of several quantity of interest (e.g. matter overdensity, local jumps in thermodynamical quantities, etc.), and letting the simulation reduce (typically, by halving) the local mesh resolution whenever such threshold values are exceeded. Examples of this include refining on the local matter overdensity (e.g. [13, 68]), on the velocity shear [42], on gas vorticity [35], on 1-dimensional velocity jumps tracking

shocks [78], on magnetic field intensity [89], etc. Each of these choices depends on the simulator's prior knowledge and expectations about the physics in the simulation, and it is thus biased to refine on behaviours that can be predicted (or at least guessed) before the run begins. Moreover, dedicated tests have shown that the employment of too aggressive AMR techniques introduces un-physical perturbations to simulated systems (e.g. [63]).

The possible application of statistical complexity, as a new method to trigger new refinements during the simulation, can offer an unbiased way of improving calculations, independently on the observer's prior expectations on the problem under study.

Moreover, the usual workflow requires first to run low or moderate resolution versions of such simulations, and to apply increasingly aggressive AMR in a second step, by restarting the first run from an interesting epoch. Information Theory thus allows a more elegant solution to this challenge (which can be a challenge for large simulations), because statistical complexity only relies on the symbolic analysis of the data-stream generated by the simulation, without any a-priori knowledge of what is in the data, nor of the relevant physical threshold to exceed.

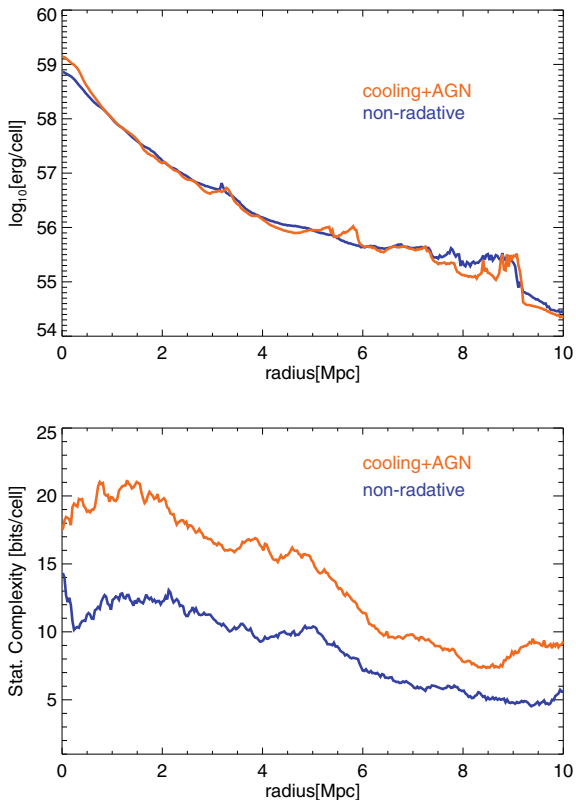
3.1.2 The Impact of Galaxy Formation on Cluster Complexity

The impact of galaxy formation physics on the dynamical evolution of the intra-cluster medium (namely radiative cooling leading to the collapse of overdense gas clumps and their later feedback on the surrounding gas distribution via feedback) is a wide field of research for numerical simulations, as it significantly modifies the (self-similar) scaling relations between the total mass of clusters and their thermodynamical or observable properties (e.g. [52, 71]).

I investigated the role of galaxy formation on the emergence of complexity with variations of the physical model discussed in the previous section. In detail, radiative re-simulations of the same galaxy clusters include equilibrium gas cooling from primordial chemical composition, and effective model for the (large-scale) energy release by AGN bursts in the course of the simulation. In this case, each feedback event (triggered whenever the gas density within a cell exceeded 10^{-2} part/cm³) releases 10^{60} erg of thermal energy and 10^{59} erg of magnetic energy (as bipolar structure). The above simplistic model bypasses the problem of following prohibitively small scales involved in the accretion of gas onto super massive black holes. However, it has been shown to adequately reproduce the thermodynamical properties of the observed ICM on ≥ 100 kpc scales (e.g. [75]).

The combined effect of cooling and AGN feedback on simulated clusters is typically to increase the gas density in the core, to produce large transients in gas temperature, to promote the significant expulsion of baryons from the innermost

Fig. 3 Top panel: radial profile of the average total (thermal, kinetic and magnetic) energy for a simulated galaxy cluster at $z = 0.01$, in a non-radiative setup or in a model including radiative cooling and AGN feedback. Bottom panel: radial profile of the average statistical complexity of all energy fields for the same two resimulations



cluster regions, as well as to introduce more turbulence and clumpiness in the ICM (see discussion in [75]).

As an example, Fig. 3 shows the average radial distribution of the total energy (kinetic, thermal and magnetic) for the gas in the simulated cluster of Fig. 1 at $z = 0.01$. On the one hand, the overall radial energy distribution is similar to the non-radiative re-simulation, but the radial profile shows more substructures (due to the enhanced clumping of gas) as well as a higher energy budget in the cluster core (as a mixed effect of gas compression and extra heating from the central AGN). A more detailed analysis of re-simulations with AGN feedback can also be found in [73]. Overall, despite the additional physics included in the simulation, the volumetric distribution of all fields is quite similar for ≥ 1 Mpc radii, indicating that the equilibration of non-gravitational perturbations within the cluster atmosphere is overall efficient enough to erase most of the signatures from AGN across the entire cluster volume.

Remarkably, the radial distribution of total statistical complexity (bottom panel of Fig. 3) shows large differences across the entire radial extent, out to ~ 10 Mpc, leading to an overall $\sim 50 - 100\%$ increase of C_μ everywhere. All energy fields show an almost equally increased complexity, with a larger difference in all cases for the

innermost cluster regions. The fact that more complexity is found even at large radii suggests that the extra complexity is not entirely due to the central AGN in the cluster, but that it probably was contributed by the activity of several AGN in the volume, and/or by volume filling processes produced before the cluster was fully assembled.

Why is complexity useful to study the evolution of intergalactic gas

When and how did the extra complexity in the cluster arise?

The block entropy analysis introduced in Sect. 2.4, is a monotonically increasing parametrisation of the information content of a system, hence it is very suitable for time-integrated studies, in which past and recent events can be dynamically related.

Following [73], I computed the block entropy, $H(L)$, and its source rate term, $h_\mu(L)$, for the entire sequence of kinetic, thermal and magnetic energy from $z = 30$ to $z = 0$ within a volume centre on the formation region of a cluster. The full analysis of the sequence of symbols (e.g. energy levels) computed by the simulation requires a huge amount of data (e.g. ≥ 2 Tb to follow all cells in the simulation at high resolution for the entire sequence of 440 root grid timesteps), hence the analysis is here restricted to a small representative data set in cluster formation region, comprising $\sim 1.6 \cdot 10^5$ cells. More optimised algorithms will need to be developed, in order to process the entire data flow of existing and future large simulations. While the absolute value of block entropy at a specific epoch may depend on the volume being investigated (as well on specific choices of the binning of energy levels and on the time sampling frequencies) the relative growth of block entropy in the energy fields is more robust to model variations (e.g. see [73] for a discussion).

Figure 4 (top panel) gives the evolution of the total energies contained in the selected sub-region in the cluster, from $z = 30$ to $z = 0$. By the end of the simulation, the total thermal and kinetic energy of gas in the cluster volume are nearly identical, as implied by the radial profile while in the first half of the simulation (and before the efficient heating by AGN) the thermal energy in the radiative simulation is lower, due to the effect of radiative losses on the densest clumps in the region. On the other hand, the magnetic energy in the same region is larger in the radiative run, due to the combined effect of gas compression (induced by cooling) and of the additional magnetisation induced by AGN feedback.

The middle panel of Fig. 4 gives the evolution of block entropy for the kinetic, thermal and magnetic energy of both re-simulations as a function of time. In this case, larger differences are visible: the block entropy increases in a monotonic way, as expected, reaching $\langle H(L) \rangle \approx 22.5$ bits/cell in the non-radiative case and $\langle H(L) \rangle \approx 28.1$ bits/cell in the radiative run. Despite the similar thermal and kinetic energy distribution across most of the simulation, the complexity of all energy fields is consistently larger in the radiative run.

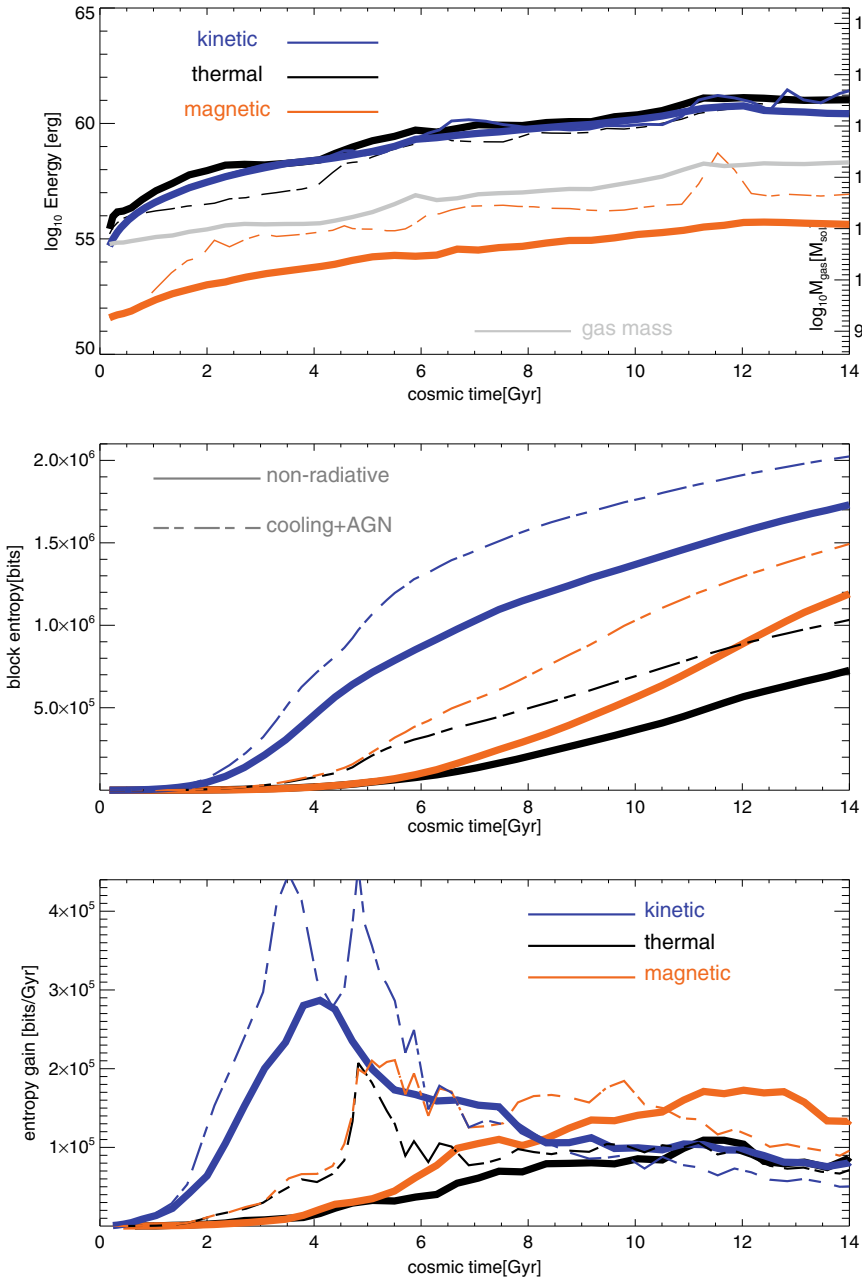


Fig. 4 Top panel: evolution of the kinetic, thermal and magnetic energy fields (as well as of the gas mass) for a sample of $1.9 \cdot 10^5$ cells in the central region of the galaxy cluster of Figs. 1 and 2. The thick lines refer to the non-radiative simulation while the thin lines are for the radiative simulation including feedback from AGN. Central panel: evolution of block entropy for the same selection of cells. Bottom panel: evolution of the entropy gain for the same selection of cells

Cooling and feedback have overall a little impact on the relative complexity of the energy fields after the cluster assembled, for $t \geq 4$ Gyr, and their role is more evident at earlier times. Longer before contributing to the mass of the cluster in this region at late epochs, the cosmic gas in the simulation was subject to a complex dynamical evolution of all fields, long before there was (approximate) equilibrium between the forming gravitational well of the cluster and the thermal gas energy.

The entropy gain, $h(L)$ (bottom panel of Fig. 4) better illustrates when and how complexity gets increased in the two runs: shortly after mergers and matter accretions experienced by the cluster (e.g. see the spike in gas mass at $t \sim 6$ Gyr), as well as after AGN bursts inducing outflows when this is included (e.g. bursts at $t \sim 3$ Gyr, $t \sim 5$ Gyr). Since the local fluctuations driven by AGN in all fields are more violent than in mergers, the entropy gain also is manifestly more significant after AGN bursts. As a consequence, the largest spikes in entropy gain are reached well before the cluster has fully assembled, i.e. for $t \leq 5$ Gyr, when its gas mass was $\leq 10\%$ of its $z = 0$ value and a large fraction of infall kinetic energy still had to be virialized.

This test well illustrates the power of Information Theory applied to astrophysical simulations, in which several different mechanisms operate together: complexity analysis can detect and expose large differences related to the underlying complexity of the adopted physical models being tested, even when detecting such differences is difficult with standard analysis. For example, the impact of AGN on the kinetic perturbations away from cluster cores is hard to detect in simulations (e.g. [23, 36, 56, 75]), owing to the rather fast dissipation of turbulent motions in the ICM).

Which scales contains most information?

The volume of galaxy clusters is large, and it comprises so many different spatial scales that it is natural to ask whether there is there a preferred scale at which the emergence of complexity is maximum, and which is the scale that contains the most information on the evolution of such systems.

In [73] I studied the evolution of energy fields in $\approx 1.9 \cdot 10^5$ cells in the centre of a forming galaxy cluster, computing their average $H(L)$ and $E(L)$, for different levels of linear coarse-interpolation of the data, from the coarsest $\Delta x_{max} = 634$ kpc resolution to finest $\Delta x_{min} = 32$ kpc one. Figure 5 gives the trend of efficiency, e , measured in the galaxy cluster simulation analysed above, depending on the adopted interpolation scale, which displays a similar trend for all energy fields.

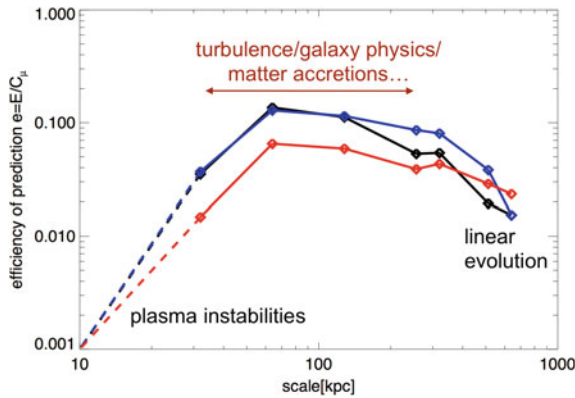


Fig. 5 Relation between the efficiency of prediction (Sect. 2.5) at different interpolation scales, for sample of $1.9 \cdot 10^5$ cells in a forming galaxy cluster. The dashed lines connects the values of e measured in the simulation, with the e values estimated from plasma physics on unresolved scales (see text for explanation). The black line is referred to the thermal energy, the blue and the red to the kinetic and magnetic energy, respectively

The maximum e is found in the range of $\sim 63 - 190$ kpc, with $e \approx 0.1 - 0.2$. This range of scale is the typical one of turbulent eddies in the simulated ICM (e.g. [62, 83]), of the typical outer correlation scale of observed and of simulated magnetic fields in the ICM (e.g. [8, 89]), as well as of measured projected density fluctuations in X-ray (e.g. [30]). It appears therefore reasonable that e is the largest on where the ICM presents the highest degree of dynamical self-organisation, which is also routinely targeted by telescope observations at different wavelengths. On the other hand, the coarse-grained evolution on much larger scales allows a more robust prediction of future evolution as on such scales the evolution approaches the linear regime of small density perturbations. However, this also makes such evolution relatively easier to predict, making the E/C_μ ratio lower than for smaller scales.

This simulation cannot probe scales $\ll \Delta x_{min} = 32$ kpc, however a few basic considerations suggest that e should decrease again for such “microscopic” scales. The efficiency of prediction at these scales can be estimated by considering that the dynamics of thermal gas in the ICM can be assumed to be a Markovian process, which means that the thermodynamic value of single particles only depends on their last micro-state. Hence $E = C_\mu - Lh_\mu \approx C_\mu - h_\mu$ because $L \approx 1$ (e.g. [58, 64]).

In this regime, the thermodynamic entropy also gives the statistical complexity, which for the thermal particles of the ICM is $S \sim 10$ keV/particle (e.g. [9]). On the other hand, the source entropy rate crucially depends on how energy is exchanged between particles on very small scales. The ICM plasma is expected to be weakly collisional on these scales, hence energy gets mostly exchanged

by collective plasma effects (including a wide range of possible plasma instabilities) acting on \sim seconds timescales (e.g. [12, 44]). In this scenario, the extremely fast action of plasma collective motions implies that on microscopic scales the efficiency of prediction is ≈ 1 only in the scale of *seconds*, while it must rapidly drop to zero for any other longer timescale. Even in the rather standard (and probably out-dated) model of a collisional ICM [61], where Coulomb collisions between particles solely exchange energy, the entropy rate can be estimated to $h \approx 10^{-7}$ keV/particle/yr based on the expected proton-proton Coulomb collision frequency, implying that $e \approx 1$ only for $\leq 10^5$ yr timescales.

In summary, a detailed thermodynamic view of single particle interactions in the ICM appears to irrelevant to predict the evolution of the ICM on any astronomically relevant scale, given the enormous difference in scale between microscopic and macroscopic processes involved. Collective processes emerge on \geq kpc scales, which are routinely observed by telescopes and usually are simulated with numerical simulations represent the best range of scales at which the “emergent” properties of the ICM are evident, and where the evolution of such systems can be effectively described using a (magneto) hydrodynamical model. In particular, the $\sim 50 - 200$ kpc range of scales appears to be the one that maximises the efficiency of prediction in the investigated cosmological simulations, and that is used in the subsequent investigation of complexity in the entire cosmic web.

3.2 How Complex Is the Formation of the Cosmic Web?

With a different set of numerical simulations, I measured the distribution of complexity in a full cosmological volume $(40 \text{ Mpc}/h)^3 \approx 57^3 \text{ Mpc}$, simulated with *ENZO* using 400^3 cells and DM particles, at the constant resolution of $\Delta x = 141 \text{ kpc}/\text{cell}$. With this setup, I investigated several variations of gas physics and of cosmological parameters in order to assess their impact on the emergence of complexity, as detailed in [74]. All simulations employed the numerical MHD scheme of Dedner [19] as in the previous case, with a magnetic field initialised to be $B_0 = 0.1 \text{ nG}$ (comoving) along all magnetic field components at the begin of the simulation ($z = 40$).

The choice of this spatial resolution was motivated by the measured trend of the efficiency of prediction Sect. 3.1.2, which ensures the most “information rich” view of the emerging complexity of the cosmic web.

The statistical complexity, C_μ , is here measured as in [79] (and similar to the previous section), by employing equal logarithmic energy bins, ranging from the maximum and the minimum of each energy field, respectively, and considering a time spacing of $dt = 5$ timesteps ($\approx 200 \text{ Myr}$) between snapshots.

A couple of visualisation examples of C_μ for the full cosmological volume, and of its spatial relation with the entire cosmic web on scales much larger than the ones probed in the previous cluster analysis (Sect. 3.1) are given in Figs. 6 and 7.

Figure 6 give a 2-dimensional view of the distribution of the thermal and magnetic energy fields for a thin slice through the simulation at $z = 0$, and the corresponding distribution of statistical complexity for the same volume. A 3-dimensional rendering of the entire simulated volume at the same epoch is given in Fig. 7, which shows the total gas density in red, the gas temperature in blue, and additionally the total complexity (in green) in the right panel.

While the large-scale distribution of all energy fields closely trace the matter distribution of the cosmic web and its related gravitational potential, with maxima located in self-gravitating matter halos, the spatial distribution of complexity appears

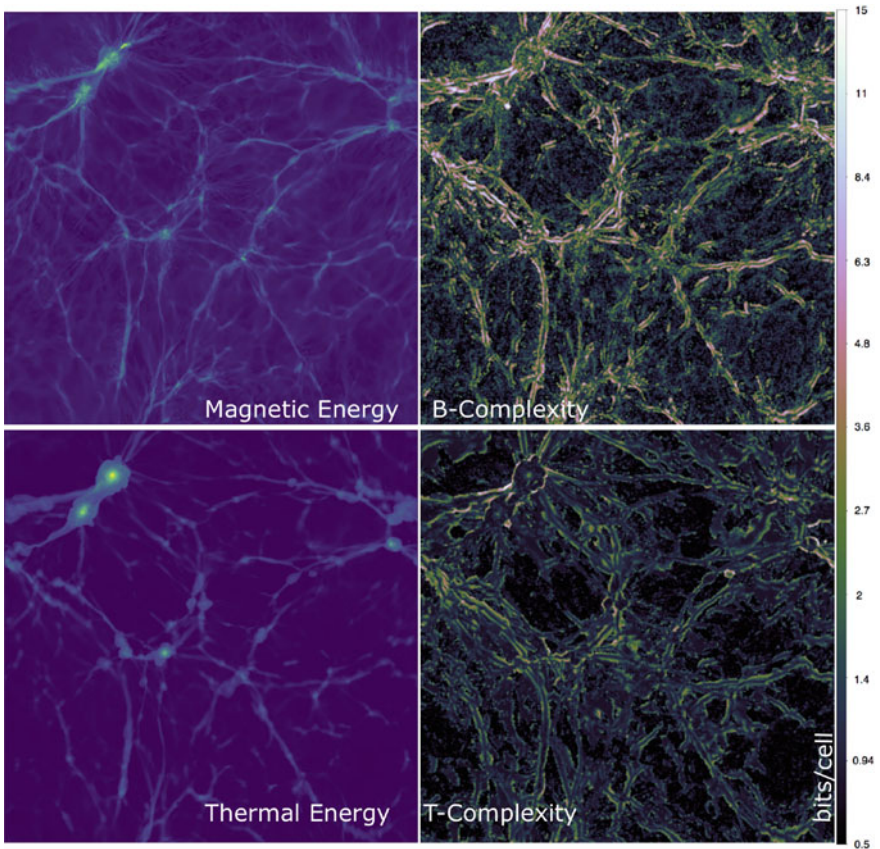


Fig. 6 Slices through a 57^3 Mpc^3 simulated volume at $z = 0$. The top panels show the magnetic energy (left) or the magnetic complexity (right), the lower panels show the thermal energy (left) or the thermal complexity (right)

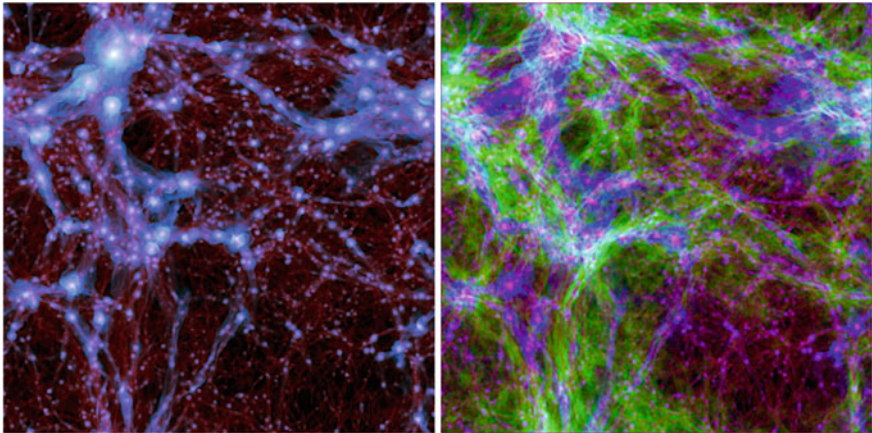


Fig. 7 Left panel: total gas density (red) and gas temperature (blue) for the 57^3 Mpc^3 simulated volume at $z = 0$. Right panel: total complexity (green) over imposed to the previous map

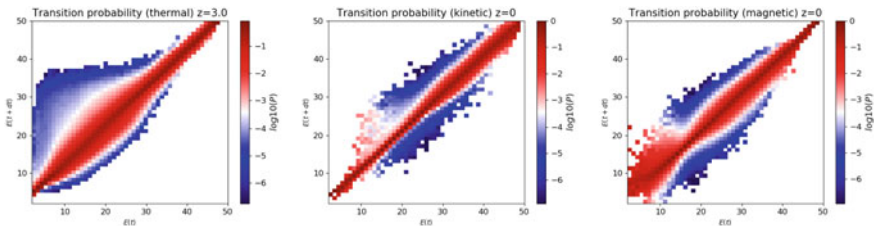


Fig. 8 Matrix of transition probabilities, P_{xyz} measured between the energy states at timesteps t and $t + dt$, considering transitions of thermal (left), kinetic (centre) and magnetic (right) energy at $z = 0.0$

broader. This means that, across the full range of cosmic environment, regions with significantly different energies may have an equally complex evolution, depending on their local dynamics and past history. The visual inspection also shows that prominent spikes of complexity are associated with shocks, marked as sharp contours around filaments or at the periphery of halos in the volume, in line with the previous section.

The complexity in different environments directly follows from the transition probability matrix (Sect. 2.3) across the entire range of cosmic overdensities. For example, the 3-dimensional structure of the complexity traces shocks around filaments and massive halos, for which a large spread in $P(E_2|E_1)$ can be expected. Figure 8 gives the transition probability matrix, $P(E_2|E_1)$ (see Sect. 2.3) measured in the full volume at $z = 0.0$ and referred to 50 logarithmic energy bins. Here the diagonal 1-to-1 relation corresponds to little complexity transitions, in which $E_i(t)$ states are mapped onto the same level at the following timestep. On the other hand, a large spread around the 1-to-1 correlation hints at complex transitions which require

more information to predict, like large thermodynamic jumps associated with shocks (see also Sect. 3.1.1).

Indeed, most energy levels in the intermediate range are spread in the probability distribution. This is consistent with the fact that strong structure formation shocks typically change the energy content of gas particles in the linear overdensity regime, with $T \sim 10^4$ K (mostly related to the most filamentary part of the cosmic web), even on the short timescale of the simulation timestep. On the other hand, as already observed in Sect. 3.1.1, strong shocks are able to cause only smaller transitions of kinetic and magnetic energy levels within cells, following from shock jump conditions. Across most environments, we can observe a spread of magnetic energy levels in the probability matrix, mostly associated with the fact that the magnetic field can be changed both by compression or by magnetic field amplification, via small-scale dynamo—albeit at a rate limited by the modest spatial dynamical range that is achieved by turbulence and dynamo in this simulation (e.g. [22]).

The distribution of complexity as a function of the cosmic environment is better quantified by the phase diagrams in Fig. 9, which give the average statistical complexity of energy fields, for the reference epochs of $z = 3$ and $z = 0$ as a function of ρ and T . The range of $n/\langle n \rangle \geq 10 - 100$ approximately marks groups or clusters of galaxies, while $T \leq 10^4$ K marks cosmic voids. Intermediate ranges of values are the location of linear or mildly non-linear structures of the cosmic web, i.e. matter sheets and filaments.

From the phase diagrams, we can observe that the peak of complexity in the cosmic volume moves across the environment as a function of time.

At $z = 3$, most of halos in the simulation are still being assembled, leading to a significant conversion of infall kinetic energy into thermalisation and magnetic field amplification, prior to the establishment of (approximate) hydrostatic equilibrium. The complexity thus peaks for the high densities at which the conversion of infall kinetic energy is more prominent, i.e. in halos undergoing mass growth, typically reaching $C_\mu \sim 10^2$ bits/cell.

Later on, it is the periphery of galaxy clusters or filaments to become the most complex environment in the volume, with $\langle C_\mu \rangle \geq 10 - 10^2$ bites/cell. Conversely, the internal volume of halos becomes less complex ($\langle C_\mu \rangle \sim 10$ bits/cell) by $z = 0$, because only rare and extreme perturbations (e.g. major mergers) can change pre-existing energy levels by a large amount. Despite the significantly different numerical resolution, these trends are in line with the highest resolution of galaxy clusters view previously discussed in Sect. 3.1.

Finally, the average complexity is measured to be small in voids, $\langle C_\mu \rangle \ll 10$ bits/cell, due to their relatively simple evolution, mostly ruled by adiabatic gas expansion. Nevertheless, a residual amount of complexity is also found in low density regions, resulting from the expansion of the structure formation shocks released during the very first stage of halo formation, and still expanding into lower densities.

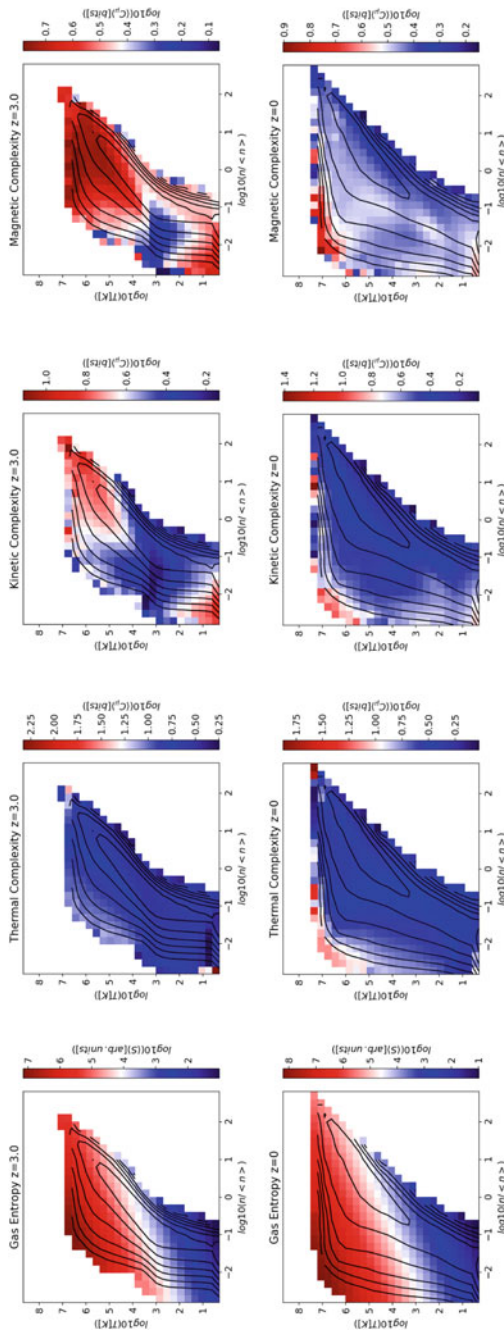


Fig. 9 Phase diagrams showing the average complexity of the thermal, kinetic and magnetic energy at $z = 3$ and at $z = 0$ for the simulated cosmic web. The first column shows the average gas entropy for the same boxes. The grey contours in each panel show the gas density distribution (with logarithmic spacing of contours)

The first column of Fig. 9 additionally shows the phase diagram of gas entropy ($S \propto T/\rho^{2/3}$), which overall is similar to the one of thermal complexity. The latter is not surprising considering that there is indeed a duality between entropy and Shannon information (as defined in Eq. 1), which stems from the basic definition of entropy in statistical thermodynamics:

$$S = -k_B \sum P_i \log P_i \quad (9)$$

(where $P_i \propto e^{\frac{\epsilon_i}{k_B T}}$ is the probability of the energy state ϵ_i with temperature T , in a Boltzmann distribution).

In the non-radiative simulations considered here, the gas entropy in the volume is only increased by the irreversible dissipation at shocks, or by spurious numerical effects. The fact that the high-temperature envelope of the phase diagrams corresponds to the location of maximum entropy production and maximum complexity, confirms that dissipative processes related to structure formation are indeed the main agents of emerging complexity for intergalactic gas in the simulated Universe.

How can complexity analysis measure different cosmologies?

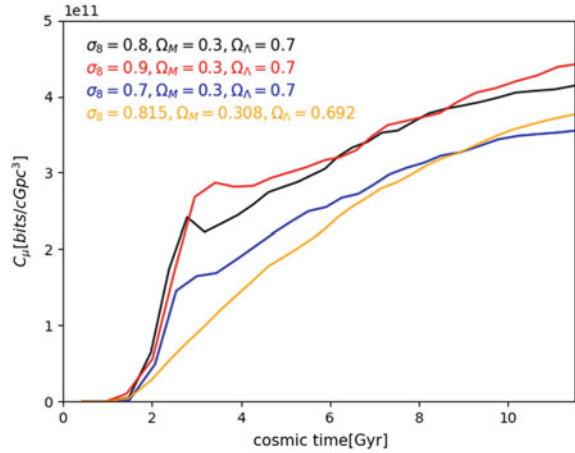
Thanks to statistical complexity, it is also possible to investigate whether even small variations on the set of cosmological parameters can affect the level of complexity in a simulated universe. In detail, here I restrict the analysis to a few relevant variations of the baseline concordance Λ CDM model (see Fig. 10 for the detail on assumed parameters) while in [74] I have also explored the simpler CDM case.

Figure 10 gives the total statistical complexity for gas residing in the cosmic web, $T \geq 10^4$ K (normalised to a comoving Gpc^3 volume) as a function of time for four different resimulations of the same volume considered above, for different variations of the σ_8 parameters, or the fiducial set of parameters from PLANCK [55].

The difference between all Λ CDM models is overall tiny ($\leq 30\%$) at most epochs and for all energies. There clearly is a dependence between the amplitude of σ_8 (which indicates the amplitude of the initial matter power spectrum within a reference scale of 8 comoving Mpc) and the final complexity, with a quite regular increase of complexity going from $\sigma_8 = 0.7$ to $\sigma_8 = 0.9$.

This follows from the fact that σ_8 is known to correlate with the rate of structure formation, and hence with the frequency of perturbations to the gas, driven by mergers. A higher σ_8 implies that the collapse of self-gravitating halos can begin earlier in time, and also more massive substructures within halos are present at all redshifts (e.g. [40]). In the case of a high σ_8 , also more shocks are launched in the cosmic volume by the infall of gas matter (see Appendix of [76]). All these factors make the high σ_8 Universe slightly

Fig. 10 Evolution of the total complexity in the cosmic web, for four different variations of cosmological parameters for the same volume



more complex than a low σ_8 one, at all epochs. In [74] I have also shown that complexity analysis can easily detect a large difference between Λ CDM and CDM models, even if the CDM model is calibrated to reproduce approximately the same number of halos in the volume. While the total number of clusters forming in the volume approximately scale as $\Omega_M \sigma_8^0.5$ (e.g. [59]), the growth of cluster is significantly delayed in the CDM cosmology (e.g. [7]), due to the lower σ_8 (0.43 in this case), and complex pattern driven by matter accretions in halos only emerge at later times.

4 Conclusions

In the present epoch of Big Data in astrophysics, as a result of existing and future multi-messenger surveys (as well as by increasingly more sophisticated numerical simulations), the tasks of identifying the complex chain of processes which lead to observed astronomical phenomena is an open challenge, calling for new and powerful analysis approaches.

In this contribution, I have discussed how new statistical tools based on Information Theory (e.g. [2, 28, 58]) may allow us to objectively measure the level of complexity of modern astrophysical simulations, as well as to identify patterns and sequences of events otherwise impossible to identify with standard approaches.

Complexity enables us to identify which physical processes are mostly responsible for the emergence of observed complex dynamical behaviour across cosmic epochs and environments, and possibly to improve numerical refinement strategies in future simulations attempting to reproduce the Universe.

With this method, I have shown that the complexity of cosmic structures has emerged early in time, when most of seeds of halos in the cosmic web started to collapse and convert their gravitational infall energy into thermal energy and magnetic field. The process is mostly mediated by violent fluid perturbations, often in the form of strong shocks and turbulent motions. On smaller scales, and before the formation of halos, the activity connected with the formation of galaxies (e.g. radiative gas cooling and feedback from active galactic nuclei) can introduce more complexity to the evolution of baryons in the cosmic web, which can be identified even at later simulated epochs.

It must be noticed that the concept of complexity used in this work is a dynamical, rather than a geometrical/topological one; the latter approach has been instead explored in works that studied cosmic structure using Minkowski functionals and Betti numbers as a proxy for topological persistence of structures (e.g. [57] and references there in).

In passing, we can also remark that the dynamical view of complexity exposed here is different from the definition of *maximum* information usually adopted by the holographic description of the Universe (e.g. [6, 70]), which yields the astounding maximum information capacity of $\sim 10^{100}$ bits which can be stored in the entire fabric of space-time using all available Planck lengths thereby contained (e.g. [5]).

Likewise, a similar astounding difference of orders of magnitude is found by comparing the theoretical maximum computing speed of all matter enclosed by large-scale structures, and their effective computing speed on the large scales considered here. The (classic) Bremermann's limit (which limits to $\approx Mc^2/h \approx 10^{47}$ bits/(s · g) the maximum processing rate of any physical system) allows a maximum of $\sim 10^{95}$ bits/s processing rate for the most massive galaxy clusters in the Universe, while the entropy gain computed in Sect. 3.1.2 can be extrapolated to $\sim 10^{28} - 10^{30}$ bits/s, depending on the different phases of cluster growth.

Based on the complexity measured in my simulation [74], it is possible to extrapolate the total statistical complexity within the entire observable Universe:

$$C_{\text{Universe}} = 4\pi \int_0^\infty \langle C_\mu(z) \rangle r^2 \frac{dr}{dz} dz \approx 3.56 \cdot 10^{16} \text{ bits} \approx 4.3 \text{ Pb.} \quad (10)$$

where the integration is done using the redshift evolution of C_μ measured in the fiducial Λ CDM simulation, and dr/dz has been measured as a function of redshift and of the cosmological model (e.g. [16]). Interestingly, this is of the same order of the total amount of data daily generated by social media,⁴ and this is also similar to the latest estimates of the maximum memory capacity of the human brain, which follows from the extrapolation of the information that can be stored by synaptic plasticity [4].⁵

⁴<http://res.cloudinary.com/yumyoshoin/image/upload/v1/pdf/future-data-2019.pdf>.

⁵See also Vazza & Feletti [80] for a recent semi-quantitative comparison between the structural properties of the cosmic web and of the human neuronal network.

The above memory capacity estimate represents the minimum amount of information required to describe the evolution of the entire visible cosmic web with a spatial detail of $\sim 10^2$ kpc, which is the one that maximises the efficiency of prediction (Sect. 2.5).

This information can be crucial to define the optimal approaches for future simulations, aiming at matching the sky and redshift coverage of incoming wide and deep multi-band surveys of the sky wavelengths (e.g. from Euclid to the Square Kilometre Array), which will define the future of astronomical data.

The expected flurry of complex data produced by such surveys will keep challenging simulators to produce Universes containing an equal amount of complexity. The new analysis methods offered by Information Theory, and described in this work, promise to offer simulators with objective tools to embark on this exciting challenge that awaits in the near future.

Acknowledgements I acknowledge financial support from the ERC Starting Grant “MAGCOW”, no. 714196. The cosmological simulations were performed with the ENZO code (<http://enzo-project.org>), which is the product of a collaborative effort of scientists at many universities and national laboratories. The simulations on which this work is based have been produced on the Jülich Supercomputing Centre (JFZ) under project HHH42 and *stressim*, as well as on Marconi at CINECA (Bologna, Italy), under project INA17_C4A28 (in all cases with F.V. as Principal Investigator). I also acknowledge the usage of online storage tools kindly provided by the INAF Astronomical Archive (IA2) initiative (<http://www.ia2.inaf.it>).

References

1. Abraham S., Aniyan A.K., Kembhavi A.K., Philip N.S., Vaghmare K.: MNRAS (2018)
2. Adami, C.: BioEssays **24**, 1085 (2002)
3. Alger, M.J., et al.: MNRAS **478**, 5547 (2018)
4. Bartol, T.M., Bromer C., Kinney J., Chirillo M.A., Bourne J.N., Harris K.M., Sejnowski T.J.: ELIFE **4** (2015)
5. Bekenstein, J.D.: Sci. Am. **289**, 58 (2003)
6. Bekenstein, J.D.: Contemp. Phys. **45**, 31 (2004)
7. Bode, P., Bahcall, N.A., Ford, E.B., Ostriker, J.P.: ApJ **551**, 15 (2001)
8. Bonafede, A., Vazza, F., Brüggem, M., Murgia, M., Govoni, F., Feretti, L., Giovannini, G., Ogrean, G.: MNRAS **433**, 3208 (2013)
9. Borgani, S., Diaferio, A., Dolag, K., Schindler, S.: Sci. Space Rev. **134**, 269 (2008)
10. Brown, S., Emerick, A., Rudnick, L., Brunetti, G.: ApJL **740**, L28+ (2011)
11. Brunetti, G., Jones, T.W.: Int. J. Mod. Phys. D **23**, 1430007 (2014)
12. Brunetti, G., Lazarian, A.: MNRAS **412**, 817 (2011)
13. Bryan, G.L., Norman, M.L.: ApJ **495**, 80 (1998)
14. Bryan, G.L., et al.: ApJS **211**, 19 (2014)
15. Chaitin, G.J.: 1995, in eprint [arXiv:chao-dyn/9509014](https://arxiv.org/abs/chao-dyn/9509014), p. 9014
16. Condon, J.J., Matthews, A.M.: Publ. Astron. Soc. Pac. **130**, 073001 (2018)
17. Crutchfield, J.P., Feldman, D.P.: Chaos: Interdiscip. J. Nonlinear Sci. **13**, 25 (2003)
18. de Avellar, M., Horvath, J.: Phys. Lett. A **376**, 1085 (2012)
19. Dedner, A., Kemm, F., Kröner, D., Munz, C.-D., Schnitzer, T., Wesenberg, M.: J. Comput. Phys. **175**, 645 (2002)
20. Dolag, K., Vazza, F., Brunetti, G., Tormen, G.: MNRAS **364**, 753 (2005)

21. Domínguez-Fernández, P., Vazza, F., Brüggén, M., Brunetti, G.: *MNRAS* **486**, 623 (2019)
22. Donnert, J., Vazza, F., Brüggén, M., ZuHone, J.: *ArXiv e-prints* (2018)
23. Dubois, Y., Devriendt, J., Teyssier, R., Slyz, A.: *MNRAS* **417**, 1853 (2011)
24. Efstathiou, G., Davis, M., White, S.D.M., Frenk, C.S.: *APJS* **57**, 241 (1985)
25. Ensslin, T.A.: *Phys. Rev. E* **87**, 013308 (2013)
26. Farnes, J., Mort, B., Dulwich, F., Salvini, S., Armour, W.: *Galaxies* **6**, 120 (2018)
27. Feldman, D.P.: Department of Physics, University of California, July (1997)
28. Feldman, D.P., Crutchfield, J.P.: *Phys. Lett. A* **238**, 244 (1998)
29. Fernandez, N., Maldonado, C., Gershenson, C.: *ArXiv e-prints* (2013)
30. Gaspari, M., Churazov, E.: *A & A* **559**, A78 (2013)
31. Gheller, C., Vazza, F., Bonafede, A.: *MNRAS* **480**, 3749 (2018)
32. Glattfelder, J.B.: *A Universe Built of Information*, pp. 473–514. Springer International Publishing, Cham (2019)
33. Grassberger, P.: *J. Stat. Phys.* **153**, 289 (2013)
34. Hoffman, F.M., et al.: *Procedia Comput. Sci.* **4**, 1450 (2011)
35. Iapichino, L., Niemeyer, J.C.: *MNRAS* **388**, 1089 (2008)
36. Kang, H., Ryu, D., Cen, R., Ostriker, J.P.: *ApJ* **669**, 729 (2007)
37. Kauffmann, G., Colberg, J., Diaferio, A., White, S.: *MNRAS* **303**, 188 (1999)
38. Keshet, U., Waxman, E., Loeb, A.: *ApJ* **617**, 281 (2004)
39. Kolmogorov, A.N.: *Int. J. Comput. Math.* **2**, 157 (1968)
40. Kravtsov, A.V., Borgani, S.: *ARAA* **50**, 353 (2012)
41. Kremer, J., Stensbo-Smidt, K., Gieseke, F., Pedersen, K.S., Igel, C.: *IEEE Intell. Syst.* **32**, 16 (2017)
42. Kritsuk, A.G., Norman, M.L., Padoan, P.: *ApJL* **638**, L25 (2006)
43. Krizhevsky, A., Sutskever, I., Hinton, G.E.: In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105. Curran Associates, Inc. (2012)
44. Kunz, M.W., Schekochihin, A.A., Cowley, S.C., Binney, J.J., Sanders, J.S.: *MNRAS* **410**, 2446 (2011)
45. Larson, J.W., Briggs, P.R., Tobis, M.: *Procedia Comput. Sci.* **4**, 1592 (2011); In: *Proceedings of the International Conference on Computational Science, ICCS 2011*
46. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: *Proc. IEEE* **86**, 2278 (1998)
47. Locatelli, N., Vazza, F., Domínguez-Fernández, P.: *Galaxies* **6**, 128 (2018)
48. Lukic, V., Brüggén, M., Banfield, J.K., Wong, O.I., Rudnick, L., Norris, R.P., Simmons, B.: *MNRAS* **476**, 246 (2018)
49. Lukic, V., Brüggén, M., Mingo, B., Croston, J.H., Kasieczka, G., Best, P.N.: *MNRAS* **487**, 1729 (2019)
50. Ball, M.N., Brunner, R.J.: *Int. J. Mod. Phys. D* **19** (2009)
51. Mesarcik, M., Boonstra, A.-J., Meijer, C., Jansen, W., Rangelova, E., van Nieuwpoort, R.V.: *MNRAS* **496**, 1517 (2020)
52. Nagai, D., Kravtsov, A.V., Vikhlinin, A.: *ApJ* **668**, 1 (2007)
53. Peebles, P.J.E.: *Principles of Physical Cosmology* (1993)
54. Pfrommer, C., Springel, V., Enßlin, T.A., Jubelgas, M.: *MNRAS* **367**, 113 (2006)
55. Collaboration, Planck., et al.: *A & A* **594**, A13 (2016)
56. Planelles, S., et al.: *MNRAS* **467**, 3827 (2017)
57. Pranav, P., Edelsbrunner, H., van de Weygaert, R., Vegter, G., Kerber, M., Jones, B.J.T., Wintraecken, M.: *MNRAS* **465**, 4281 (2017)
58. Prokopenko, M., Boschetti, F., Ryan, A.J.: *Complexity* **15**, 11 (2009)
59. Rosati, P., Borgani, S., Norman, C.: *ARAA* **40**, 539 (2002)
60. Ryu, D., Kang, H., Hallman, E., Jones, T.W.: *ApJ* **593**, 599 (2003)
61. Sarazin, C.L.: *X-ray Emission from Clusters of Galaxies*. Cambridge University Press, Cambridge (1988)
62. Schmidt, W., Engels, J.F., Niemeyer, J.C., Almgren, A.S.: *MNRAS* **459**, 701 (2016)
63. Schmidt, W., Schulz, J., Iapichino, L., Vazza, F., Almgren, A.S.: *Astron. Comput.* **9**, 49 (2015)

64. Shalizi, C.R., Shalizi, K.L., Haslinger, R.: *Phys. Rev. Lett.* **93**, 149902 (2004)
65. Shannon, C.E.: *IEEE Proc.* **37**, 10 (1949)
66. Shannon, C.E., Weaver, W.: *The Mathematical Theory of Communication* (1949)
67. Simonyan, K., Zisserman, A.: 2014, CoRR, [arXiv:abs/1409.1556](https://arxiv.org/abs/1409.1556)
68. Springel, V.: *MNRAS* **401**, 791 (2010)
69. Springel, V., et al.: *Nature* **435**, 629 (2005)
70. Suskind, L., Lindesay, J.: *An Introduction to Black Holes. The Holographic Universe, Information and the String Theory Revolution* (2005)
71. Teyssier, R., Moore, B., Martizzi, D., Dubois, Y., Mayer, L.: *MNRAS* **414**, 195 (2011)
72. van Weeren, R.J., de Gasperin, F., Akamatsu, H., Brüggen, M., Feretti, L., Kang, H., Stroe, A., Zandanel, F.: *Sci. Space Rev.* **215**, 16 (2019)
73. Vazza, F.: *MNRAS* **465**, 4942 (2017)
74. Vazza, F.: *MNRAS*, 2968 (2019)
75. Vazza, F., Brüggen, M., Gheller, C.: *MNRAS* **428**, 2366 (2013)
76. Vazza, F., Brunetti, G., Gheller, C.: *MNRAS* **395**, 1333 (2009)
77. Vazza, F., Brunetti, G., Gheller, C., Brunino, R., Brüggen, M.: *A & A* **529**, A17+ (2011)
78. Vazza, F., Brunetti, G., Kritsuk, A., Wagner, R., Gheller, C., Norman, M.: *A & A* **504**, 33 (2009)
79. Vazza, F., Etori, S., Roncarelli, M., Angelinelli, M., Brüggen, M., Gheller, C.: *A & A* **627**, A5 (2019)
80. Vazza, F., Feletti, A.: The quantitative comparison between the neuronal network and the cosmic web. *Front. Phys.* **8**(491), 491 (2020). <https://doi.org/10.3389/fphy.2020.525731>. (Nov) (Provided by the SAO/NASA Astrophysics Data System). <https://ui.adsabs.harvard.edu/abs/2020FrP....8.491V>
81. Vazza, F., Ferrari, C., Bonafede, A., Brüggen, M., Gheller, C., Braun, R., Brown, S.: 2015, *ArXiv e-prints*
82. Vazza, F., Ferrari, C., Brüggen, M., Bonafede, A., Gheller, C., Wang, P.: *A & A* **580**, A119 (2015)
83. Vazza, F., Roediger, E., Brueggen, M.: 2012, *ArXiv e-prints* [1202.5882](https://arxiv.org/abs/1202.5882)
84. Vernstrom, T., Gaensler, B.M., Brown, S., Lenc, E., Norris, R.P.: *MNRAS* **467**, 4914 (2017)
85. Vogelsberger, M., et al.: *Nature* **509**, 177 (2014)
86. Vogelsberger, M., Marinacci, F., Torrey, P., Puchwein, E.: 2019, *arXiv e-prints*, [arXiv:1909.07976](https://arxiv.org/abs/1909.07976)
87. Wittor, D., Vazza, F., Brüggen, M.: *MNRAS* **464**, 4448 (2017)
88. Wolfram, S.: *Phys. D* **10**, 1 (1984)
89. Xu, H., Li, H., Collins, D.C., Li, S., Norman, M.L.: *ApJL* **698**, L14 (2009)

The Voronoi Tessellation Method in Astronomy



Iryna Vavilova, Andrii Elyiv, Daria Dobrycheva, and Olga Melnyk

Abstract The Voronoi tessellation is a natural way of space segmentation, which has many applications in various fields of science and technology as well as in social sciences and visual art. The varieties of the Voronoi tessellation methods are commonly used in computational fluid dynamics, computational geometry, geolocation and logistics, game dev programming, cartography, engineering, liquid crystal electronic technology, machine learning, etc. The very innovative results were obtained in astronomy, namely for a large-scale galaxy distribution and cosmic web pattern, for revealing the quasi-periodicity in a pencil-beam survey, for a description of constraints on the isotropic cosmic microwave background and the explosion scenario likely supernova events, for image processing, adaptive smoothing, segmentation, for signal-to-noise ratio balancing, for spectrography data analysis as well as in the moving-mesh cosmology simulation. We briefly describe these results paying more attention to the practical application of the Voronoi tessellation related to the spatial large-scale galaxy distribution.

I. Vavilova (✉) · A. Elyiv · D. Dobrycheva · O. Melnyk
Main Astronomical Observatory of the NAS of Ukraine, 27 Akademik Zabolotny St.,
Kyiv 03143, Ukraine
e-mail: irivav@mao.kiev.ua

A. Elyiv
e-mail: elyiv@mao.kiev.ua

D. Dobrycheva
e-mail: daria@mao.kiev.ua

O. Melnyk
e-mail: melnykol@gmail.com

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021
I. Zelinka et al. (eds.), *Intelligent Astrophysics*, Emergence, Complexity
and Computation 39, https://doi.org/10.1007/978-3-030-65867-0_3

1 The Voronoi Tessellation in a Spatial Galaxy Distribution: First Works and Basic Approach

The geometrical methods based on the Voronoi diagram deal with a partitioning of space into regions in a specific subset of generators. It was named after Georgy F. Voronoi (April 28, 1868, Zhuravka village, Chernihiv region, Ukraine—Nov 20, 1908, Warsaw, Poland), the outstanding Ukrainian mathematician [76, 89], who studied the general n -dimensional case of these diagrams [101, 102].

In 1984, Matsuda and Shima advanced the idea to apply the Voronoi tessellation method for describing the cellular structure of the Local Universe [63], finding a topological tendency of galaxies “to cluster at the vertices, edges and faces of polyhedral shaped voids”. In 1987, Ling demonstrated that the Voronoi tessellation and the Minimal Spanning Tree being applied to the CfA Redshift Survey of galaxies (the first survey to map the large-scale structure of the Universe) are able to detach filamentary structures and voids [60]. In 1989, Yoshioka and Ikeuchi proposed three-dimensional Voronoi tessellation as a model of the evolution of the negative density perturbations regions, which resulted in the overlapping of shells while the modeled skeleton can be compared with real observed structures and with mass distribution correlation functions [107].

For the first time, the Voronoi tessellation was considered in detail as a pattern of matter distribution in the Universe in work by Icke and Weygaert [48] and series of their following works [49, 50, 104]. These authors concluded that the regions of lower density become more spherical with evolution and matter floods away from expansion centers and accrues at the borders of packing of spheres. This leads to the partition of space on the Voronoi tessellation with nuclei in the centers of low-density regions called the voids. High-density regions—clusters of galaxies—lie at the crossing of vertexes of adjacent cells, filaments at the edge of cells, and pancakes of large-scale structure (LSS) are faces of cells (Fig. 1, right). Sheth et al. [83] have developed its idea and considered the model of a void created in the frame of the Voronoi tessellation paradigm.

The Voronoi tessellation can be constructed as follows. Let us consider a Voronoi cell of finite size in N -dimensional space (usually $N = 2$ or $N = 3$), where a fixed number of points is distributed according to some statistical law (for example, the Poisson law). Suppose that each point is the center of a spherical expanding bubble structure. If all these structures begin to expand at the same moment with the same rate, the bubbles will be touched in planes that perpendicularly bisect the lines connecting the centers of expansion. These bisecting planes, in turn, intersect each other. As a result of this process, new lines will be generated, which in turn intersect each other and form a network. Using an adopted terminology, we will call such a center of the cell as a nucleus. So, each nucleus will be enclosed by a set of $(N - 1)$ —dimensional planes forming a convex cell. Distribution of nuclei forms the Voronoi tessellation.

The realization of Voronoi tessellations for a certain number of expanding nuclei, which is known as the Voronoi foam, can be found in [48, 104]. In the case of two-

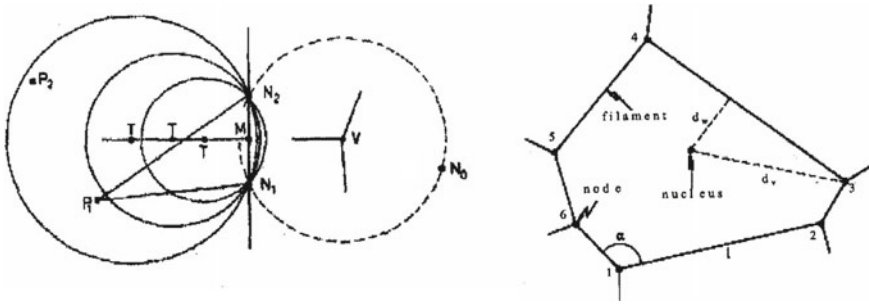


Fig. 1 (Left) The construction of a new Delaunay triangle from two known nuclei N_1 such that (N_1, N_2, N_3) forms a triangle whose circumsphere does not contain any other nucleus in the Voronoi tessellation. (Right) Identification of the four quantities which were calculated in each Voronoi cell: l_i , the length of wall i ; α , the angle between two walls meeting at vertex; d_w , the distance between the nucleus and a wall, where the projection of the nucleus doesn't necessarily lie on the wall ([48], open astronomy)

dimensional realization, the construction of a Voronoi cell consists of the search for all the Delaunay triangles having three nuclei (the center of the circumscribing circle is a vertex of the Voronoi foam). The program proposed by the authors [48], allows one to find all the Delaunay triangles having N_1 as a corner and construct the Voronoi cell belonging to N_1 by joining the circumcenters of the Delaunay triangles. Having applied this procedure to all nuclei, we obtain the Voronoi tessellation.

The process of forming the Voronoi tessellation is shown in Fig. 1 (left). The points N_0, N_1, N_2 form a Delaunay triangle obtained in a previous search; corresponding Voronoi vertex V is shown within the (dashed) circumcircle of N_0, N_1, N_2 as well as stubs of the Voronoi cell walls. On the left hand side of the diagram, the T are a sequence of trial points, the third of which produces a circle that encompasses two nuclei, P_1 and P_2 . The radius of the circumcircle of (N_1, N_2, P_1) being smaller than that of (N_1, N_2, P_2) , the point P_2 is N_3 , i.e. the third corner of the Delaunay triangle. Thus, the circumcenter of (N_1, N_2, P_1) is the next Voronoi vertex which, if connected with V , produces a complete Voronoi cell wall [48].

The obtained results could explain the heuristic models that supposing Voronoi tessellations as 3D templates for the galaxy distribution as well as could reproduce a variety of galaxy clustering properties. In an ideal scenario, the LSS is organized by equal spherical voids expanding at the same rate. The walls and filaments would be found precisely between expanding voids, and the resulting LSS web skeleton would the Voronoi tessellation.

The Voronoi tessellation method was picked up and also thrived in our research on a spatial galaxy distribution since 1990-is [92] that allowed us to obtain several priority results. Namely, we elaborated three main approaches in Voronoi tessellation application: (1) to describe a cosmic web skeleton in matter distribution as a Voronoi tessellation with nuclei at low-density regions; (2) to use Voronoi tessellation as a tool for direct measurement of galaxy local concentration and environmental description

of low-populated galaxy systems such as triplets, pairs, and isolated galaxies; (3) to apply Voronoi diagrams altogether with machine learning methods for 3D mapping of the Zone of Avoidance of our Galaxy [97, 99], where Generative Adversarial Network (GAN) algorithms are very useful [1, 36]. In particular, Coutinho et al. [19] performed verification of various algorithms that can reproduce the cellular structure of the Universe. By comparing the simulated distributions with real observational data, these authors showed that the best algorithm uses the nearest neighbour parameter between galaxies, and that network algorithms can be improved to reproduce the large-scale structure of the Universe.

We give examples in Sect. 2, how manner our developed approach is working. We briefly overview in Sect. 3 various astronomical research with the Voronoi diagrams, accentuating the papers related to the large-scale structure of the Universe, as well as we highlight in Sect. 4 several works and software, where the Voronoi tessellation and machine learning get along well with each other.

2 Voronoi Tessellation of the First, Second and Third Orders: Identification of the Low-Populated Galaxy Groups, Environment Effect, and Dark Matter Content

Because of Voronoi tessellation is a geometrical method based only on galaxy positions, it allows detaching overdensity regions of galaxies in comparison with the background [93]. We tested it with various samples of galaxies. First of all, we used the Local Supercluster of galaxies, which is well studied among other galaxy superclusters, for identifying galaxy groups of various populations. It was revealed that Voronoi's tessellation method depends weakly on the richness-parameter of groups, and the number of galaxies in the rich structures is growing rather than in the weak structures with an increase of this parameter [64].

In the first-order Voronoi tessellation, the critical parameter is the volume of the galaxy's Voronoi cell V . This parameter characterizes an environmental galaxy density. The condition of cluster/group membership of a particular galaxy is the relatively small V . This condition is actual when close neighbouring galaxies surround the galaxy. That is why the first order Voronoi tessellation is not corrected for the identification of small isolated galaxy systems [64].

We used the second-order Voronoi tessellation for the identification of pairs and single galaxies. Each galaxy i from set S forms the common cells with a certain number of neighbouring galaxies (Fig. 2). So, under neighbouring galaxies of galaxy i , we understand only galaxies that create common cells with this galaxy. For example, galaxy 1 creates only 4 common cells ($V_{1,2}$, $V_{1,3}$, $V_{1,4}$, $V_{1,5}$) with neighbouring galaxies 2, 3, 4, and 5, respectively. Each pair of galaxies i, j is characterized by the dimensionless parameters $p_{i,j}$:

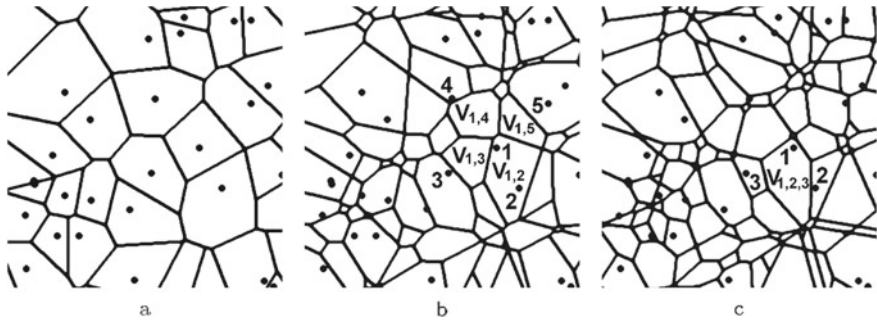


Fig. 2 2D Voronoi tessellation of the first- (a), second- (b) and third- (c) order for the same distribution of the random nuclei ([34], open astronomy)

$$p_{i,j} = \frac{\sqrt[D]{V_{i,j}}}{m_{i,j}}, \quad (1)$$

where D —space dimension, $V_{i,j}$ —the area (for 2D) or volume (for 3D) of cell, $m_{i,j}$ —distance between galaxies i and j . So, contrary to the first-order tessellation, the second-order tessellation for set S distribution of nuclei is the partition of the space which associates a region $V_{1,2}$ with each pair of nuclei 1 and 2 from S in such a way that all points in $V_{1,2}$ are closer to 1 and 2 than other nuclei from S . Region $V_{1,2}$ is a common cell for nuclei 1 and 2. However, these nuclei do not need to lie in the common cell. For example, nuclei 1 and 5 create the common cell $V_{1,5}$, and they do not lie in this cell. In such a way, the second-order Voronoi tessellation is available for the identification of single galaxies and pairs (Fig. 2b).

Let us introduce the parameter p_e , which describes only pair environment and does not depend on the distance between pair members directly. We define it as the mean value of $p_j(1)$ and $p_l(2)$ parameters of the first and second galaxy, excepting p from both sets:

$$p_e = \frac{\sum_{j=2}^k p_j(1) + \sum_{l=2}^n p_l(2)}{k + n - 2}, \quad (2)$$

where k and n —number of neighbouring galaxies for 1 and 2 galaxies of geometric pair, respectively. We started sums from $j = 2$ and $l = 2$ for excepting $2p$, because the first galaxy is neighbour for the second galaxy and vice versa. Therefore $(k + n - 2)$ is sum of neighbouring galaxies of pair members excepting of pair galaxies as neighbouring for each other. Parameter p_e depends on the distribution of neighbouring galaxies. A small value of p_e points out that such a pair is located in a loose environment. In such case the average volume of common cells of pair components with neighbouring galaxies is relatively small, and distance between them is significant, see formula (1) and Fig. 3a.

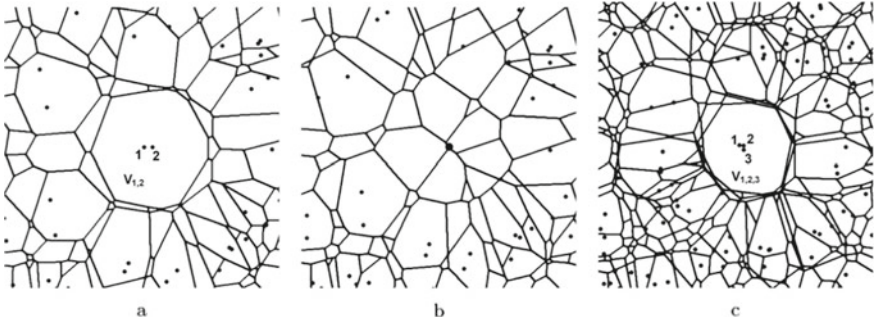


Fig. 3 Different configurations of the galaxies: isolated close pair (a) and isolated single galaxy (b) in the second-order tessellation; isolated close triplet in the third-order tessellation (c) ([34], open astronomy)

A single galaxy is a galaxy, which is not a member of any geometric pair. The single galaxies are field galaxies in the environment of geometric pairs. Every single galaxy has the own neighbours; single galaxies and geometric pair members can be among them. According to the second-order Voronoi tessellation, the larger is the degree of galaxy isolation, the larger is the number of neighbours (see Fig. 2b in comparison with Fig. 3b), but these neighbours locate farther. The best parameter that describes the isolation degree of the single galaxy, s , is the mean value of all parameters p_j of this galaxy:

$$s = \frac{\sum_{j=1}^k p_j}{k} \quad (3)$$

The third-order Voronoi tessellation is appropriate for the identification of galaxy triplets. It is the partition of the space which associates a region $V_{1,2,3}$ with each triplet of nuclei 1, 2, 3 in such a way that all points in $V_{1,2,3}$ are closer to nuclei 1, 2, 3 than other nuclei from S [59]. All points of the common triplet's cell are closer to galaxies of this triplet than to other galaxies. Similarly to the parameter $p_{i,j}$ for pairs, we can set up the parameter $t_{i,j,u}$ for triplets:

$$t_{i,j,u} = \frac{D \sqrt{V_{i,j,u}}}{\max(m_{i,j}, m_{i,u}, m_{j,u})}, \quad (4)$$

where D is the space dimension, $V_{i,j,u}$ is the area (for 2D) or volume (for 3D) of the cell, and $m_{i,j}$, $m_{i,u}$, $m_{j,u}$ are the distances between galaxies in the triplet. A geometric triplet in the third-order Voronoi tessellation contains three galaxies that have a common cell and the same maximal parameters $t_{\max}(1) = t_{\max}(2) = t_{\max}(3) = t$. The parameter t characterizes a degree of geometric triplet isolation. We can define the parameter of triplet environment t_e as the mean value of parameters $t_i(1)$, $t_j(2)$, and $t_u(3)$, except t from three sets:

$$t_e = \frac{\sum_{i=2}^k t_i(1) + \sum_{j=2}^n t_j(2) + \sum_{u=2}^q t_u(3)}{k + n + q - 3} \quad (5)$$

here in the case of the third-order Voronoi tessellation, k , n , and q denote the number of neighbouring triplets which contain galaxies 1, 2, and 3, respectively. Therefore, $(k + n + q - 3)$ is the number of neighbouring triplets for a certain triplet that contain at least one galaxy from this triplet (see, Fig. 3).

Parameters p , s , and t are the basic ones and define the isolation degree of a galaxy pair, single galaxy, or triplet compared to the background, respectively. Parameters p_e and t_e are additional ones and contain information about the distribution of the neighbouring galaxies (environment). Similar to the second- and third-order Voronoi tessellations, it is possible to apply more high-order Voronoi tessellations to identify galaxy quartets and quintets, etc.

So, one can use galaxies as the nuclei of the Voronoi tessellation taking into account the equatorial coordinates α , δ and radial velocities of galaxies V_h only. For the construction of the 3D Voronoi tessellations, it is necessary to determine the distances in 3D space. The spatial distance between two galaxies can be decomposed into projected (tangential) distance r and radial component v (difference of the radial velocities). We can determine the projected distance with a relatively high accuracy. Simultaneously, the radial component has errors due to the inaccuracy of radial velocity measurement of each galaxy and existing strong peculiar velocities (due to virial motions of galaxies in groups and clusters). As a result, the galaxy distribution in the radial velocities space is extended along the radial component, the so-called fingers-of-God effect. This is attributed to the random velocity dispersion in a galaxy volume-limited sample that cause a galaxy's velocity to deviate from pure Hubble flow, stretching out a group of galaxies in redshift space [54, 66]. Various authors take into account this effect in their way, depending on the specifics of their problem. For example, Marinoni et al. [62] chose some cylindrical window of clustering, which is extended along the radial component. We introduced the weight for a radial component [34], avoiding the problem of tangential and radial distance in equivalence to apply the high-order 3D Voronoi tessellation method.

An efficient way to show Voronoi tessellation advantages was to apply it to the galaxy samples from the Local Supercluster [64, 94, 96] and the Sloan Digital Sky Survey (SDSS), where at the first time we examined it for spectroscopic aims [34, 65, 67, 70, 77, 95]. We did not consider galaxies that located within 2° near borders, because the correct estimation of Voronoi cell volume is not possible in this case. Selecting single galaxies and pairs by the second-order Voronoi tessellation, as well as triplets by the third-order Voronoi tessellation method, we obtained 2196 geometric pairs, 1182 triplets, and 2394 single galaxies. We did not make a clear division between physical gravitationally bound systems and non-physical ones, following the supposition that the more isolated a system is, the higher probability that it is physical (compact pairs are with $R_h < 150$ kpc and triplets are with $R_h < 200$ kpc).

Estimating the dark matter content in the low-populated groups, we obtained the median values of mass-to-luminosity ratio [34]: $12M_{Sun}/L_{Sun}$ for the isolated

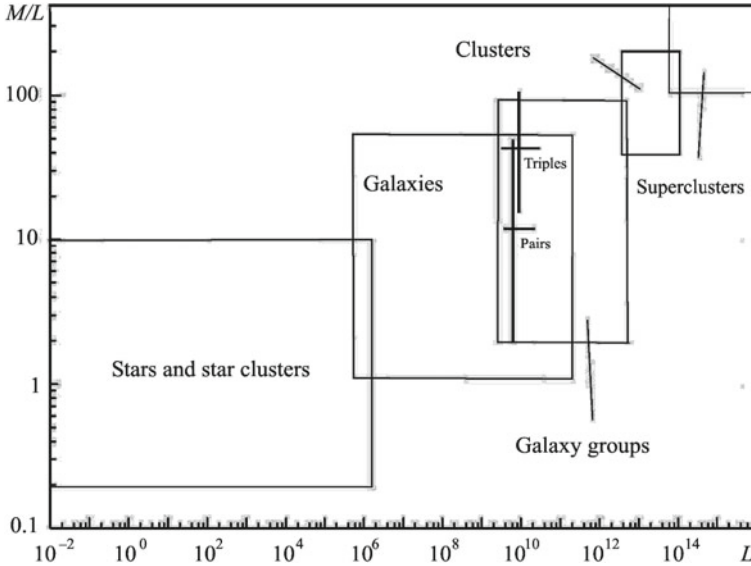


Fig. 4 Mass-to-luminosity ratio diagram for galaxy systems of different population (star clusters, galaxies, galaxy groups, clusters, and superclusters), where the result for the low-populated groups [66] is pointed ([96], open astronomy)

pairs and $44M_{Sun}/L_{Sun}$ for the isolated triplets. Note that for the most compact pairs and triplets (with $R < 50$ (100) kpc, respectively) there is not a very large difference in dark matter content for pairs and triplets: $7 M_{Sun}/L_{Sun}$ and $8 M_{Sun}/L_{Sun}$. The mass-to-luminosity ratio diagram for galaxy systems of different population (star clusters, galaxies, galaxy groups, including the low-populated ones, clusters, and superclusters) is presented in Fig. 4. Several examples of isolated triplets of galaxies are given in Fig. 5. We conclude about the dark matter distribution that for the dynamically younger sparsely groups (triplets), dark matter is more likely associated with the individual galaxy halos, for the interacting and late sparsely groups the dark matter lies in a common halo of galaxy groups.

Using an inverse volume of Voronoi cell ($1/V$) as a parameter describing the local environmental density of a galaxy, we considered the volume-limited SDSS (DR5 and DR9) galaxy samples ($0.02 < z < 0.1$, $-24 < M_r < -19.4$) [26, 27, 30, 67] and found that

- the early type galaxies prefer to reside in the Voronoi cells of smaller volumes (i.e., dense environments) than the late type galaxies, which are located in the larger Voronoi cells (i.e., sparse environments);
- the relationships between the morphological types and the $u - r$, $g - i$, and $r - z$ color indices of pairs of galaxies with radial velocities $3000 < V < 9500$ km/s evident that the Holmberg effect is not revealing, by the other words, it can be considered only in historical aspect [28];

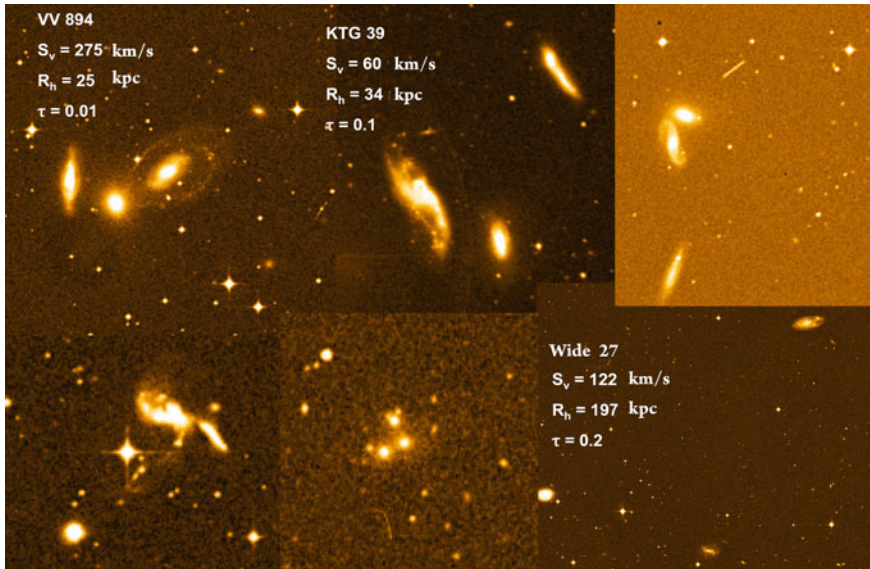


Fig. 5 The interacting (VV894), most compact (KTG39), and wide triplets of galaxies, where S_v is the rms velocity of galaxies with respect to the triplet centre, R_h —harmonic mean radii of the triplet, $\tau = 2H_0R_h/S_v$ —its dimensionless crossing time ([96], open astronomy)

- properties of such small groups as pairs and triplets, where segregation by luminosity was clearly observed, are fit well to Dressler effect: galaxies in isolated pairs and triplets are on average two times more luminous than isolated galaxies;
- the dependence of the color indices and stellar magnitudes is effective for the automated morphological classification of the galaxies (E —early types, L —late types).

The morphological types of the galaxies were divided into two classes: Early— E (from elliptical to lenticular) and Late— L (from spiral Sa to irregular Irr types). The absolute magnitude

$$M_r = m_r - 5 \log(V/H_0) - 25 - K(z) - ext \quad (6)$$

could be corrected for Galactic absorption ext in accordance with [81] and K —correction $K(z)$ according to [16]. Here we used the Λ CDM model of the Universe with the Λ CDM cosmological parameters ($\Omega_M = 0.27$, $\Omega_\Lambda = 0.73$, $\Omega_k = 0$, $H_0 = 0.71$). In order to apply the Voronoi tessellation method we should done transition from equatorial coordinates and velocities to the comoving x , y , z coordinates for each central galaxy in the sample ($M_r < -20.7$). To do this we can transform the redshift z to the corresponding distance $\chi(z)$ for each galaxy by integrating as follows

$$\chi(z) = D_H \int_0^z \frac{dz'}{E(z')} \quad (7)$$

where $D_H = c/H_0$ is the Hubble distance and $E(z')$ is the Hubble parameter defined as follows

$$E(z') = \sqrt{\Omega_M(1+z)^3 + \Omega_k(1+z)^2 + \Omega_\Lambda} \quad (8)$$

The coordinates x, y, z of the galaxies in the comoving space are determined as follows

$$x = \chi(z) \cos(\theta) \cos(\phi) \quad (9)$$

where (θ) is the declination of each galaxy, (ϕ) is the right ascension, and $(\chi(z))$ is the corresponding distance for redshift z . After getting the three-dimensional Cartesian coordinates of the galaxies, we divided the geometrical space occupied by the galaxy sample in mosaic cells (volumes V in the 3D case). Each cell has a galaxy as a nucleus and consists of elementary volumes of space closer to this galaxy than to any other galaxy [63]. The use of the Voronoi tessellation to isolate groups of galaxies in three dimensions has been described in detail by Melnyk et al. [64]. Figure 2a shows an example of the Voronoi tessellation in a two dimensional case. Let us use the value of inverse volume ($1/V$) of the Voronoi cells to describe the density of galaxy environments; when $1/V$ is higher, a galaxy is less isolated.

Examples of the distributions of E and L galaxies versus inverse volume of the Voronoi cells that contain them are shown in Fig. 6. In work [27] we grouped galaxies from the SDSS sample at $z < 0.1$ into 4 logarithmic intervals $1/V < 0.001$, $0.001 < 1/V < 0.01$, $0.01 < 1/V < 0.1$, and $1/V > 0.1$ for four ranges of the redshift $0.02 < z < 0.04$, $0.04 < z < 0.06$, $0.06 < z < 0.08$, and $0.08 < z < 0.1$ (in the rows) and for different ranges of absolute stellar magnitude, $-21.5 < M_r < -20.7$, $-22.5 < M_r < -21.5$, and $M_r < -22.5$. The number of galaxies in each

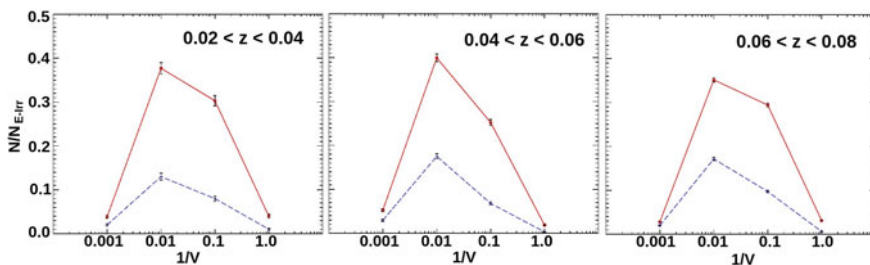


Fig. 6 The distribution of the number of galaxies versus inverse volume of the Voronoi cell (local density parameter), with early morphological type E indicated by red lines and late type L indicated by blue lines, for different ranges of redshift; absolute stellar magnitude of galaxies selected from the SDSS at $z < 0.1$ is $-22.5 < M_r < -21.5$. The number of galaxies in each bin is normalized to the total number of $E \div L$ within the given subsample. The number of central bright E and L galaxies is as follows: $E = 1636$, $L = 459$ for $0.02 < z < 0.04$, $E = 3609$, $L = 1247$ for $0.04 < z < 0.06$, $E = 9432$, $L = 3596$ for $0.06 < z < 0.08$ [27]

bin for the E and L types is normalized to the total number of $E \div L$ galaxies within the given subsample. Figure 6 shows that the fraction of galaxies of spiral and late types becomes larger while redshift increasing, while the fraction of early types, on the contrary, is smaller. It explains the well known evolutionary trend of a reduction in the number of galaxies with suppression of star formation for increasing redshift [21, 90], even at comparatively low redshifts down to $z < 0.1$. Also, for the brighter galaxies in the sample, the fraction of galaxies of earlier types is larger since, on the average, earlier types have higher luminosities (the well-known morphological type versus colour indices/luminosity relation) [8, 44, 73]. The brightest galaxies of earlier types with $M_r < -22.5$ appear preferentially in denser environments: the peak of the distribution of the inverse volumes of the Voronoi cells for the E types lie within the interval $0.01 < 1/V < 0.1$, while in other intervals of M_r , for the L types the peak of the distribution always is within $0.001 < 1/V < 0.01$ (the morphology-density relation [8, 28, 32, 44]).

We can also determine the density of galaxies in a Voronoi cell, including their faint satellites, i.e., galaxies with $M_r > -20.7$: $(n + 1)/V$, where n is the number of faint galaxies in the Voronoi cell, and V is the volume of the Voronoi cell. We also constructed distributions of early E and late L types galaxies in dependence on the parameter $(n + 1)/V$ in four intervals: $(n + 1)/V < 0.01$, $0.01 < (n + 1)/V < 0.1$, $0.1 < (n + 1)/V < 1$, and $(n + 1)/V > 1$. The number of galaxies is normalized to the number of $E \div L$ galaxies within the given range of $(n + 1)/V$. We examined the density of galaxies only in the first two redshift intervals, since we cannot evaluate the evolution of their properties at a higher z because there are not enough faint galaxies. However, we can compare the galaxies' environmental density as a function of the absolute magnitude and morphological type of the bright central galaxy. Thus, the fraction of early types of central galaxies increases with increasing environmental density, while, on the other hand, the fraction of late types decreases; that is, the earlier types are in a denser environment than the late types. When the central galaxy is brighter, the fraction of early types in a subsample will be larger [27, 100].

3 The Voronoi Tessellation in Astrophysical Research

Ebeling and Wiedenmann [33] were the first to apply the Voronoi tessellation for finding galaxy groups and clusters. Later such an approach was used by Ramella et al. [78], Kim et al. [56], Lopes et al. [61], Barrena et al. [6], Melnyk et al. [65], Panko and Flin [71]. Doroshkevich [31] introduced its for filaments and walls (1D and 2D LSS structures) as well as Neyrinck [68] for the search of voids in a spatial galaxy distribution.

We note some important earlier works as concerns with other applications of Voronoi diagrams to the large-scale galaxy distribution: for revealing the quasi-periodicities in a pencil-beam survey [51, 87], for a description of constraints on the Voronoi model when applied to the isotropic cosmic microwave background [17]. A

significant contribution for Voronoi tessellation application to various astronomical tasks was made by Zanninetti, who considered two- and three-dimensional cases of the explosion scenario likely supernova events and developed a dynamical method allowing to describe the explosion phases [108, 109].

Ramella et al. [78] created a Voronoi Galaxy Cluster Finder, which uses positions and magnitudes of galaxies to define galaxy clusters and extract its parameters: size, richness, central density, etc. The 3D Voronoi tessellation for galaxy group identification was realized by Marinoni et al. [62] and Cooper et al. [18]. Weygaert et al. prepared a useful review of the spatial galaxy distribution with Delaunay and Voronoi tessellations [42, 105]. They discussed the Delaunay Tessellation Field Estimator (DTFE) and the concept of Alphashapes for matter distribution; the Multiscale Morphology Filter (MMF), which uses the DTFE for detachment of filaments, sheets, and clusters; the Watershed Voidfinder (WVF) to identify voids.

The era of big data surveys (see, for example, review in work by Vavilova et al. [98] accelerated the Voronoi diagrams application on a spatial galaxy distribution properties and environment influence: $z = 0.1 - 3.0$, COSMOS survey [82]; $z \leq 0.5$, Herschel-ATLAS/GAMA [9]; $z < 0.1$, Coma Supercluster [20]; $z < 0.3$, ALHAMBRA survey [79]. Söchting et al. used Voronoi tessellation within overlapping slices in the photometric redshift space ($0.2 < z < 3.0$). It allowed them to detach region $z \sim 0.4$ with a slow emergence of virialized clusters accordingly to the hierarchical scenario and to detect new superclusters as the peaks of a matter distribution up to $z = 2.9$ [85]. As for the Voronoi tessellation cluster finder algorithms, we note the work by Soares et al., who developed it to produce reliable cluster catalogs up to $z = 1$ or beyond and down to $10^{13.5}$ Msun. They built the Voronoi tessellation cluster finder in photometric redshift shells and used the two-point correlation function of the galaxies in the field to determine the density threshold for the detection of cluster candidates and to establish their significance [84].

A principal new galaxy cluster finder based on a 3D Voronoi Tessellation plus a maximum likelihood estimator, followed by gapping-filtering in radial velocity ($VoML + G$), was created. Pereira et al. [74, 75] applied it successfully to find optical clusters (R_{200}) in the Local Universe as well as Santiago-Batista et al. for the identification of continuous filaments in the environment of superclusters [80]. Grokhovskaya et al. developed filtering algorithms of multiparameter analysis of the large-scale distribution of galaxies (identification of galaxy systems and voids) in narrow slices in the entire range of redshifts of HS 47.5-22 constructing density contrast maps, namely with adaptive kernel and Voronoi tessellation [40]. The 3D Voronoi tessellation application to the DEEP2 survey was first introduced by Gerke et al. [38]. Meanwhile, Shen Ying et al. [106] proposed an algorithm which computes the cluster of 3D points by applying a set of 3D Voronoi cells and allows a 3D point cluster pattern can be highlighted and easily recognized.

Hung et al. have demonstrated that Voronoi tessellation Monte-Carlo mapping is beneficial for studying the environment effect on galaxy evolution in high-redshift large-scale structures ($z \sim 1$) in the ORELSE survey (Observations of Redshift Evolution in Large Scale Environments) [47]. An exciting application of Voronoi tessellation was proposed by Lam et al. [57]: for constructing the white dwarfs lumi-

nosity functions they used parameters of proper motion and colours from the Pan-STARRS 1 3π Steradian Survey Processing Version 2; for improving the accuracy of the maximum volume method they used Voronoi tessellation space binning to recalculate photometric/astrometric uncertainties. It helped to estimate disk-to-halo dark matter ratio as 100. Another a non-parametric method for estimating halo concentration using Voronoi tessellation, TesseRACT, was proposed by Lang et al. [58], who showed that it fits well with non-spherical halos and more accurate at recovering intermediate concentrations for N-body halos than techniques that assume spherical symmetry.

The very interesting algorithm, Void Finder ZOBOV (ZOnes Bordering On Voidness), based on Voronoi tessellation, was proposed by Neyrinck et al. [68]. This algorithm finds density depressions galaxy distribution without free parameters. To estimate local density, it uses the Voronoi tessellation. One of the output of this algorithm is the probability that each void arises from Poisson fluctuations. However, Elyiv et al. [35] have demonstrated a weak spot for ZOBOV void finder. Voids are the lowest density regions, so any method that uses the positions of galaxies directly to measure density for identifying the voids is then prone to shot noise error since voids are the regions with a very low concentration of galaxies by definition (Fig. 7). The Void IDentification and Examination toolkit (VIDE) developed by Sutter et al. [88] includes the parameter-free void finder ZOBOV, where “Voronoi tessellation of the tracer particles is used to estimate the density field followed by a watershed algorithm to group Voronoi cells into zones and subsequently voids”.

Zaninetti in series of works [110, 112] developed a practical statistics for the voids between galaxies with two new survival functions and considered the 3D distribution of the volumes of Poissonian Voronoi Diagrams to their 2D cross-sections in the assumption of gamma-function for the 3D statistics of the volumes of the voids in the Local Universe. He also conducted simulations [111] of a spatial galaxy distribution using the Poissonian Voronoi polyhedra and the 2dF Galaxy Redshift Survey and the Third Reference Catalog of Bright Galaxies; Zaninetti gives a brief overview of a current status of the research on the statistics of the Voronoi Diagrams in [113].

Among other astronomical tasks, the Voronoi diagrams have been used for image processing, adaptive smoothing, segmentation, for signal-to-noise ratio balancing [14], for spatial structure of the solar wind and solar-terrestrial connections [7], for spectrography data analysis in different electromagnetic regions [12, 13, 25], in the moving-mesh cosmology simulation [86] and [103] (AREPO Public Code), chemical evolution in the early Universe [15], star formation simulation [46], spatial distribution of lunar craters [45].

For example, Cabrera et al. [11] applied the Voronoi diagram for image reconstruction technique in the interferometric data based on the Bayesian approach. Cadha et al. proposed Voronoi compact image descriptors and showed that Voronoi partitioning improves the geometric invariance and performance of image retrieval [14] as well as they developed a Voronoi-based machine learning method (deep convolution neural network). As for the cosmological simulation, Busch and White [10] used Voronoi tessellation for a hierarchical tree structure that allowed them to associate local density peaks with disjoint subsets of particles and to analyze mass distribu-

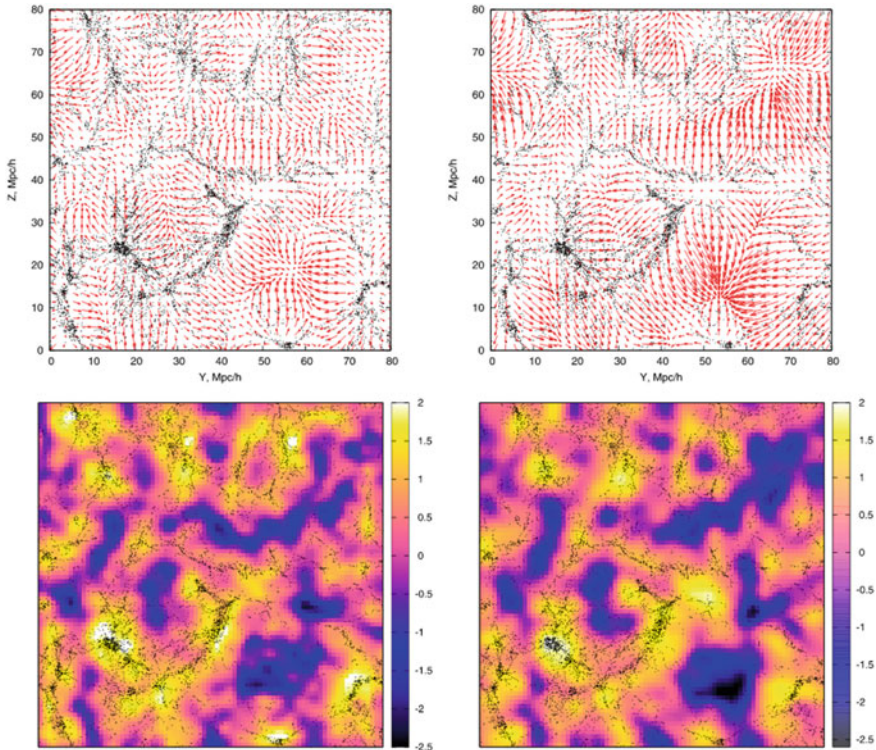


Fig. 7 The reconstructed displacement field (top panels) and its divergence (bottom panels) obtained with the two void finders, the Uncorrelating Void Finder (left-hand panels) and the Lagrangian Zel'dovich Void Finder (right-hand panels). The displayed region's size is $80 \times 80 \text{ Mpc } h^{-1}$, with a thickness of $5 \text{ Mpc } h^{-1}$. Black dots represent dark matter haloes. The amplitude of the vector field components (red arrows) is reduced by a factor of 0.75, for visual clarity ([35], open access)

tion at different levels of threshold. Similar to our work [27], when we introduced parameter of the volume of Voronoi cell to study environment influence on galaxies from the SDSS, Paranjape and Alam [72] also applied the inverse local number density parameter. They studied physical effects for such properties as halo (subhalo) mass, large-scale environment, etc. in various cosmological dark matter models and concluded that the Voronoi volume function gives a new mathematical instrument for galaxy evolution physics and dark sector study. Nightingale et al. developed the PyAutoLens software (<https://github.com/Jammy2211/PyAutoLens>), where the Voronoi tessellation is applied for reconstruction of strongly lensed galaxies.

Neyrinck developed the sectional-Voronoi algorithms in Python for cosmic-web research, because the Voronoi/Delaunay duals and origami tessellation give a wide class of spiderwebs. "Voronoi edges are perpendicular bisectors of their corresponding Delaunay edges; the 'bisector' part can be relaxed. Each Voronoi edge may

be slid along its Delaunay edge, closer to one of the generators. They may not be slid entirely independently, though, since the Voronoi edges must still join vertices. There turns out to be one extra degree of freedom per generator, causing its cell to expand or contract. The result is a sectional-Voronoi diagram, a section through a higher-dimensional Voronoi tessellation. A generator's extra degree of freedom in a sectional-Voronoi diagram can be thought of as its distance from the space being tessellated. A sectional-Voronoi diagram can also be thought of as a Voronoi tessellation in which each generator may have a different additive 'power' in the distance function used to determine which points are closest to the generator (thus an alternative term, power diagram). Ash and Bolker [2] showed that 2D spiderwebs and sectional Voronoi tessellations are equivalent" (cited by [69]). The package is available at <https://github.com/neyrinck/sectional-tess>, <https://mybinder.org/v2/gh/neyrinck/sectional-tess/master>.

In the present day, the Voronoi diagrams methods have many applications in various fields of science and technology as well as in social sciences and visual art [3, 4]. They are commonly used in computational fluid dynamics, computational geometry, geolocation and logistics, game dev programming, cartography, engineering, liquid crystal electronic technology, etc. For the first time, the Voronoi tessellation was utilized by Debnath et al. [22] for the discoveries in the particle physics beyond the Standard Model at the Large Hadron Collider at CERN. "Since such tessellations capture the underlying probability distributions of the phase space, interesting features in the data can be detected by studying the geometrical aspects of the ensemble of Voronoi cells" (cited by [23]). These methods allow identifying kinematic edges in two dimensions and generalize the technique for robust detection of phase space boundaries, which could be applied to discover new physics. An interesting library of "Voronoi Diagrams: Applications from Archaeology to Zoology" is collected by Scot Drysdale on the website <https://www.ics.uci.edu/~eppstein/gina/scot.drysdale.html>.

4 The Voronoi Tessellation and Machine Learning

Straight application of classical Voronoi diagram in Machine Learning is the k -nearest neighbors (k -NN) algorithm at the number of neighbors $k = 1$. In the case of the classifier, the output class is choosing among its k the closest neighbors. Each of them gives a contribution to the class with some weight. Normally weight is inverse to the distance between target object and neighbor (closer neighbors will have a stronger influence than further neighbors) or uniform (all points in neighborhood are weighted equally). If $k = 1$, then the object is just linked to the class of the nearest neighbor. From the other side, it could be interpreted as the building of the Voronoi diagram by training objects as nuclei of the diagram. The target object will have a class depending on which Voronoi cell it resides. Bring your data to life.

A set of programming codes for 1-NN visualization ($k = 1$) with examples (Hover Voronoi, a demonstration of d3-Delaunay, Voronoi Labels, Voronoi neigh-

bors, Voronoi update) are available on the website <https://observablehq.com/@d3/> by Mike Bostock (2018). For the color image segmentation problem in computer vision, an adaptive and unsupervised clustering approach with Voronoi diagrams was introduced, which outperforms the existing algorithms [43]. A Python library “Pycobra” contains several ensemble machine learning algorithms and visualization tools based on the Voronoi tessellations [41]. It can be downloaded from the Python Package Index (PyPi) and Machine Learning Open Source Software (MLOSS) at <https://github.com/bhargavvader/pycobra>.

In the case of $k > 1$, we should use the concept of high order Voronoi diagram, where a cell represents the set of points in space closer to a given k nuclei than to all others (see, Sect. 2 and works by Elyiv et al. [34, 99]). In this case, k -order Voronoi space dividing can help us to find k -near neighbors directly. The crossing of high-order Voronoi diagram borders represent the set of k near neighbors. In k -NN regression, the output value for the target object is the average of the values of k nearest neighbors. If each neighbor has equal weight, it means that for each cell could be assigned pre-calculated averaged value. Next, if the target object resides in this cell, automatic assigned could be done. In all these cases, creating a Voronoi diagram on the training sample could make a faster k -NN algorithm application.

For example, Inkulu and Kapoor [53] presented an algorithm covering the Voronoi diagram with hyperboxes, which provides ANN queries. Another parallel spatial range query algorithm based on Voronoi diagrams and MR-tree, which is benefiting from the k -NN, is developed by Fu and Liu [37].

Voronoi diagram also has a wide application in deep learning. In work [5], the authors studied the geometry of Deep Artificial Neural Networks with piecewise affine and convex nonlinearities. The authors demonstrated that each layer’s input space partition corresponds to the Voronoi diagram with several regions that grow exponentially with increasing neurons. Numerical experiments support these theoretical results, which are expressed by the Deep ANN decision boundary in the input space, a measure of its curvature that depends on the network architecture, activation functions, and weights.

In work [52] the authors presented a Deep Convolution Neural Network (CNN) constructed on a Voronoi tessellation of 3D molecular structures of proteins (VoroCNN model). Both convolution and pooling operations were used as a part of network architecture to predict local qualities of 3D protein folds. They computed Voronoi tessellation of molecular 3D structures and converted them into a protein interaction graph. The graph’s critical property is that it implicitly keeps the information about the spatial relationship between the atoms of the protein model. The authors claim that for presently available amounts of data and computational resources, Voronoi tessellation is the best representation of 3D protein structure than raw volumetric data.

far back in time, but builds up structure continuously, using not-very-special initial conditions and gravity (plus perhaps other forces) to develop what we see today. In between come some remarkable constructs, including Thomas Wright’s hierarchy, Descartes’s Voronoi tessellation of whirlpools in the ether, Alfred Russel Wallace’s (yes, the evolution guy) “Goldilocks” location for the Solar system, Cornelis Easton’s off-center spiral arms, and the Kapteyn Universe” (cited by Trimble [91]). We have combined this representation, which is consonant with ours, in Fig. 8 as an illustration of partitioning the space into cells for the subsequent extraction of the physical essence of the phenomena: one of them displays classical physics, Vortex theory applied to the Solar system (Descartes, 1644), the other gives a visualization of galaxy distribution through the 2D-Voronoi tessellation.

Acknowledgements This work was partially conducted in the frame of the budgetary program of the NAS of Ukraine “Support for the development of priority fields of scientific research” (CPCEL 6541230).

References

1. Ambrogioni, L., Güçlü, U., van Gerven, M.: k-GANs: ensemble of generative models with semi-discrete optimal transport (2019). [arXiv:1907.04050](https://arxiv.org/abs/1907.04050)
2. Ash, P.F., Bolker, E.D.: Generalized Dirichlet tessellations. *Geom. Dedicata*. **20**(2), 209–243 (1986)
3. Aurenhammer, F.: Voronoi diagrams. A survey of a fundamental geometric data structure. *ACM Comput. Surv.* **23**(3), 345–405 (1991)
4. Aurenhammer, F., Klein, R.: In: Sack, J.-R. (ed.) *Voronoi Diagrams*. North-Holland, Amsterdam (2000)
5. Balestrieri, R., Cosentino, R., Aazhang, B., and Baraniuk R.G.: The Geometry of deep networks: power diagram subdivision. In: 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada, pp. 1–10 (2019)
6. Barrena, R., Ramella, M., Boschin, W., et al.: VGCF detection of galaxy systems at intermediate redshifts. *Astron. Astrophys.* **444**(3), 685–695 (2005)
7. Borovsky, J.E.: The spatial structure of the oncoming solar wind at Earth and the shortcomings of a solar-wind monitor at L1. *J. Atmos. Solar Terr. Phys.* **177**, 2–11 (2018)
8. Blanton, M.R., Eisenstein, D., Hogg, D.W., et al.: Relationship between environment and the broadband optical properties of galaxies in the sloan digital sky survey. *Astrophys. J.* **629**, 143–157 (2005)
9. Burton, C.S., Jarvis, M.J., Smith, D.J.B., et al.: Herschel-ATLAS/GAMA: the environmental density of far-infrared bright galaxies at $z \leq 0.5$. *Mon. Notic. R. Astron. Soc.* **433**(1), 771–786 (2013)
10. Busch, P., White, S.D.M.: The tessellation-level-tree: characterizing the nested hierarchy of density peaks and their spatial distribution in cosmological N-body simulations. *Mon. Notic. R. Astron. Soc.* **493**, 5693–5712 (2020)
11. Cabrera, G.F., Casassus, S., Hitschfeld, N., et al.: Bayesian image reconstruction based on Voronoi diagrams. *Astrophys. J.* **672**, 1272–1285 (2008)
12. Cappellari, M., Copin, Y.: Adaptive spatial binning of 2D spectra and images using Voronoi tessellations. In: *Galaxies: The Third Dimension*, ASP Conference Proceedings, vol. 282, p. 515 (2002)
13. Cappellari, M., Copin, Y.: Adaptive spatial binning of integral-field spectroscopic data using Voronoi tessellations. *Mon. Notic. R. Astron. Soc.* **342**, 345–354 (2003)

14. Chadha, A., Andreopoulos, Y.: Voronoi-based compact image descriptors: efficient region-of-interest retrieval with VLAD and deep-learning-based descriptors (2016). [arXiv:1611.08906](https://arxiv.org/abs/1611.08906)
15. Chiaki, G., Yoshida, N., Hirano, S.: Gravitational collapse and the thermal evolution of low-metallicity gas clouds in the early Universe. *Mon. Notic. Roy. Astron. Soc.* **463**(3), 2781–2798 (2016)
16. Chilingarian, I.V., Melchior, A.L., Zolotukhin, I.Y.: Analytical approximations of K-corrections in optical and near-infrared bands. *Mon. Notic. R. Astron. Soc.* **405**, 1409–1420 (2010)
17. Coles, P., Barrow, J.D.: Microwave background constraints on the Voronoi model of large-scale structure. *Mon. Notic. R. Astron. Soc.* **244**, 557–562 (1990)
18. Cooper, M.C., Newman, J.A., Madgwick, D.S., et al.: Microwave background constraints on the Voronoi model of large-scale structure. *Astrophys. J.* **634**(2), 833–848 (2005)
19. Coutinho, B.C., Hong, S., Albrect, K., et al.: The network behind the cosmic web (2016). [arXiv:1604.03236v2](https://arxiv.org/abs/1604.03236v2)
20. Cybulski, R., Yun, Min S., Fazio, G.G., et al.: From voids to coma: the prevalence of pre-processing in the local universe. *Mon. Notic. Roy. Astron. Soc.* **439**(4), 3564–3586 (2014)
21. Cucciati, O., Iovino, A., Marinoni, C., et al.: The VIMOS VLT deep survey: the build-up of the colour-density relation. *Astron. Astrophys.* **458**, 39–52 (2006)
22. Debnath, D., Gainer, J.S., Kim, D., Matchev, K.: Edge detecting new physics the Voronoi way (2015). [arXiv:1506.04141](https://arxiv.org/abs/1506.04141)
23. Debnath, D.: Generic and sensitive searches for new physics. A dissertation presented to the graduate school of the University of Florida. 293 p. (2018)
24. Descartes, R.: *Principia Philosophiae*. Ludovicus Elzevirius, Amsterdam (1644)
25. Diehl, S., Statler, T.S.: Adaptive binning of X-ray data with weighted Voronoi tessellations. *Mon. Notic. R. Astron. Soc.* **368**, 497–510 (2006)
26. Dobrycheva, D.V.: The new galaxy sample from SDSS DR9 at $0.003 < z < 0.1$. *Odessa Astron. Publ.* **26**, 187–189 (2013)
27. Dobrycheva, D.V., Melnyk, O.V., Vavilova, I.B., Elyiv, A.A.: Environmental density versus colour indices of the low redshifts galaxies. *Astrophysics* **58**, 168–180 (2015)
28. Dobrycheva, D.V., Vavilova, I.B.: No the Holmberg effect for galaxy pairs selected from the SDSS DR9 at $z \leq 0.06$. *Odessa Astron. Publ.* **29**, 37–41 (2016)
29. Dobrycheva, D.V.: Morphological content and color indices bimodality of a new galaxy sample at the redshifts $z < 0.1$. PhD Thesis in Phys.-Math. Sciences, Main Astronomical Observatory, NAS of Ukraine, 132pp. (2017)
30. Dobrycheva, D.V., Vavilova, I.B., Melnyk, O.V., Elyiv, A.A.: Machine learning technique for morphological classification of galaxies at $z < 0.1$ from the SDSS (2017). [arXiv:1712.08955](https://arxiv.org/abs/1712.08955)
31. Doroshkevich, A., Gottlober, S., Madsen, S.: The accuracy of parameters determined with the core-sampling method: applications to Voronoi tessellations. *Astron. Astrophys., Suppl. Ser.* **123**, 495–506 (1997)
32. Dressler, A.: Galaxy morphology in rich clusters - implications for the formation and evolution of galaxies. *Astrophys. J.* **236**, 351–365 (1980)
33. Ebeling, H., Wiedenmann, G.: Detecting structure in two dimensions combining Voronoi tessellation and percolation. *Phys. Rev. E (Stat. Phys., Plasmas, Fluids, Relat. Interdiscip. Top.)* **47**(1), 704–710 (1993)
34. Elyiv, A., Melnyk, O., Vavilova, I.: High-order 3D Voronoi tessellation for identifying isolated galaxies, pairs and triplets. *Mon. Notic. R. Astron. Soc.* **394**, 1409–1418 (2009)
35. Elyiv, A., Marulli, F., Pollina, G., et al.: Cosmic voids detection without density measurements. *Mon. Notic. R. Astron. Soc.* **448**, 642–653 (2015)
36. Elyiv, A.A., Melnyk, O.V., Vavilova, I.B., et al.: Machine-learning computation of distance modulus for local galaxies. *Astron. Astrophys.* **635**, id.A124, 7 pp. (2020)
37. Fu, Z., Liu, S.: A Voronoi-tree based parallel range query method on distributed spatial database. *ISPRS Ann. Photogramm., Remote. Sens. Spat. Inf. Sci.* **12**, 37–43 (2012)
38. Gerke, B.F., Newman, J.A., Davis, M., et al.: The DEEP2 galaxy redshift survey: first results on galaxy groups. *Astrophys. J.* **625**(1), 6–22 (2005)

39. Gregul, A.I., Mandzhos, A.V., Vavilova, I.B.: The existence of the structural anisotropy of the Jagiellonian field of the galaxies. *Astrophys. Space Sci.* **185**, 223–235 (1991)
40. Grokhovskaya, A.A., Dodonov, S.N.: Large scale distribution of galaxies in the field HS 47.5-22. I. Data analysis technique. *Astrophys. Bulletin.* **74**, 379–387 (2019)
41. Guedj, B., Srinivasa Desikan, B.: Pycobra: a python toolbox for ensemble learning and visualisation. [arXiv:1707.00558](https://arxiv.org/abs/1707.00558)
42. Hidding, J., van de Weygaert, R., Vegter, G., Jones, B.J.T.: Adhesion and the geometry of the cosmic web. In: Thirteenth Marcel Grossmann Meeting: On Recent Developments in Theoretical and Experimental General Relativity, Astrophysics and Relativistic Field Theories, pp. 2142–2144 (2015)
43. Hettiarachchi, R., Peters, J.F.: Voronoi region-based adaptive unsupervised color image segmentation (2016). [arXiv:1604.00533](https://arxiv.org/abs/1604.00533)
44. Hogg, D.W., Blanton, M.R., Brinkmann, J., et al.: The dependence on environment of the color-magnitude relation of galaxies. *Astrophys. J.* **601**, L29 (2004)
45. Honda, C., Yasuda, Y., Yokota, Y.: Lunar crater spatial distribution for each surface model age. In: American Geophysical Union, Fall Meeting 2019, abstract P31C–3473 (2019)
46. Hubber, D.A., Ercolano, B., Dale, J.: Observing gas and dust in simulations of star formation with Monte Carlo radiation transport on Voronoi meshes. *Mon. Notic. Roy. Astron. Soc.* **456**(1), 756–766 (2016)
47. Hung, D., Lemaux, B.C., Gal, R.R., Tomczak, A.R., et al.: Establishing a new technique for discovering large-scale structure using the ORELSE survey. *Mon. Notic. Roy. Astron. Soc.*, Advance Access, pp. 1–39 (2019)
48. Icke, V., van de Weygaert, R.: Fragmenting the universe. 1. Statistics of two-dimensional Voronoi foams. *Astron. Astrophys.* **184**, 16–32 (1987)
49. Icke, V., van de Weygaert, R.: Voronoi foam as a model of the medium-scale universe. In: Large-Scale Structures in the Universe Observational and Analytical Methods: Proceedings of a Workshop, Held at the Physikzentrum Bad Honnef, vol. 310, pp. 257–266 (1988)
50. Icke, V., van de Weygaert, R.: The galaxy distribution as a Voronoi foam. *R. Astron. Soc., Q. J.* **32**, 85–112 (1991)
51. Ikeuchi, S., Turner, E.I.: Quasi-periodic structures in the large-scale galaxy distribution and three-dimensional Voronoi tessellation. *Mon. Notic. R. Astron. Soc.* **250**, 519–522 (1991)
52. Igashov, I., Olechnovic K., Kadukova, M., et al.: VoroCNN: deep convolutional neural network built on 3D Voronoi tessellation of protein structures (2020). [Arxiv:063586v1](https://arxiv.org/abs/063586v1)
53. Inkulu, R., Kapoor, S.: ANN queries: covering Voronoi diagram with hyperboxes. [arXiv:1111.5893](https://arxiv.org/abs/1111.5893)
54. Jackson, J.C.: A critique of Rees’s theory of primordial gravitational radiation. *Mon. Notic. Roy. Astron. Soc.* **156**, 1P (1972)
55. Karachentsev, I.D., Makarov, D.A.: The galaxy motion relative to nearby galaxies and the local velocity field. *Astron. J.* **111**, 794 (1996)
56. Kim, R.S.J., Kepner, J.V., Postman, M., et al.: Detecting clusters of galaxies in the sloan digital sky survey. I. Monte Carlo comparison of cluster detection algorithms. *Astron. J.* **123**(1), 20–36 (2002)
57. Lam, M.C., Hambly, N.C., Rowell, N., Chambers, K.C., et al.: The white dwarf luminosity functions from the Pan-STARRS 1 3π steradian survey. *Mon. Notic. R. Astron. Soc.* **482**(1), 715–731 (2019)
58. Lang, M., Holley-Bockelmann, K., Sinha, M., et al.: Voronoi tessellation and Non-parametric halo concentration. *Astrophys. J.* **811**(2), 9 (2015)
59. Lindenbergh, R.: Limits of Voronoi diagrams. PhD thesis, 132 (2002)
60. Ling, E.N.: New statistical approaches to galaxy clustering. PhD thesis, Sussex University, Brighton (England) (1987)
61. Lopes, P.A.A., de Carvalho, R.R., Gal, R.R., et al.: The northern sky optical cluster survey. IV. An Intermediate-redshift galaxy cluster catalog and the comparison of two detection algorithms. *Astron. J.* **128**(3), 1017–1045 (2004)

62. Marinoni, C., Davis, M., Newman, J.A., et al.: Three-dimensional identification and reconstruction of galaxy systems within flux-limited redshift surveys. *Astrophys. J.* **580**(1), 122–143 (2002)
63. Matsuda, T., Shima, E.: Topology of supercluster-void structure. *Progress Theoret. Phys.* **71**, 855–858 (1984)
64. Melnyk, O.V., Elyiv, A.A., Vavilova, I.B.: The structure of the local supercluster of galaxies detected by three-dimensional Voronoi's tessellation method. *Kinemat. Fiz. Neb. Tel* **22**, 283–296 (2006)
65. Melnyk, O.V., Elyiv, A.A., Vavilova, I.B.: 3-D Voronoi's tessellation as a tool for identifying galaxy groups. Galaxy evolution across the hubble time. In: Combes, F., Palous, J. (eds.) *Proceedings of IAU Symposium*, vol. 235, pp. 223–223 (2006)
66. Melnyk, O.V., Elyiv, A.A., Vavilova, I.B.: Mass-to-light ratios for galaxy pairs and triplets in various environments. *Kinemat. Phys. Celest. Bodies.* **25**, 43–47 (2009)
67. Melnyk, O.V., Dobrycheva, D.V., Vavilova, I.B.: Morphology and color indices of galaxies in Pairs: criteria for the classification of galaxies. *Astrophysics* **55**, 293–305 (2012)
68. Neyrinck, M.C.: ZOBOV: a parameter-free void-finding algorithm. *Mon. Notic. R. Astron. Soc.* **386**, 2101–2109 (2008)
69. Neyrinck, M.C.: The cosmic spiderweb and general origami tessellation design (2008). [arXiv:1809.00015](https://arxiv.org/abs/1809.00015)
70. O'Mill, A.L., Duplancic, F., Lambas, G.D., et al.: Galaxy triplets in sloan digital sky survey data release 7 - I. Catalogue. *Mon. Notic. R. Astron. Soc.* **421**, 1897–1907 (2012)
71. Panko, E., Flin, P.: A catalogue of galaxy clusters and groups based on the muenster red sky survey. *J. Astron. Data* **12**, 1P (2006)
72. Paranjape, A., Alam, S.: Voronoi volume function: a new probe of cosmology and galaxy evolution. *Mon. Notic. R. Astron. Soc.* **495**, 3233–3251 (2020)
73. Park, C., Choi, Y.-Y., Vogeley, M.S., et al.: Environmental dependence of properties of galaxies in the sloan digital sky survey. *Astrophys. J.* **658**, 898–916 (2007)
74. Pereira, S., Campusano, L.E., Hitschfeld-Kahler, N., et al.: A 3D Voronoi+Gapper galaxy cluster finder in redshift space to $z \approx 0.2$ I: an algorithm optimized for the 2dFGRS. *Astrophys. J.* **838**(2), 109 (2017)
75. Pereira, S., Campusano, L.E., Hitschfeld-Kahler, N., et al.: A 3D Voronoi+Gapper galaxy cluster finder in redshift space to $z \approx 0.2$ I: an algorithm optimized for the 2dFGRS. *Astrophys. J.* **838**(2), 18 (2017)
76. Pratsyovity, M.V., Syta, H.M.: Geometric mosaics of the Great Ukrainian (to the 150th anniversary of Professor G. Voronoi). *Visnyk of the NAS of Ukraine*, 4, 92–101 (2018)
77. Pulatova, N.G., Vavilova, I.B., Sawangwit, U., et al.: The 2MIG isolated AGNs - I. General and multiwavelength properties of AGNs and host galaxies in the northern sky. *Mon. Notic. R. Astron. Soc.* **447**, 2209–2223 (2015)
78. Ramella, M., Boschin, W., Fadda, D., Nonino, M.: Finding galaxy cluster using Voronoi tessellations. *Astron. Astrophys.* **368**, 776–786 (2001)
79. San Roman, I., Cenarro, A.J., Díaz-García, L.A., et al.: The ALHAMBRA survey: 2D analysis of the stellar populations in massive early-type galaxies at $z < 0.3$. *Astron. Astrophys.* **609**, A20
80. Santiago-Bautista, I., Caretta, C.A., Bravo-Alfaro, H., et al.: Identification of filamentary structures in the environment of superclusters of galaxies in the local universe. *Astron. Astrophys.* **637**, id.A31, 26 (2020)
81. Schlegel, D.J., Finkbeiner, D.P., Davis, M.: Maps of dust infrared emission for use in estimation of reddening and cosmic microwave background radiation foregrounds. *Astrophys. J.* **500**, 525–553 (1998)
82. Scoville, N., Arnouts, S., Aussel, H., et al.: Evolution of galaxies and their environments at $z = 0.1 - 3$ in COSMOS. *Astrophys. J. Suppl.* **206**(1), 3 (2013)
83. Sheth, R.K., van de Weygaert, R.: A hierarchy of voids: much ado about nothing. *Mon. Notic. R. Astron. Soc.* **350**, 517–538 (2004)

84. Soares-Santos, M., de Carvalho, R.R., Annis, J., Gal, R.R., et al.: The Voronoi tessellation cluster finder in 2+1 dimensions. *Astrophys. J.* **727**(1), 14 (2011)
85. Söchtig, I.K., Coldwell, G.V., Clowes, R.G., et al.: Ultra deep catalogue of galaxy structures in the cosmic evolution survey field. *Mon. Notic. R. Astron. Soc.* **423**, 2436–2450 (2012)
86. Springel, V.: E pur si muove: galilean-invariant cosmological hydrodynamical simulations on a moving mesh. *Mon. Notic. R. Astron. Soc.* **401**, 791–851 (2010)
87. Subba, R.M.U., Szalay, A.S.: Statistics of pencil-beams in Voronoi foams. *Astrophys. J.* **391**, 483–493 (1992)
88. Sutter, P.M., Lavaux, G., Hamaus, N., Pisani, A., et al.: VIDE: the void identification and examination toolkit. *Astron. Comput.* **9**, 1–9 (2015)
89. Syta, H., van de Weygaert, R.: Life and Times of Georgy Voronoi (2009). [arXiv:0912.3269](https://arxiv.org/abs/0912.3269)
90. Tal, T., Dekel, A., Oesch, P., et al.: Observations of environmental quenching in groups in the 11 Gyr since $z = 2.5$: different quenching for central and satellite galaxies. *Astrophys. J.* **789**, 1–11 (2014)
91. Trimble, V.: Nor yet the last to lay the old aside: structuring the something. In: American Astronomical Society Meeting Abstracts, vol. 223, p. 90.01 (2014)
92. Vavilova, I.B.: An investigation of large-scale galaxy distribution in the Local Supercluster and the Jagellonian Field by the methods of cluster, fractal and wavelet analysis. PhD Thesis in Phys.-Math. Sciences, Main Astronomical Observatory, NAS of Ukraine, 222 p. (1995)
93. Vavilova, I., Melnyk, O.: Voronoi tessellation for galaxy distribution. In: Proceedings of the Third Voronoi Conference on Analytic Number Theory and Spatial Tessellations, vol. 55, pp. 203–212 (2005)
94. Vavilova, I.B., Karachentseva, V.E., Makarov, D.I., Melnyk, O.V.: Triplets of galaxies in the local supercluster. I. Kinematic and virial parameters. *Kinemat. Fiz. Neb. Tel.* **1**, 3–20 (2005)
95. Vavilova, I.B., Melnyk, O.V., Elyiv, A.A.: Morphological properties of isolated galaxies vs. isolation criteria. *Astron. Nachr.* **330**, 1004–1009 (2009)
96. Vavilova, I.B., Bolotin, YuL., Boyarsky, A.M., et al.: Dark matter: Observational Manifestation and Experimental Searches. *Akadempriodyka*, Kyiv (2015)
97. Vavilova, I.B., Elyiv, A.A., Vasylenko, MYu.: Behind the zone of avoidance of the milky way: what can we restore by direct and indirect methods? *Radiophys. Radioastron.* **23**, 244–257 (2018)
98. Vavilova, I., Pakuliak, L., Babyk, I., et al.: Surveys, catalogues, databases, and archives of astronomical data. In: Scoda, P., Adam, F. (eds.) *Knowledge Discovery in Big Data from Astronomy and Earth Observation*, pp. 57–102. Elsevier, Amsterdam (2020)
99. Vavilova, I., Dobrycheva, D., Vasylenko, M., et al.: Multiwavelength extragalactic surveys: examples of data mining. In: Scoda, P., Adam, F. (eds.) *Knowledge Discovery in Big Data from Astronomy and Earth Observation*, pp. 307–323. Elsevier, Amsterdam (2020)
100. Vavilova, I.B.: Astroinformatics of the large-scale structures of the Universe. Dr. Hab. Thesis in Phys.-Math. Sciences, Main Astronomical Observatory, NAS of Ukraine, 388 p. (2020)
101. Voronoi, G.: Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier Mémoire. Sur quelques propriétés des formes quadratiques positives parfaites. *J. reine angew. Math.* **133**(2), 97–156 (1907); **133**(3), 157–158 (1907)
102. Voronoi, G.: Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième Mémoire. Recherches sur les parallélogrammes primitifs. *J. reine angew. Math.* **134**(3), 198–246 (1908); **134**(4), 247–287 (1908); **136**(2), 67–178 (1909)
103. Weinberger, R., Springel, V., Pakmor, R.: The AREPO public code release. *Astrophys. J. Suppl.* **248**(2), 32 (2020)
104. van de Weygaert, R., Icke, V.: Fragmenting the universe. II - Voronoi vertices as Abell clusters. *Astron. Astrophys.* **213**, 1–9 (1989)
105. van de Weygaert, R., Aragon-Calvo, M.A., Jones, B.J.T., et al.: Geometry and morphology of the cosmic web: analyzing spatial patterns in the universe (2009). [arXiv:0912.3448](https://arxiv.org/abs/0912.3448)
106. Ying, S., Guang, X., Chengpeng, L., et al.: Point cluster analysis using a 3D Voronoi diagram with applications in point cloud segmentation. *ISPRS Int. J. Geo-Inf.* **4**, 1480–1499 (2015)

107. Yoshioka, S., Ikeuchi, S.: The large-scale structure of the universe and the division of space. *Astrophys. J.* **341**, 16–25 (1989)
108. Zaninetti, L.: Dynamical Voronoi tessellation. I. *Astron. Astrophys.* **224**, 345–350 (1989)
109. Zaninetti, L.: Dynamical Voronoi tessellation. II. *Astron. Astrophys.* **233**, 293–300 (1990)
110. Zaninetti, L.: Practical statistics for the voids between galaxies. *Serb. Astron. J.* **181**, 19–29 (2010)
111. Zaninetti, L.: A geometrical model for the catalogs of galaxies. *Revista Mexicana de Astronomía y Astrofísica* **46**, 115–134 (2010)
112. Zaninetti, L.: New analytic results for poissonian and non-poissonian statistics of cosmic voids. *Revista Mexicana de Astronomía y Astrofísica* **48**, 209–222 (2012)
113. Zaninetti, L.: Photometric effects and Voronoi-diagrams as a mixed model for the spatial distribution of galaxies. *Open Astron. J.* **6**, 48–71 (2013)

Statistical Characterization and Classification of Astronomical Transients with Machine Learning in the era of the Vera C. Rubin Observatory



Marco Vicedomini, Massimo Brescia, Stefano Cavuoti, Giuseppe Riccio, and Giuseppe Longo

Preprint version of the manuscript to appear in the Volume “Intelligent Astrophysics” of the series “Emergence, Complexity and Computation”, Book eds. I. Zelinka, D. Baron, M. Brescia, Springer Nature Switzerland, ISSN: 2194-7287

Abstract Astronomy has entered the multi-messenger data era and Machine Learning has found widespread use in a large variety of applications. The exploitation of synoptic (multi-band and multi-epoch) surveys, like LSST (Legacy Survey of Space and Time), requires an extensive use of automatic methods for data processing and interpretation. With data volumes in the petabyte domain, the discrimination of time-critical information has already exceeded the capabilities of human operators and crowds of scientists have extreme difficulty to manage such amounts of data in multi-dimensional domains. This work is focused on an analysis of critical aspects related to the approach, based on Machine Learning, to variable sky sources classification, with special care to the various types of Supernovae, one of the most important subjects of Time Domain Astronomy, due to their crucial role in Cosmology. The work is based on a test campaign performed on simulated data. The classification was carried out by comparing the performances among several Machine Learning algorithms on statistical parameters extracted from the light curves. The results make in evidence some critical aspects related to the data quality and their parameter space characterization, propaedeutic to the preparation of processing machinery for the real data exploitation in the incoming decade.

M. Vicedomini · S. Cavuoti (✉) · G. Longo

Department of Physics, University of Naples Federico II, Strada Vicinale Cupa Cintia, 21, 80126 Napoli, Italy

e-mail: stefano.cavuoti@inaf.it

M. Brescia · S. Cavuoti · G. Riccio

INAF - Astronomical Observatory of Capodimonte, Salita Moiariello 16, 80131 Napoli, Italy

e-mail: massimo.brescia@inaf.it

1 Introduction

The scientific topics covered in this work falls within what is called Time Domain Astronomy. This is the study of variable sources, i.e. astronomical objects whose light changes with time. Although the taxonomy of such sources is extremely rich, there are two main kinds of objects, respectively, transients and variables. The first changes its nature during the event, while the second presents just a brightness variation. The study of these phenomena is fundamental to identify and analyze either the mechanisms causing light variations and the progenitors of the various classes of objects.

Since ancient times the phenomenon of Supernovae (SNe) has fascinated human beings, but only recently we understood, in most cases, why and how this explosion happens [1]. Obviously there are still many open questions, but the knowledge about the type of galaxy hosting various kinds of Supernova and at which rate they take place, could help us to better understand this phenomenon and many other related properties of the Universe [2].

For example, the observed luminosity dispersion of SNe is evidenced through inhomogeneities in the weak lensing event and this is an upper limit on the cosmic matter power spectrum. Massive cosmological objects like galaxies and clusters of galaxies can magnify many times the flux of events like SNe that would be too faint to detect and bring them into our analysis scope. Studies on lensed SNe type Ia by clusters of galaxies may be used to probe the distribution of dark matter on them. Time delay between the multiple images of lensed SNe could provide a good estimates of its high redshift. Furthermore there are two factors that makes SNe better than other sources, like quasars, in measuring time delay [3]: (i) if the Supernovae is taken before the peak, the measurements are easier and on short timescale compared to the quasars; (ii) the SN light fade away with time, so we can measure the lens stellar kinematics and the dynamics lens mass modeling. In the next decade, the Vera C. Rubin Observatory will perform the Rubin Observatory Legacy Survey of Space and Time (LSST), using the Rubin Observatory LSST Camera and the Simonyi Survey Telescope. LSST will play a key role in the discovery of new lensed SNe Ia [4]. LSST will help to find apparently host-less SNe of every type, and this may help to study dwarf galaxies with a mass range of $10^4 \div 10^6$ solar masses. These galaxies, indeed, play a key role in large scale structure models, and despite their very big predicted population, over 1 Mpc we cannot see them until now. Same story for the theorized intracluster population of stars stripped from their galaxies, which could be seen through the SNe host-less events.

In order to understand and push ourselves further and further into the universe, ever more powerful incoming observing instruments, like LSST, will be able to deliver impressive amounts of data, for which astronomers are obliged to make an intensive use of automatic analysis systems. Methods that fall under the heading Data Mining and Machine Learning have now become commonplace and indispensable to the work of scientists [5–7]. But then, where the human work is still needed? For sure in terms of final analysis and validation of the results. This thesis work is therefore based

on this virtuous combination, by exploiting data science methodology and models, such as Random Forest [8], Nadam, RMSProp and Adadelta [9], to perform a deep investigation on time domain astronomy, by focusing the attention on Supernovae classification, performed on realistic sky simulations. Furthermore, a special care has been devoted to the parameter space analysis, through the application of the method Φ LAB [10, 11] to the various classification experiments, in order to evaluate the commonalities among them in terms of features found as relevant to solve the recognition of different types of transients.

In Sect. 2 we describe the two data simulations used for the experiments and the extracted statistical features composing the data parameter space. In Sect. 3 we give a brief introduction of the ML methods used, while in Sect. 4 the series of experiments performed are deeply reported. Finally, in Sect. 5 we analyze the results and draw the conclusions.

2 Data

In this work two simulation datasets were used; the Supernova Photometric Classification Challenge (hereafter SNPhotCC, [12]) and the Photometric LSST Astronomical Time-Series Classification Challenge (hereafter PLAsTiCC, [13–15]).

2.1 The SNPhotCC Simulated Catalogue

This catalogue was the subject of a challenge performed in 2010 and consists of a mixed set of simulated SN types, respectively, Ia, Ibc and II, selected by respecting the relative rate (Table 1). The volumetric rate was found by Dilday et al. [16] as $r_v = \alpha(1+z)^\beta$, where for SNe Ia parameters we have $\alpha_{Ia} = 2.6 \times 10^{-5} \text{Mpc}^{-3} h_{70}^3 \text{yr}^{-1}$, $\beta_{Ia} = 1.5$ and $h_{70} = H_0/(70 \text{ kms}^{-1} \text{Mpc}^{-1})$. H_0 is the present value of the Hubble parameter. For non Ia SNe, the parameters come from Bazin et al. [17] and are $\alpha_{NonIa} = 6.8 \times 10^{-5} \text{Mpc}^{-3} h_{70}^3 \text{yr}^{-1}$ and $\beta_{NonIa} = 3.6$. The simulation is based on four bands, *griz*, with cosmological parameters $\Omega_M = 0.3$, $\Omega_\Lambda = 0.7$ and $\omega = -1$, where Ω_M is the density of barionic and dark matter, Ω_Λ is the density of dark energy and ω is the cosmological constant. Moreover, the point-spread function,

Table 1 SNPhotCC dataset composition

Types	Bands	Sampling	%	Amount
SN Ia	g, r, i, z	Uneven	23,86	5088
SN Ibc	g, r, i, z	Uneven	13,14	2801
SN II	g, r, i, z	Uneven	63	13430

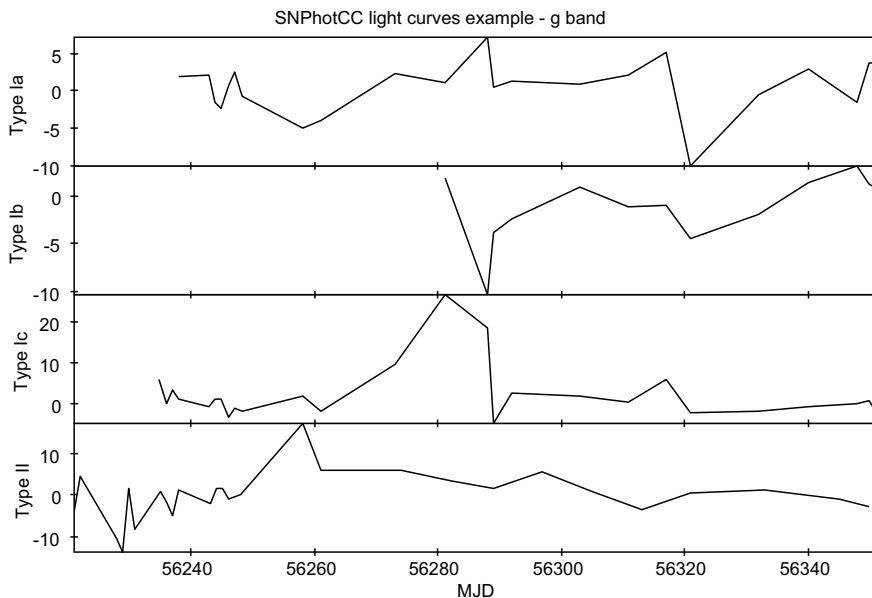


Fig. 1 Examples of SNPhotCC light curves in g band. From the top to the bottom: SN004923(Ia), SN000760(Ib), SN003475(Ic), SN001986(II)

atmospheric transparency and sky-noise were measured in each filter and epoch using the one-year chronology.

The dataset sources are based on two variants, respectively, with or without the host-galaxy photometric redshift. For this work only the samples without redshift information were used.

Every simulated light curve has at least one observation, in two or more bands, with signal-to-noise ratio >5 and five observations after the explosion (Fig. 1). A spectroscopically confirmed training subset was provided; it was based on observations from a 4 m class telescope with a limiting r -band magnitude of 21.5 and on observations from an 8 m class telescope with a limiting i -band magnitude of 23.5.

2.2 The PLAsTiCC Simulated Catalogue

This catalogue arises from a challenge focused on the future use of the LSST,¹ by simulating the possible objects on which science will be based. In particular, most of these objects are transients.

LSST will be the largest telescope specialized for the Time Domain Astronomy, whose first light is foreseen in late 2020. Its field of view will be ~ 3.5 degrees (the

¹<https://www.kaggle.com/c/PLAsTiCC-2018>

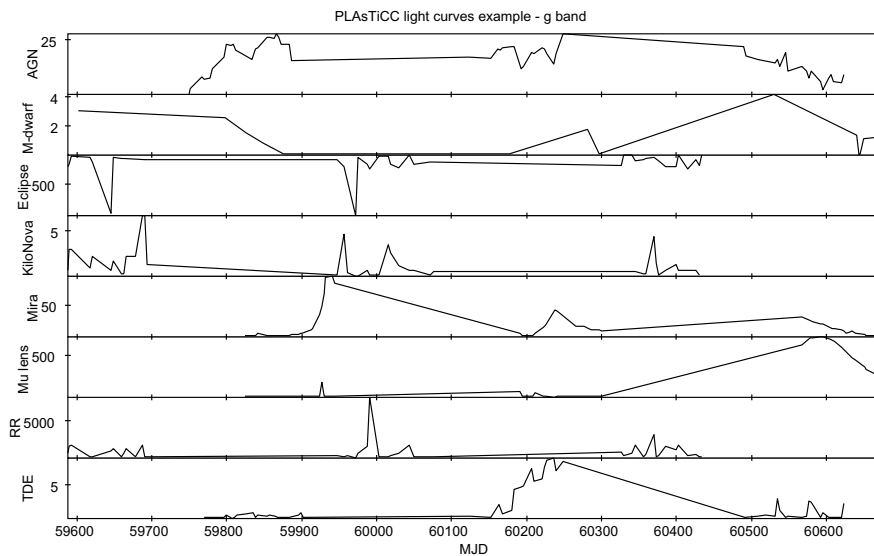


Fig. 2 Examples of PLAsTiCC light curves in g band. From the top to the bottom: 2198 (AGN), 2157270 (M-Dwarf), 22574 (Eclipsing Binary), 139362 (Kilonova), 80421 (Mirae), 45395 (μ -lens), 184176 (RR Lyrae), 9197 (TDE)

diameter will be about seven full moons side by side), with a 6.5m effective aperture, a focal ratio of 1.23 and a camera of 3.2 Gigapixel.

Every four nights it will observe the whole sky visible from the Chile (southern hemisphere). Therefore, it will find an unprecedented amount of new transients: Supernovae Ia, Ia-91bg, Iax, II, Ibc, SuperLuminous (SL), Tidal Disruption Events, Kilonova, Active Galactic Nuclei, RR Lyrae, M-dwarf stellar flares, Eclipsing Binary and Pulsating variable stars, μ -lens from single lenses, μ -lens from binary lenses, Intermediate Luminosity Optical Transients, Calcium Rich Transients and Pair Instability Supernovae.

LSST data will be used for studying stars in our Galaxy, understanding how solar systems and galaxies formed and the role played by massive stars in galaxy chemistry as well as measuring the amount of matter in the Universe. PLAsTiCC includes light curves with realistic time-sampling [15], noise properties and realistic astrophysical sources.

Each object has observations in six bands: u (300 \div 400 nm), g (400 \div 600 nm), r (500 \div 700 nm), i (650 \div 850 nm), z (800 \div 950 nm), and y (950 \div 1050 nm). The training set is a mixture of what we can expect to have before LSST, so it is a quite homogeneous ensemble of ~ 8000 objects; the test set, instead, is based on what we expect to have after 3 years of LSST operations and it is formed by ~ 3.5 million of objects. The observations are limited in magnitude in single band to 24.5 in the r band and to 27.8 r stacked band (see Figs. 2 and 3 for examples of light curves). By combining training and test, we collected the objects per class as listed in Table 2.

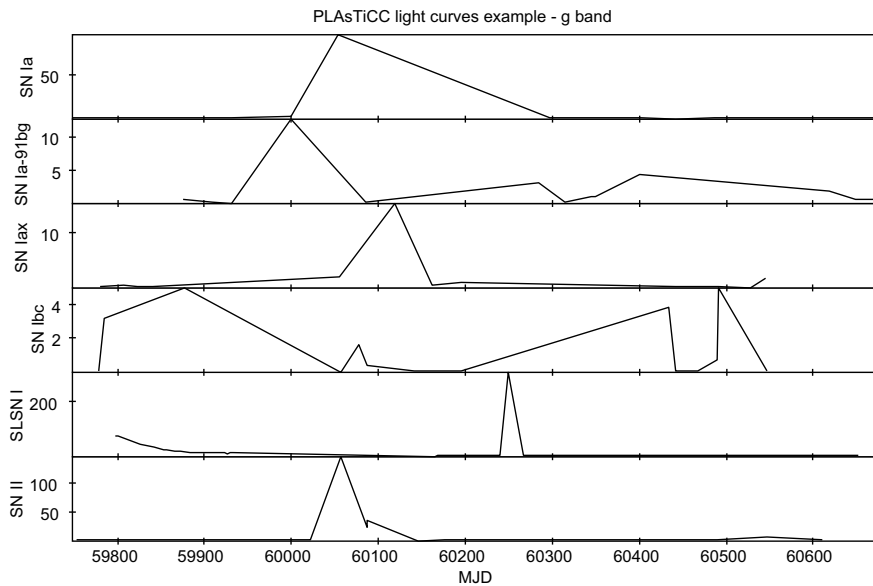


Fig. 3 Examples of PLAsTiCC light curves in g band. From the top to the bottom: 15461391(SNIa), 1143209(SNIa-91bglike), 1019556(SNIax), 1076072(SNIbc), 73610(SLSN I), 1028853(SNIi)

Table 2 PLAsTiCC dataset composition

Types	Training	Test	Bands	Sampling	%	Amount
SNIa	2313	1659831	u, g, r, i, z, y	Uneven	47.57	1662144
SNIax	183	63664	u, g, r, i, z, y	Uneven	1.81	63847
SNIa 91bglike	208	40193	u, g, r, i, z, y	Uneven	1.15	40401
SNIbc	484	175094	u, g, r, i, z, y	Uneven	5.00	175578
SNIi	1193	1000150	u, g, r, i, z, y	Uneven	28.65	1001343
SLSN I	175	35782	u, g, r, i, z, y	Uneven	1.02	35957
AGN	370	101424	u, g, r, i, z, y	Uneven	2.89	101794
M-Dwarf	981	93494	u, g, r, i, z, y	Uneven	2.68	94475
RR Lyrae	239	197155	u, g, r, i, z, y	Uneven	5.63	197394
Mirae	30	1453	u, g, r, i, z, y	Uneven	0.04	1483
Eclipse	924	96572	u, g, r, i, z, y	Uneven	2.77	97496
KN	100	131	u, g, r, i, z, y	Uneven	0.01	231
TDE	495	13555	u, g, r, i, z, y	Uneven	0.38	14050
μ Lens	151	1303	u, g, r, i, z, y	Uneven	0.04	1454
Other	0	13087	u, g, r, i, z, y	Uneven	0.36	13087

2.3 The Statistical Parameter Space

In order to evaluate the classification performances, the light curves of the objects have been subject of a statistical approach, by transforming them into a set of features representing some peculiar characteristics of the astrophysical objects. Within this work we used the following features (already used in a similar task in [18]), resulting from a preliminary mapping of variable object light curves into a statistical parameter space:

- Amplitude (ampl): the arithmetic average between the maximum and the minimum magnitude,

$$ampl = \frac{mag_{max} - mag_{min}}{2} \quad (1)$$

- Beyond1std (b1std): the fraction of photometric points above or under one standard deviation from the weighted average,

$$b1std = P(|mag - \overline{mag}| > \sigma) \quad (2)$$

- Flux Percentage Ratio (fpr): the ratio between two flux percentiles $F_{n,m}$. The flux percentile is defined as the difference between the flux value at percentiles n and m , respectively. For this work, the following fpr values have been used:

$$\begin{aligned} fpr20 &= F_{40,60}/F_{5,95} \\ fpr35 &= F_{32,5,67,5}/F_{5,95} \\ fpr50 &= F_{25,75}/F_{5,95} \\ fpr65 &= F_{17,5,82,5}/F_{5,95} \\ fpr80 &= F_{10,90}/F_{5,95} \end{aligned}$$

- Lomb-Scargle Periodogram (ls): the period obtained by the peak frequency of the Lomb-Scargle periodogram.
- Linear Trend (lt): the slope a of the light curve in the linear fit,

$$\begin{aligned} mag &= a * t + b \\ lt &= a \end{aligned} \quad (3)$$

- Median Absolute Deviation (mad): the median of the deviation of fluxes from the median flux,

$$mad = median_i(|x_i - median_j(x_j)|) \quad (4)$$

- Median Buffer Range Percentage (mbrp): the fraction of data points which are within 10% of the median flux,

$$mbrp = P(|x_i - median_j(x_j)| < 0.1 * median_j(x_j)) \quad (5)$$

- Magnitude Ratio (mr): an index to see if the majority of data points are above or below the median of the magnitudes,

$$mr = P(mag > median(mag)) \quad (6)$$

- Maximum Slope (ms): the maximum difference obtained measuring magnitudes at successive epochs,

$$ms = max(|\frac{(mag_{i+1} - mag_i)}{(t_{i+1} - t_i)}|) = \frac{\Delta mag}{\Delta t} \quad (7)$$

- Percent Difference Flux Percentile (pdfp): the difference between the fifth and the 95th percentile flux, converted in magnitudes, divided by the median flux,

$$pdfp = \frac{(mag_{95} - mag_5)}{median(mag)} \quad (8)$$

- Pair Slope Trend (pst): the percentage of the last 30 couples of consecutive measures of fluxes that show a positive slope,

$$pst = P(x_{i+1} - x_i > 0, i = n - 30, \dots, n) \quad (9)$$

- R Cor Bor (rcb): the fraction of magnitudes that is above 1.5 magnitudes with respect to the median,

$$rcb = P(mag > (median(mag) + 1.5)) \quad (10)$$

- Small Kurtosis (kurt): the ratio between the 4th order momentum and the square of the variance. For small kurtosis it is intended the kurtosis on a small number of epochs,

$$kurt = \frac{\mu_4}{\sigma^2} \quad (11)$$

- Skewness (skew): the ratio between the 3rd order momentum and the variance to the third power,

$$skew = \frac{\mu_3}{\sigma^3} \quad (12)$$

- Standard deviation (std): the standard deviation of the flux.

3 Machine Learning Models

A classifier can be used as a descriptive model to distinguish among objects of different classes, and as a predictive model to predict the class label of input patterns. Classification techniques work better for predicting or describing data sets with binary or nominal categories. Each technique uses a different learning algorithm to find a model that fits the relationship between the feature set and class labels of the input data. The goal of the learning algorithm is to build models with good generalization capability. The typical approach of machine learning models is to randomly shuffle and split the given input dataset with known assigned class labels into three subsets: training, validation and blind test sets. The validation set can be used to validate the learning process, while the test set is used blindly to verify the trained model performance and generalization capabilities. In the following sections we briefly introduce the methods used to perform the classification experiments, together with the statistical estimators adopted to evaluate their performances.

3.1 *The Random Forest Classifier*

A Random Forest (RF, [8]) is a classifier consisting of a collection of tree-structured classifiers $\{h(x, \Theta_k), k = 1, \dots\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x [19]. The generalization error for this algorithm depends on the strength of single trees and from their correlations through the raw margin functions. The upper bound, instead, tell us that smaller the ratio of those quantities, better the RF performance. To improve the model accuracy by keeping trees strength, the correlation between trees is decreased and bagging with a random selection of features is adopted. Bagging or Bootstrap Aggregating, is a method designed to improve the stability and accuracy of machine learning algorithms. It also reduce variance and minimizes the risk of overfitting. Given a training set of size n , bagging generates m new training sets, each of size p , by sampling from the original one uniformly and with replacement. This kind of sampling is known as a bootstrap sample. The m models are fitted using the m bootstrap samples and combined by averaging the output (for regression) or voting (for classification). Bagging is useful because, in addition to improving accuracy when using random features, it provides an estimate of the generalized error of the set of trees and the strength and correlation of trees. The estimation is done out-of-bag. Out-of-bag means that the error estimate of each pair (x, y) is made on all those bagging datasets that do not contain that given pair.

3.2 The Nadam, RMSProp and Adadelta Classifiers

The simplest optimization algorithm is the *Gradient Descent*, in which the gradient of the function to be minimized is calculated. This depends on the parameter θ_{t-1} . Only a portion of the gradient is used to update the parameters; this portion is given by the parameter η :

$$\begin{cases} g_t \leftarrow \nabla_{\theta_{t-1}} f(\theta_{t-1} - \eta \mu m_{t-1}) \\ m_t \leftarrow \mu m_{t-1} + g_t \\ \theta_t \leftarrow \theta_{t-1} - \eta m_t \end{cases}$$

where m is the so-called *momentum vector*, used to accelerate the update of the learning function, while μ is the decay constant. These two terms increase the speed of gradient decreasing in the direction where the gradient tends to remain constant, while reducing it where the gradient tends to oscillate.

Nadam is a modified version of the Adam algorithm, based on the combination between the momentum implementation and the L_2 normalization. This type of normalization changes the η member, dividing it by the L_2 norm of all previous gradients.

Adadelta is a variant that tries to reduce the aggressive, monotonically decreasing learning rate. In fact, instead of accumulating all past squared gradients, it restricts the window of accumulated past gradients to some fixed size w . This has the advantage of compensating for the speeds along the different dimensions by stabilizing the model on common features and allowing the rare ones to emerge. A problem of this algorithm comes from the norm vector that could become so large to stop the training, preventing the model from reaching the local minimum. This problem is solved by RMSProp, a L_2 normalization based algorithm, which replaces the sum of n_t with a decaying mean, characterized by a constant value ν . This allows the model to avoid any stop of the learning process. For a detailed description of these models, see [9].

3.3 Parameter Space Exploration

The choice of an optimal set of features is connected to the concept of feature importance, based on the measure of a feature's relevance [11]. Formally, the importance or relevance of a feature is its percentage of informative contribution to a learning system. We approached the feature selection task in terms of the *all-relevant* feature selection, able to extract the most complete parameter space, i.e. all features considered relevant for the solution to the problem. This is appropriate for problems with highly correlated features, as these features will contain nearly the same information. With a minimal-optimal feature selection, choosing any one of them (which could happen at random if they are perfectly correlated), means that the rest will never be selected. The method Φ LAB, deeply discussed in [11], includes properties of both

embedded and wrappers categories of feature selection to optimize the parameter space, by solving the *all-relevant* feature selection problem, thus indirectly improving the physical knowledge about the problem domain.

3.4 Classification Statistics

In this work, the performance of the classification models is based on some statistical estimators, extracted from a matrix known as *confusion matrix* [20].

The example shown in Table 3 is a confusion matrix for a binary classification. Each entry a_{ij} in this table is the number of records from class i predicted to be of class j . The numbers a_{00} and a_{11} show correct classified records. The a_{01} records named *False Positive* indicate wrong records classified in class 0, when their correct classification was class 1; instead, a_{10} named *False Negative* show the records classified in class 1 but belonging to class 0. The total number of correct predictions is $a_{11} + a_{00}$, and the total number of wrong ones is $a_{10} + a_{01}$. For a better comparison between different models, summarizing the results through a confusion matrix is the common way. We can do this using a *performance metric*, such as *accuracy*, defined as follows:

$$Accuracy = \frac{a_{00} + a_{11}}{a_{00} + a_{11} + a_{01} + a_{10}}$$

A highest accuracy is the target of every classifier. Other important statistical estimators, for a better understanding of the results for each class, are:

$$Purity = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Completeness = \frac{TruePositive}{TruePositive + FalseNegative}$$

$$Contamination = 1 - Purity = \frac{FalsePositive}{TruePositive + FalsePositive}$$

Table 3 Example of a binary confusion matrix

		Predicted	
		P=0	N=1
Target	p=0	a_{00}	a_{10}
	n=1	a_{01}	a_{11}

$$F1_{score} = \frac{2}{(Purity)^{-1} + (Completeness)^{-1}}$$

Purity of a class is the percentage of correctly classified objects in that class, divided by the total classified objects in that class. Also named as precision of a class.

Completeness of a class is the percentage of the correctly classified objects in that class divided by the total amount of objects belonging to that class. Also named as recall of a class.

Contamination of a class is the dual measure of purity.

F1-Score of a class is the harmonic mean between purity and completeness of that class and it is a measure of the average trade-off between purity and completeness.

4 Experiments

In order to pursue the main goal of the present work, related to a deep analysis of SNe in terms of their classification and characterization of the parameter space required to recognize their different types, we relied on the two simulation datasets, one in particular developed and specialized within the LSST project (see Sects. 2.1 and 2.2). We preferred a statistical approach, by mapping the light curves into a set of statistical features. The classification with statistical data have been performed through the comparison of different types of classifiers, respectively, Nadam, RMSProp, Adadelata and Random Forest.

A data pre-processing phase was carried out on the PLAsTiCC dataset, based on a pruning on the flux and related error, in order to reduce the amount of negative fluxes present within data, which could affect the learning capability of the machine learning models. On the SNPhotCC dataset, both the errors in the flux and the quantity of negative fluxes were such that it was not deemed necessary to perform the pruning. The curves in the PLAsTiCC dataset were selected in successive steps so as to minimize the presence of negative fluxes, reaching, where possible, a subset of about 35,000 light curves per type. In the SNPhotCC dataset, on the other hand, all the given 5088 SN-Ia curves were selected and the type II curves were reduced so as to balance the classes; the other types of SNe have been discarded, due to their negligible amount available.

The sequence of classification experiments followed an incremental complexity, starting from the most simple exercise on the PLAsTiCC dataset, i.e. the separation between periodic and non-periodic objects (*P Vs NP*), expected to be well classified due to their very different features within any parameter space. In terms of initial minimization of negative fluxes, it was decided to apply the following replacement: for each class of objects, the observations related to the same day were grouped, by taking the least positive flux value. This value has been replaced to all the negative fluxes of that day.

As expected, the classifiers revealed a high capability to disentangle periodic from non-periodic objects. Therefore, in all further experiments we excluded periodic sources, by focusing the exclusive attention to variable objects, increasing the complexity of classification, by considering different sub-classes of transients and evaluating the performances of the selected machine learning classifiers.

The next step was, in fact, to recognize the SNe from all the other non-periodic objects available in the dataset (*SNe Vs All*). But, preliminarily, we tested different methods for replacing the negative fluxes. For instance, in addition to the first mentioned method (e.g. minimum positive flux extracted from observations within the same day), a second method was chosen, in which negative fluxes were replaced by the constant number 0.001, considered as the absolute minimum flux emitted by the sources. We tried also a third method, in which the negative fluxes were simply excluded from the input dataset, without any replacement. In theory, such third method was considered the worst case, since it would cause a drastic reduction of the light curve sample available. As we will show, the second method (the constant minimum flux value), obtained the best classification performances for all classifiers. Therefore, it was used as the reference for all further classification experiments.

The subsequent experiments concerned some fine classifications of most interesting SNe types, starting from the classic case of *SNIa versus SNII* types, followed by a mix of *SNIa versus Superluminous SNe I (SNIa Vs SL-I)*, concluding with the most complex case, based on the multi-class experiment, in which we tried to simultaneously classify all six different types of SNe (*six-class SNe*).

Besides the negative flux replacement, we investigated also the feature selection problem, in order to identify the most significant parameter space able to recognize different types of SNe. After the selection process we verified that such reduced amount of data dimensions could maintain sufficiently high the classification performances. We tried also to maintain uniform the number of features among the different use cases, although respecting their statistical importance, exploring the possibility to find a common parameter space, suitable for all classification cases.

The *SNIa versus SNII* use case was also performed on the SNPhotCC dataset, since this dataset was composed almost exclusively by such two types of SNe. The results were then compared with those performed on the PLAsTiCC dataset, deprived of the *u* and *y* bands for uniformity with the SNPhotCC dataset bands, in order to maximize the fair comparison.

In summary, in this work five series of experiments were performed on the PLAsTiCC dataset and one on the SNPhotCC dataset. Such experiments were chosen hierarchically and considering the most important goal of this work, i.e. the fine classification of SNe types. An overview of the followed procedure is shown in Fig. 4.

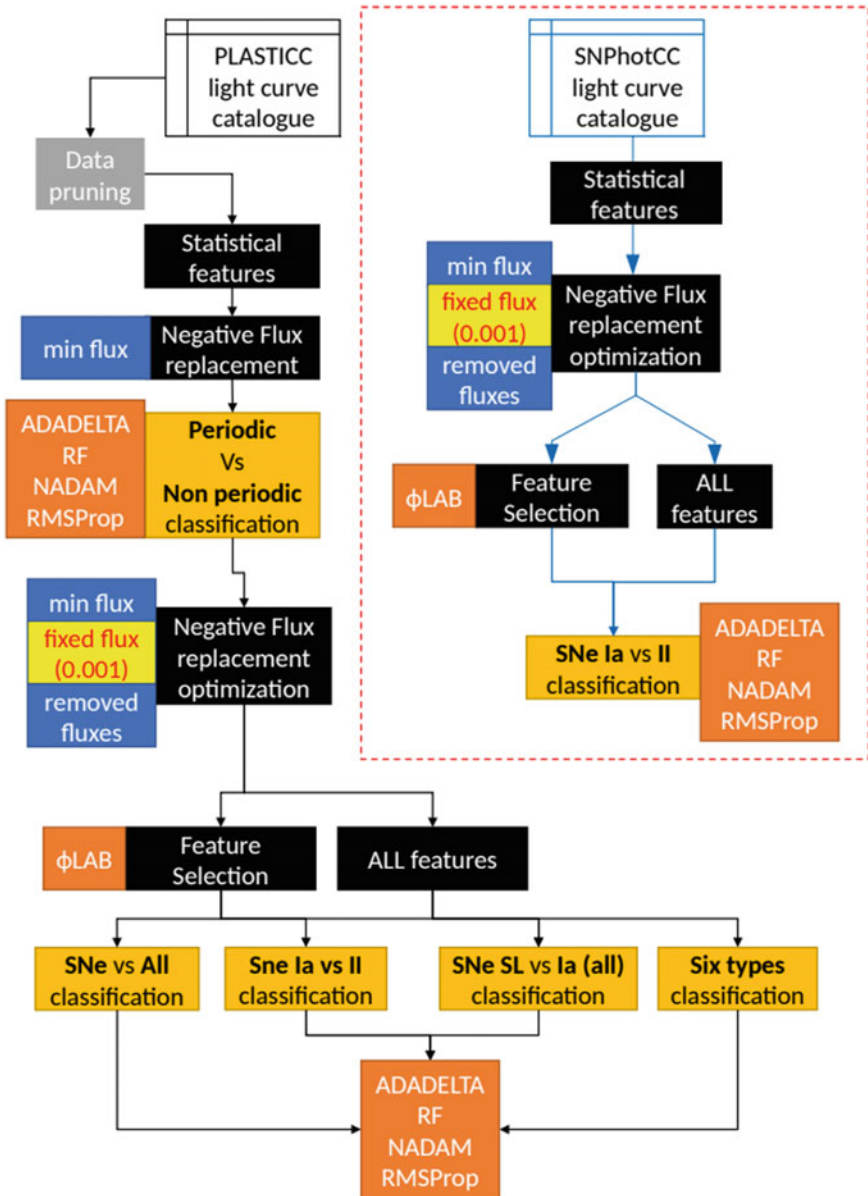


Fig. 4 Summary of the procedure designed and followed along the experiments

4.1 Data Pre-processing

From the whole PLAsTiCC dataset a maximum of 200,000 objects per class was randomly extracted (whenever possible). For each class, a pruning in flux and its error was performed. While, no any pruning was done on the SNPhotCC dataset. The Table 4 shows the limits derived from pruning.

After this first skimming, the amount of objects for the various classes was reduced to a maximum of about 35,000 curves. The reduction for classes with more than 35 K objects was driven by the choice of the curves with the least number of observations with negative fluxes and with at least 6 observations per band.

Due to the residual presence of negative fluxes, we started their handling by trying the following replacement method. By considering all the curves of a class, we checked all the observations of a given day. If in that day there was a negative or zero flux, then it was replaced with the lowest positive flux present. Else if only negative fluxes were present, they were replaced with the lowest positive flux of the previous day. This replacement has been applied to every day, for all curves and for all classes. An example of the replacing method is shown in Table 5.

Since 19 features have been chosen for our statistical approach, by considering 6 bands in PLAsTiCC, a total of 114 features composed the original parameter space.

After the composition of statistical datasets, some light curves included some missing entries, or *NaN* (Not-a-Number), causing the exclusion of those objects from the datasets, due to their unpredictable impact on the training of classifiers. The total amount of light curves per class is reported in Table 6.

4.2 Periodic Versus Non Periodic

This was the first classification experiment, performed only on PLAsTiCC simulation. Having no need, at this level, to optimize the treatment of negative fluxes, we used only the method previously described (Sect. 4.1). We had RR lyrae, Mirae variables and Eclipsing Binaries in the periodic class (*P*) and all the others in the non periodic (*NP*) class. To balance the classes we excluded some objects in the second class, as shown in Table 7. The random partitioning percentage between training and test sets was fixed, respectively, to 80% and 20%.

This series of experiments, as expected, being the simplest given the intrinsic difference of the objects involved, did not reveal any surprise. All estimators showed a great efficiency to recognize periodic objects from the variables (non periodic) ones (Table 8).

Table 4 Table of values retained after data pruning on the classes of PLAsTiCC dataset

Object	Band	Flux	Flux error	Object	Band	Flux	Flux error
AGN	u	>-50	<160	M-Dwarf	u	>-60	<300
	g, r	>-50	<160		g, r	>-60	<100
	i, z	>-50	<160		i, z	>-60	<80
	y	>-50	<160		y	>-60	<180
E. Binary	u	>-200	<800	Kilonova	u	>-10	<60
	g	>-800	<800		g	>-10	<20
	r	>-900	<800		r	>-10	<20
	i	>-800	<800		i	>-10	<25
	z	>-1100	<800		z	>-20	<40
	y	>-800	<650		y	>-30	<70
Mirae	u	>-30	<2500	μ Lens	u	>-40	<1700
	g	>-20	<800		g	>-20	<250
	r	>-50	<900		r	>-30	<400
	i	>-1200	<1700		i	>-40	<300
	z	>-8000	<3000		z	>-60	<400
	y	>-11000	<3300		y	>-90	<500
RR Lyrae	u	>-1300	<1500	SN Ia	u	>-50	<1350
	g	>-6000	<1500		g	>-20	<500
	r	>-6000	<1500		r	>-20	<400
	i	>-4500	<1500		i	>-40	<170
	z	>-4500	<1200		z	>-60	<200
	y	>-5500	<1200		y	>-100	<300
SN Iax	u	>-30	<550	SN Ia91bg	u	>-30	<800
	g	>-10	<150		g	>-20	<200
	r	>-20	<150		r	>-20	<200
	i	>-30	<100		i	>-30	<150
	z	>-50	<125		z	>-40	<150
	y	>-90	<200		y	>-90	<325
SN Ibc	u	>-50	<800	SN II	u	>-40	<200
	g	>-20	<200		g	>-20	<100
	r	>-20	<150		r	>-20	<100
	i	>-30	<100		i	>-30	<100
	z	>-60	<125		z	>-60	<100
	y	>-110	<350		y	>-110	<150
SL SN I	u	>-30	<1000	TDE	u	>-20	<200
	g	>-10	<150		g	>-10	<50
	r	>-15	<125		r	>-10	<50
	i	>-20	<100		i	>-20	<50
	z	>-40	<100		z	>-30	<75
	y	>-70	<175		y	>-60	<150

Table 5 Example of the negative fluxes replacement within the PLAsTiCC catalogue

ID	MJD	Flux	
		Before	After
1	59820.0015	-25.154862	0.284215
2	59820.0238	15.458932	15.458932
3	59820.1234	-5.848961	0.284215
4	59820.4451	-20.548951	0.284215
5	59820.8251	0.284215	0.284215
6	59820.0234	-9.542318	0.284215
7	59820.6234	10.854215	10.854215

Table 6 Summary of the light curves composing the simulated datasets

Dataset	Object	Curves	Object	Curves
PLAsTiCC	AGN	34666	E. Binary	34484
	Kilonova	232	M-Dwarf	34849
	Mirae	1154	μ Lens	1187
	RR Lyrae	32698	SN Ia	34953
	SN Iax	34977	SN Ia 91bg	34923
	SN Ibc	34932	SN II	34828
	SL SN I	34959	TDE	14023
	Total objects		361711	
SNPhotCC	SN Ia	5088	SN II	12027
	Total objects		17115	

Table 7 Summary of the sources belonging to the PLAsTiCC dataset in the P (periodic class) versus NP (non periodic class) use case divided in training (80%) and test (20%) sets

Object	Number of curves		Object	Number of curves	
	Training	Test		Training	Test
RR Lyrae	26158	6540	Kilonova	187	46
E. Binary	27587	6897	M-Dwarf	6001	1501
Mirae	923	231	μ Lens	950	238
AGN	6001	1501	SN Ia	6001	1501
SN Iax	6001	1501	SN Ia 91bg	6001	1501
SN Ibc	6001	1501	SN II	6001	1501
SL SN I	6001	1501	TDE	6001	1501
Total P training		54668	Total NP training		55146
Total P test		13668	Total NP test		13793

Table 8 Summary of the best results (in percentages) for the 4 classifiers in the classification experiment *P versus NP*. For Nadam, RMSProp and Adadelta models, a decay value of 10^{-5} and a learning rate of 0.0005 were assigned

%	Type	RF	Nadam	RMSProp	Adadelta
Accuracy	–	99	97	98	96
Purity	NP	99	97	99	95
	P	99	98	98	97
Completeness	NP	99	98	98	97
	P	99	97	99	95
F1 Score	NP	99	98	98	96
	P	99	97	98	96

4.3 Handling of Negative Fluxes

In both simulated catalogues, as introduced in Sect. 4, the presence of negative fluxes required an investigation on how to replace them in order to minimize their negative impact on the learning efficiency of machine learning models. Therefore, it was decided to approach this problem in three ways.

The first (named as *M1*) was to replace their value as introduced in Sect. 4 (and preliminarily used for the Periodic versus Non Periodic classification experiment, described in Sect. 4.2): for each class of objects, the observations related to the same day were grouped, by taking the least positive flux value. This value has been replaced to all the negative fluxes of that day.

The second approach (named as *M2*) was to replace the negative fluxes with a constant value of 0.001, considered as the minimum flux emitted by the sources.

The third solution (*M3*) consisted into the total rejection of negative fluxes from the dataset, without any replacement.

The impact on classification accuracy has been analyzed by comparing the three solutions in the *SNe versus All* (the class *All* includes the rest of transient types) classification experiment on the Plastic dataset and the *SNIa versus SNII* experiment on the SNPhotCC dataset. In both cases, the data have been treated with the three replacement types, producing different amount of objects per class. The entire composition of the datasets for the three methods is shown in Table 9, while the composition of the classes of SN, All, SNIa and SNII are shown in Table 10.

The results of the two experiments are shown, respectively, in Tables 11 and 12. The results indicated that, on average, in the case of the PLAsTiCC dataset, the second method (*M2*) obtained a better accuracy, with some exception in favor of *M3*. In the case of SNPhotCC dataset, on the other hand, *M2* and *M3* resulted more close in terms of classification efficiency. Therefore, since we were mostly interested

Table 9 Summary of sources of datasets for each replacing method adopted for negative fluxes

	Object	Number of curves			Object	Number of curves		
		<i>M1</i>	<i>M2</i>	<i>M3</i>		<i>M1</i>	<i>M2</i>	<i>M3</i>
PLAsTiCC	AGN	34666	34666	34082	Kilonova	232	232	229
	μ Lens	1187	1187	1144	M- Dwarf	34849	34849	34191
	SN Ia	34953	34891	34423	SL SN I	34959	34959	34750
	SN Iax	34977	34977	34680	SN Ia 91bg	34923	34923	34559
	SN Ibc	34932	34932	34437	SN II	34828	34771	34393
	TDE	14023	14023	13985				
Total objects <i>M1</i> :		294529		Total objects <i>M2</i> : 294410		Total objects <i>M3</i> : 290873		
SNPhotCC	SN Ia	5088	5088	5086	SN II	5088	5088	5077
Total objects <i>M1</i> :		10176		Total objects <i>M2</i> : 10176		Total objects <i>M3</i> : 10163		

Table 10 Summary of sources of training and test sets for each negative flux replacing method

	Object	<i>M1</i>		<i>M2</i>		<i>M3</i>	
		Training	Test	Training	Test	Training	Test
PLAsTiCC	AGN	27732	6934	27732	6934	27266	6816
	Kilonova	186	46	186	46	183	46
	μ Lens	949	238	949	238	915	229
	M-Dwarf	27879	6970	27879	6970	27353	6838
	SN Ia	12001	3001	11975	2994	11802	2954
	SL SN I	12001	3001	12001	3001	11935	2979
	SN Iax	12001	3001	12001	3001	11900	2976
	SN Ia 91bg	12001	3001	12001	3001	11866	2975
	SN Ibc	12001	3001	12001	3001	11828	2951
	SN II	12001	3001	11983	2992	11835	2970
	TDE	11218	2805	11218	2805	11188	2797
	Total SN	72006	18006	71962	17990	71166	17805
	Total All	67964	16993	67964	16993	66905	16726
SNPhotCC	SN Ia	4071	1017	4071	1017	4062	1016
	SN II	4071	1017	4071	1017	4070	1015

to directly compare the classification performances between the two datasets, by considering also the drastic reduction of available sources using the *M3* method, we definitely selected and applied the *M2* to both datasets.

Table 11 Comparison among the three replacing methods for negative fluxes on the PLAsTiCC dataset in the classification case *SNe versus All*. For Nadam, RMSProp and Adadelata a learning rate of 0.001 and a decay value of 10^{-5} were chosen. The statistics are expressed in percentages

Dataset	Use case	Algorithm	Class	Estimator	$M1$	$M2$	$M3$	
PLAsTiCC	<i>SNe versus All</i>	RF	SN	Purity	86	91	85	
				Completeness	94	93	91	
				F1-score	90	92	88	
			All	Purity	93	92	90	
				Completeness	83	90	83	
				F1-score	88	91	86	
			Nadam	SN	Purity	77	84	83
					Completeness	82	78	85
					F1-score	79	81	84
		All		Purity	80	78	84	
				Completeness	73	85	82	
				F1-score	76	81	83	
		RMSProp		SN	Purity	85	89	87
					Completeness	83	89	91
					F1-score	84	89	89
			All	Purity	83	88	90	
				Completeness	85	89	86	
				F1-score	84	88	88	
			Adadelata	SN	Purity	80	85	85
					Completeness	84	86	87
					F1-score	82	86	86
		All		Purity	82	85	86	
				Completeness	78	84	84	
				F1-score	80	85	85	

4.4 Optimization of the Parameter Space for Transients

After choosing how to handle the negative fluxes, we investigated the statistical parameter space (PS) of the two simulated datasets, in order to explore the possibility to reduce the dimensionality of the classification problem (feature selection) and to analyze the impact of the resulting optimized PS on the classification efficiency for each particular type of classes involved in all cases, as well as the possibility to find a common set of relevant features, suitable to separate different types of transients. We applied the Φ LAB algorithm, introduced in Sect. 3.3, to both datasets in various classification use cases (except the preliminary experiment *P Vs NP*), obtaining an optimized parameter space for each of them. The analysis of feature commonalities among all classification experiments is shown in Fig. 5. In particular, the feature selection of the *SNIa versus SNI* use case has been done on the PLAsTiCC dataset deprived of the *u* and *y* bands, for uniformity with the SNPhotCC dataset in terms of direct comparison.

Table 12 Comparison among the three replacing methods for negative fluxes on the SNPhotCC dataset in the classification case *SNIa versus SNII*. For Nadam, RMSProp and Adadelata a learning rate of 0.001 and a decay value of 10^{-5} were chosen. The statistics are expressed in percentages

Dataset	Use case	Algorithm	Class	Estimator	M1	M2	M3	
SNPhotCC	<i>SNIa versus SNII</i>	RF	SNIa	Purity	91	95	91	
				Completeness	94	97	93	
				F1-score	93	96	92	
			SNII	Purity	94	97	93	
				Completeness	91	95	91	
				F1-score	92	96	92	
			Nadam	SNIa	Purity	86	91	92
					Completeness	92	92	94
					F1-score	89	92	93
		SNII		Purity	91	92	94	
				Completeness	86	91	92	
				F1-score	88	91	93	
		RMSProp		SNIa	Purity	91	92	93
					Completeness	93	96	94
					F1-score	92	94	94
			SNII	Purity	93	96	94	
				Completeness	91	92	93	
				F1-score	92	94	94	
			Adadelata	SNIa	Purity	89	86	92
					Completeness	92	88	92
					F1-score	91	87	92
		SNII		Purity	92	88	92	
				Completeness	89	85	92	
				F1-score	90	87	92	

From the analysis of the histogram of Fig. 5 it was possible to extract a common optimized parameter space, composed by relevant features with higher percentage of common occurrences among various classification use cases (the cumulative measurement process is explained in the caption of the Fig. 5). The extraction was done trying also to balance the different amount of relevant features provided by Φ LAB in every classification case with their percentage of commonality among different cases, with the aim at extracting the same number of relevant features in all cases. The best compromise found is reported in Table 13 and corresponds to 78 extracted features (on a total of 114) suitable for the six-band cases (*ugriz* in PLAsTiCC) and 52 (on a total of 76) for the four-band cases (*griz* in SNPhotCC). These two resulting optimized (reduced) parameter spaces have been used in the classification cases described in the next sections, each time by comparing the classification efficiency between the complete and the reduced parameter spaces.

By looking at the optimized parameter spaces obtained (Fig. 5), extremely interesting is the presence of some common features among the various classification

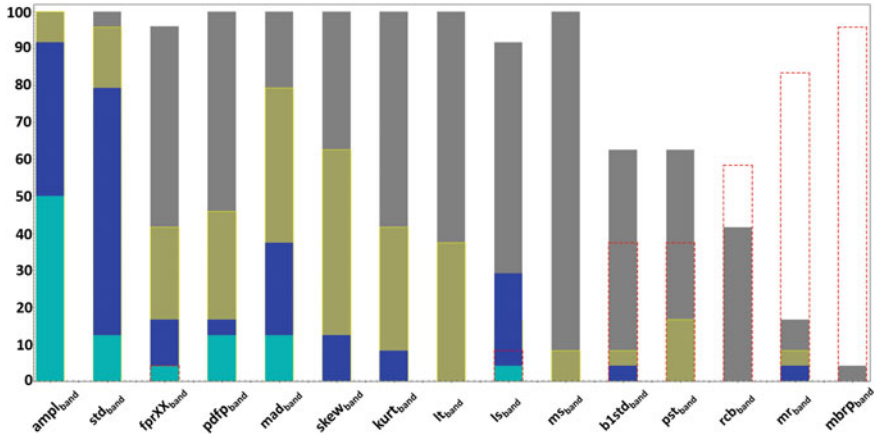


Fig. 5 Cumulative statistical analysis of the feature selection performed with the method Φ LAB on different use cases. The results include the four classification cases on PLAsTiCC (*SN Vs All*, *SN Ia Vs SN II*, *SN Ia Vs SL-I*, *six-class SNe*) and the single case *SN Ia versus SN II* on SNPhotCC. The indicated features are grouped per statistical type, including all their available bands. After having calculated the various feature rankings for each classification case with Φ LAB, ordered by decreasing importance, the vertical bars shown in the histogram represent the percentage of common occurrences of each feature type, among various classification use cases, within, respectively, the first 25% (cyan), 50% (blue), 75% (yellow) and 100% (gray) of feature rankings. While dotted red bars indicate the percentage of common occurrences of rejection among various feature rankings

cases. In particular, the Amplitude (*ampl*) shows a crucial role for the classification of various SNe types. Also important is the Standard Deviation (*std*), which reaches 79.2% of common occurrences. Equally interesting appears the high percentage of common rejections of Median Buffer Range Percentage (*mbrp*), Magnitude Ratio (*mr*) and R Cor Bor (*rcb*). Within most of the light curves of the datasets used, the average value of the *mbrp*, which is the percentage of points in an interval of 10% of the median flux, is very high. This shows that most of the light curves are relatively contained within the flux extension. The *mr* feature, the percentage of points above the median magnitude, has always values greater than 40%, with a standard deviation of a lower order of magnitude, except in the case of the *six-class SNe* problem, in which the standard deviation is comparable with the *mr* value. This shows that most of the light curves are basically symmetrical in magnitude. Finally, the *rcb* has an average value of about 30% with a comparable standard deviation. Therefore, it ranges over the whole spectrum of possible values without any class distinction.

The *ampl*, which from a physical point of view represents the half-amplitude, in magnitude, of the light curves, is the most important feature in all use cases and it is related to the different distribution of objects in the classes. In the *SNe versus All* use case, the class of SNe shows a bi-modal distribution, while the class *All* shows an alternation between bi-modal and uni-modal distributions, with different peaks from the SNe distributions. In the *SN Ia versus SL-I* use case, the SNe Ia have a bi-modal distribution, unlike the SL-I type, which instead is uni-modal. The *six-class SNe* use

Table 13 Summary of the resulting common optimized parameter spaces from the analysis of the feature selections. Each feature listed is intended to include all its available bands. First four use cases (columns 2 to 5) refer to the classification cases approached on PLAsTiCC with such optimized PS in six bands (*ugrizy*), while last column is referred to the classification experiment done with SNPhotCC in four bands (*griz*). For PLAsTiCC the optimized PS include 78 features, while SNPhotCC is composed by 52. Take into account that the feature $fprXX_{band}$ includes 5 different types per band group (See Sect. 2.3 for details)

Feature	[<i>SNe Vs All</i>]	[<i>SNIa Vs SNII</i>]	[<i>SNIa Vs SL-I</i>] PLAsTiCC	[<i>six-class SNe</i>]	[<i>SNIa Vs SNII</i>] SNPhotCC
$ampl_{band}$	x	x	x	x	x
$pdfp_{band}$	x	x	x		x
ms_{band}				x	
mad_{band}	x	x	x	x	x
std_{band}	x	x	x	x	x
$skew_{band}$	x	x	x	x	x
$fprXX_{band}$	x	x	x	x	x
$kurt_{band}$	x	x	x	x	x
ls_{band}	x	x	x	x	x
lt_{band}	x	x	x	x	x
Totals		PLAsTiCC: 78		SNPhotCC: 52	

case shows that the SNe Ia have a similar peak w.r.t. the sub-types *Iax*, *Iabg91*, *SL* and the *Ibc*. The SNe II instead, show an unexpected shape similarity with the SNe Ia in the PLAsTiCC simulation, and this should explain a classification efficiency in the *SNIa versus SNII* case smaller than what obtained on the SNPhotCC data (see Sect. 4.6).

The *std*, the deviation from the mean flux, has the same trend of the *ampl*, with bi-modal and uni-modal distributions and with peaks at different values.

The *fpr*, the flux percentage ratio, related to the sampling of the light curve assuming a relevance with the higher flux values, shows that, in the *SNe versus All* case and with the *griz* bands, there are two distributions with distinguishable peaks. In the *six-class SNe* case, the *riz* bands, with the wider flux ratios, contribute to solve the envelope of the 6 classes. In the *SNIa versus SL-I* case, the different distributions can be particularly identified in the *rizy* bands, again in the broader flux ratios such as 50, 65 and 80. Finally, in the *SNIa versus SNII* problem the distinction is more complex and only in few *riz* band cases it is possible to see the two different distributions.

In the other relevant features shown in Fig. 5, we do not infer distinct distributions in the various use cases, but only different fluctuations around the same distribution. This means that all the curves of all the classes share, more or less, the same distribution w.r.t. the flatness of the curve (*kurt*), the symmetry of the curve (*skew*), the slope deriving from the linear fit (*lt*), the period obtained from the peak frequency of

the Lomb Scargle Periodogram (ls), the ratio of difference between percentiles and the median ($pdfp$) and finally the median of deviations from the median (mad). Since these features have proved to be highly relevant, this implies that those fluctuations in the class distributions contribute substantially to the classification of different types of SNe. Finally, in the *six-class SNe* problem, another feature appears relevant, which is the maximum difference in magnitude between two successive epochs (ms), providing, slightly in the u band and in a more consistent way in the y one, fluctuations suitable in principle for the resolution of the more complex classification.

4.5 Supernovae Versus All

In this use case we had SNe type Ia, Iax, Ia 91bg-like, Ibc, II and SL-I within the *SNe* class and all the other object types, except the excluded periodic ones, in the *All* class. We performed the experiments on the PLAsTiCC dataset with the 4 classifiers using, respectively, the entire set of statistical features available (114) and with the optimized parameter space (78). The amount of objects for each type included in the two classes is shown in Table 14.

Among Nadam, RMSProp and Adadelta, the best performances were obtained with the RMSProp in both cases (whole and optimized parameter spaces). While Random Forest reached the best classification performances. The statistical results are shown in Table 15.

Table 14 Summary of the objects belonging to the PLAsTiCC dataset, used for the *SNe versus All* experiment, randomly partitioned in training (80%) and test (20%) sets

Type	Training	Test
SN Ia	11975	2994
SN Iax	12001	3001
SN Ia91bg	12001	3001
SN Ibc	12001	3001
SN II	11983	2992
SL SN I	12001	3001
Kilonova	186	46
M-Dwarf	27879	6970
μ Lens	949	238
TDE	11218	2805
AGN	27732	6934
Total SN	71962	17990
Total All	67964	16993

Table 15 Summary of the statistical results for the 4 classifiers with, respectively, all the features and the 78 selected. For Nadam, RMSProp and Adadelta, the values of 10^{-5} and 0.0005 were assigned to the decay and learning rate hyper-parameters, respectively

Features		Random forest		Nadam		RMSProp		Adadelta	
		All	78	All	78	All	78	All	78
% Accuracy	–	92	92	85	86	90	90	86	85
% Purity	SN	91	91	85	86	91	91	86	84
	All	92	92	85	86	89	90	86	86
% Completeness	SN	93	93	86	87	90	90	87	87
	All	90	90	84	85	90	90	85	83
% F1 Score	SN	92	92	86	87	90	87	87	86
	All	91	91	84	86	90	86	86	84

4.6 Supernovae Ia Versus II

In this experiment we considered only SNe of type Ia and II. In this case it was possible to use both SNPhotCC and PLAsTiCC datasets, since in the case of SNPhotCC, these two types of SN were available. The amount of objects used is shown in Table 16.

We performed the experiment with the 4 classifiers using, respectively, all the features and the amounts related to the two optimized feature sets, respectively, 78 for PLAsTiCC and 52 for SNPhotCC. For a direct comparison between the SNPhotCC and PLAsTiCC datasets, we also considered a reduced version of the PLAsTiCC dataset, by excluding the *u* and *y* bands for uniformity with the SNPhotCC catalogue in terms of bands available. The statistical results are reported in Table 17.

Table 16 Summary of the objects belonging to the datasets used for the *SNIa versus SNIi* experiment on PLAsTiCC and SNPhotCC, randomly partitioned in training (80%) and test (20%) sets

Dataset	Type	Number of curves	
		Training	Test
PLAsTiCC	SN Ia	27964	6990
	SN II	27983	6966
	Total	55947	13956
SNPhotCC	SN Ia	4071	1017
	SN II	4071	1017
	Total	8142	2034

Table 17 Summary of the statistical results for the 4 classifiers in the *SM/a versus SMI* experiment. For each classifier it is reported the statistics related to the PLAsTiCC (PLA columns) and SNPhotoCC (SNP columns) datasets. In the case of PLAsTiCC, the columns are related to the whole original feature space (All) and the optimized one (78) using 6 bands (*u_{griz}*), together with the reduced feature space (52) using 4 bands (*griz*) for a direct comparison with the corresponding optimized parameter space obtained on SNPhotoCC. For Nadam, RMSProp and Adadelta, the values of 10^{-5} and 0.0005 were assigned to, respectively, the decay and learning rate hyper-parameters, in the cases of 78 features. While 10^{-5} and 0.001 values have been assigned for the cases with 52 features

	Random forest						Nadam						RMSProp						Adadelta					
	PLA			SNP			PLA			SNP			PLA			SNP			PLA			SNP		
	6	4	4	6	4	4	6	4	4	6	4	4	6	4	4	6	4	4	6	4	4	6	4	4
Bands used features	All	78	79	All	52	All	All	52	All	52	All	All	52	All	52	All	All	52	All	52	All	52	All	52
% Accuracy	-	78	79	78	96	96	71	72	71	93	94	76	76	78	94	96	74	74	73	90	96	74	73	90
% Purity	Ia	76	76	76	95	95	70	70	69	90	92	74	74	75	93	94	72	73	72	89	93	72	73	89
	II	81	81	80	97	97	72	73	74	95	96	78	78	80	95	98	75	74	75	92	96	75	74	92
% Completeness	Ia	82	83	82	97	97	75	76	77	96	96	79	80	82	95	98	76	75	77	92	96	75	77	92
	II	74	74	74	95	95	67	67	65	89	92	73	71	73	93	93	71	72	70	88	93	72	70	88
% F1 Score	Ia	79	79	79	96	96	72	73	73	93	94	77	77	79	94	96	74	74	74	90	95	74	74	90
	II	77	78	77	96	96	70	70	69	92	94	75	74	76	94	95	73	73	72	90	95	73	72	90

In terms of classification performance, it appears evident the discrepancy between the two datasets. The capability of classifiers to recognize the two classes is higher on SNPhotCC and this implies a strong dependency of learning models from the overall accuracy of the simulations. Furthermore, the very similar percentages among the whole feature set and the optimized versions probes the capability of the feature selection method Φ LAB to extract a set of relevant features, able to preserve the level of classification efficiency.

4.7 Superluminous SNe Versus SNe I

In the *SNIa versus SL-I* experiment, the three sub-classes of SNe, Ia, Ia91bg and Iax have been mixed in the same percentage and then classified against Superluminous SNe I. We performed the experiments with the 4 classifiers using all the features and the 78 selected with Φ LAB. The amount of objects per type is shown in Table 18.

The statistical results of the classification are shown in Table 19.

By analyzing the results, it is noticeable the lower performance of Adadelta w.r.t. other classifiers, where Random Forest appeared the best one for all estimators. The similar results obtained for both parameter spaces confirm the validity of the feature selection.

In terms of error percentages on the SN I class (all Ia sub-types), Table 20 reports the level of contamination for each sub-type in the experiment with all features and using Random Forest.

As shown in Table 20, the most contaminated sub-class is SNIax, which indicates its high difficulty of recognition among other SN types.

Table 18 Summary of the objects belonging to the dataset used for the *SNIa versus SL-I* experiment on PLAsTiCC, randomly partitioned in training (80%) and test (20%) sets

Type	Number of curves	
	Training	Test
SN Ia	9323	2331
SN Iax	9323	2331
SN Ia91bg	9323	2331
SLSN I	27967	6992
Total Ia	27969	6993
Total SL	27967	6992

Table 19 Summary of the statistical results for the 4 classifiers on the *SNIa versus SL-I* experiment, with, respectively, all the features and the 78 of the optimized parameter space of PLAsTiCC dataset. For Nadam, RMSProp and Adadelata, the values of 10^{-5} and 0.0005 were assigned to the decay and learning rate hyper-parameters, respectively

Features		Random forest		Nadam		RMSProp		Adadelata	
		All	78	All	78	All	78	All	78
% Accuracy	–	88	87	81	82	85	82	71	70
% Purity	SL-I	83	80	77	74	81	77	71	70
	SN Ia	93	93	85	89	90	87	71	70
% Completeness	SL-I	94	95	87	92	91	89	71	70
	SN Ia	80	76	74	69	79	74	72	71
% F1 Score	SL-I	88	87	82	82	86	83	71	70
	SN Ia	86	84	79	78	84	80	71	70

Table 20 Summary of the contamination analysis among all SN Ia sub-types obtained by the Random Forest, with the complete parameter space, on the *SNIa versus SL-I* experiment

Class	Total	Correctly classified	Wrongly classified	
SN Ia	2331	2328	3	≈0%
SN Iax	2331	1508	823	35%
SN Ia91bg	2331	1779	552	24%

4.8 Simultaneous Classification of Six SNe Sub-Types

Last classification experiment performed was the most complex, because we tried to classify simultaneously all the six classes of SNe available in the PLAsTiCC dataset. The experiments with the 4 models were performed using all the features and the 78 selected by the optimization procedure. The amount of objects per class is shown in Table 21.

The statistical results of the six-class classification is reported in Table 22.

Also in this case, the Random Forest obtained best results and the similar statistics between the whole and optimized parameter space confirm the good performances of the feature selection method. By analyzing the classification estimators for the single classes, the SNIa91bg showed a high difficulty to be recognized, while Ia91bg

Table 21 Summary of the objects belonging to the dataset used for the *six-class SNe* experiment on PLAsTiCC, randomly partitioned in training (80%) and test (20%) sets

SN class	Number of curves	
	Training	Test
Ia	27912	6979
Ia91bg	27938	6985
Iax	27981	6996
II	27816	6955
Ibc	27945	6987
SL I	27967	6992

Table 22 Summary of the statistical results for the 4 classifiers on the *six-class SNe* experiment, with, respectively, all the features and the 78 of the optimized parameter space of PLAsTiCC dataset. For Nadam, RMSProp and Adadelta, the values of 10^{-4} and 0.001 were assigned to the decay and learning rate hyper-parameters, respectively

Features		Random forest		Nadam		RMSProp		Adadelta	
		All	78	All	78	All	78	All	78
% Accuracy	–	66	62	53	55	60	61	48	48
% Purity	SN Ia	79	79	68	71	73	76	62	59
	SN Ia 91bg	82	78	64	70	79	81	52	58
	SN Iax	58	57	46	48	52	51	39	34
	SN II	74	75	58	61	68	66	56	55
	SN Ibc	40	42	32	34	34	35	32	32
	SL SN I	62	59	48	47	56	56	47	50
% Completeness	SN Ia	77	77	56	57	68	67	55	48
	SN Ia 91bg	25	30	27	20	20	17	21	16
	SN Iax	33	37	16	20	26	27	21	15
	SN II	79	79	79	77	77	78	67	67
	SN Ibc	64	57	47	47	54	58	38	47
	SL SN I	91	91	76	88	88	85	85	87
% F1 Score	SN Ia	78	78	62	63	71	71	58	53
	SN Ia 91bg	39	44	38	31	32	28	30	26
	SN Iax	42	45	23	29	35	35	27	21
	SN II	76	77	67	68	72	72	61	60
	SN Ibc	49	48	38	40	42	44	34	38
	SL SN I	73	72	59	62	68	67	61	63

Table 23 Percentages of contamination in the *six-class SNe* classification results

		Random Forest Classification						Contamination %
		Predicted %						
		Ia	Iabg	Iax	II	Ibc	SL	
T r u e	Ia	77.3	-	-	22.6	-	0.1	22.7
	Iabg	-	25.3	17.5	-	47.6	9.6	74.7
	Iax	-	2.5	32.6	0.01	45.1	19.7	67.4
	II	21.0	-	-	78.7	0.01	0.3	21.3
	Ibc	0.1	3.0	6.0	0.03	63.9	27.0	36.1
	SL	0.1	0.1	0.3	4.6	4.1	90.8	9.2

and Iax types were often confused for SNIbc. SL type resulted the most complete, although the purity was reduced by the contamination of SNIbc and SNIax (Table 23).

Finally SNIa and SNII types, although reducing their efficiency w.r.t. the dedicated two-class experiment, maintained a sufficient level of classification.

5 Discussion and Conclusions

The present work is related to the important problem of classification of astrophysical variable sources, with special emphasis to SNe. Their relevance in terms of cosmological implications is well known, causing a special attention to the problem of recognizing different types of such astronomical explosive events.

To face this challenge, the SNPhotCC dataset and the PLAsTiCC dataset have been chosen to have a statistical sample, albeit of simulations, as wide as possible. Based on the objects in the datasets, a test campaign with increasing complexity has drawn up. To approach the problem we have chosen 4 machine learning methods that require a transformation of light curves into a series of statistical features, potentially suitable to recognize different source types.

In the construction of statistical datasets, the presence of negative fluxes within the observations had to be solved, due to their negative impact on the learning capability of ML models. Working directly with the light curves, their shape is relevant, thus the presence of negative fluxes is not a big problem, because it is always possible to translate the curve along the ordinate axis. In the statistical parameter space instead, since there are features requiring the conversion to magnitudes and since the translation would alter the features values in an unpredictable way, the negative fluxes must be replaced in some way. To solve this problem we tried three approaches, as described in Sect. 4.3. In the first one, the atmospheric and instrumental setup conditions were respected, by grouping the observations taken in the same day; this solution evidently introduced noise, by altering the phase within groups of light curves. The second solution, which proved to be the best candidate, replaced negative fluxes

with a positive number, by maintaining unchanged the sampling, and introducing a lower contribution of noise within data. Finally, the third method removed the observations with negative fluxes, thus highly sub-sampling the light curves. From the classification results obtained adopting the second solution, we were confident that the deformations undergone by the light curves were not able to alter their original nature nor to significantly reduce the performances in both simulation datasets used.

The parameter space analysis was approached with the Φ LAB algorithm to perform a reduction of dimensionality of the classification use cases and to investigate the possibility to identify a common set of features that could be considered suitable to recognize different types of transients. From the comparison between the original and optimized feature spaces, in terms of classification performance, the adopted method resulted extremely reliable to find a reduced set of relevant features, able to preserve the amount of information required to maintain the same level of classification efficiency. Starting from the Φ LAB results, a statistical analysis was performed, which highlighted some interesting aspects related to the physical nature of transients and SNe in particular. The Amplitude feature, representing the semi-difference between the minimum and maximum of the light curve, resulted the most relevant. Since the various classes of SNe have different light peaks, the semi-difference of the amplitude of the curve is typical of each different type of object. Also relevant resulted the standard deviation, MAD, and all features related to the percentiles or characterizing the light curve shape, such as skewness and kurtosis. The relevance of percentiles is related to the different decay time of the light radiation for the various types of SNe. Although a SN is not a periodic event, the feature related to the Lomb-Scargle periodogram has a high importance, because it is able to classify the SNe with a different periods of light decay. On the other hand, all the feature related with thresholds on the number of points around the median (such as the rcb, mr and mbrp), were rejected by our feature analysis method, probably due to their average values too close to their limits.

The most important outcome of the parameter space analysis was the identification of a feature set common to all classification use cases that revealed a coherent behavior in terms of classification performances obtained in all cases, always well close to the efficiency arising from the original parameter spaces.

In terms of pure classification among different types of sources, the high capability to distinguish between Periodic and Non Periodic objects confirmed what expected and posed ML methodology as a good candidate to approach the transient classification problem in Astronomy.

Once removed periodic objects, the high completeness (93%) reached in classifying SNe in the *SNe versus All* case (Fig. 6), confirmed that ML methods, in particular the Random Forest, could be suitable to distinguish SNe from other transients.

We wanted to verify in the remaining 7%, which was the most contaminating among the different sub-types of SNe; both SNe Ia91bg and SNe Iax were found to have the highest misclassification rate (12% of their test set). Moreover, from this analysis it was revealed that SNIa and SNIId have an error rate of about 1 per thousand, a remarkable result compared to the other SNe error rates. For completeness, the contamination was also verified for the *All* class, revealing that the AGN type has an

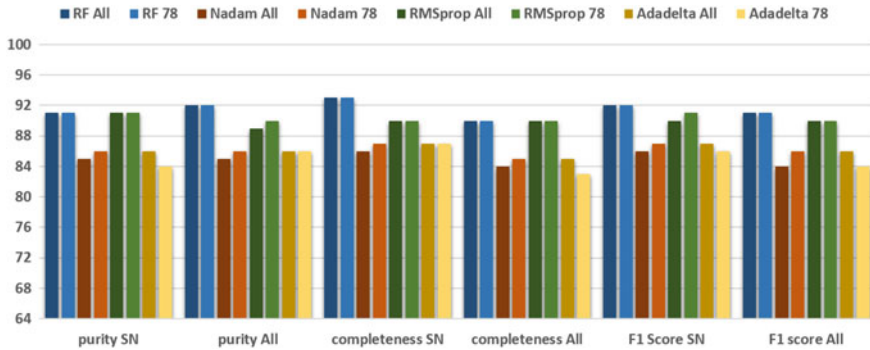


Fig. 6 Histogram of the statistical results (in %) for the *SNe versus All* classification problem

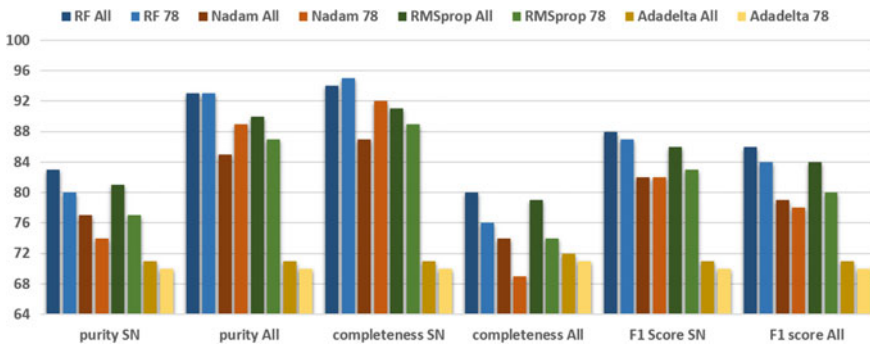


Fig. 7 Histogram of the statistical results (in %) for the *SNIa versus SL-I* classification problem

error rate of 1 per thousand, while the M-Dwarf and the TDE are the classes with the highest error rates (16%). Further experiments should be carried out to identify the SNe classes with which these two different types of transients are confused and to verify which features play a key role in their classification.

Another interesting case was the classification between Super Luminous (SL) SNe and the mixed Ia types, from which ML appeared able to recognize the SL category with a completeness, in the best case, of 95% (Fig. 7).

Although further experiments could be in principle performed, from the results obtained in this work, we can suppose to have identified a set of features suitable to help the classification of the SNe Ia, II and SL. However, those able to classify the sub-types Ia91bg and Iax are still unclear. Hopefully, with the availability of real LSST data in the near future, tests with only these two sub-classes could be conducted, with at most the addition of SNIbc, to evaluate which features could result relevant to recognize such types of SNe.

Acknowledgements The software package of machine learning models used in this work was developed within the DAME project [21]. MB and GR acknowledge the financial contribution from

the agreement *ASI/INAF 2018-23-HH.0, Euclid ESA mission - Phase D*, while MB acknowledges also the *INAF PRIN-SKA 2017 program 1.05.01.88.04* and the *MIUR Premiale 2016: MITIC*. Topcat [22] has been used for this work.

References

1. Branch, D.: *Nature* **465**(7296), 303 (2010). <https://doi.org/10.1038/465303a>
2. Goobar, A., Leibundgut, B.: *Ann. Rev. Nuclear Particle Sci.* **61**(1), 251 (2011). <https://doi.org/10.1146/annurev-nucl-102010-130434>
3. Huber, S., Suyu, S.H., Noebauer, U.M., et al.: *A&A* **631**, A161 (2019). <https://doi.org/10.1051/0004-6361/201935370>
4. Ivezić, Z., Strauss, M.A., Tyson, J.A., et al.: American Astronomical Society Meeting Abstracts #217, American Astronomical Society Meeting Abstracts, American Astronomical Society Meeting Abstracts, vol. 217, p. 252.01 (2011)
5. Brescia, M., Cavuoti, S., Amaro, V., et al.: *Data Analytics and Management in Data Intensive Domains*, Kalinichenko, L., Manolopoulos, Y., Malkov, O., Skvortsov, N., Stupnikov, S., Sukhomlin, V., (eds.), pp. 61–72. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-96553-6_5
6. Brescia, M., Djorgovski, S.G., Feigelson, E.D., et al.: *Astroinformatics*, IAU Symposium, vol. 325 (2017)
7. Brescia, M., Longo, G.: *Nuclear Instrum. Methods Phys. Res. A* **720**, 92 (2013). <https://doi.org/10.1016/j.nima.2012.12.027>
8. Breiman, L., Last, M., Rice, J.: *Statistical Challenges in Astronomy*, pp. 243–254. Springer, New York (2003)
9. Dozat, T.: *Proceedings of ICLR Workshop*, vol. 1, pp. 2013–2016 (2016)
10. Brescia, M., Salvato, M., Cavuoti, S., et al.: *MNRAS* **489**(1), 663 (2019). <https://doi.org/10.1093/mnras/stz2159>
11. Delli Veneri, M., Cavuoti, S., Brescia, M., et al.: *MNRAS* **486**(1), 1377 (2019). <https://doi.org/10.1093/mnras/stz856>
12. Kessler, R., Bassett, B., Belov, P., et al.: *PASP* **122**(898), 1415 (2010). <https://doi.org/10.1086/657607>
13. Ponder, K., Hlozek, R., Allam, T., et al.: American Astronomical Society Meeting Abstracts, vol. 52, p. 203.15 (2020)
14. Malz, A.I., Hložek, R., Allam, T., et al.: *Astron. J.* **158**(5), 171 (2019). <https://doi.org/10.3847/1538-3881/ab3a2f>
15. The PLAsTiCC team, Allam, T., Bahmanyar, A., Biswas, R., et al.: *arXiv e-prints arXiv:1810.00001* (2018)
16. Dilday, B., Kessler, R., Frieman, J.A., et al.: *ApJ* **682**(1), 262 (2008). <https://doi.org/10.1086/587733>
17. Bazin, G., Palanque-Delabrouille, N., Rich, J., et al.: *A&A* **499**(3), 653 (2009). <https://doi.org/10.1051/0004-6361/200911847>
18. D’Isanto, A., Cavuoti, S., Brescia, M., et al.: *MNRAS* **457**(3), 3119 (2016). <https://doi.org/10.1093/mnras/stw157>
19. Breiman, L.: *Mach. Learn.* **45**(1), 5 (2001). <https://doi.org/10.1023/a:1010933404324>
20. Stehman, S.V.: *Remote Sens. Environ.* **62**(1), 77 (1997). [https://doi.org/10.1016/S0034-4257\(97\)00083-7](https://doi.org/10.1016/S0034-4257(97)00083-7)
21. Brescia, M., Cavuoti, S., Longo, G., et al.: *Publicat. Astron. Soc. Pac.* **126**(942), 783 (2014). <https://doi.org/10.1086/677725>
22. Taylor, M.B.: *Astronomical Data Analysis Software and Systems XIV*, Astronomical Society of the Pacific Conference Series, Shopbell, P., Britton, M., Ebert, R., (eds.), vol. 347, p. 29 (2005)

Application of Machine and Deep Learning Methods to the Analysis of IACTs Data



Alessandro Bruno, Antonio Pagliaro, and Valentina La Parola

Abstract The Imaging Atmospheric Cherenkov technique opened a previously inaccessible window for the study of astrophysical sources of radiation in the very high-energy regime (TeV) and is playing a significant role in the discovery and characterization of very high-energy gamma-ray emitters. However, the data collected by Imaging Atmospheric Cherenkov Telescopes (IACTs) are highly dominated, even for the most powerful sources, by the overwhelming background due to cosmic-ray nuclei and cosmic-ray electrons. For this reason, the analysis of IACTs data demands a highly efficient background rejection technique able to discriminate gamma-ray induced signal. On the other hand, the analysis of ring images produced by muons in an IACT provides a powerful and precise method to calibrate the overall optical throughput and monitor the telescope optical point-spread function. A robust muon tagger to collect large and highly pure samples of muon events is therefore required for calibration purposes. Gamma/hadron discrimination and muon tagging through Machine and Deep Learning techniques are the main topics of the present work.

Keywords Machine learning · Deep learning · Atmospheric Cherenkov telescopes · Muons · Gamma/hadron separation · Image analysis · Pattern recognition · Computer vision

A. Bruno · A. Pagliaro (✉) · V. La Parola
INAF IASF Palermo, via Ugo La Malfa 153, 90146 Palermo, Italy
e-mail: antonio.pagliaro@inaf.it
URL: <http://www.iasf-palermo.inaf.it/>

A. Bruno
e-mail: alessandro.bruno@inaf.it

V. La Parola
e-mail: valentina.laparola@inaf.it

1 Introduction

When high energy particles and photons enter the Earth's atmosphere, they initiate a chain reaction of particles known as an atmospheric cascade. Secondary relativistic charged particles in the cascade emit Cherenkov light.

Imaging Atmospheric Cherenkov Telescopes (IACTs) can detect and image the Cherenkov radiation, allowing the observation of gamma-rays from the ground. However, gamma-rays contribute only to a small fraction of the flux of cosmic rays. The difficulty in suppressing the vast number of cosmic ray background events is one of the aspects that limit the sensitivity of IACTs. In order to detect gamma-ray sources an IACT analysis method must be able to perform an efficient background rejection, that is to separate the gamma-ray induced signal from the much more prevalent background of hadron induced showers, through the identification of shape features in the image. Luckily, the images generally contain sufficient information to separate the gamma-ray signal from the dominant cosmic rays background and reconstruct the arrival direction and energy of the primary.

Electromagnetic showers are characterized by an elliptically shaped shower image whose major axis is directed towards the source. If the primary particle is a cosmic ray, a hadronic shower develops. Although such hadronic showers often have electromagnetic sub-shower components as well, they lead to a typically more irregular shape. Analysis of IACT images relies on the extraction of relevant features from the camera pixel data. Whether those features are a vector of parameters representing the image, such as the image moments, or the full photo-electron intensity count in each pixel, in a Deep Learning approach they are automatically chosen by the network.

In general, Deep Learning concerns the application of complex Artificial Neural Networks to hierarchical learning tasks. For computer vision, Convolutional Neural Networks were designed specifically to perform image recognition tasks. We aim to develop and test several Convolutional Neural Networks Deep Learning architecture to determine the effectiveness of Deep Learning solutions over gamma/hadron separation.

High energy muons generated by air showers can be detected via their ring signature. In this paper, muon tagging is achieved by means of Machine Learning based on the extraction of relevant parameters from the data. We propose a feature set for muon tagging in order to automatically identify them.

2 The Simulations

The production of Cherenkov light in a shower induced either by a photon or by a particle in the atmosphere is a stochastic process. The shape, intensity and dimension of the observed Cherenkov pool depend on several factors, and most of them cannot be known with enough precision. The (unknown) height of the first interaction in the atmosphere, for example, introduces a large spread in the dimension of the light pool

at ground level, that reflects in large uncertainty in the evaluation of the energy of the primary particle. For this reason, the use of detailed simulation sets with different starting parameters and in different observing conditions is fundamental both for the calibration of the telescopes and for the analysis of the real data.

Throughout this work, all data sets used for training the different networks comprise air showers produced by high energy photons and cosmic rays generated by Monte Carlo simulations. These events are obtained by simulating the interaction of muons, gamma-rays and protons with the atmosphere using the CORSIKA software tool [Heck, Pierog and Knapp, 2012, Astrophysics Source Code Library, ascl:1202.006], that tracks the particles through the atmosphere while they undergo reactions with the air nuclei. An appropriate ray-tracing code follows the Cherenkov light produced by each shower when it is focused by the telescope mirrors. Finally, the image of the shower as it is recorded by the IACT camera on the focal plane is produced. The final image also includes the contribution of the night sky background. Simulated data trigger the telescope sensors giving rise to images. The camera images we produce at the end of the simulations are of size 56×56 pixels. Each pixel value represents the signal intensity. We produced:

- Muons: a set of muon events with energy between 6 and 1 TeV. Our final sample consists of 1500 muons and 1500 non-muons (both photons and hadrons).
- Gamma-ray photons and hadrons: two sets of Gamma-ray photons and hadrons simulations: one with standard night sky background, the second one with high night sky background. Events are distributed according to a realistic spectrum (with index -2.49 for the photons and -2.72 for the hadrons) between 3 and 100 TeV. We simulated 5075 photons and 5075 hadron events for each set.

3 The Muon Case

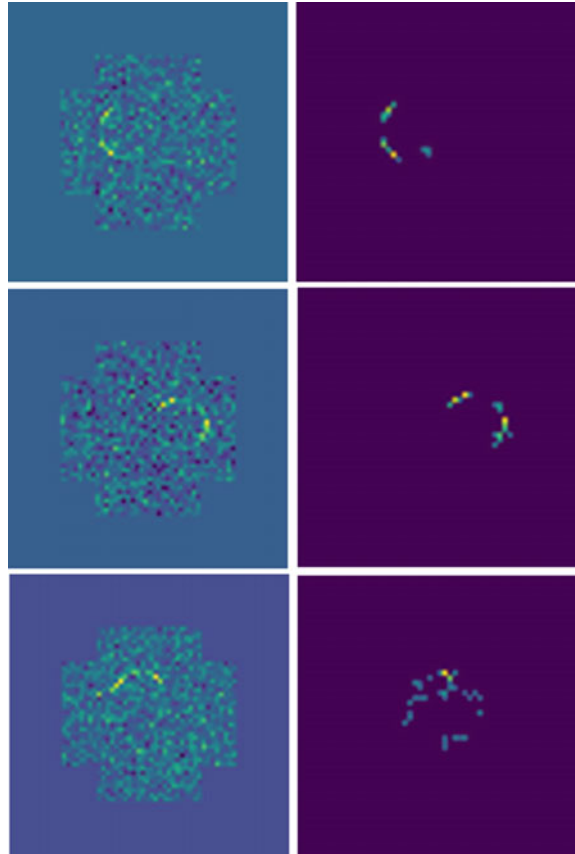
The optical throughput of an IACT is calibrated analyzing the image produced by highly energetic muons. Muons induce Cherenkov light emission, which, if the muon impact point is close to the optical axis, is imaged onto the focal plane as an arc or a ring.

Muon tagging is achieved by means of Machine Learning based on the extraction, from the camera data, of relevant parameters such as the image moments, the full photo-electron intensity count, the fullness and others. This task is not very computationally demanding, while the choice of the best parameters is crucial.

3.1 Image Cleaning Method

Muon tagging is applied to images cleaned from the Night Sky Background (Fig. 1). The cleaning maintains the basic shape of the signal and cancels isolated pixels. Our

Fig. 1 Examples of muon events before and after the application of the cleaning method



method is based on a two-step cut algorithm. The first step cancels out all the pixel under a threshold τ . The threshold depends on the mean and standard deviation of the data and is computed as:

$$\tau = \bar{I} + k \cdot \sigma_I$$

where I average and root mean square are computed on the intensity of the image pixels, with no distinction between signal and Night Sky Background. Our best choice from trial and error for the value of k is $k = 2.5$.

The second step of the cleaning algorithm applies a 3×3 pixels window and selects only structures containing at least three pixels. Outliers are then removed: structures farther than eight pixels from the centre of mass are cancelled.

3.2 Choice of the Parameters

A set of sixteen discriminating parameters computed on the cleaned image has been chosen as input for our neural network. These are:

- The fractal dimension computed by means of the box-counting method (for a description of the box-counting method see [1]);
- The standard deviation of the fractal dimensions computed on different scales by means of wavelet methods as described in [2];
- The circularity computed as

$$\text{Circularity} = 4\pi(\text{Area}/\text{Perimeter}^2)$$

on the approximate polygon. We approximate a polygon from the largest connected structure by means of the Douglas-Peucker algorithm. Connected structures are selected defining feature connections with a 3x3 structuring element;

- The number of sides of the approximate polygon;
- The total intensity computed as the sum of the values of all the pixels;
- The first four Hu's moments computed on the approximate polygon (see [3]). Hu's moment invariants are a set of numbers calculated using central moments that are invariant to image transformations. The first six moments have been proved to be invariant to translation, scale, rotation, and reflection. If

$$\mu_{ij} = \sum_x \sum_y (x - \bar{x})^i (y - \bar{y})^j I(x, y)$$

and we define normalized central moments as

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{(i+j)/2+1}}$$

the first four Hu's moments are:

$$h_0 = \eta_{20} + \eta_{02}$$

$$h_1 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$h_2 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$h_3 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

We found that Hu's moments higher than the fourth are not enough discriminating for our purpose;

- The total number of pixels over $3\sigma_I$, being σ_I the standard deviation of the uncleaned image;

- The average distance of the connected structures with three or more pixels from the centre of mass;
- The total number of non zero pixels in connected structures with two or more pixels;
- The maximum value of the radii of the connected structures with three or more pixels;
- The maximum distance of a connected structure from the centre of mass;
- The radius of the best fit circle computed on the largest connected structure;
- The fullness, defined as the number of non zero pixels inside a 2.5 pixels radius from the centre of the best fit circle computed on the largest connected structure.

The histograms of the values of the parameters for 1000 muon events and 1000 non-muon events are shown in Fig. 2. Histograms show how some of them are strongly discriminating, some are slightly.

3.3 *Architecture and Results*

The Machine Learning architecture used in our muon tagger is a linear stack of four layers, the input shape is made of the previously described sixteen parameters, hidden layers are made of 240 and 120 neurons, with a dropout of 0.2 after each stage, while the output stage is a single number between zero and one. Dropout is a technique widely used in machine learning methods to improve performances. It is a regularization technique applied to the output of some layers; some neurons are randomly dropped from the network during the training process. The dropout technique aims to prevent neurons of a network from relying on some other neurons during the training step.

We use the rectifier activation function on the first layers and the sigmoid function, that ensures our network output is between zero and one, in the output layer.

The algorithm has been applied on a mixed data set of muon and non-muon (both protons and gamma) events.

For the learning process we choose the following arguments: for the optimizer, we use the Nesterov Adam optimizer, which is essentially RMSprop with Nesterov momentum; for the loss function, we use binary cross-entropy.

We trained our neural network to identify muons. The learning process is made on a set of 1000 muons and 1000 non-muons events. After only 200 epochs the validation accuracy (on a 20% random samples of the data) is over 99%.

Our results for a test set of 1000 muons and 1000 non-muons events is as follows:

- True identification of non-muons: 96.1 %
- True identification of muons: 99.5 %

The selection power for muons is very high, while being not very computing demanding and pave the way for the development of high performance machine learning techniques run directly on the telescope camera server for the pre-selections of muon events.

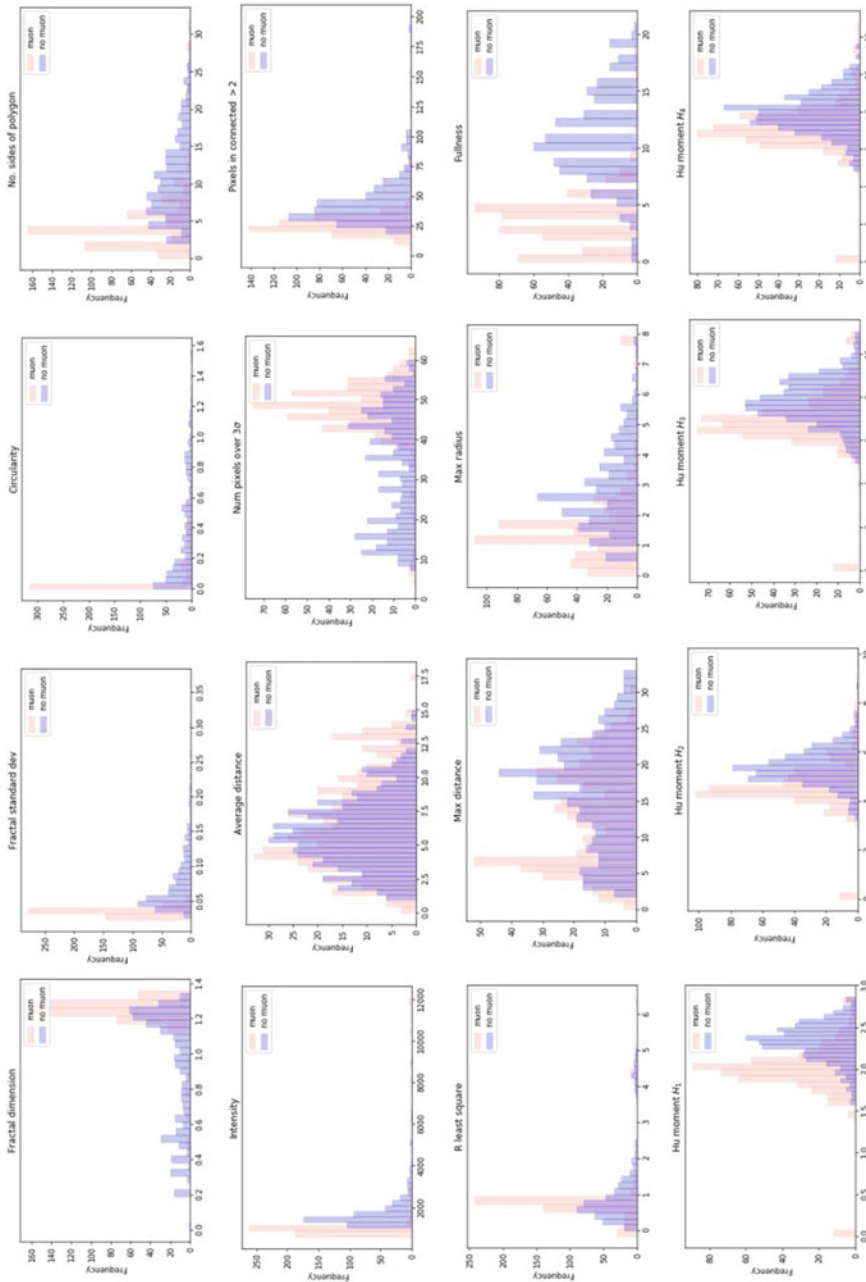


Fig. 2 Histograms for parameters used as inputs for the neural network. Clearer colour are moons, darker colour represents non-moons

3.4 *Transfer Learning for the Muon Case*

Transfer Learning is a Deep Learning approach which relies on pre-trained networks trained over a specific domain to be used over a new task domain (see [5]). A faster preliminary training step is needed rather than training networks from scratch. The latter scenario involves a large number of images to set-up a lot of parameters characterizing each CNNs.

The objective of this specific study is to analyse the power of the Transfer Learning paradigm to infer knowledge from a little number of images with constrained condition such as a strong background signal like the one present in the IACT domain. To this aim, we built a sample of 1500 muons plus 1500 non-muons images. The choice of 3000 as number of samples is not given by chance.

We want to find out a sort of trade-off between the number of images per class (muons and non-muons) and the test-accuracy of several Deep Learning architectures based on different principles and operations, trained over image classification tasks such as GoogLeNet, ResNet, SqueezeNet, MobileNet, putting in evidence how they affect the efficiency of the system.

Different pre-built Convolutional Neural Networks (CNNs), extensively described in the appendices, have been employed to find correlations between network depths, layers, hyper-parameters and performances over the detection topic. We conduct the study using a gradual approach, that is, first employing simpler networks made up of a lower number of layers and then, using networks with a higher number of layers. The list of the adopted networks follows:

- Flattened Deep Learning Architecture [6], which is one of the first basic and straightforward (Appendix A);
- GoogLeNet [8], based upon a graph with a large number of layers, characterized with higher depth than many state of the art approaches (Appendix D);
- ResNet-50 [7], made up of an ensemble of Residual Nets combining a graph (Appendix E).

In our work we customised CNNs pre-trained on ImageNet whose implementations are with Python packages such as TensorFlow Keras, PyTorch, and Torchvision. The latter is focused on image classification, semantic segmentation, and object detection.

To compare different performances among the above mentioned architectures we go through a multiple iterative refinement step to assess the best configurations for each given CNN. The step of parameters tuning relies upon trials with different epochs, number of images, number of classes for training.

3.5 *Experimental Results of Transfer Learning over the Muon Case*

First experimental results show a variety of performances which mostly depend on some parameters such as the number of images, the type of architecture and the number of epochs during training. Since we wanted to evaluate the performances of pre-built networks, we applied the Transfer Learning paradigm to transfer knowledge from a general domain to the specific IACT one. To this aim we focused our attention on CNNs pre-trained over ImageNet [11], an image database organized according to a semantic hierarchy called WordNet. The database counts in almost 14 millions of images. Several algorithms for object detection and image classification at large scale have been continuously tested and evaluated through the ImageNet Large Scale Visual Recognition Challenge (from 2010 up to 2017). In our work, we use four CNNs (GoogLeNet, ResNet-50, MobileNet, SqueezeNet) pre-trained over ImageNet to be specialized over the muon case. As briefly mentioned before, the choice of Transfer Learning rather than training from scratch is required if we want to assess the power of Deep Learning methods using a strict constraint such as the number of images. In fewer words, we want to find out whether a connection can be found between the level of depth of the architectures we test and the level of knowledge inference abilities over a new application domain such as the IACT. In order to provide the artificial intelligence model with a sufficient number of images to set up parameters describing the application domain patterns and representations closely, we used Data Augmentation [12], a common practice in Transfer Learning.

We compared the test accuracy of the training process of the architecture: Flattened Network, GoogLeNet, ResNet-50 and SqueezeNet along with a growing number of epochs. We observe that different architecture-based CNNs have dissimilar performances. The experiments are conducted with a number of epochs up to 50 because the test accuracy function is observed not to increase its values, reaching out a sort of plateau. As we notice in Table 1, the first model we test is the one with the worst performance over the task of muon detection since it gets stuck with percentages around 60%. As we go along with a growing number of layer architectures, we notice better performances. In order, SqueezeNet, GoogLeNet and ResNet-50 reach out test

Table 1 Validation accuracy values of CNN architectures for the muon case study

Validation accuracy of CNNs				
No. of epochs	Flattened network	GoogLeNet	ResNet-50	SqueezeNet
2	0.67	0.43	0.57	0.52
3	0.65	0.85	0.82	0.65
10	0.63	0.88	0.87	0.70
20	0.63	0.89	0.89	0.73
50	0.62	0.89	0.90	0.80

accuracy values equal or more significant than 80%. While SqueezeNet can achieve up to 80%, models with even a more-in-depth architecture, such as GoogleNet and ResNet-50 score top test accuracy values, 89% and 90% respectively. That said, all performances of GoogleNet and ResNet-50 being almost equal, ResNet-50 turns out to be preferable in terms of the computational burden. Validation accuracy is used as metrics to evaluate the performances of four CNN architectures and is defined as follows:

$$\text{Validation Accuracy} = \frac{\text{Number of correct predictions on validation dataset}}{\text{Total number of predictions made}}$$

4 The Gamma / Hadron Case

The capability to reduce the hadron background is one of the key aspects that determine the sensitivity of an IACT. Currently, the main data reduction and analysis software of the operating IACT facilities adopt methods based on Machine Learning algorithms. Although the performance of these methods has proven over the years to be rather robust and reliable, it may be overcome by new methods, in particular those based on Deep Learning.

To assess the performance of Deep Learning architectures we conducted several experimental sessions over raw data, whose Night Sky Background has not been suppressed.

Since in a Deep Learning approach the relevant parameters of the image are automatically chosen by the network, and the relevant shape features of the image are sorted out, Night Sky Background suppression is automatically achieved and no preliminary cleaning is requested.

4.1 Models

This section is devoted to the description of the Deep Learning architectures employed to test the Transfer Learning paradigm on the gamma/hadron separation from IACT data. In greater detail, we adopt the Convolutional Neural Networks (CNNs) [14] which are already pre-trained over the ImageNet repository. It is worth mentioning that CNNs have been widely adopted as one of the most accurate methods over tasks such as object recognition, suspicious region detection in biomedical imaging, speech recognition and semantic analysis. Each architecture needs to abide by some rules and constraints given by its own layers size, the number of layers, pooling, stride and hyper-parameters, which characterise the overall structure of the CNN

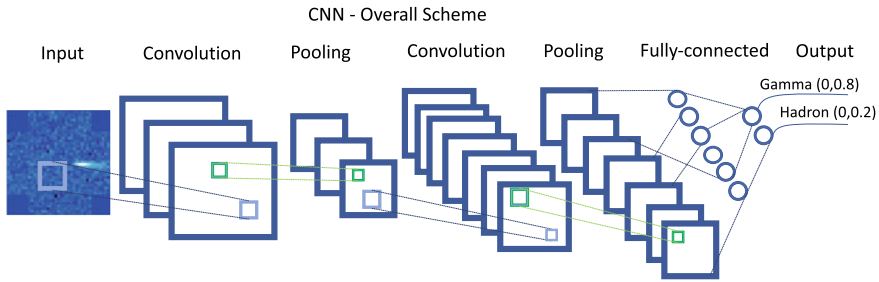


Fig. 3 The graphical representation of a basic CNN with convolutional and pooling layers

stack. Here we describe the most important detail of Flattened Network, GoogLeNet, ResNet-50, MobileNet, SqueezeNet. More specific descriptions can be found in the appendices. Figure 3 shows a graphical representation of a basic Convolutional Neural Network with convolutional, pooling and fully-connected layers, which are shared through the most of CNN architectures. Apart from the above-mentioned layers, a fundamental role is played by the so-called feature maps [15], that is, the output of filtering at each of the layers composing a CNN. A feature map is the result of the spatial filtering operation of input images with kernels such as the convolution and the pooling ones. The pooling layer and the stride parameter generally act to downsample the input image and extract features from the image itself, giving rise to a certain number of feature maps. The computational burden of the above steps is shared through the whole architecture and concerns mostly the size of kernels and pooling filters along with stride parameters. The role played by the size of filters is fundamental for the extraction of sized features in images. Filters with size 3×3 , 5×5 or even 7×7 are conventional in CNN stacks. It is expected that the size of filters increases along with the dimensions of the images to inspect. The convolutional layer is meant to filter out some particular spatial features from input images. A spatial filter mask slides across the image using a sliding step parameter called stride. One can briefly state the output size of the filtering as well as the number and size of feature maps to be affected by two factors such as the kernel size and the stride value. The Flattened Network has been proposed by Jin et al. [6] to achieve the state-of-the-art results in terms of classification and recognition and to obtain a speed-up in the training time. All of that is achieved using a more lightweight architecture, that is, a reduced number of parameters to avoid redundancy of filters in the Convolutional Network. Jin et al. separated the conventional 3D convolution filters into three consecutive 1D filters. This step goes under the name of 3D filter separation under rank one. A graphic scheme is given in Fig. 4. The main idea behind it is all over the Convolution layer. The output of three consecutive 1D filterings gives an equivalent representation of the 3D filter if the rank is one. In greater details, Jin et al. chose a CNN model architecture as the one proposed by Srivastava and Salakhutdinov [13] with a smaller multilayer perceptron.

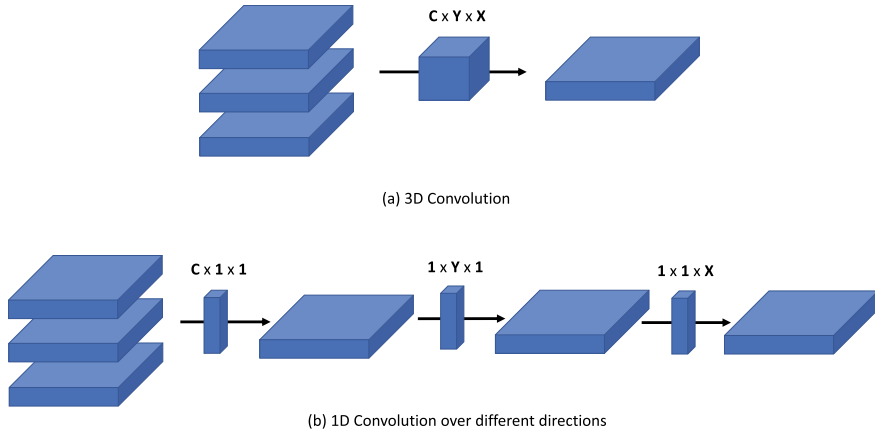


Fig. 4 In order, a graphical scheme of 3D convolution (upper row) and the corresponding 1D convolutions over different directions

5 Results

In this section, we give a detailed description of the experiments we ran with different CNN architectures and models. On the first instance, we detail the experimental sessions concerning the training process to draw some considerations about how diverse models suit the gamma/hadron separation. The dataset we use during our experimental trials consists of 10150 images with size 56×56 pixels and a standard Night Sky Background value. We split the whole dataset into two parts: the first 80% is used as a training set; the remaining images are used during the test trials. The training set is, in turn, split into two parts so that 20% of training images are used as a validation set. We highlight that in order to work with CNN architectures we need to abide by some requirements, such as the size of the input layer. Both test and training images go through resize transforms to make them CNN input layer compliant. In the mentioned architectures, SGD (Stochastic gradient descent) is adopted as an iterative learning algorithm to carry out the process of training over a dataset. Epochs and batch size are hyper-parameters which initialise SGD; the number of epochs controls the number of complete steps through the training set. The batch size handles the number of samples the training step goes through before the internal model parameters are updated.

We assess the accuracy of the CNN architectures using a different number of epochs (2, 3, 10, 20, 50). The batch size is experimentally fixed to 64. Dropout regularization parameter is set to 0.2. As a first step, we evaluate the performances of CNNs in the task of the gamma/hadron separation using the validation accuracy as described in Sect. 4. As it can be observed in Fig. 5, ResNet-50 achieves better performances than the other CNNs. Upon that, we decide to keep going on with our tests using ResNet-50 architecture and trying to assess the ability of knowledge inference over data apart from validation and training sets. As well as standard

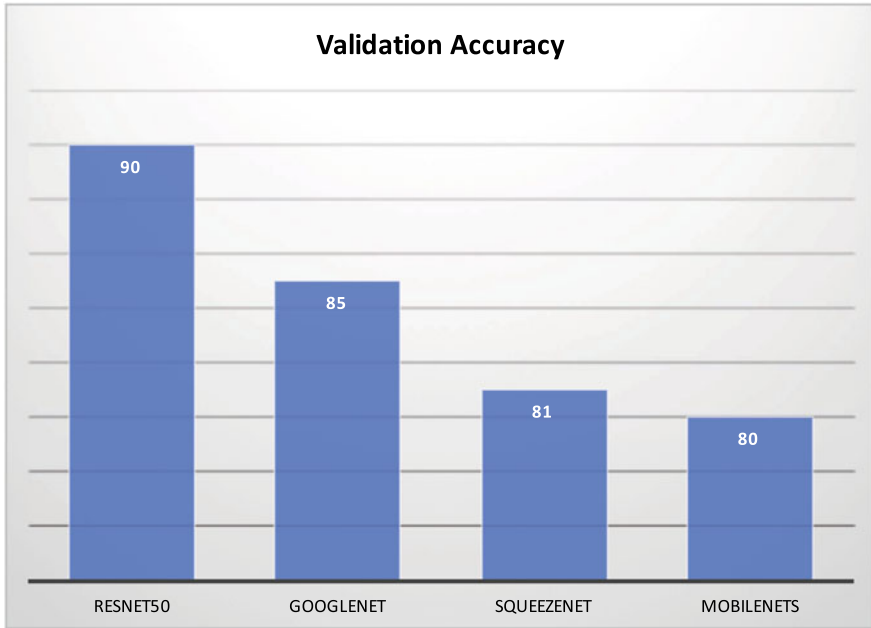


Fig. 5 Validation accuracy rates of four architectures are given with 50 epochs

Machine Learning metrics such as validation and test accuracy, we report results using the Quality Factor as defined down below:

$$Q = \frac{\epsilon_{\gamma}}{\sqrt{\epsilon_{bkg}}}$$

From the equation above, ϵ_{γ} is the γ rate while ϵ_{bkg} represents the hadron acceptance rate. The γ acceptance rate is defined as the correctly classified γ events out of the total number of γ events. The hadron acceptance rate is defined as the ratio of proton events which behave like γ events after the γ -hadron classification. On the other side, the hadron rejection rate is the number of hadron events which have been correctly classified out of the total number of hadron events. The larger Quality Factor, the better gamma/hadron discrimination capabilities of the method.

We set up two runs with different Night Sky Background levels, described as it follows:

- Run no. 1 consists of 10150 images with standard Night Sky Background value (the same Night Sky Background as in the training set);
- Run no. 2 consists of 10150 images with one and a half times the Night Sky Background value in Run 1.

The network we fine-tuned over the gamma/hadron application domain appears to be Night Sky Background sensitive (see Fig. 6). Zooming in on Fig. 6a, b, our trained



Fig. 6 Gamma Acceptance and Hadron Rejection rates are given for different experimental runs (a) and their corresponding Quality Factor values. The experiments have been conducted with the ResNet-50 model which is shown to perform better during the training step as shown in Fig. 11

model achieves respectively 98.7% and 99.3% of gamma and hadron acceptance rates on the run 1 while, as long as we analyse experimental trials with higher Night Sky Background we observe decreases in both rates coefficients (gamma and hadron acceptance rates). The corresponding Quality Factor values shown in Fig. 6a directly derived from gamma and hadron acceptance rates show the performance of ResNet-

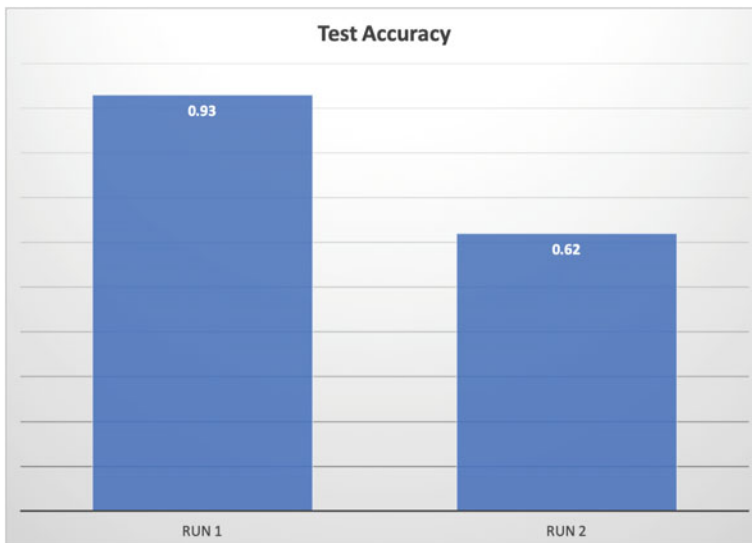


Fig. 7 Test accuracy rates of ResNet-50 model fine-tuned on gamma/hadron images for the gamma/hadron separation task. The test accuracy is defined as $Test Accuracy = \frac{Number\ of\ correct\ predictions\ on\ test\ dataset}{Total\ number\ of\ predictions\ made}$

50 equal respectively to 11.80 and 1.15 on run 1 and run 2. We want to highlight that ResNet-50 performances over run 1 simulation data is highly competitive with many state-of-the-art techniques as described in [19]. Because of our experimental results, Deep Learning architectures might resent from Night Sky Background values. From a pure detection viewpoint, looking at Fig. 7 higher values of test accuracy are shown to be over run no. 1 while results on run no. 2 dramatically decrease (the model is not able to make enough correct predictions of hadrons). Much more efforts would be necessary to clearly disambiguate the role played by the training set over the performances of the model on the test runs.

6 Conclusions

We have studied the application of different Machine and Deep Learning techniques to IACT images, with the aim of assessing their effectiveness in discriminating the images produced by muons, gamma-rays, and protons, respectively.

The muon case is relatively simple, since muon images have a well defined either circular or arc-shaped shape. This case was studied using a Machine Learning architecture, consisting of a linear stack of four layers. We selected a set of 16 image parameters, and we find that this approach is highly effective, reaching a 99.5% of correct identification over the muon sample, and 96.1% over the non-muon sample. We have also tested the effectiveness of a Transfer Learning approach over the domain of muon images. We find that the best results are achieved using the architecture with the highest level of depth. However Machine Learning is still more efficient.

The gamma/hadron discrimination case is more complicated: the difference between the two domains is vaguer whereas showers produced by hadrons may have an electromagnetic component, while showers produced by photons, especially by the low energy ones, or falling very close to the telescope axis may have a more irregular shape. In this case our study shows that an approach based on Transfer Learning may be effective in the discrimination, achieving a Quality Factor of ~ 12 , highly competitive with many state-of-the-art techniques as described in [19]. Considering that we have trained the network only on a limited case sample, we expect that a more efficient training, including, e.g., images with different levels of Night Sky Background, off-axis photons, and/or a spectral distribution optimized for the training, may improve this result.

Appendices: Deep Learning architectures

A. Flattened Network

The Flattened Network [6] consists of a consecutive sequence of one-dimensional filters across all directions. The architecture of the network is simple and lightweight, and we use it in order to test whether such a simple network is affected by overfitting using a limited number of images. Models affected by overfitting usually make predictions that fit the data at hand perfectly, but are not able to generalize knowledge from larger datasets. This generally happens when the system does not discriminate information from bias or background noise embedded with data.

Each output channel requires a filter $W \in \mathbb{R}^{C \times X \times Y}$ described as:

$$F_f(x, y) = I * W_f = \sum_{c=1}^C \sum_{x'=1}^X \sum_{y'=1}^Y I(c, x - x', y - y') W_f(c, x', y')$$

where f is an index of output channel, $I \in \mathbb{R}^{C \times N \times M \times F}$ is the input map, N and M are the spatial dimensions of the input. We assume the stride parameter to be one.

A rule of thumb to accelerate multi-dimensional convolution is to apply filter separation. To accomplish filter separation some constraints need to be considered. Under unit rank of the filter W_f , the unit rank filter \widehat{W}_f can be separated into cross-products of three one-dimensional filters as follows:

$$\widehat{W}_f = \alpha_f \times \beta_f \times \gamma_f$$

It is necessary to highlight that separability of filters is a strong condition. The rank of filter W_f is usually larger than one in practice. The problem mentioned above might affect the performance of the network over classification tasks. In our work we used Flattened Networks abiding by the condition that one or more convolutional layers are converted to a sequence of 1-dimensional convolutions.

B. MobileNets

MobileNets [9] represent a class of efficient models based on streamlined architectures employing depth-wise separable convolutions to set-up lightweight deep neural networks. As an innovative solution Howard et al. [9] introduced two different global hyper-parameters as a trade-off between latency and accuracy of the network. MobileNets have different application domains, even though their ideal destination is to allow developers and computer scientists for testing and training CNNs over mobile devices and embedded vision applications. Depthwise separable filters represent the base on which MobileNets are built. Depthwise separable convolutions

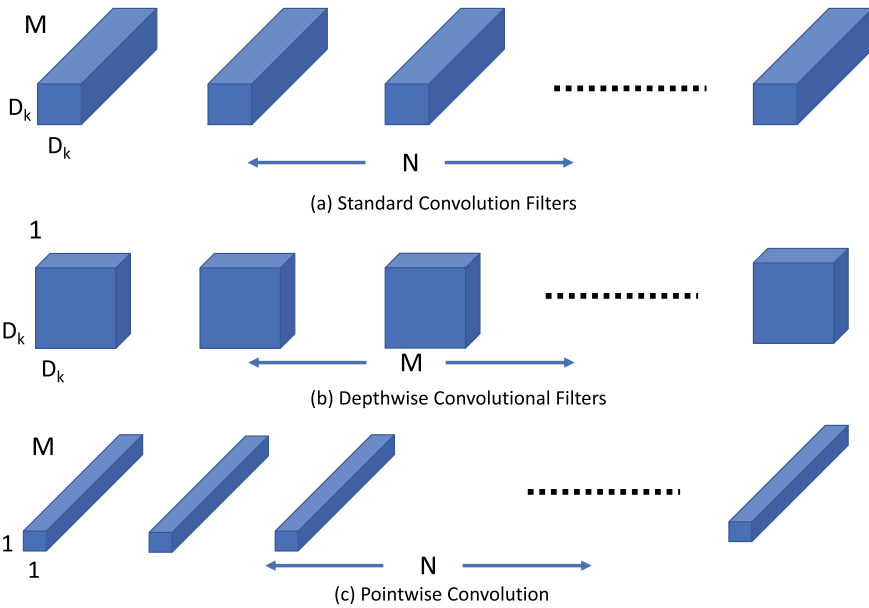


Fig. 8 MobileNets are based on Depthwise Convolutional Filters (b) and Pointwise Convolution (c), which make a noticeable parameter reduction over the architectures based on Standard Convolution Filters (a)

make use of factorization of convolutions into a depthwise and a 1×1 pointwise convolution. A standard convolution layer both filters and combines the input into a new set of outputs. The depthwise separable convolution splits this into two layers, the first one is devoted to filtering inputs while the second one is for the combination of inputs into a new series of outputs. The innovation of the proposed scheme in MobileNets is depicted as in Fig. 8 where standard Convolutional Filters are substituted with Depthwise Convolutional Filters and Pointwise Convolution allowing for a reduction of a great number of the architecture parameters. By zooming in Fig. 8 we notice that standard convolutions carry out a computational burden of:

$$D_k \cdot D_k \cdot M \cdot N \cdot D_f \cdot D_f$$

where M represents the number of input channels, N is the number of output channels, $D_k \cdot D_k$ represents the size of the kernel while $D_f \cdot D_f$ is the feature map size. Depthwise convolution with one filter per input channel turns out to have a cost of:

$$D_k \cdot D_k \cdot M \cdot D_f \cdot D_f$$

Depthwise convolution filters input channels but it does not combine the input into a new set of outputs. For this reason another layer is needed to combine the results of

Depthwise Convolutions filtering. That is accomplished with a linear combination of Depthwise Convolution using a 1×1 convolution. The latter one is called depthwise separable convolution whose computational cost is as it follows:

$$D_k \cdot D_k \cdot M \cdot D_f \cdot D_f + M \cdot N \cdot M \cdot D_f \cdot D_f$$

As described in [9], a reduction in computation is achieved expressing convolution as a two-step process of filtering. By adopting 3×3 Depthwise separable convolutions, MobileNet is able to achieve up to 9 times less the computation than standard convolution. In our work we are interested in assessing the performance of MobileNets in the topic of IACT. For a more-in-depth description of the overall architecture the reader is remanded to [9].

C. SqueezeNet

SqueezeNet [10] architecture aims to leverage the reduction of parameters to deliver proper levels of accuracy in classification tasks with a short latency time as well as MobileNets. As well as in MobileNets, Iandola et al. [10] aim to identify a model that has fewer parameters in such a way to carry out experiments with more efficient training, to have less overhead time in client-server Deep Learning-based applications and to rely upon available architecture in embedded deployment. Iandola et al. [10] focused their efforts on the so-called model compression which has recently arisen around the objective of compressing existing CNN models in a lossy way. Denton et al. [17] applied SVD (Singular Value Decomposition) to pre-trained CNN models, Han et al. [18] used a pruning algorithm over networks to compress model dimensions. When it was introduced, SqueezeNet represented an innovation in the field of model compression for CNN architecture because of the introduction of the so-called fire module out of which the architecture itself is built.

In greater detail, Iandola et al.[10] proposed SqueezeNet by leveraging three strategies. The first strategy consists of replacing 3×3 filters with 1×1 filters (1×1 convolution filters have nine times fewer parameters than 3×3 convolution filters). The second strategy is to decrease the number of input channels to 3×3 filters. The third strategy consists of apply downsampling late in CNN to achieve larger feature maps as convolution layer output across the most of layers in the network. The first two strategies concern mainly decreases of parameters in CNN architectures, and the third one is focused on accuracy maximization with fewer parameters. A *Fire* module consists of a squeeze convolution layer sized 1×1 which feeds into an expand layer that is a combination of 1×1 and 3×3 convolution layers. In Fig. 9 a simplified scheme of this module is given. Furthermore, the module can be tuned up using three different hyper-parameters. In our work, we adopt the architecture of SqueezeNet with a standalone convolution layer, followed by a series of 8 *Fire* modules and a final convolution layer. Max pooling is performed along with the

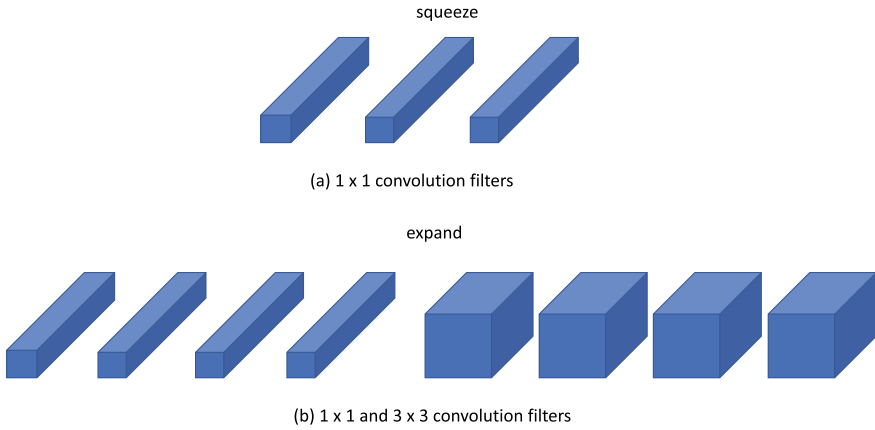


Fig. 9 SqueezeNet CNN architecture allows for a huge reduction of parameters, which is mainly based on the employment of squeeze (a) and expand (b) steps. The usage of 1×1 convolution filters make the CNN architecture more lightweight

network while the stride parameter is set to 2. The reader who is interested in further and more detailed description is remanded to the reference paper [10].

D. GoogLeNet

GoogLeNet is crafted to be an efficient deep neural network for computer vision tasks and its performances over the context of classification have been widely assessed [8]. Szegedy et al. [8] proposed an architecture whose main hallmark is the improved utilization of computing resources in the network itself. The main idea behind the GoogLeNet architecture is to find out how sparse structures in a convolutional network can be approximated by dense components. The authors of GoogLeNet engineered a network using a layer-by-layer approach where high statistic correlation values of the last layer are used to group visual features (boundaries, edges, contours, motifs) in clusters. These clusters give rise to units of the next layer and, at the same time, are connected to the previous layer. *Inception* module is described in Fig. 10. The current version of *Inception* includes layers with filters sized 1×1 , 3×3 , 5×5 . Furthermore, max-pooling filters are added in the architecture as successful elements in state-of-the-art CNNs. Each unit from the earlier layers corresponds to a region of the input image. As depicted in Fig. 10, the visual information coming out of the previous layer is conveyed onto the next one through both convolution and pooling filters which are combined using a filter concatenation. *Inception* modules are piled up on top of each other. The outputs of inception modules are statistically correlated with the corresponding layers of the network: features of higher levels are expected to decrease in spatial density when captured by higher layers. It is necessary

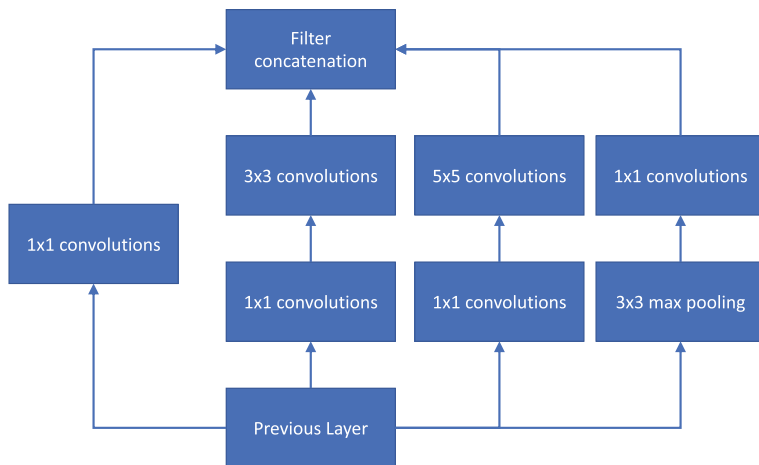


Fig. 10 The Inception module allows for multiple outputs coming out of the previous layer to be combined using a filter concatenation

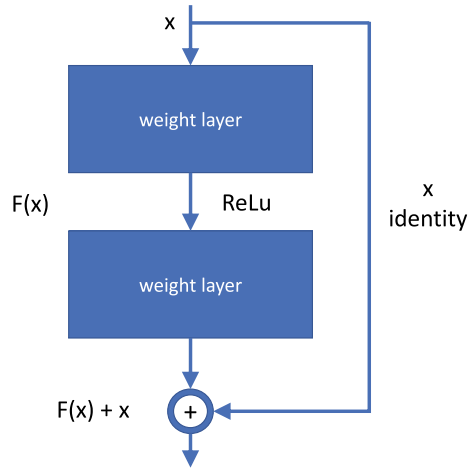
to highlight that because of technical reasons, such as memory efficiency, *Inception* modules are added only at higher layers. The overall network counts in 22 layers when only layers with parameters are considered (no pooling layers are counted in). If we consider each block inside *Inception* modules the overall networks can count up to 100 layers (this number is affected by the infrastructure system set-up). Whoever interested in further insights on the architecture of GoogLeNet is remanded to the reference paper [8].

E. ResNet-50

He et al. [7] proposed Deep Residual Learning to generally ease the process of training of deep networks, which are likely to get through the problem of degradation. When the depth of networks increases accuracy gets saturated and then degrades rapidly. This kind of degradation is not caused by overfitting, and what sounds more surprisingly is that adding more layers makes the training accuracy even lower. He et al. [7] addressed the issue of training accuracy degradation with the introduction of Deep Learning Residual. Rather than considering Deep Learning models fitting a particular mapping between input and output, they approach the improvement of training accuracy from a different perspective. They let a stack of layers fit a residual mapping function. Denoting the original mapping function as $H(x)$, the Deep Residual Learning is based on the position of letting the layer stack fit the residual function defined as down below.

$$F(x) := H(x) - x$$

Fig. 11 A building block of Residual Net



Following this approach, the original mapping problem is proposed in a new form as $F(x) + x$. As a new formulation based on a simple summation, it can be realised using feedforward networks with shortcut connections as in Fig. 11. Shortcut connections simply add new layers with identity mapping. Their outputs are added to the outputs of the stack. Using shortcut connections is an excellent way to avoid new parameters because they are based on identity mapping. The intuition of Deep Residual Learning authors is that multiple nonlinear layers can asymptotically approximate complicated function representing the input of the training process. The reformulation of learning is meant to deal with the training accuracy degradation problem. He et al. named ResNet (Residual Network) after the Deep Residual Learning formulation. The standard version of ResNet is 34 parameter layer sized where shortcut connections turn the network structure into its residual counterpart. A more in-depth version of ResNet has also been proposed with a bottleneck building block whose each residual function F involves a number of three layers rather than two. In our work we conducted different experiments with ResNet-50, which counts a number of 50 parameter layers.

References

1. Li, J., Du, Q., Sun, C.: An improved box-counting method for image fractal dimension estimation. *Pattern Recognit.* **42**(11), 2460–2469 (2009)
2. Pagliaro, A., D’Anna, F, D’Alí Staiti, G.: A multiscale, lacunarity and neural network method for γ/h discrimination in extensive air showers. In: *Proceedings of the 32nd International Cosmic Ray Conference* (2011)
3. Huang, Z., Leng, J.: Analysis of Hu’s moment invariants on image scaling and rotation. In: *Proceedings of 2nd International Conference on Computer Engineering and Technology* (2010)

4. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. *J. Mol. Biol.* **147**, 195–197 (1981)
5. Weiss, K., Khoshgoftaar, T., M., Wang, D.: A survey of transfer learning. *J. Big Data* **3**(1), 9 (2016). SpringerOpen
6. Jin, J., Dundar, A., Culurciello, E.: Flattened convolutional neural networks for feedforward acceleration (2014). [arXiv:1412.5474](https://arxiv.org/abs/1412.5474)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
8. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
9. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., and Wang, W., Weyand, T., and Andretto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications (2017). [arXiv:1704.04861](https://arxiv.org/abs/1704.04861)
10. Iandola, F. N., Song, H., Moskewicz, M. W., Khalid, A., Dally, W. J., Keutzer, K.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size (2016). [arXiv:1602.07360](https://arxiv.org/abs/1602.07360)
11. Russakovsky, O., Deng J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis. (IJCV)* **115**(3), 211–252. <https://doi.org/10.1007/s11263-015-0816-y> (2015)
12. Wang, J., Perez, L.: The effectiveness of data augmentation in image classification using deep learning. In: Computer Vision and Pattern Recognition (2017)
13. Srivastava, N., Salakhutdinov, R.R.: Discriminative transfer learning with tree-based priors. In: Advances in Neural Information Processing Systems, pp. 2094–2102 (2013)
14. Agarwal, R., Diaz, O., Llad’o, X., Yap Moi, H., Mart’i, R.: Automatic mass detection in mammograms using deep convolutional neural networks. *J. Med. Imaging* **6**(3), 031409 (2019)
15. LeCun, Y., Boser, B., Denker, J.S. et al.: Handwritten digit recognition with a back-propagation network. In: Advances in Neural Information Processing Systems (1990)
16. LeCun, Y., Bottou, L., Bengio, Y., et al.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
17. Denton, E.L., Zaremba, W., Bruna, J., LeCun, Y., Fergus, R.: Exploiting linear structure within convolutional networks for efficient evaluation. In: NIPS (2014)
18. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural networks. In: NIPS (2015)
19. Sharma, M., Nayak, J., Koul, M. K., Bose, S., Mitra, A.: Gamma/hadron segregation for a ground based imaging atmospheric Cherenkov telescope using machine learning methods: random Forest leads. *Res. Astron. Astrophys.* **14**(11), 1491 (2014)

Intelligent Photometric Identification of Extragalactic Objects from AllWISE × Pan-STARRS DR1 Data



Vladislav Khramtsov, Volodymyr Akhmetov, Peter Fedorov, Sergii Khlamov, Artem Dmytrenko, and Anna Velichko

Abstract We present the results of identification of extragalactic objects within cross-matching result of the two photometric catalogues, AllWISE and Pan-STARRS DR1. To separate galaxies and quasars from stars, we constructed a machine learning model, trained on photometric information from optical and infrared wavelength ranges. The model was based on three essential procedures: construction of the autoencoder artificial neural network, separation of galaxies and quasars from stars with Support Vector Machine (SVM) classifier, and cleaning the AllWISE × PS1 sample from sources with unusual colour indexes with one-class SVM. As a training sample, we used a set of spectroscopically confirmed sources from Sloan Digital Sky Survey Data Release 14. In the result of applying the classification model to the AllWISE × PS1 sample, we created catalogue, containing 40 million extragalactic objects, and covering 3/4 of the celestial sphere up to $g=23^m$. A number of tests to examine the classification quality using different methods, namely data from spectroscopic surveys, and astrometric examination, are in agreement with each other. Tests indicate high purity ($\sim 98.0\%$) and completeness ($> 98\%$) of resulting catalogue at the faint end of magnitude range; classification quality still retains acceptable levels of 70% for purity and 97% for completeness for the brightest and faintest parts of g magnitude range.

1 Introduction

Classification of photometrically observed sources is a crucial stage to produce reliable astrophysical results. Notably, the identification of the extragalactic objects is a crucial point for improvement of the celestial reference frame [25, 26, 42, 58], morphological classification of galaxies [9], finding strongly lensed quasars [35, 61]; the large-scale structure of Universe analysis [12], and supporting the identification of

V. Khramtsov · V. Akhmetov · P. Fedorov · S. Khlamov (✉) · A. Dmytrenko · A. Velichko
Institute of Astronomy, V. N. Karazin Kharkiv National University, 35 Sumska Str., Kharkiv,
Ukraine
e-mail: sergii.khlamov@gmail.com

the host candidates for gravitational-wave events [22] also could be supplemented with the extragalactic catalogue, attracting the photometric redshifts [55].

The simplest method of identifying the extragalactic objects is applying the morphological criterion on images of sources. However, this method has significant limitations [65]. Though the low-redshift galaxies are resolved, images of the faint and/or distant galaxies as well as the quasars appear to be point-like in the main. Besides, in the high-density sky areas (for example, in the galactic plane), the blending phenomenon results in the false classification of the pairs of stars, the nearest neighbours, which are classified with any morphological criteria as extended sources, i.e., galaxies. Another classification method is in using the colour index values (after this referred to as colour). On the colour-colour or colour-magnitude diagrams (then—colour diagrams) locations of galaxies and quasars sharply differ from the locations of stars due to different forms of spectral energy distribution. It is known that utilisation of optical magnitudes only is not enough to perform the classification of sources with both high purity and completeness with colour diagrams [35]. Therefore, infrared information often complements the optical one, because of the combination of these two wavelength ranges allows describing the spectral energy distribution of sources more distinctly [3, 38]. Furthermore, using mid-infrared colours is a very effective way to distinguish quasars from the vast majority of passive galaxies and stars, because of quasars emit strongly at mid-infrared wavelengths range [7, 23, 63].

Although it is possible to separate low-redshift quasars from stars with using optical magnitudes (see e.g. [1, 14]), utilization of mid-infrared colour diagrams also helps in separation of quasars from stars and galaxies (e.g. the two-colour criteria [20, 32, 41, 44, 63] and the one-colour criteria [7, 62]). However, separating the galaxies in the same way is harder because of stars overlap with non-active galaxies at these colour diagrams (e.g., Fig. 12 in [67]). Kovács & Szapudi [37] presented approach enabling to identify galaxies with given near-infrared and mid-infrared colour indexes—with using of photometry from Two Micron All Sky Survey (2MASS) [59] and Wide-field Infrared Survey Explorer (WISE) [67], it is possible to separate extragalactic objects from the stars almost clearly.

For the problem of optimal source separation on the colour diagrams, the Support Vector Machine (SVM) [64] classifier has been established as one of the most popular geometrical classifiers. Classification of sources into groups with SVM comes down to the construction of separating hyperplane in a high-dimensional space of features. This method has found an application in the many works dedicated to the identification of extragalactic objects, including Solarz et al. [60], where the classification of objects on the infrared colour diagrams within AKARI [45] North Ecliptic Pole Deep Field (AKARI NEP) has been implemented; in Malek et al. [47] work, SVM was used to classify objects into stars, galaxies and quasars with their colours within optical-infrared wavelength range for the VIMOS Public Extragalactic Redshift Survey. SVM algorithm has been used in Kovács & Szapudi [37] to analyse the colours, which contribute to the separation of objects into extragalactic sources and stars with using the 2MASS×WISE catalogue. The re-adaptation procedure for the WISE×SuperCOSMOS galaxy catalogue [10], performed to identify the galaxies

using SVM separation in colour space, is proposed in paper from Krakowski et al. [38].

We present the method of the identification of the extragalactic objects within the wide-field photometric surveys. The resulting catalogue of galaxies and quasars is named Northern Extragalactic WISE×Pan-STARRS (NEWS), which covers almost 3/4 of the sky in the broad optical-infrared wavelength range up to $g=23^m$. To create the NEWS catalogue, we combined data from the AllWISE [17] survey in the mid-infrared range and Pan-STARRS [15] digital survey in optical and near-infrared ranges. Our catalogue is a result of the classification of the AllWISE×Pan-STARRS objects with SVM machine learning algorithm. The classification has been done on photometric information only, with automatic feature engineering using an autoencoder neural network.

2 Data and Sample Selection

2.1 AllWISE

The Wide-field Infrared Survey Explorer (WISE) [67] is a space NASA Medium Class Explorer mission that surveyed the whole celestial sphere in mid-infrared wavelength range using four filters— $W1$, $W2$, $W3$, $W4$ (3.4, 4.6, 12 and 22 μm respectively with angular resolutions of 6.1", 6.4", 6.5", and 12" in bands respectively). We used data from AllWISE catalogue [17] of 747.6 million sources. The catalogue is 95% complete up to $W1 < 17.1^m$, $W2 < 15.7^m$, $W3 < 11.5^m$, and $W4 < 7.7^m$ (in Vega system).

AllWISE photometry is problematic in high-dense regions because of the effects of saturation and blending. Also, the aperture photometry is insecure for the extended sources in AllWISE because of the lack of aperture corrections on source compactness. Keeping this in mind, we selected sources with existent measurements in $W1$, $W2$ profile-fit magnitudes (that are signed in the catalogue as $w1mpro$ and $w2mpro$, hereafter $W1$ and $W2$) and limited this sample by the following criteria: $snr_{W1} > 5$, $snr_{W2} > 2$ and $W1 > 13.8^m$, where snr_{W1} , snr_{W2} —signal-to-noise ratio of measurements in $W1$, $W2$ bands respectively. We used no information in $W3$, $W4$ bands because of its low sensitivity. In consequence, a sample comprising 508 854 970 million objects has been obtained.

2.2 Pan-STARRS DRI

Pan-STARRS Data Release 1 (PS1) catalogue [15] is the first release of Panoramic Survey Telescope and Rapid Response System (Pan-STARRS1), located in the Haleakala Observatory on the island of Maui in Hawaii. Within the Pan-STARRS1

survey, the observations of almost 30 000 sq.deg. ($90^\circ > \delta > -30^\circ$) involving optical and near-infrared filters (g, r, i, z, y filters with average wavelengths 481, 617, 752, 866, and 962 nm accordingly) were conducted up to the 23.3^m , 23.2^m , 23.1^m , 22.3^m , 21.4^m in each band respectively (on the 5σ limiting sensitivity level). PS1 catalogue consists of ≈ 1.9 billions of sources, mostly with measured photometry.

We used mean PSF magnitudes (in AB system) from the PS1 catalogue. These magnitudes provide more accurate measurement for the point-like sources than Kron-like or aperture photometry. We selected the sources within $14^m < g < 23^m$ and $14^m < r < 23^m$ magnitude ranges, and limited the sample by the errors of magnitudes ($\sigma_x < 0.5$ mag, where x denote one of the five filters). Reasons for selecting such limits are evident. First of all, extragalactic sources are almost absent in the bright part of optical magnitudes $< 14^m$. Besides, images of bright sources are saturated, leading to unreliable magnitude measurements. And from another side, limiting by $g < 23^m$ and $r < 23^m$ results in excluding the sources with low signal-to-noise ratio and more uniform limiting magnitude distribution on sky. After filtering, we received the sample of 835 713 490 objects.

2.3 Positional Cross-Matching

For each source from AllWISE catalogue, we have chosen the nearest one from the PS1 in the fixed circular window with $1.5''$ radius employing the software described in [4]. Thus, all of the sources from AllWISE, which have more than one counterpart in PS1, were excluded. To avoid cross-matching of fast nearby to the Sun objects (for example, brown dwarfs) with the absolute value of proper motion greater than ~ 10 mas yr $^{-1}$, the radius in use was found as optimal. In the result, we got cross-matched AllWISE \times PS1 photometric sample, consisted of ~ 200 million sources, for each of which the positions from the PS1 and seven magnitudes in optical-infrared filters were presented. The resultant sample will be below referred to as the investigated sample.

2.4 Training Sample: SDSS DR14

As a training sample, we have chosen the Sloan Digital Sky Survey Data Release 14 (SDSS DR14) [2]—a catalogue of spectroscopically confirmed sources. SDSS DR14 is the second release of Sloan Digital Sky Survey IV phase [11]. It is a cumulative dataset, including extended Baryon Oscillation Spectroscopic Survey [19] data and data from the second phase of Apache Point Observatory Galactic Evolution Experiment observations [43].

SDSS DR14 catalogue consists of 4 851 200 spectroscopically confirmed objects among which 2 541 242 objects are galaxies, 680 843 are quasars, and 928 859 are stars. To create a training sample, we have used well-confirmed objects only

($z_{\text{Warning}}=0$), and removed all stars with CLASS=GALAXY morphological flag, corresponding to the extended sources. Training sample was cross-matched with the investigated sample using $1.0''$ radius. As a result, we have obtained SDSS \times AllWISE \times PS1 sample composed of 1 784 610 objects for which seven magnitudes and confirmed classes are available. A resultant sample contains mainly extragalactic (1 474 458 galaxies and quasars) objects but also comprises stars (310 152 objects). This entire sample will be below referred to as the training sample.

3 Classification Model

3.1 Formal Description

The proposed classification model performs a probabilistic classification of objects into two classes: stars and extragalactic sources. As the input features, we have used the optical-infrared colour indexes [34]. We have used the machine learning to create the classification model, namely: the form of transformation from original colours into the feature space, outlier detector and separating hyperplane.

First of all, we have performed the construction of the feature space, in which the location of extragalactic sources differs from the location of the stars. Usually, only sure of the original colours are used to building feature space for classification [3, 37, 38, 66]. This process can be done through different feature selection methods (for example, with Principal Component Analysis). In our model, we used feature engineering as the approach to build feature space, that is fully automatic, instead of manual colour selection. A distinctive speciality of our classification model is an automatic creation of features, that are connected to all colours. To perform the creation of such features, an autoencoder [51] neural network was used.

Autoencoder, as a particular architecture of an artificial neural network, consists of two parts: encoder and decoder. Autoencoder represents data into a feature vector having lower dimensionality than the input vector (encoding) and recovers original data from these features (decoding). Quality of representation is determined by the accuracy of the recovering the colours in the output of the decoder. The main goal of the autoencoder learning within our model is the creation of the most informative features, that allow us to represent original colours most accurately.

In our model, we used the result of encoder only; decoder has been used only to learn the autoencoder and to control the quality of representation. In order to create the feature space, we used the deep autoencoder: encoding was performed through a four-layer fully-connected neural network. The encoder input consisted of 21 colours, and the output layer of encoder reproduced it as five features.

Decoding was performed using symmetrical, concerning the encoder, network. In order to provide the representation of 21 colours, calculated through $W1, W2, g, r, i, z, y$ magnitudes, we have used the data on all of the 1.8 million objects from the training sample. As a result of encoding, we successfully represented 21-dimensional

vector of colours by a 5-dimensional feature vector for each training object with recovering error (mean squared error) on the order of 10^{-5} mag.

The AllWISE×PS1 sample could contain sources with unusual (concerning the training sample) sources. In order to perform correct classification, we have conducted the preliminary excluding of objects with unusual colours in the AllWISE×PS1 catalogue. That was done via One-Class Support Vector Machine (OCSVM) [57] within the constructed feature space.

The next step of classification model construction is a separation of extragalactic objects from the stars. We applied the Support Vector Machine (SVM) [64] method for our purpose. SVM classification has a simple geometrical interpretation. The algorithm receives a set of training objects with known features and classes at the input and returns the equation of optimally separating hyperplane. Under the term ‘optimally separating hyperplane’ we mean a hyperplane, which is located between different classes and which is the most distant from the nearest points of both classes (so-called, support vectors). To avoid the shift of separating hyperplane due to class imbalance [37], we have attributed weights for the sources from one of the class. The weights were equal to the ratio of the number of sources from the unweighted group to the number of sources from the weighted group.

3.2 *Technical Details*

The final architecture of autoencoder was constructed with simple searching over parameters of artificial neural network (number of layers, number of neurons, activation functions, etc.) with the aim to encode input colour vector in such feature space, where extragalactic objects are separated from stars. Our artificial neural network was constructed manually, without adopting the automatic algorithms, e.g., Neural Architecture Search methods¹. Autoencoder in use consisted of seven layers in total (with 21-15-10-5 neurons in the encoder, and with non-linear connections between layers; decoder was constructed symmetrically). Input values were scaled from 0 to 1. Weights were initialised by random orthogonal matrices [54] and were optimised with Adaptive Moment Estimation (Adam) algorithm [36], with a mean squared error loss function.

To separate extragalactic objects from stars, we used linear SVM, trained to classify sources within five-dimensional representative feature space. In this context, SVM had to be tuned over one free parameter, namely C . We have searched for the optimal C parameter with the simple grid-search method and five-fold cross-validation; parameter tuning was done concerning the value of area under receiver operating characteristic (ROC AUC) [13]. We have chosen $C=1$ as the optimal value, corresponding to the maximal ROC AUC metric equal to $\approx 99.3\%$.

Then, we have trained SVM with five-fold cross-validation on SDSS training data. Important to note, that with using five-fold cross-validation, it is possible to get

¹See, for example, <https://www.automl.org/automl/literature-on-neural-architecture-search>.

validation estimations of the classes for all of the training sources, without worrying about overfitting. Thus, for validation with SDSS DR14 sample, we used out-of-fold predictions, received after five-fold cross-validation.

OCSVM with radial basis function kernel was optimised over two parameters (γ and ν) via simple grid-search with using five-fold cross-validation on the training sample. We choose $\nu=10^{-5}$ and $\gamma=10^{-3}$, with which the fraction of training sources, predicted as unusual objects, was minimal and equal to 0.004%.

In result of the training, we obtained the algorithm which is able to classify sources into the stars and extragalactic objects with the ROC AUC=99.32%; the purity and the completeness of the extragalactic sample are estimated to be about $\approx 99.75\%$ and $\approx 99.85\%$ respectively, as we directly obtained with five-fold cross-validation.

4 Application of Approach

In this section, we present the results of our approach application to the extragalactic objects identification within AllWISE \times PS1 sample. To create the final galaxy catalogue, we mapped colours of each source into the five-dimensional feature space, provided outlier detection with OCSVM, and classified the sources with SVM.

4.1 Classification Result

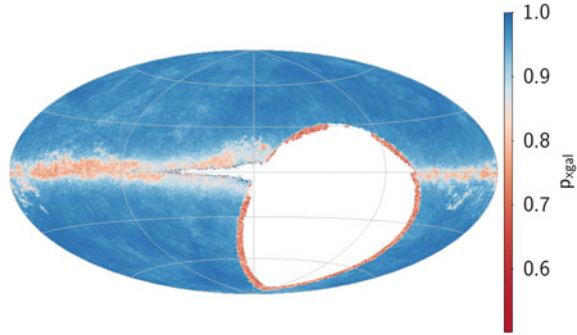
After training the OCSVM as an anomaly detector, we applied it to the AllWISE \times PS1 sample. As a result, $\sim 300\,000$ sources were identified as objects with unusual features (colours in fact). These sources were located outside the region of feature space, occupied by training objects.

The sources, which were not excluded with OCSVM, were classified with trained SVM. To decide the belonging of sources to a particular class, we used additional parameter, released by the SVM classifier, namely a probability, that source belongs to extragalactic group p_{xgal} . In terms of the SVM, probability p_{xgal} is a distance, expressed through the sigmoid function [50], between a feature vector of a source and the separating hyperplane.

We considered that a source is extragalactic objects if the corresponding probability is $p_{xgal} > 0.5$. This criterion allowed us to derive the NEWS catalogue of 40350492 objects. The g magnitude distribution of sources from NEWS peaks at $g \sim 21.7^m$, showing an obtained catalogue of extragalactic objects is one of the most in-depth catalogues, covering significant fraction of the sky.

NEWS catalogue consists of the extended sources in the main. We calculated the amount of point-like sources using $g - g_{Kron}$ value, where g_{Kron} —Kron-like magnitudes [39]. Difference between PSF and Kron-like magnitudes for a source is a criterion of its ‘pointness’ [24]. This difference is almost zero for the point-like sources, and, in most cases, greater than zero for the extended sources. We assumed

Fig. 1 Sky distribution of the p_{xgal} for 40 million sources from the NEWS catalogue in Aitoff projection (galactic coordinates, the origin of the coordinate system is at the centre)



the pointiness with the following criteria: $g - g_{Kron} < 0.05$ and $g - g_{Kron} > -0.25$ within $g < 21^m$; all sources, failed criteria, we assumed as extended sources. In result, ~ 2 million sources were identified as point-like among ~ 38 million sources with available g_{Kron} magnitude in PS1 catalogue.

Besides extended sources, NEWS catalogue contains quasars also. By virtue of our classification model construction, we are not able to separate quasars from galaxies naturally, but we can do this postfactum, with using criteria allowing us to identify quasars with mid-infrared colours. For example, with using the criterion [62], namely $W1 - W2 > 0.8$, we identified 2 286 418 quasars in the NEWS, $\sim 1\,000\,000$ of which are point-like sources. We note, that rest $\sim 1\,000\,000$ sources could be, in fact, stars, which could be removed from the NEWS catalogue as extremely suspicious sources.

The sources within the high-extinction region of the sky from the NEWS galaxy catalogue could have insecure classification. This is clearly shown in Fig. 1—probability p_{xgal} is strongly dependent on sky location and is $p_{xgal} < 0.9$ at $\delta < -30^\circ$ and within the Zone of Avoidance.

5 Validation with External Data

Various spectroscopic surveys contain precise information about classes of the observed sources. In this regard, we are especially interested in catalogues of confirmed galaxies as well as in catalogues of confirmed stars. The classes of such sources could be treated as the ground truth.

To validate NEWS catalogue, we, first of all, need to match in AllWISE \times PS1 sample spectroscopically confirmed galaxies and stars, and then compare the actual classes (received by spectroscopic data) with predicted by our model classes. This approach allows estimation of the main parameters of classification quality—purity P_G and completeness C_G .

The purity of the NEWS catalogue was estimated not with using this definition, due to unreliability of this estimation in the case of high-class imbalance in validation sample. As it is shown below, the catalogues in use contain an order of magnitude

fewer stars than galaxies. This remark is not relevant for completeness estimation, where the count information about galaxies only is used. Thus, to estimate a purity of NEWS catalogue, we used the Matthews correlation coefficient (MCC) [46].

5.1 Spectroscopic Datasets

As validation samples, we used the following catalogues of spectroscopically confirmed sources:

- Deep Imaging Multi-Object Spectrograph (DEIMOS) [31];
- Galaxy And Mass Assembly (GAMA) Data Release 3 [8, 21];
- Large Sky Area Multi-Object Fiber Spectroscopic Telescope (LAMOST) Data Release 4 [16];
- SDSS DR14 (see Sect. 2.4);
- VIMOS Public Extragalactic Redshift Survey (VIPERS) Data Release 2 [56];
- zCOSMOS [18].

We cross-matched each catalogue with AllWISE×PS1 within 1.0'' circular radius for subsequent validation. The cross-matching results, as well as criteria of selecting sources from validation data, we describe in Table 1. In Fig. 2, we present g -band magnitude distribution of the validation samples within AllWISE×PS1 catalogue. This clearly shows different magnitude distributions for each validation dataset and allowing the investigation of classification quality at different magnitudes.

Table 1 Number of stars and galaxies within validation catalogues, matched in AllWISE×PS1. To select reliable data, we used criteria, separately described in the second column

Survey	Criteria	Galaxies	Stars
DEIMOS	z is not $-1, Q = 2$	438	285
GAMA	$z > 0.05$ $z < 0.90$ $NQ > 1$ <i>survey</i> is not <i>SDSS</i>	97 820	0
LAMOST	<i>class</i> is not <i>None</i>	30 093	1 107 982
SDSS	see Sect. 2.4	1 474 458	310 152
VIPERS	$zflg > 2$	2 517	312
zCOSMOS	$z > 0$	2 188	0

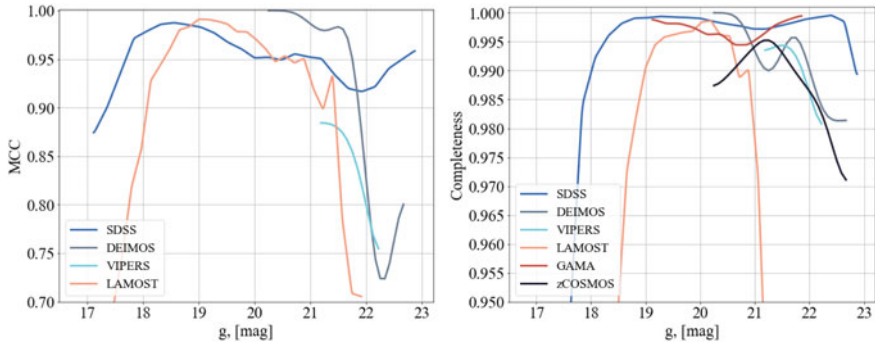


Fig. 2 Dependence of classification quality (MCC —left plot, completeness—right plot) for the AllWISE \times PS1 spectroscopically confirmed sources on g -band magnitude

5.2 Validation with Galaxy Catalogues

We have estimated the purity and completeness of NEWS catalogue with using spectroscopic datasets, presented in Sect. 5.1. These catalogues, in matching with AllWISE \times PS1, are variously distributed over the g -band magnitude, allowing the estimation of classification quality starting from the bright ($g \sim 17^m$) part and finishing at the faint ($g \sim 23^m$) end of magnitude range. We stress that the classification quality is a function of magnitude because of the dependence between the precision of features, used in classification, and precision of magnitudes, which, in its turn, degrades with magnitude increasing.

To analyse the purity of the NEWS catalogue, we used MCC metric, with which we compared the predicted by our model classes with actual classes from catalogues, containing galaxies and stars, namely from VIPERS, DEIMOS, SDSS, and LAMOST samples. Other catalogues (zCOSMOS and GAMA) could not be used to analyse purity due to the absence of stars in these samples. Completeness of the NEWS catalogue was estimated employing all of the validation catalogues.

The result of purity estimation is shown in Fig. 2 (left plot) as dependencies of MCC values against the magnitude for six various validation samples. MCC curves are entirely consistent for the SDSS, LAMOST, DEIMOS, and VIPERS datasets, showing the similar trends, namely low classification quality ($MCC < 0.9$) at bright range ($g < 18^m$), satisfactory quality ($MCC > 0.9$) in the range $18.0^m < g < 21.5^m$ with monotone decreasing of classification quality to the faintest magnitudes ($g \sim 23^m$). Sharp declines of MCC curves at the edges of magnitude ranges for almost all of the validation samples are the result of a relatively small number of sources at these magnitude bins. Obviously, that smaller number of sources results in more enormous contribution in decreasing MCC of each wrong classified by our model source.

We also carried out a similar manipulation to investigate the completeness of the NEWS catalogue. Result of completeness estimation of the NEWS with using various

samples of galaxies is shown in Fig. 2 (right plot). Completeness of the NEWS is 95% at $g \sim 18^m$, > 99% at $g \sim 20^m$, and 98% at $g \sim 22.5^m$.

Estimated purity and completeness of the NEWS are mitigated at the bright $g < 18^m$ and the faint $g > 22^m$ parts magnitude range and reach 70% and 97% respectively.

Of course, the validation samples are not exhaustive, because of the validation samples cover restricted regions of the five-dimensional feature space (in virtue of specific target selection for spectroscopic observations), and the sources, included in the validation samples, could not fully reflect the variation nature of objects, comprising AllWISE \times PS1 dataset. Therefore, validation with such spectroscopic samples could not guarantee the exact concurrence of purity and completeness estimations with the real ones. However, despite the substantial difference between sources of validation samples, the behaviour of estimated purity and completeness is, in general, coincided between catalogues, that is an indicator of stable classification quality estimation.

5.3 Astrometric Validation

Validation with external spectroscopic data gives the direct, consistent estimations of purity and completeness of the NEWS catalogue. However, the stars and extragalactic objects, besides spectroscopic properties, have other features of their nature, using of which could provide validation of the galaxy catalogue.

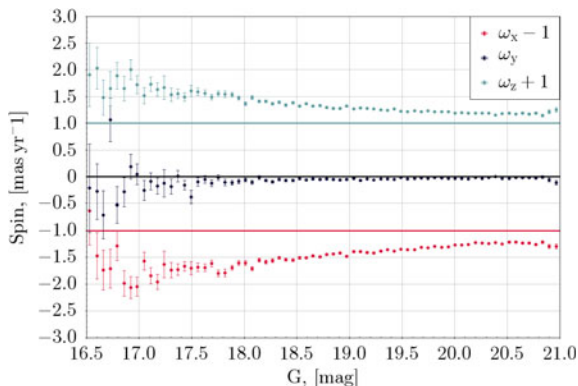
Examples of such features are proper motions. Extragalactic objects, due to their remoteness, are motionless sources on the sky, despite the stars. At the precision level ~ 1 mas, positions of extragalactic sources could be assumed independent on time, approximately during ~ 100 years. Great platform to carry out such test of NEWS sources is the second data release of Gaia astrometric mission (Gaia DR2) [28–30, 42].

The reliability of astrometric validation was shown in [48], where candidate quasars were validated with an analysis of the mean values of proper motions and parallaxes. But for our analysis, we used a model of rigid-body rotation of the reference frame, with which we can estimate the ω_x , ω_y , ω_z —the components of the angular velocity vector, knowing the components of proper motion for reference sources.

If the NEWS catalogue contains extragalactic objects (galaxies and quasars) only, the rotation of the system, defined by positions and proper motions of these sources in respect with reference frame Gaia-CRF2 will be absent. However, in the case of nonzero components of rotation vector, one could assume some effects, the most likely of which is the existence of stars in the NEWS sample.

An important feature of Gaia DR2 is that point-like sources only are included in this catalogue [52]. Thus, cross-matching NEWS and Gaia DR2 result in the sample of point-like sources—quasars, and, in the case of wrong classification, stars. This allows estimation of the contamination in the NEWS by stars.

Fig. 3 Dependence of components of rotation vector ω_x , ω_y , ω_z of the system, defined by positions and proper motions of 1.2 million NEWS sources concerning the Gaia-CRF2 on the G magnitude



We notice that Gaia DR2 contains measured proper motions for the sources that meet the following conditions:

1. mean G -band magnitude $G < 21^m$;
2. $astrometric_sigma5d_max < 1.2 \times \gamma$,

where $astrometric_sigma5d_max$ —is a longest principal axis in the five-dimensional error ellipsoid (correspond to the errors of five astrometric parameters), and $\gamma = \max[1, 10^{0.2 \times G - 3.6}]$ [42].

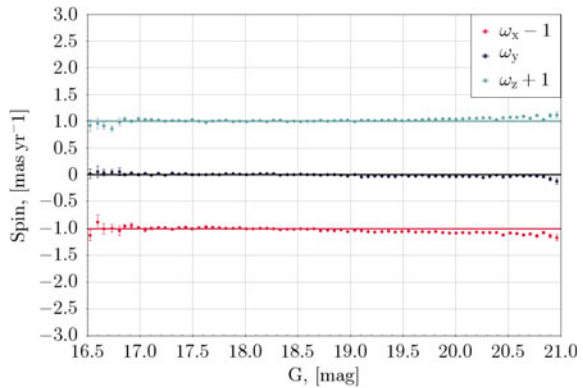
After cross-matching of the NEWS catalogue with Gaia DR2 within $0.5''$ radius, we have found 4.3 million common sources. For most of the matched in Gaia DR2 galaxies from NEWS had the measured positions only, possibly due to the high errors in photocentre determination. Thus, we constructed a sample, containing 1 242 635 sources from NEWS in Gaia DR2, which passes the criteria listed above (and, as a result, have the measured proper motions), are located above $\delta = -30^\circ$, and are outside the regions with high extinction ($E(B - V) < 0.1^m$).

In addition to that, 95% of the matched sources are within the magnitude range $g < 21.6^m$. Accordingly, the NEWS catalogue, except the sources within the Zone of Avoidance ($E(B - V) > 0.1^m$), consists of 20.05 million sources up to the $g=21.6^m$ magnitude limit.

Assuming our sample comprising extragalactic objects only, we expect the zero components of rotation vector with using the rigid-body rotation model. In Fig. 3 the dependencies of components of rotation vector, defined by positions and proper motions of 1.2 million sources, on G -band magnitude are shown. Vertical error bars represent the dispersion of components within the magnitude bin. The components ω_x , ω_z were shifted by $\pm 1.0 \text{ mas yr}^{-1}$ for better visibility.

As we can see, the values of ω_x , ω_y , ω_z reach $(-0.8, -0.1, 0.6) \text{ mas yr}^{-1}$ at $G=17^m$, $(-0.4, 0.0, 0.3) \text{ mas yr}^{-1}$ at $G=19^m$, and $(-0.3, 0, 0.2) \text{ mas yr}^{-1}$ at $G=21^m$. Reason of such behaviour of components of rotation vector could be magnitude equation in proper motions, as well as stellar proper motions. Even if the magnitude equation exists in the Gaia DR2 data, its effect, in the case of a relatively uniform

Fig. 4 Dependence of components of rotation vector ω_x , ω_y , ω_z of the system, defined by positions and proper motions of 1.0M NEWS sources, passed astrometric criterion, concerning the Gaia-CRF2 on the G magnitude



distribution of sources across the wide field on the sky, should be rather random than systematic. Thus we assume, stars in NEWS catalogue cause this magnitude dependence (and nonzero components of rotation vector).

To exclude the stars from NEWS×Gaia DR2 sample, we used the following criterion [42]: $(\frac{\mu_{\alpha*}}{\zeta_{\mu_{\alpha*}}})^2 + (\frac{\mu_{\delta}}{\zeta_{\mu_{\delta}}})^2 < 25$, which allows to pick up the ‘motionless’ sources only. Choosing the sources from the 1.2 million sample, proper motions of which passed this criterion, and the probability of being extragalactic objects for which was $p_{xgal} > 0.95$, we received the sample of 1 019 687 sources. Repeating the procedure of estimating rotation vector, we found that magnitude dependence of the rotation vector acquired a weaker trend, resulting in zero components of rotation vector almost at full magnitude range (see Fig. 4). Vertical error bars represent the dispersion of components within the magnitude bin. The components ω_x , ω_z were shifted by $\pm 1.0 \text{ mas yr}^{-1}$ for better visibility. Important to note, that such a probability limit was chosen as a compromise between the purity of the resulting sample (as far as components of rotation vector are concerned), and its completeness (i.e., the total number of the sources in the resulting sample).

We can transform this result into the numerical estimation of the NEWS catalogue purity, supposing most of the extended galaxies from NEWS were not cross-matched in the Gaia DR2. This assumption is based on the observational strategy of the Gaia mission [52]. Assuming the number of 20.05 million sources, as a total number of galaxies and stars in NEWS up to $g=21.6^m$ (this magnitude limit corresponds to the one, at which NEWS×Gaia DR2 cross-matching result contains 95% of sources) and also assuming $\sim 220\,000$ sources, which have significant proper motions are stars, we can estimate the purity of NEWS as $P_G = (20.05 - 0.22)/20.05 \approx 98.9\%$. We conclude that applying astrometric validation shows the estimated purity of the NEWS catalogue, consistent with the estimations via external spectroscopic datasets.

6 Conclusions

In this work, we presented the approach for the identification of the extragalactic objects. Our approach incorporates the feature engineering with autoencoder artificial neural network, outlier detection with OCSVM, and binary classification with SVM. As a result of our approach application, we have created the NEWS catalogue, comprising 40 350 492 candidates in extragalactic objects (quasars and galaxies), which passed the classification, based on the proposed classification model. NEWS catalogue covers 3/4 of the sky up to $g=23^m$ and is one of the most in-depth catalogues of extragalactic objects, covering most of the sky.

We identified with a high-quality extragalactic objects within AllWISE \times PS1 sample, as confirmed by several checks. Namely, with using independent validation data—catalogues of spectroscopically confirmed sources and astrometric catalogue Gaia DR2—we estimated purity ($MCC > 0.95$ at $19^m < g < 21.5^m$ with spectroscopic data, $P_G \sim 99\%$ at $17^m < g < 21.6^m$ with Gaia DR2, and $P_G \sim 99\%$ at $15^m < g < 23^m$ with five-fold cross-validation) and completeness ($C_G > 99\%$ at $18^m < g < 21.5^m$) of the NEWS sample.

The proposed method for identifying extragalactic objects is flexible and fully-automatic approach that could become an alternative to traditional colour cuts with manually-selected colours for classifying sources into two classes: stars and extragalactic objects. Using this model, we created the NEWS catalogue, which can be used for cosmological, astrophysical [5], astrometric [6, 27, 53], photometric studies [40, 49] and for any astronomical data sets processing [33].

Soon we plan to analyze the methods of classifying the sources with a few missed magnitudes. Successful solving this task could help in increasing the number of identified extragalactic sources, because of some of them were excluded at the stage of data preparation due to leaving on the sources with all available magnitudes only.

Acknowledgements This publication makes use of data products from the Wide-field Infrared Survey Explorer, which is a joint project of the University of California, Los Angeles, and the Jet Propulsion Laboratory/California Institute of Technology, and NEOWISE, which is a project of the Jet Propulsion Laboratory/California Institute of Technology. WISE and NEOWISE are funded by the National Aeronautics and Space Administration.

The Pan-STARRS1 Surveys (PS1) and the PS1 public science archive have been made possible through contributions by the Institute for Astronomy, the University of Hawaii, the Pan-STARRS Project Office, the Max-Planck Society and its participating institutes, the Max Planck Institute for Astronomy, Heidelberg and the Max Planck Institute for Extraterrestrial Physics, Garching, The Johns Hopkins University, Durham University, the University of Edinburgh, the Queen’s University Belfast, the Harvard-Smithsonian Center for Astrophysics, the Las Cumbres Observatory Global Telescope Network Incorporated, the National Central University of Taiwan, the Space Telescope Science Institute, the National Aeronautics and Space Administration under Grant No. NNX08AR22G issued through the Planetary Science Division of the NASA Science Mission Directorate, the National Science Foundation Grant No. AST-1238877, the University of Maryland, Eotvos Lorand University (ELTE), the Los Alamos National Laboratory, and the Gordon and Betty Moore Foundation.

Funding for the Sloan Digital Sky Survey IV has been provided by the Alfred P. Sloan Foundation, the U.S. Department of Energy Office of Science, and the Participating Institutions. SDSS-IV

acknowledges support and resources from the Center for High-Performance Computing at the University of Utah. The SDSS web site is www.sdss.org.

This work has made use of data from the European Space Agency (ESA) mission Gaia (<https://www.cosmos.esa.int/gaia>), processed by the Gaia Data Processing and Analysis Consortium (DPAC, <https://www.cosmos.esa.int/web/gaia/dpac/consortium>). Funding for the DPAC has been provided by national institutions, in particular the institutions participating in the Gaia Multilateral Agreement.

The work of V. Akhmetov was supported under the special program of the NRF of Ukraine “Leading and Young Scientists Research Support” – “Astrophysical Relativistic Galactic Objects (ARGO): life cycle of active nucleus”, No. 2020.02/0346.

References

1. Abraham, S., Philip, N., Kembhavi, A., Wadadekar, Y.G., Sinha, R.: MNRAS **419**, 80 (2012)
2. Abolfathi, B., Aguado, D., Aguilar, G., et al.: ApJS **235**, 42 (2018)
3. Akhmetov, V., Fedorov, P., Velichko, A., Shulga, V.: MNRAS **469**, 763 (2017)
4. Akhmetov, V., Khlamov, S., Dmytrenko, A., et al.: AISC **871**, 3 (2019)
5. Akhmetov, V., Khlamov, S., Tabakova, I., et al.: IEEE Int. Symp. Ind. Electron. **2019**, 4 (2019)
6. Akhmetov, V., Khlamov, S., Khramtsov, V., et al.: AISC **1080**, 896 (2020)
7. Assef, R.J., Stern, D., Kochanek, C.S., et al.: ApJ **772**, 26 (2013)
8. Baldry, I.K., Liske, J., Brown, M.J.I., et al.: MNRAS **474**, 3875 (2018)
9. Baldry, I.K., Glazebrook, K., Brinkmann, J., et al.: ApJ **600**, 681 (2004)
10. Bilicki, M., Peacock, J., Jarrett, T., et al.: ApJ **225**, 1 (2016)
11. Blanton, M., Bershad, M., Abolfathi, B., et al.: AJ **154**, 28 (2017)
12. Blake, C., Bridle, S.: MNRAS **363**, 1329 (2005)
13. Bradley, A.: Pattern Recognit. **30**(7), 1145–1159 (1997)
14. Brescia, M., Cavuoti, S., Longo, G.: MNRAS **450**, 3893 (2015)
15. Chambers, K., Magnier, E., Metcalfe, N., et al.: (2016). [arXiv:1612.05560](https://arxiv.org/abs/1612.05560)
16. Cui, X.Q., Zhao, Y.H., Chu, Y.Q., et al.: RA&A **12**, 1197 (2012)
17. Cutri, R., Wright, E., Conrow, T., et al.: (2013). 2013yCat.2328....0C
18. Davies, L.J.M., Driver, S.P., Robotham, A.S.G., et al.: MNRAS **447**, 1014 (2015)
19. Dawson, K., Kneib, J.-P., Percival, W., et al.: AJ **151**, 44 (2016)
20. Donley, J.L., Koekemoer, A.M., Brusa, M., et al.: ApJ **748**, 142 (2012)
21. Driver, S.P., Hill, D.T., Kelvin, L.S., et al.: MNRAS **413**, 971 (2011)
22. Dállya, G., Galgóczi, G., Dobos, L., et al.: MNRAS **479**, 2374 (2018)
23. Elvis, M., Wilkes, B.J., McDowell, J.C. et al.: ApJS **95**, 1 (1994)
24. Farrow D.J., Daniel J.m Cole, S. et al.: MNRAS **437**, 748 (2014)
25. Fedorov, P.N., Akhmetov, V.S., Bobylev, V.V., Gontcharov, G.A.: MNRAS **415**, 665 (2011)
26. Fedorov, P.N., Akhmetov, V.S., Shulga, V.M.: MNRAS **440**, 624 (2014)
27. Fedorov, P.N., Akhmetov, V.S., Velichko, A.B.: MNRAS **476**, 2743 (2018)
28. Gaia Collaboration, (Brown, A., et al.): A&A **616**, A1 (2018)
29. Gaia Collaboration, (Mignard, F., et al.): A&A **616**, A14 (2018)
30. Gaia Collaboration, (Prusti, T., et al.): A&A **595**, A1 (2016)
31. Hasinger, G., Capak, P., Salvato, M., et al.: ApJ **858**, 77 (2018)
32. Jarrett, T.H., Cohen, M., Masci, F., et al.: ApJ **735**, 112 (2011)
33. Khlamov, S., Savanevych, V., Briukhovetskyi, O. et al.: Proceedings of IEEE 2nd International Conference on Data Stream Mining and Processing (DSMP), p. 227 (2018)
34. Khramtsov, V., Akhmetov, V.: Proc. IEEE XIIIth Int. Sci. Tech. Conf. (CSIT) **2018**, 72 (2018)
35. Khramtsov, V., Sergeev, A., Spiniello, C., et al.: A&A **632**, A56 (2019)
36. Kingma D.P., Ba J.: (2014). [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
37. Kovács, A., Szapudi, I.: MNRAS **448**, 1305 (2015)

38. Krakowski, T., Małek, k., Bilicki, M. et al.: *A&A* **596**, A39 (2016)
39. Kron, R.G.: *ApJS* **43**, 305 (1980)
40. Kudzej, I., Savanevych, V., Briukhovetskyi, O., et al.: *Astronomische Nachrichten* **340**(1–3), 68 (2019)
41. Lacy, M., Storrie-Lombardi, L.J., Sajina, A., et al.: *ApJS* **154**, 166 (2004)
42. Lindegren, L., Hernandez, J., Bombrun, A.: *A&A* **616**, A2 (2018)
43. Majewski, S., APOGEE Team, & APOGEE-2 Team: *Astronomische Nachrichten* **337**, 863 (2016)
44. Mateos, S., Alonso-Herrero, A., Carrera, F.J., et al.: *MNRAS* **426**, 3271 (2012)
45. Matsuhara, H., Shibai, H., Onaka, T., Usui, F.: *AdSpR* **36**, 1091 (2005)
46. Matthews, B.: *Biochimica et Biophysica Acta (BBA) - Protein Structure* **405**(2), 442–451 (1975)
47. Małek, K., Solarz, A., Pollo, A., et al.: *A&A* **557**, A16 (2013)
48. Nakoneczny, S., Bilicki, M., Solarz, A., et al.: *A&A* **624**, A13 (2019)
49. Parimucha, S., Savanevych, V., Briukhovetskyi, O., et al.: *Contributions of the Astronomical Observatory Skalnaté Pleso* **49**, 151 (2019)
50. Platt, J.C.: In: Smola, A.J., Bartlett, P., Schölkopf & D.Schuurmans, B. (eds.) *Advances in Large Margin Classifiers*, vol. 61. MIT Press, Cambridge (1999)
51. Rumelhart, D. E., Hinton, G. E., and Williams, R. J., 1986, in *Parallel distributed processing: explorations in the microstructure of cognition*, ed. D. E. Rumelhart, J. L. McClelland, & CORPORATE PDP Research Group(Cambridge, USA: MIT Press), 1, 318
52. Robin, A., Luri, X., Reylyé, C., et al.: *A&A* **543**, A100 (2012)
53. Savanevych, V., Khlamov, S., Vavilova, I., et al.: *A&A* **609**, A54 (2018)
54. Saxe, A., McClelland, J., Ganguli, S.: (2013) [arXiv:1312.6120](https://arxiv.org/abs/1312.6120)
55. Salvato, M., Ilbert, O., Hoyle, B.: *NatAs* **3**, 212 (2019)
56. Scodreggio, M., Guzzo, L., Garilli, B., et al.: *A&A* **609**, A84 (2018)
57. Schölkopf, B., Smola, A., & Müller, K., 1999, in *Advances in Kernel Methods*, ed. B. Schölkopf, C. J. C. Burges, & A. J. Smola (Cambridge, USA: MIT Press), 327
58. Secrest, N., Dudik, R., Dorland, B., et al.: *ApJ* **2015**(221), 1 (2015)
59. Skrutskie, M., Cutri, R., Stiening, R., et al.: *AJ* **131**, 1163 (2006)
60. Solarz, A., Pollo, A., Takeuch, T., et al.: *A&A* **541**, A50 (2012)
61. Spiniello, C., Agnello, A., Napolitano, N.R., et al.: *MNRAS* **480**, 1163 (2018)
62. Stern, D., Assef, R.J., Benford, D.J., et al.: *ApJ* **753**, 30 (2012)
63. Stern, D., Eisenhardt, P., Gorjian, V., et al.: *ApJ* **631**, 163 (2005)
64. Vapnik, V.: *The nature of statistical learning theory*. Springer-Verlag, New York, Inc., New York, USA (1995)
65. Vasconcellos, E.C., de Carvalho, R.R., Gal, R.R., et al.: *AJ* **141**, 189 (2011)
66. Vickers, J., Röser, S., Grebel, E.: *AJ* **151**, 99 (2016)
67. Wright, E.L., Eisenhardt, P.R.M., Mainzer, A.K., et al.: *AJ* **140**, 1868 (2010)

Ensemble Classifiers for Pulsar Detection



Jakub Holewik and Gerald Schaefer

Abstract A common way of identifying pulsars is based on their emitted radio waves. To allow for automated screening of a large number of radio signals, machine learning methods can be adopted. One challenge here though is that training such methods is hampered by the inherent imbalance in the available data since signals related to actual pulsars are relatively rare. In this chapter, we show that ensemble classification methods that specifically address class imbalance can be successfully employed for pulsar detection. Ensemble classifiers combine several base classifiers to provide more robust and typically improved classification performance, while class imbalance can be addressed through careful sampling or through cost-sensitive classification. Our results demonstrate that such dedicated ensembles yield better results compared to methods that do not consider class balance.

1 Introduction

Pattern classification is an important area of machine learning and a variety of techniques for designing classifiers have been proposed in the literature while applications in industry include, among many others, speech recognition, fingerprint identification and data analysis [10]. A classifier is trained on a set of training samples and tries to learn the correspondence between features in the data and the given classes. The goal is to train a classifier that, when presented with a previously unseen sample, is able to assign it to its correct class.

In astrophysics there are various problems where a classifier can be applied including classifying objects such as photometric variable stars [31], supernovas [26] or globular clusters [5], as well as for catalogue matching [34].

Our interest in this chapter is the automatic identification of pulsars. Pulsars are rare neutron stars detectable through the radio waves they emit [33]. To allow for automated methods of screening a large number of radio signals, machine learning methods can be adopted. However, the recorded signals contain a considerable

J. Holewik · G. Schaefer (✉)

Department of Computer Science, Loughborough University, Loughborough, UK

amount of noise as well as radio frequency interference. In fact, interference is so common that the large majority of candidate signals detected turn out to not stem from pulsars. This makes the use of common machine learning algorithms challenging since they are not designed to take into account such a class imbalance.

Class imbalance is a common issue in classification tasks. When collecting data for training, in particular for binary classification (i.e., tasks where patterns are separated into exactly two classes as in “pulsar” or “not pulsar”), the ideal scenario is that the split of patterns between the classes will be approximately equal. Unfortunately, when collecting data for pulsar candidates, only a handful of observations correspond to true pulsars. Consequently, conventional classifiers struggle to learn effectively from such an imbalanced dataset and in particular to learn well from the minority class, which for our problem here (real pulsars) is the one of interest.

In this chapter, an extension of our earlier work in [18], we show that ensemble classification methods, i.e. methods that combine several base classifiers, that specifically address class imbalance can be successfully employed for pulsar detection. Our results demonstrate that such dedicated ensembles yield better results compared to methods that do not consider class balance, and suggest their use for other applications in the field of astrophysics.

The remainder of the chapter is organised as follows. Section 2 covers some of the background on pulsar detection, while Sect. 3 looks at the problem of class imbalance and some common approaches to overcome it. Ensemble classification is described in Sect. 4, and the ensemble classifiers that we employ to address class imbalance in Sect. 5. Experimental results are presented in Sect. 6, while Sect. 7 concludes the chapter.

2 Background

Searching for pulsars is conducted through collecting pulsar candidates, that is, sets of statistical information about certain radio emissions captured from space [28]. Search techniques used to isolate these look for periodic broadband signals that appear dispersed. The collected signals are then converted into numerical values which can be analysed to determine which of them are actual pulsars. Signals that are determined to be likely coming from pulsars can then be passed on for further observation. Traditionally, this analysis was conducted manually by human experts. Unfortunately, the majority of captured signals does not come from pulsars, leading to a lot of time dedicated to discarding noisy candidates. Also, technological advancements have significantly increased the number of candidates being discovered [37], leading to the manual approach becoming infeasible.

Consequently, various automated approaches have been developed for pulsar classification, with [19] describing a program for candidate selection as early as 1992. However these methods were not intelligent enough, since after initial filtering of candidates, the selected samples still needed to be manually checked. The use of advanced machine learning approaches leads to more reliable detection. Examples

include an artificial neural network for pulsar classification [11], PICS, a method which utilises image pattern classification approaches to recognise pulsars from diagnostic plots [44], and SPINN, a high-performance solution that is also based on neural networks [29].

While such machine learning approaches have shown potential to greatly reduce the work needed for identifying pulsars, none of them address the fundamental problem of data imbalance present in the candidate selection task with true pulsars being greatly outnumbered by noisy samples. It is this aspect that we specifically address in this chapter.

3 Imbalanced Classification

Many real-life datasets are imbalanced so that patterns of interest (commonly referred to as the “positive class” or “minority class”) are outnumbered by “other” patterns (referred to as the “negative class” or “majority class”). In our pulsar detection task, there are many more noise samples compared to those that represent a true pulsar. This is challenging since classification algorithms typically try to maximise accuracy over all samples and thus tend to be biased towards the majority class, leading to often poor recognition of minority class samples.

3.1 *A Closer Look at Rarity*

While we can draw a distinction between “rare classes” and “rare cases” in datasets [43], our issue of imbalance typically refers to rare classes where the positive class as a whole represents a smaller portion of the dataset (rare cases are singular patterns that differ from most other patterns in the same class).

We can also distinguish between absolute and relative rarity [43]. Relative rarity means that the positive class contributes a much lower percentage to the dataset, whereas absolute rarity suggests that the overall number of patterns in a class is just generally low. Absolute rarity is often a reflection of lack of data, while, not surprisingly, classes lacking a sufficient number of samples tend to suffer from higher misclassification rates [42]. Relative rarity can still cause problems even in big datasets, for example when using greedy search heuristics [43].

3.2 *Addressing Imbalance—Sampling*

The most common approach to address class imbalance is through sampling. The aim here is trying to “fix” the data by creating a new training dataset with equal class distribution. In undersampling approaches, this is achieved by removing samples of

the majority class, at the cost of discarding potentially useful data [43]. On the other hand, in oversampling, the number of patterns of the minority class is increased to match more closely that of the majority class. The key obstacle here is how to obtain useful new minority class samples.

SMOTE (for Synthetic Minority Over-sampling TEchnique) [6] generates new, artificial patterns of the minority class that are supposed to be similar to the actual patterns in the dataset. For this, it uses a nearest neighbour approach, creating new patterns by combining features from existing neighbouring patterns. SMOTE has been widely used and is known to help with generalisation in imbalanced classifiers [6, 43].

3.3 Addressing Imbalance—Cost-Sensitive Classification

An alternative approach to address class imbalance is cost-sensitive classification which is based on the idea of assigning a penalty to misclassifications [30]. Conventional classifiers try to reduce the number of misclassifications but do not pay attention to which class these belong to. Defining class weights is a common approach to reflect the varying degrees of importance among classes. In most imbalanced classification problems, the minority class is the class of interest and is thus assigned a higher weight so that the resulting classifier will focus more on reducing the error rate of that class.

4 Ensemble Classification

Typically, a single classifier is used in pattern recognition problems. However, classifiers are rarely perfect and designing a classifier that will generalise well is a difficult problem. On the other hand, different classifiers can complement each other when it comes to achieving high performance [15]. This observation leads to the development of multiple classifier systems, also known as ensemble classification. By using a set of base classifiers and then combining their outputs, these methods can deal particularly well with noisy inputs and yield more robust classification [16].

4.1 Rationale Behind Ensembles

Before exploring ensemble classification in depth, it is useful to ask whether it is worth the effort in the first place [14]. In general there are three reasons for why ensemble methods are worth using [9]:

- *Statistical argument*: This is relevant in problems that suffer from data sparsity. With a limited number of samples, training could produce many different classifiers with similar performance. But some of them will be better than others at generalisation. Combining these classifiers into an ensemble is better than picking one and risking that it will not perform well on unseen data.
- *Computational argument*: This argument applies to methods that use some sort of hill-climbing or random search, for example neural networks with gradient descent or decision trees with greedy splitting rules. These techniques are inherently difficult to optimise and a common issue is getting stuck in a local optimum. This is where an ensemble can be beneficial by employing multiple classifiers which begin searching in different places, thus improving the likelihood of finding the global optimum.
- *Representational argument*: It may be impossible to obtain an optimal classifier. For example, for a binary dataset with a non-linear decision boundary, there is no linear classifier that can yield perfect classification. In a situation like this, there are two possible solutions. One is to train a classifier of higher complexity, while the other solution is to combine some imperfect classifiers with the aim of increasing the overall performance.

4.2 Voting

The simplest approach to create an ensemble is to train a set of classifiers independently, and then have them vote for the classification result. The decision about how to classify a pattern is made by every member of the ensemble performing classification, and then conducting a vote to determine the final class. Different voting schemes such as selecting the class that receives most votes or weighted voting can be employed.

While voting yields a straightforward method to combine different classifiers, for better performance more sophisticated approaches are needed, the most common of which are bagging and boosting.

4.3 Bagging

In bagging, short for “bootstrap aggregating”, each classifier is trained not on the full dataset, but the dataset is randomly sampled multiple times (with replacement) and each classifier trained on a different random subset [1]. The classifiers are then combined, usually by plurality vote (i.e., the sample assigned to the class that receives most votes).

Unstable classifiers are ones where even a small change in the training set can drastically affect the end result [2], and bagging was intended for unstable classifiers such as neural networks and decision trees. If used with stable classifiers, for example

nearest neighbour classification, there is a risk that the ensemble members trained on slightly different data subsets will end up very similar to each other so that the ensemble will not be much different from a single classifier.

The best-known adaptation of bagging is the Random Forest algorithm [3]. Here, decision trees are used as base classifiers for the ensemble, and training sets are obtained from independent and identically distributed random vectors which can be random samples of the dataset, samples with random features or even randomly varying parameters of the tree itself. A common strategy is to sample both the data space and the feature space, i.e., grow the trees from a random subset of the dataset, with patterns containing only a random subset of all features [23].

4.4 *Boosting*

Boosting, whose origin stems from the weighted majority algorithm proposed in [24], has the idea of “weak learners”—classifiers that are only slightly better than a random guess—at its core with the aim of transforming, or “boosting”, them into strong learners [35]. In boosting algorithms, this is accomplished by iteratively training ensemble members, with each of them focussing on specific data patterns that were difficult to learn for the previous classifier.

In AdaBoost [13], short for “Adaptive Boosting”, a weight is assigned to each pattern. These weights can be used in two different ways. One approach is to treat the weight as a probability that the given pattern will be selected in the random training subset generated for the next classifier. Another approach is to feed all classifiers the entire dataset and use the weights directly as weights for training, which requires classifiers that support weighted learning and also makes the algorithm deterministic as no sampling occurs [23]. The key feature of AdaBoost is that the weights are updated after a classifier has been trained, with weights increasing if the classifier predicts the sample incorrectly, and decreasing otherwise.

5 Ensembles for Imbalanced Classification

In this chapter, we investigate the use of ensemble classifiers that specifically address class imbalance for pulsar detection. In the following we briefly describe the algorithms we evaluate.

5.1 *SMOTEBagging*

SMOTEBagging combines a bagging ensemble with various sampling strategies [41]. The main idea is to employ a bagging scheme that trains base classifiers on subsets

of the training data so that each class is equally represented. SMOTE is applied to the minority class, and original patterns and generated patterns are used together with a random subsample of the majority class when training each classifier. Experimental results in [41] show that SMOTE effectively improves the diversity and performance of a bagging ensemble.

5.2 *SMOTEBoost*

SMOTEBoost [7] uses SMOTE in combination with an AdaBoost classifier with SMOTE employed to improve performance on the minority class, and boosting used to make up for the loss of general accuracy. However, SMOTEBoost is more sophisticated than just running SMOTE on a dataset and then training an ensemble on it. SMOTE is instead used separately for each classifier, and all synthetically generated patterns are discarded before training the next classifier. Unlike standard AdaBoost, which treats all misclassifications equally, misclassified minority class patterns are focussed on. Particularly hard to learn minority patterns will have “similar” synthetic patterns with similar weights added to the training set, thus enabling classifiers to better learn them, while implicitly creating more diversity in the ensemble (since every classifier is trained on a number of exclusive patterns that will be discarded afterwards).

5.3 *EasyEnsemble*

Liu et al. [25] proposes an effective ensemble which focusses on undersampling rather than oversampling. EasyEnsemble can be seen as a fusion between bagging and boosting but is somewhat unique in that it technically generates an ensemble of ensembles. During each iteration, it uses random undersampling with replacement to generate a subset of the majority class training data. This subset is then used together with the full minority class data as a training set for an AdaBoost ensemble. This way, a set of diverse AdaBoost ensembles is generated, each trained on different majority class data. Finally, the outputs of all classifiers predicting the same class are summed, and the class with higher support is chosen.

5.4 *Balanced Random Forest*

In a similar fashion to EasyEnsemble, the Balanced Random Forest algorithm [8] adapts Random Forest classification for imbalanced classification problems. In standard Random Forest, for an imbalanced dataset, there is a high risk of selecting very few minority class patterns in the bootstrap samples selected for learning. Possible

solutions to this issue include sampling stratified subsets (i.e., subsets that preserve the distribution of classes), or utilising a sampling technique like oversampling or undersampling. Balanced Random Forest uses undersampling on the majority class, and so each tree is based on a subset that is balanced between the classes.

5.5 *AdaC2*

With a rationale similar to the one behind SMOTEBoost, AdaC2 [38] uses AdaBoost with a cost-sensitive approach to address data imbalance. The aim is to adjust the weights so that misclassified minority class patterns are the main focus, and the algorithm uses a cost value for each pattern which represents the penalty to the classifier for misclassifying that pattern with minority class costs higher than majority class costs. These costs are incorporated to define the weight w_i^{k+1} of pattern i in the next classifier as

$$w_i^{k+1} = \frac{w_i^k e^{\beta_i C_i}}{Z}, \quad (1)$$

where w_i^k is the pattern's weight in the current classifier, $\beta_i = \alpha_k$ if pattern i was misclassified by classifier k , and $\beta_i = -\alpha_k$ otherwise, with α_k a parameter which is a predefined function of the classifier's error rate, and Z is a normalisation factor.

5.6 *AdaCost*

AdaC2 and related algorithms are actually simplified variants of another cost-sensitive method, namely AdaCost [12]. Here, the AdaBoost cost function is

$$w_i^{k+1} = \frac{w_i^k e^{\beta_i D_i}}{Z}, \quad (2)$$

where, instead of introducing a constant cost for each pattern, a cost adjustment function D_i is used, which is designed to have higher values when the pattern was misclassified. An interesting aspect of this algorithm is that, unlike AdaC2, it does not reduce to AdaBoost when both the majority and minority class are given the same weight [38].

5.7 *USBE*

USBE (for UnderSampling Balanced Ensemble) uses balanced data subspaces and a classifier fusion technique to construct an ensemble [21]. The dataset is split into

subsets with balanced class distributions based on undersampling the majority class, and a base classifier trained on each balanced subsets. Additionally, feature selection is performed, separately for each classifier, which in turn further increases the diversity of the ensemble. The classifiers are then fused by deriving weights for the individual classifiers based either on a genetic algorithm or a neural network.

6 Experimental Results

6.1 Dataset

In this chapter, we perform pulsar classification based on the HTRU2 study [27], which is a large database of pulsar candidates collected in the dedicated High Time Resolution Universe Survey [20].

The features are extracted from the pulse profile which describes the longitude-resolved version of a the signal, averaged in frequency and time. The DM-SNR curve represents the correlation between the dispersion measure (DM; the integrated density of free electron columns between the pulsar and the point of observation) and the signal-to-noise ration (SNR) from the given pattern.

Specifically, we use eight attributes that represent the various features of each pulsar candidate [29], namely:

- mean of the integrated pulse profile;
- standard deviation of the integrated pulse profile;
- excess kurtosis of the integrated pulse profile;
- skewness of the integrated pulse profile;
- mean of the DM-SNR curve;
- standard deviation of the DM-SNR curve;
- excess kurtosis of the DM-SNR curve;
- skewness of the DM-SNR curve.

The dataset comprises 16,259 bogus patterns (caused by radio frequency interference and noise) and 1,639 real pulsar patterns which have been manually verified [27], thus exhibiting significant class imbalance.

6.2 Performance Measures

In binary classification problem, there are four basic measures which form the basis of various performance metrics [36], namely:

- *True positives (TP)*: the number of patterns from the positive class that are correctly classified;

- *True negatives (TN)*: the number of patterns from the negative class that are correctly classified;
- *False positives (FP)*: the number of patterns from the negative class that are incorrectly classified as the positive class;
- *False negatives (FN)*: the number of patterns from the positive class that are incorrectly classified as the negative class.

For conventional classification, the basic performance measures are [36]:

- *Accuracy*: the overall percentage of correctly classified patterns, i.e.

$$Acc = \frac{TP + TN}{TP + FN + FP + TN}; \quad (3)$$

- *Precision*: the percentage of patterns classified as positive that are correctly classified, i.e.

$$Prec = \frac{TP}{TP + FP}; \quad (4)$$

- *Recall (or Sensitivity)*: the percentage of positive patterns that are classified correctly, i.e.

$$Rec = Sen = \frac{TP}{TP + FN}; \quad (5)$$

- *Specificity*: the percentage of negative patterns that are classified correctly, i.e.

$$Spec = \frac{TN}{TN + FP}. \quad (6)$$

However, for imbalanced classification problems, these standard measures are not sufficient [39]. In particular, using accuracy to describe the overall efficacy of a classifier can be misleading since it will largely describe performance on the majority class. On the other hand, precision and recall express how well the classifier is doing on the positive (minority) class [43]. The F-score integrates these two measures into a single one as [32]

$$F = \frac{2 \times Prec \times Rec}{Prec + Rec}. \quad (7)$$

Another useful measure is the G-Mean [22] which is defined as the geometric mean between sensitivity and specificity, i.e.

$$G = \sqrt{Sen \times Spec}, \quad (8)$$

and thus corresponds to a point on the receiver operator characteristic (ROC) curve.

6.3 Results

In our experiments we evaluate the classifiers detailed in Sect. 5 and do so based on two types of base classifiers, decision trees based on CART [4], and support vector machines (SVMs) [40] with polynomial kernels and parameters optimised using a grid search. For comparison, we also tested 3 classifier ensembles that are not dedicated to imbalanced problems, namely an ensemble based on the random subspace method [17], with 80% of features in each subspace and majority voting, standard Random Forest, and AdaBoost.

We perform 20-fold cross validation, where the dataset is divided into 20 partitions, and 19 are used for training while the remaining partition is employed as the test set. The process is repeated 20 times so that each partition is once used for testing and the results averaged. All ensembles are based on 5 base classifiers.

The obtained results are given in Table 1 for tree classifiers and in Table 2 for SVMs.

From Tables 1 and 2, we can see that already the standard ensembles give fairly good performance, except for AdaBoost with SVMs as base classifiers. This confirms that the extracted features provide a good basis for successful pulsar identification.

However, ensembles that are dedicated to imbalanced classification problems do indeed yield better results, in particular in terms of both a high G-mean and good F-score results combined with high sensitivity, thus confirming the usefulness of the presented approaches. While the Balanced Random Forest only slightly outperforms standard Random Forest classification, bigger gains are achieved by AdaC2 and AdaCost compared to AdaBoost, in particular when using SVMs as individual classifiers. Still better performance is achieved by SMOTEBagging, EasyEnsemble and USBE. Overall, USBE gives the best results in terms of G-mean, which is the most appropriate single measure for imbalanced problems, with the neural fuser slightly

Table 1 Results using decision trees as base classifiers

	Accuracy	Precision	Recall	Specificity	F-score	G-mean
Random subspace ensemble	97.85	93.41	81.21	99.45	86.88	89.87
Random Forest	98.07	93.60	84.81	99.42	88.92	91.79
AdaBoost	97.96	94.80	81.21	99.57	87.48	89.92
SMOTEBagging	96.66	76.64	89.11	97.39	82.40	93.16
SMOTEBoost	96.63	76.48	88.93	97.37	82.24	93.06
Balanced Random Forest	96.15	73.84	86.96	97.04	79.86	91.86
EasyEnsemble	95.89	71.10	89.60	96.50	79.29	92.99
AdaC2	97.82	90.41	84.08	99.14	87.13	91.30
AdaCost	97.82	90.41	84.08	99.14	87.13	91.30
USBE (genetic fuser)	95.92	71.87	91.58	96.36	80.48	93.93
USBE (neural fuser)	95.98	72.25	91.70	96.41	80.75	94.01

Table 2 Results using SVMs as base classifiers

	Accuracy	Precision	Recall	Specificity	F-score	G-mean
Random subspace ensemble	97.93	93.48	82.17	99.45	87.46	90.40
AdaBoost	96.82	94.64	67.52	99.63	78.81	82.02
SMOTEBagging	97.33	81.84	89.44	98.09	85.48	93.67
SMOTEBoost	96.97	81.29	84.95	98.12	83.08	91.30
EasyEnsemble	97.22	80.76	89.68	97.95	84.99	93.72
AdaC2	98.04	94.20	82.80	99.51	88.14	90.77
AdaCost	97.96	92.58	83.44	99.36	87.77	91.05
USBE (genetic fuser)	97.30	83.01	88.96	98.14	85.81	93.42
USBE (neural fuser)	96.03	72.65	91.64	96.48	80.95	94.03

outperforming the one based on a genetic algorithm. In addition, we can notice that the use of SVMs as base classifiers is generally superior to decision trees, leading to the highest G-mean of 94.03 by USBE with a neural fuser and SVMs as individual classifiers, representing an excellent performance on this challenging problem.

7 Conclusions

In this chapter, we have investigated the use of ensemble classifiers that address class imbalance for finding true pulsars among the candidates in the HTRU2 study, and have shown that they indeed provide a useful approach for this challenging task and improved performance compared to standard ensembles. Since the investigated methods are essentially agnostic with respect to the application, we expect that they can also be successfully employed for other astrophysical applications such classification of photometric variable stars [31], supernovas [26] or globular clusters [5].

References

1. Breiman, L.: Bagging predictors. *Mach. Learn.* **24**, 123–140 (1996)
2. Breiman, L.: Heuristics of instability and stabilization in model selection. *Ann. Stat.* **24**, 2350–2383 (1996)
3. Breiman, L.: Random forests. *Mach. Learn.* **45**, 5–32 (2001)
4. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: *Classification and Regression Trees*. Chapman and Hall, London (1984)
5. Cavuoti, S., Garofalo, M., Brescia, M., Paolillo, M., Pescape, A., Longo, G., Ventre, G.: Astrophysical data mining with GPU. A case study: genetic classification of globular clusters. *New Astron.* **26**, 12–22 (2014)
6. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002)

7. Chawla, N., Lazarevic, A., Hall, L., Bowyer, K.: SMOTEBoost: improving prediction of the minority class in boosting. In: 7th European Conference on Principles and Practice of Knowledge Discovery in Database, pp. 107–119 (2003)
8. Chen, C., Liaw, A., Breiman, L.: Using random forest to learn imbalanced data. Technical report, UC Berkeley (2004)
9. Dietterich, T.: Ensemble methods in machine learning. In: Multiple Classifier Systems, pp. 1–15. Springer, Berlin (2000)
10. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. Wiley, Hoboken (2001)
11. Eatough, R., Molkenthin, N., Kramer, M., Noutsos, A., Keith, M., Stappers, B., Lyne, A.: Selection of radio pulsar candidates using artificial neural networks. *Mon. Not. R. Astron. Soc.* **407**, 2443–2450 (2010)
12. Fan, W., Stolfo, S., Zhang, J., Chan, P.: AdaCost: misclassification cost-sensitive boosting. In: 16th International Conference on Machine Learning, vol. 99, pp. 97–105 (1999)
13. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **55**, 119–139 (1997)
14. Ho, T.: Multiple classifier combination: lessons and the next steps. In: Hybrid Methods in Pattern Recognition, pp. 171–198 (2002)
15. Ho, T., Hull, J., Srihari, S.: Combination of structural classifiers. In: IAPR Workshop on Syntactic and Structural Pattern Recognition, pp. 123–136 (1990)
16. Ho, T., Hull, J., Srihari, S.: Decision combination in multiple classifier systems. *IEEE Trans. Pattern Anal. Mach. Intell.* **16**, 66–75 (1994)
17. Ho, T.K.: The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**, 832–844 (1998)
18. Holewik, J., Schaefer, G., Korovin, I.: Imbalanced ensemble learning for enhanced pulsar identification. In: 11th International Conference on Swarm Intelligence, pp. 515–524 (2020)
19. Johnston, S., Lyne, A., Manchester, R., Kniffen, D., D’Amico, N., Lim, J., Ashworth, M.: A high-frequency survey of the southern galactic plane for pulsars. *Mon. Not. R. Astron. Soc.* **255**, 401–411 (1992)
20. Keith, M., Jameson, A., van Straten, W., Bailes, M., Johnston, S., Kramer, M., Possenti, A., Bates, S., Bhat, N., Burgay, M., Burke-Spolaor, S., D’Amico, N., Levin, L., McMahon, P., Milić, S., Stappers, B.: The high time resolution universe pulsar survey I. System configuration and initial discoveries. *Mon. Not. R. Astron. Soc.* **409**, 619–627 (2010)
21. Krawczyk, B., Schaefer, G.: Breast thermogram analysis using classifier ensembles and image symmetry features. *IEEE Syst. J.* **8**, 921–928 (2014)
22. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: one-sided selection. 14th International Conference on Machine Learning, pp. 179–186 (1997)
23. Kuncheva, L.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience, Hoboken (2004)
24. Littlestone, N., Warmuth, M.: The weighted majority algorithm. *Inf. Comput.* **108**, 212–261 (1994)
25. Liu, X., Wu, J., Zhou, Z.: Exploratory undersampling for class-imbalance learning. *IEEE Trans. Syst. Man Cybern. Part B* **39**, 539–550 (2009)
26. Lochner, M., McEwen, J., Peiris, H., Lahav, O., Winter, M.: Photometric supernova classification with machine learning. *Astrophys. J. Suppl. Ser.* **225**, 31 (2016)
27. Lyon, R.: HTRU2 data set. <https://archive.ics.uci.edu/ml/datasets/HTRU2/>
28. Lyon, R.J., Stappers, B., Cooper, S., Brooke, J., Knowles, J.: Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach. *Mon. Not. R. Astron. Soc.* 1104–1123 (2016)
29. Morello, V., Barr, E., Bailes, M., Flynn, C., Keane, E., van Straten, W.: Spinn: a straightforward machine learning solution to the pulsar candidate selection problem. *Mon. Not. R. Astron. Soc.* **443**, 1651–1662 (2014)
30. Nakashima, T., Yokota, Y., Ishibuchi, H., Schaefer, G., Drastich, A., Zaviscek, M.: Constructing cost-sensitive fuzzy rule-based classification systems for pattern classification problems. *J. Adv. Comput. Intell. Intell. Inform.* **11**, 546–553 (2007)

31. Richards, J., Starr, D., Brink, H., Miller, A., Bloom, J., Butler, N., James, J., Long, J., Rice, J.: Active learning to overcome sample selection bias: application to photometric variable star classification. *Astrophys. J.* **744**, 192 (2011)
32. Rijsbergen, C.J.V.: *Information Retrieval*, 2nd edn. Butterworth-Heinemann, Oxford (1979)
33. Roberts, N., Lorimer, D., Kramer, M., Ellis, R., Huchra, J., Kahn, S., Rieke, G., Stetson, P.: *Handbook of Pulsar Astronomy*. In: *Cambridge Observing Handbooks for Research Astronomers*. Cambridge University Press, Cambridge (2005)
34. Rohde, D.J., Drinkwater, M., Gallagher, M., Downs, T., Doyle, M.: Applying machine learning to catalogue matching in astrophysics. *Mon. Not. R. Astron. Soc.* **360**, 69–75 (2005)
35. Schapire, R.E.: The strength of weak learnability. *Mach. Learn.* **5**, 197–227 (1990)
36. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **45**(4), 427–437 (2009)
37. Stovall, K., Lorimer, D., Lynch, R.: Searching for millisecond pulsars: surveys, techniques and prospects. *Class. Quantum Gravity* **30**, (2013)
38. Sun, Y., Kamel, M., Wong, A., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognit.* **40**, 3358–3378 (2007)
39. Tang, Y., Zhang, Y., Chawla, N., Krasser, S.: SVMs modeling for highly imbalanced classification. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **39**, 281–288 (2009)
40. Vapnik, V.N.: *Statistical Learning Theory*. Wiley, Hoboken (1998)
41. Wang, S., Yao, X.: Diversity analysis on imbalanced data sets by using ensemble models. In: *IEEE Symposium on Computational Intelligence and Data Mining*, pp. 324–331 (2009)
42. Weiss, G.: Learning with rare cases and small disjuncts. In: *12th International Conference on Machine Learning*, pp. 558–565 (1995)
43. Weiss, G.: Mining with rarity: a unifying framework. *SIGKDD Explor.* **6**, 7–19 (2004)
44. Zhu, W., Berndsen, A., Madsen, E., Tan, M., Stairs, I., Brazier, A., Lazarus, P., Lynch, R., Scholz, P., Stovall, K., Ransom, S., Banaszak, S., Biwer, C., Cohen, S., Dartez, L., Flanigan, J., Lunsford, G., Martinez, J., Mata, A., Rohr, M., Walker, A., Allen, B., Bhat, N., Bogdanov, S., Camilo, F., Chatterjee, S., Cordes, J., Crawford, F., Deneva, J., Desvignes, G., Ferdman, R.D., Freire, P., Hessels, J., Jenet, F., Kaplan, D., Kaspi, V., Knispel, B., Lee, K., van Leeuwen, J., Lyne, A., McLaughlin, M., Siemens, X., Spitler, L., Venkataraman, A.: Searching for pulsars using image pattern recognition. *Astrophys. J.* **781**, 117 (2014)

Periodic Astrometric Signal Recovery Through Convolutional Autoencoders



Michele Delli Veneri, Louis Desdoigts, Morgan A. Schmitz, Alberto Krone-Martins, Emille E. O. Ishida, Peter Tuthill, Rafael S. de Souza, Richard Scalzo, Massimo Brescia, Giuseppe Longo, and Antonio Picariello

Abstract Astrometric detection involves precise measurements of stellar positions, and it is widely regarded as the leading concept presently ready to find Earth-mass planets in temperate orbits around nearby sun-like stars. The TOLIMAN space telescope [39] is a low-cost, agile mission concept dedicated to narrow-angle astrometric monitoring of bright binary stars. In particular the mission will be optimised to search for habitable-zone planets around α Centauri AB. If the separation between these two stars can be monitored with sufficient precision, tiny perturbations due to the gravitational tug from an unseen planet can be witnessed and, given the configuration of the optical system, the scale of the shifts in the image plane are about one-millionth of a pixel. Image registration at this level of precision has never been demonstrated (to our knowledge) in any setting within science. In this paper, we demonstrate that a Deep Convolutional Auto-Encoder is able to retrieve such a signal from simplified simulations of the TOLIMAN data and we present the full experimental pipeline to recreate our experiments from the simulations to the signal analysis. In future works,

M. D. Veneri (✉) · A. Picariello
University of Naples Federico II, DIETI, Naples, Italy
e-mail: michele.delliveneri@unina.it

A. Picariello
e-mail: antonio.picariello@unina.it

L. Desdoigts · P. Tuthill
School of Physics, The University of Sydney, Sydney, NSW 2006, Australia
e-mail: louis.desdoigts@sydney.edu.au

P. Tuthill
e-mail: peter.tuthill@sydney.edu.au

M. A. Schmitz
Department of Astrophysical Sciences, Princeton University, 4 Ivy Ln., Princeton
NJ08544, USA
e-mail: morgan.schmitz@astro.princeton.edu

A. Krone-Martins
Donald Bren School of Information and Computer Sciences, University of California,
Irvine, CA 92697, USA
e-mail: algol@uci.edu

all the more realistic sources of noise and systematic effects present in the real-world system will be injected into the simulations.

1 Introduction

Astronomy seeks to answer our deepest questions. Where did it all begin and how is it going to end? Are we alone in the Universe? Is there life beyond our biosphere—or conversely is Earth and our planetary system in some way unique? Such inquiries have given rise to the fields of astrobiology and exoplanetary research.

Despite our long term commitment to explore these questions, the development of instruments capable of detecting planets around distant stars has proven to be one of the most challenging astronomical quests [22]. The first exoplanet orbiting a Sun-like star was detected through small deviations caused in radial velocity measurements of its host [27, this work was subsequently awarded the 2019 Nobel Prize in Physics]. A little more than twenty years later, there are more than 4000 confirmed exoplanets.¹ The celestial garden is therefore a fertile ground for discovery, and the synergy between new astronomical missions and modern statistical learning techniques promises an exceptionally bright future for this rapidly expanding field. Discovery and characterisation of exoplanets is particularly suited to combinations of approaches that can push the boundaries in both the acquisition of exceptionally clean, low-noise data, as well as the ability to sift large volumes of observations in order to extract subtle signals that are often submerged under orders of magnitude by statistical and systematic noise. Every technology in this area has to face these

CENTRA/SIM, Faculdade de Ciências, Universidade de Lisboa, Ed. C8, Campo Grande, 1749-016 Lisboa, Portugal

E. E. O. Ishida

Université Clermont Auvergne, CNRS/IN2P3, LPC, 63000 Clermont-Ferrand, France
e-mail: emille.ishida@clermont.in2p3.fr

R. S. de Souza

Key Laboratory for Research in Galaxies and Cosmology, Shanghai Astronomical Observatory, Chinese Academy of Sciences, 80 Nandan Road, Shanghai 200030, China
e-mail: drsouza@shao.ac.cn

R. Scalzo

Centre for Translational Data Science, University of Sydney, Darlington, NSW 2008, Australia
e-mail: richard.scalzo@sydney.edu.au

G. Longo

University of Naples Federico II, Department of Physics E. Pancini, Naples, Italy
e-mail: giuseppe.longo@unina.it

M. Brescia

INAF - Astronomical Observatory of Capodimonte, Naples, Italy
e-mail: massimo.brescia@inaf.it

¹<http://exoplanet.eu/catalog/>.

problems because, on a cosmic scale, exoplanets are almost completely irrelevant. They contribute only infinitesimally to the mass or energy budget of galaxies. Even in our own solar system major gas-giant planets such as Neptune and Uranus evaded detection until the advent of the modern telescope; the challenge of discovery at light-year distance scales can seem forbidding.

The most successful techniques to reveal exoplanets are *indirect* in that they do not witness signals from the planet itself, but rather the planet's influence on its host star. One is the transit method which witnesses a dip in starlight as the planet traverses the observer's line-of-sight to the star. An alternative method is the radial velocity, which records to-and-from perturbations in the velocity of the star, as it is perturbed by the gravitational field of the planet. The TOLIMAN (Telescope for Orbital Locus Interferometric Monitoring of our Astrometric Neighbourhood) program was motivated by the realisation that neither of these methods are suited to answer a fundamental question: are there any potentially habitable exoplanets around the Sun's nearest neighbour twin system— α Centauri AB? Unfortunately, the transits require an alignment, a very rare event, while radial velocity can find massive gas-giant planets, but not small rocky exo-Earths in the habitable zone of the system.

Arguably, a very promising alternative method is the most traditional branch of *Astrometry*: the study of deviations in the position of the star in the plane of the celestial sphere that, in this case, are imposed by the motion of the star and the exoplanet around a common center of mass. Like all signals in this domain of science, the deviations in position are very small, of the order of one micro-arcsecond. To give a sense of scale, for an observer on Earth, this is the angle subtended by a coin held edge-on (~ 2 mm) while standing on the moon. For the specific case we are interested in, the situation is even more interesting.

α Centauri is a binary star system (thus the A/B), with two stars constantly in motion one around each other. If their motion could be monitored, for example by taking a series of images at different times, one would see the distance between the center of the stars changing as their orbit evolves. After their equivalent of a year this pattern would repeat—thus, by observing the separation between the stars during some time one would detect a periodic signal. This expected signal would be slightly different if one considers the presence or absence of an Earth-like planet as the third element in this system—and that is the type of perturbation we aim at measuring. One can imagine that at such scales even the smallest deviations in the position of the satellite or thermal effects in its structure and instruments are enough to build up noise in each image, which is orders of magnitude higher than the signal. TOLIMAN has been designed to implement innovative optical principles to deliver a robust estimate of this signal, despite the inevitable presence of many competing random processes and systematic noise. Details can be found in [39] and in Sect. 3 of the present work. A critical component for the success of the mission is our ability to extract periodic signals at the milliarcsecond level from a data stream consisting of over a million of images downlinked from the satellite.

The general process to solve this problem has at least two major stages: first, it is necessary to estimate the period of one cycle for the binary star system; then, a

more careful analysis of the amplitude deviations at the relevant periods enables the discovery of additional clues about the presence of the planet. In this chapter, we present some first concepts of one of the possible strategies to solve the first stage directly from raw, imaging data.

Given a series of images of a binary star system as observed by the TOLIMAN mission, our framework uses an unsupervised neural network to learn an abstract (latent), low dimensional representation of the data (the raw images). Here we use a deep convolutional autoencoder [15]. This step reduces the dimensionality of the problem from 256×256 pixels (size of the images) at each sampling time to 1 parameter of the latent space, which can then be analyzed as a traditional time series. An overview of the workflow is given in Fig. 1. We use simplified simulated versions of the images to be measured by TOLIMAN to show how one can use concepts of neural networks to construct a data analysis pipeline that may be able to extract periodic astrometric signals with an amplitude up to a million times smaller than the pixel size.

In this chapter, we shall guide the reader through all the modules illustrated in Fig. 1. Section 2 gives an overview of the astrometric principles that inspired the TOLIMAN mission, presented in more details in Sect. 3. We then show how the simulations were constructed in Sect. 4, with a brief review of the principles of traditional dimensionality reduction techniques in Sect. 5. We introduce basic concepts of deep learning, and how they can be used to learn a meaningful non-linear representation of the input data, in Sect. 6. In Sect. 7 we analyze the architectural choices made to build the deep convolutional autoencoder and in Sect. 8 the time-series analysis tools, which allow to extract the periodic signal from the data latent space. Once most of the tools are presented, we show the performed experiment and their results in Sect. 9. We finally draw the conclusions in chapter in Sect. 10.

2 Astrometry

Before delving in the conceptual diagram of Fig. 1, we want to introduce the astrometric detection field and thus the reasons behind the TOLIMAN satellite architectural choices. Astrometric detection involves precise measurements of stellar positions and it is widely regarded as the leading concept presently ready to find Earth-mass planets in temperate orbits around nearby sun-like stars [e.g. 34, 36]. The principle for detecting a planet using astrometry is the same as that adopted by the hunters of unseen companions of stars [e.g. 2] about two hundred years ago. As a planet orbits the star, the latter is tugged in a small circle by reflex motion, thus, by careful measurements of the position of the star over time (either in a local or global frame, that must be more stable than the signal produced by the invisible companion), these tiny displacements, imposed on the host star by the gravity of orbiting exoplanets, yield a solution for the planet mass and orbit. Unlike other methods, there are few blind spots, and the signal generated by companions increases with planet-star separation, converse to both radial velocity and transit methods. These unique characteristics

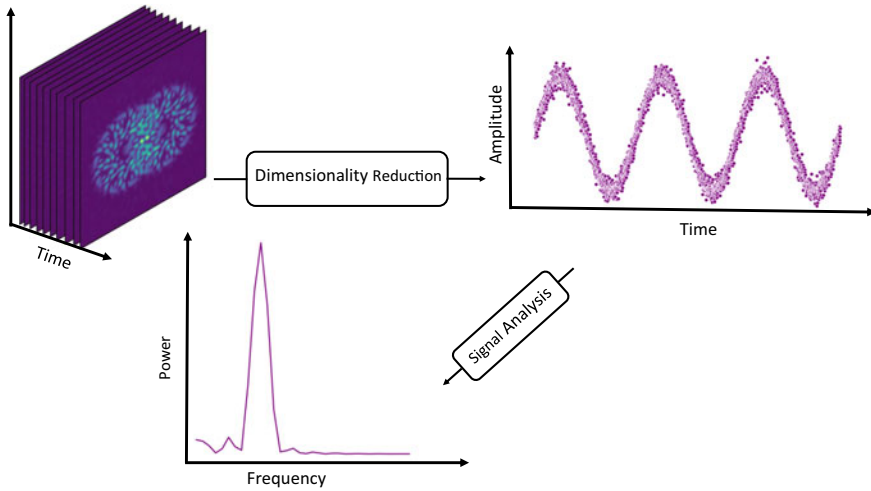


Fig. 1 Concept workflow. The underlying concept of the proposed data analysis is based on finding a lower-dimensional representation (a compression) of the raw data which preserves periodic signals. Once a suitable representation is found, the effect caused by the presence of the planet can be detected using a time series analysis

make it ideal for probing habitable zones at larger orbital radii. Furthermore, the intrinsic signal (the amplitude of the periodic angular wobble on the sky) is inversely proportional to distance, favouring stars in the immediate neighbourhood of the Sun.

However, despite the potential promise, astrometric detection for exoplanetary discovery has not yet entered the mainstream. The angular excursions induced by habitable-zone Earth-analog planets are small, of order of one micro-arcsecond even for best-case targets, such as Alpha-Centauri. Ground-based high precision astrometry campaigns must fight the considerable sources of noise, such as the starlight path through the Earth’s turbulent atmosphere. Long-baseline optical interferometers have historically delivered precisions better than 100μ -arcseconds, with a recent resurgence of interest prompted by ESO’s GRAVITY instrument [16] with accuracy an order of magnitude better, which is still not sufficient for Earth-mass planets, however. Furthermore, the nearest stars to Earth present a large apparent angular diameter and are correspondingly difficult to observe on long baselines, since they are over-resolved objects, and thus present challenges to the interferometric technique, due to low fringe contrast. These intrinsic challenges for ground-based astrometric observation have increasing the interest in space. Global, large space astrometric surveys over wide angles have proved to be extremely productive delivering fundamental stellar positions, distances and kinematics with the ESA/HIPPARCOS mission [11], and its ambitious successor ESA/Gaia [13, 14], which is now measuring billion stars with precisions of the order of $\sim 10 \mu$ as. Although the Gaia mission expected to deliver a rich harvest of gas giant planets [e.g. 3, 31], in order to detect and study

rocky planets in temperate orbits, we need to push detection thresholds down to levels better than $1 \mu\text{as}$, something that will require dedicated new concepts.

Conventional astrometry approaches measure the position of a star, using a grid of reference nearby objects. This requires relatively large fields of view since the distance between science targets and sufficiently bright reference objects are of the order of several arcminutes. However, maintaining long-term instrumental stability over such large angles is notoriously challenging. Several interesting missions have been proposed by groups in Europe [36], the US [40] and China [4], addressing the different concepts to solve this problem with highly stable and continuously monitored spacecrafts and instruments. This poses, however, an additional non-negligible problem: the instrumental cost scales significantly with the field-of-view. Thus it is natural to ask the question if it is possible to obtain micro-arcsecond level measurements for certain targets, like Alpha-Centauri, using much narrower fields-of-view, and thus avoiding the high costs associated to the stability of large field-of-view concepts.

2.1 *Narrow-Angle Astrometry*

Our ability to perform narrow field astrometric science ultimately rests on the ability to precisely register the position of the stellar image in each exposure. This meets a fundamental photon noise limit, even with a perfectly stable optical apparatus. Typically any bright nearby star will provide enough photons so that this theoretical limit is not a major problem, requiring only minutes or hours of integration with a telescope of reasonable aperture. However, the critical limitation is not set by photons from the target star but from the absolute stability of the image plane sensor required to perform the measurement; something that can only be accomplished with continual monitoring and ongoing calibration. For the practical narrow-field astrometry, registration of the images is performed by simultaneous monitoring of a constellation of background stars, which provide instantaneous information about the exact plate scale and further order deformations. Our astrometric detection error budget is therefore dominated by the accumulation of sufficient counts on these much fainter reference stars that, for a field of view of several arcminutes, are likely to be thousands of times fainter than the target star. The concept underlying the TOLIMAN mission was developed on the principle that it is possible to entirely sidestep this dilemma for the special case of observations of bright binary stars.

Where two bright stars lie close together in the sky, precise monitoring of their separation will deliver the key science with negligible photon noise. In particular, α Centauri is almost ideally tailored for a mission exploiting narrow-angle self-referenced astrometric detection. As our nearest celestial neighbour system, Alpha Cen's pair of solar-analogue stars means that habitable-zone exoplanets could be true Earth-twins in year orbits: at the sweet spot for detectability within an attainable

mission duration and yielding signals factors of 2–10 times stronger than the next-best systems. The two habitable zones have wide enough orbits to yield good signals, yet not so wide as to require an extended mission lifetime for detection.

3 TOLIMAN

The TOLIMAN space telescope [39] is a low-cost mission which aims to push the boundaries of astrometric measurements in binary star systems and to enable the detection of Earth-like planets around α Centauri, our closest extra-solar system. The mission is optimised to search for habitable-zone planets that, for α Centauri, implies deflections with amplitudes of order of $\sim 1 \mu\text{s}$ over roughly 1-year orbital periods. The detection of such a small astrometric signal has never been reported before in the astronomical literature.

To accomplish this task with an affordable spacecraft and mission profile, an innovative optical and signal encoding architecture was proposed. It explores and reformulates the idea of a Diffractive Pupil based optical system.

As originally envisaged, a diffractive pupil telescope would have a set of diffractive features, most simply a regular array of small opaque dots, embedded in the pupil of the instrument [e.g. 17]. These must be anchored to some element with extreme mechanical stability. The features cause starlight to diffract in the image, essentially forming a pattern whose features are exactly known and stable so long as the diffractive pupil remains stable. For bright sources, this simple concept offers a cunning solution to the key problem that overwhelmingly dominates astrometric error budgets: the stability of the instrument.

When trying to reference stellar positions at micro-arcsecond scales, a host of small imperfections and mechanical drifts, warps and creep of optical surfaces, generates systematic instabilities that can be orders of magnitude larger than the true signal. Rather than trying to directly contain all these errors, the Diffractive Pupil approach sidesteps them. It creates a new ruler of patterned starlight against which to register positions in the image plane. The cleverness of this approach is that the diffractive grid of starlight suffers identical distortions and aberrations to the signal that is measured. Therefore, drifts in the optical system cause identical displacements of both the object and the ruler being used to measure it, making data immune to a large class of errors that encompasses other precision relative astrometry approaches.

The opaque dots pupil proposed by Guyon et al. [17] results in a diffraction pattern where the image plane is populated by a regular grid of sidelobe images diffracted from the bright target star. However, when considering broadband illumination, bandwidth smearing of the starlight will draw each sidelobe into a narrow radial streak or ray. The signal recovery proceeds by registering the location of these rays against the background field stars. Because the diffractive ruler takes the form of long narrow radial rays, positional information recovered must be in the orthogonal ordinate. Therefore, the primary observable consists of the recovery of azimuthal positions of (a rich field of) background stars registered against the nearest diffraction rays. For

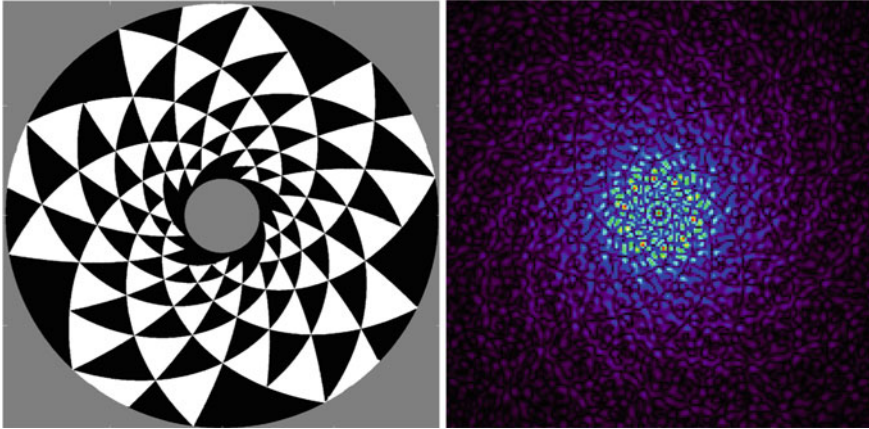


Fig. 2 Left Panel: a conceptual design pupil for the TOLIMAN mission, with white/black regions indicating discrete phase steps of $0/\pi$. Right Panel: the monochromatic PSF generated yields a complex and strongly featured pattern extending from the core, uniformly filling the region with sharp fringes

the TOLIMAN mission, the diffractive pupil formulation described above has two fatal flaws: (1) it relies on background field stars and (2) with its radially smeared ruler it is unable to yield precision measurement of the separation of any binary star. For only a single pair of stars, as is the case of Alpha-Centauri, radial information is essential. Instead, TOLIMAN proposes a novel form of diffractive ruler which generates fine-featured patterns capable of spanning the required separations between the components of a binary star system.

TOLIMAN requires diffractive pupils capable of creating patterns with a sharp structure extending in the radial direction. Our primary design driver was to find patterns that create a region on the image plane uniformly filled with features that have the highest gradient energy and that occupy the minimum span in dynamic range. Essentially, the former criteria attempt to optimise our ability to accurately register the resulting pattern—fitting algorithms rely on regions where the image has the strongest slopes or sharp edges. The latter condition is required to spread the starlight preventing saturation of the detector, and spanning the separation of the binary with diffractive features so as to enable the diffractive pupil methodology. Such a design is depicted in Fig. 2 and is now seen to meet our goal of filling the entire diffractive region, including the core, with sharp structure. Sharp gradients in the image plane optimise the ability to precisely register such an image.

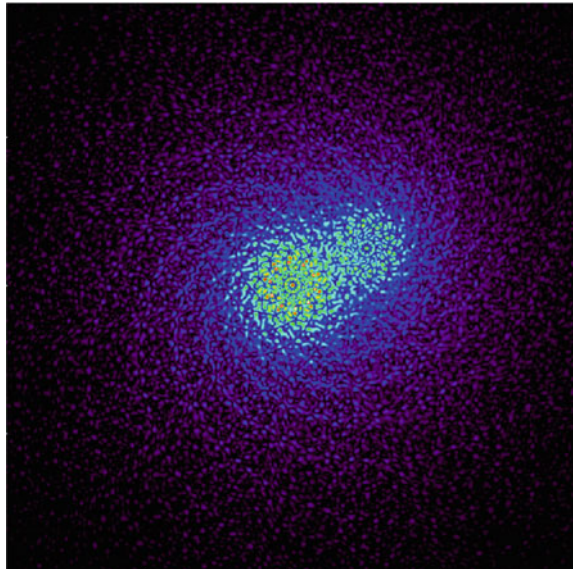
3.1 *The TOLIMAN Data Challenge*

In its simplest form, extracting the science signal arising from TOLIMAN data requires the exact registration of two overlapping point-spread functions, one for each component of the binary star, in the image sensor plane of the orbiting space telescope. If the separation between these two stellar images can be monitored with sufficient precision, tiny perturbations due to the gravitational tug from an unseen planet can be detected. Given the configuration of the optical system, the scale of the shifts in the image plane are about one-millionth of a pixel (10^{-6} pix), thus exquisite stability is required: these motions are only manifest as a sinusoidal perturbation over year timescales.

Although there are many potential sources of imperfection and error, this first study restricts itself to the most basic and fundamental one, with noise processes arising principally from photon noise and the spatial discretization of the signal. Additional terms, as imperfect spacecraft pointing, jitter and roll stability, will be addressed in future work. For the present study, simulated and laboratory testbed data were created to embody such error terms.

A pictorial illustration of the basic challenge is shown in Fig. 3: two patterns exist within the frame of data, in this case without the noise terms. High degrees of sharp image structure result in a data for which accurate image registration is possible; however, on the other hand the levels of extreme measurement precision required to obtain the science move this from a relatively routine exercise in image processing (at levels of 10^{-2} pixel) to an unsolved problem at signal fidelity levels never yet attempted (at levels of 10^{-6} pixel).

Fig. 3 A simulated binary star as observed with the conceptual design TOLIMAN pupil discussed above



4 Simulations

We are now ready to describe the simulations of the TOLIMAN data, i.e. the inputs of our conceptual workflow shown in Fig. 1. These simulations were necessary given that no testbed has been build yet to accurately reproduce the TOLIMAN data. This section, thus, describes the formalism and necessary steps to produce a mock data set that mimics the precision level required by the TOLIMAN mission. The first challenge in developing a method capable of extracting a signal as small as one-millionth of a pixel is to develop a computational model capable of emulating such signal under varying conditions of noise. Although injecting a signal into an image may seem a rather trivial task, conventional approaches fall short when pushed to the limits of precision required by the TOLIMAN mission, often resulting in large computational cost. The traditional simulation approach consists of generating a super-sampled Point Spread Function (PSF). Since stars can be considered point sources, to simulate a stellar field as would appear on the detector, we simply need to shift and downsample that PSF in the sensor grid. Thus, by assigning it to either random or specified positions within the image and repeating the procedure for many different point sources, we can recreate a stellar field. While this can be made computationally efficient today using the widespread GPU accelerators, such traditional methods unfortunately introduce errors orders of magnitude greater than the signal we expect to measure, thus requiring alternative approaches to the generation of the mock data.

4.1 The Fast Fourier Transform

The Fast Fourier Transform (FFT) has long been used as an optical simulator since it performs the same operation numerically as a focusing mirror or lens does optically. An input image will undergo a transformation from a spatial representation to a frequency representation when observed at the focal plane. The Fast Fourier Transform operates on a digitised representation of the input with an $O(n \log(n))$ computational complexity, making it a corner stone in basic computations of optical systems. In this section, presenting the basic underpinnings, we detail how one can use FFT's to create images of stellar fields with the injection of arbitrarily sized positional information.

Generating these PSF's is conceptually straightforward, requiring only the representation of the electric field at the aperture $E(x, y, \lambda) = A(x, y)e^{i\theta(x, y, \lambda)}$ as its amplitude $A(x, y)$ and phase $\theta(x, y)$ and combine these terms into a complex array. The PSF in the (u, v) focal plane is then found by taking the power of the resultant FFT of the complex array.

$$PSF(u, v) = |\mathcal{F}\{E(x, y, \lambda)\}|^2 = |\mathcal{F}\{A(x, y)e^{i\theta(x, y, \lambda)}\}|^2 \quad (1)$$

Positional information can then be injected through applying a linear gradient to the phase θ . An Optical Path Difference (OPD) is introduced across the aperture by any source off-axis from the normal of the telescope pointing. Easily calculated through the angular offset from the normal, the OPD simply translates into phase as a function of the observation wavelength.

$$\theta_{slope}(x, y, \lambda) = \frac{2\pi}{\lambda} OPD(x, y) \quad (2)$$

Having this mathematical representation of on-sky position to telescope response allows for arbitrary signal sizes to be introduced to any stellar objects. Other natural or designed phase perturbations like optical aberrations (coma, astigmatism, etc.) or devices like the TOLIMAN diffractive pupil can easily be represented and added to the other phase sources. Optical aberrations are not explored in this work but the principles underpinning their simulation follows simply from this work. Other phase devices like the TOLIMAN diffractive pupil $\theta_{pupil}(x, y, \lambda)$ follow the same general idea. Formulated as a mirror with ‘steps’ cut in, we take the height of each step $h(x, y)$ and translate to phase by taking the OPD as twice the height of the step.

$$\theta_{pupil}(x, y, \lambda) = \frac{2\pi}{\lambda} 2h(x, y) \quad (3)$$

The total phase θ is then a linear combination of these effects. Taking the field amplitude $A(x, y)$ as unity for all non-masked regions of the aperture gives the full description of the electric field $E(x, y, \lambda)$.

Having formulated the electric field response to the system, we must introduce a complete description of the optical architecture. This is described by a handful of parameters: aperture diameter D , effective focal length fl and pixel size d_{pix} . Desiring computational efficiency through the inclusion of our optical system, we define some value N_{out} to be the size of the array which we pass to the FFT. This is the primary driver behind the computation cost. Using this value and the previously described parameters, the size of the array N_E representing our electric field $E(x, y, \lambda)$ can be found. Note all arrays are taken to be of size $N \times N$. These two values necessarily differ as a way to encode optical parameters without focal plane interpolation. The ratio between N_{out} and N_E determines sampling in the focal plane matching that of our system.

$$N_E = N_{out} \frac{d_{pix} \times \lambda}{D \times fl} \quad (4)$$

Embedding this array representing the electric field into an N_{out} sized array, we use Eq. 1 to generate a PSF that requires no interpolation and can have positional signals of any size injected, limited only by floating-point precision of course. Further details and descriptions of these processes can be found in [32].

4.2 Generating Data

With the tools to simulate PSF's through our optical system we must now generate a data set. By adding basic noise processes, stellar spectrum and astrometric signals we can create a comprehensive set of images that can be used to test the recovery and reconstruction abilities of all the data-driven techniques described in this chapter. Here a balance must be struck, generating a truly comprehensive data set for the TOLIMAN mission is merely intractable. With a full signal period of order one year, any data set must present the fundamental challenges of the mission in an efficient way. Here we examine choices such as number of wavelengths, stars and images to simulate, along with the included noise processes.

One of the first things to consider is the size of the data set, the total number of images produced. The TOLIMAN signal is introduced to the α Cen system through the gravitational tug of an orbiting planet and so our signal is sinusoidal by nature. The orbital period that we are searching for is of order of a single year, and so producing a 'frame by frame' data set would be computationally intractable. Consequently, we need to generate each 'image' as a representation of a collection of multiple from the actual telescope. We chose to represent three full signal cycles over 1000 images, with each image representing approximately a full day.

Observing in the visible spectrum over a 100 nm bandwidth, the choice spectral resolution is essential. The wavelength dependence of the PSF demands that the image at each wavelength be computed individually. To represent the real world as closely as possible, a spectral resolution of 1 nm was chosen for several reasons: (1) firstly the Toliman PSF is spread over many diffraction limits ($10\lambda/D$) so at the outer reaches bandwidth smearing begins to have a substantial effect on the PSF shape (for an example see Fig. 4); (2) secondly, by choosing to maintain the stellar alignment on the detector constant and keeping one of the stars stationary, we can massively reduce the number of PSFs we must compute. A stationary star only requires the calculation of a single broadband PSF. For the moving star, since the TOLIMAN signal is sinusoidal by nature and the stellar alignment is kept stationary, we only

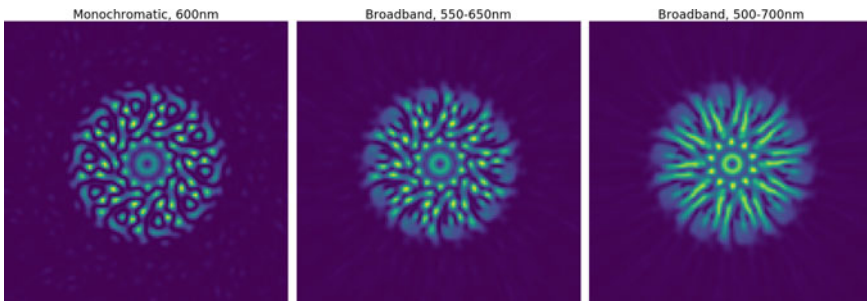


Fig. 4 TOLIMAN PSF at different bandwidths. Left: Monochromatic 600 nm. Centre: 550–600 (best resembles actual mission). Right: 500–700

need to calculate the PSFs for a single signal cycle. The result is a large overhead for small simulations but with the benefit of being able to produce large and accurate simulations efficiently.

Given our spectral resolution, we can use one of the many libraries available to generate spectra that reflect the true stellar parameters for each star. These libraries access existing stellar databases and recreate synthetic spectra for a host of variable stellar parameters such as an effective temperature, metallicity and observational flux. We used *Pysynphot* [35] to generate stellar spectra and fluxes for our system. This package uses models built from HST observations across the HR diagram to simulate atmospheric emissions from different stars. Taking the relative fluxes and total photon counts output from this system, we can scale each monochromatic PSF by its relative power to recreate accurate PSFs.

While real data will feature many varied noise processes, here we only consider two noise sources: photon and detector noise. These are dictated by Poisson and Gaussian statistics respectively. Detector noise is primarily driven by random thermal fluctuations of the discrete electrons that carry the signal through the detector. With available modern low-noise sensors, this noise is not expected to limit the extraction of the signal since it averages out to some constant value over many frames. The addition of even modest levels of this noise also serves a separate motivation: to allow for a smoother error space. This helps numerical algorithms converge faster as fine structures in the gradients are rounded and the algorithms can follow a smooth descent to the optimum. On the other hand, photon noise is an essential processes that must be examined. Arising from the discrete nature of photons, this noise is simulated at each pixel by drawing from the Poisson distribution whose mean is dictated by the PSF. When performing image registration of small signals such as those anticipated in the TOLIMAN mission, the total number of photons that arrive in each image becomes an important factor. As shown in [17], there is a fundamental relationship between the number of photons received and the positional information carried by those photons. With insufficient photons, signals can not be extracted.

Simulations proceeded with the production of a comprehensive batch of noisy image data sets, with sinusoidal signals in separation of the binary star injected with decreasing amplitude to mimic increasingly more challenging planets, up to the limiting deflection of one-millionth of a pixel. These simulations closely resemble the expected response of the TOLIMAN optical system to the observation of the α Cen system and were used to build and train machine learning algorithms.

5 Dimensionality Reduction

As it can be seen in the conceptual diagram in Fig. 1, the first and the most crucial step in the proposed data analysis scheme is to apply some transformation to the raw data produced by the instrument to allow us to unveil the periodic changes in the images through time. This transformation can be seen as a dimensionality reduction,

a compression of the imaging data into a smaller dimensional space that preserves periodic signals that may exist in the data.

Dimensionality reduction [e.g. 30], that is, of representing data in a different space than that in which it can naturally be observed, is a set of techniques that try to transform data from one representation to a lower-dimensional one with the lowest information loss. Reducing dimensionality of data with minimal information loss is important for feature extraction, compact coding and computational efficiency, to eliminate redundancies and enforce constraints. In particular image compression techniques try to take advantage of the statistical properties of the images in order to reduce their computational footprint. One of the most straightforward and widely used approaches of dimensionality reduction is the Principal Components Analysis (PCA) [9]. This approach consists in applying a linear projection of the original data on a set of orthogonal axes (the principal components), built to recover the maximum amount of information contained in the original data with as few coefficients as possible. In practice, PCA can be computed by performing a singular value decomposition of the data, contained in a matrix X . Each datapoint can then be reconstructed by a linear combination of the basis elements: $X \approx DA$, where D is a matrix containing the principal components, and A a matrix containing the coefficients used in their linear combination when reconstructing the data. Another broad class of dimensionality reduction methods, closer in heuristic to using a neural network to build the new representation space, is that of dictionary learning (DL). Instead of using PCA to select the new basis of representation D , one can instead *learn* it from the data itself. Much like in deep learning approaches, dictionary learning relies on the choice of a loss function l to quantify the difference between input data and its reconstruction. Learning the representation then amounts to solving the following optimization problem:

$$\min_{D,A} l(X, DA). \quad (5)$$

Depending on the desired properties of the representation to be learned, one can further add *constraints* to either the dictionary D or the coefficients A . A great many flavours of dictionary learning exist depending on the constraints selected. One of the most widely used is the addition of a *sparsity* constraint on A [26]. In practice, the sparsity constraint is often obtained by adding an l_1 term to the cost function:

$$\min_{D,A} l(X, DA) + \lambda \|A\|_1. \quad (6)$$

Several other constraints exist, and often lead to the resulting dictionary learning approach having its own name: non-negative matrix factorization [23] when using positivity constraints, sparse PCA [8] when the sparsity constraint is instead imposed on the dictionary, independent components analysis [19] when imposing statistical independence between the components. Both PCA and DL have been utilized, in the development of the work described in this chapter, to compress the images and a period consistent with the one of the signal injected in images could be found in the produced lower-dimensional representations up to a signal amplitude of 10^{-4} .

These techniques were not capable to detect astrometric signals with amplitudes at the order of 10^{-6} times smaller than the pixel size. However, their application showed us that, through data driven techniques, the images could be transformed into a lower-dimensionality space while preserving the temporal signal structure and thus that the challenging μ arcsecond level signals could perhaps still be recovered with the use of other techniques, such as Deep Learning. The next section makes a small, but self-contained, introduction to these other techniques.

6 Deep Learning

In this section we will review all the concepts underpinning Deep Learning needed to understand the inner workings of the deep convolutional autoencoder used in this work to create a lower-dimensional representation of the TOLIMAN simulated data. The main advantage of these techniques over classical dimensionality reduction is that the layered structures of Deep Neural Networks (DNNs) can encode an input representation with increasing levels of abstraction in successive layers [15, 21]. For such reasons, in the last decade, Deep Learning has been successfully applied across a wide range of applications including computer vision, speech recognition, bioinformatics and astroinformatics.

In this work, we make use of two classes of Neural Networks: fully connected and convolutional. Fully connected Neural Networks, also simply known as Neural Networks, can be used to approximate any nonlinear functional relationship between a set of inputs and outputs [7]. Each layer of a neural network transforms a vector of inputs $x \in R^N$ as follows:

$$\hat{y} = f(Wx + b), \tag{7}$$

where $W \in R^{(N \times K)}$ is a matrix of weights, $b \in R^K$ is a bias term, and the nonlinear activation function $f : R \rightarrow R$ is applied component-wise. The bias term shifts the baseline activation function input away from zero, providing richer behavior for modelling the functional relationship between the input and output variables. In networks with multiple layers, the output of each layer is connected to the input of the following one

$$y_{l+1}^{\hat{}} = f_{l+1}(W_l h_l + b_l) = f_{l+1}(W_l f_l(\dots(f_1(W_0 x + b_0) + b_1) \tag{8}$$

where h_l is the hidden layer or feature vector of layer l . The input is processed through all the layers until it reaches the output of the network \hat{y} .

The parameters (weights and biases) of the network are selected to minimize a *loss function*, such as a mean square error, summarizing the difference between the network output and a desired or observed target value y . Stochastic gradient descent (SGD) is a common optimization process for neural networks: at each stage of training, the network parameters are updated by a small vector proportional to the gradient of the loss function with respect to those parameters. This is straightforward for the

output layer; weights and biases in overlying layers can be efficiently calculated through successive applications of the chain rule for derivatives, in a process called *backpropagation*. The use of derivative information for efficient network training requires that the loss function be smooth.

Convolutional Neural Networks (CNNs) differ from fully connected neural nets only in that their architecture exploits the localized structure of images to reduce the number of network parameters needed. Instead of connecting each neuron in a layer to every other neuron in the next layer, the connection structure of CNN layers is sparse, and parameters are shared across a layer to enforce translation invariance of features extracted on each scale across the image. Three types of layers are typically used: (1) Convolutional Layers, (2) Pooling Layers and (3) Fully-Connected Layers. In the following, we will analyze in detail the architecture and inner workings of each one of them.

6.1 Convolutional Layer

The Convolutional Layer is the most computationally intensive part of a CNN architecture; its parameters consist of a set of learnable filters. Every filter, also known as a *kernel*, is spatially small (usual sizes are 3×3 , 5×5 and so on, where three and five are sizes in number of pixels), but includes weights for each channel of its input. For the first layer, these channels are the data channels (for example R, G, B in a three-channel image). In subsequent layers, each channel corresponds to the output of a single kernel from the previous layer. During the forward pass, each kernel slides across the spatial dimensions of the input, computing the dot product between itself and the part of the input volume that it encompasses (*convolution*). As the kernel slides, it produces a bi-dimensional activation map that encodes the responses of the kernel at every spatial position. The content of the activation map at each location is a direct response to some visual feature present in the image to which the kernel is sensitive, such as an edge or a colour. Each convolutional layer employs multiple different filters, producing a set of activation maps that are stacked along the depth to produce a multi-channel output. Due to the limited size of the filters, neurons are not connected to the full extent of the input volume but only to a small region (the *receptive field*). The connections are thus local in space (width and height of the input), but are always fully connected in-depth (i.e. across learned/extracted features).

The structure of the output volume of a convolutional layer is controlled by three hyper-parameters:

- *Depth*: the number of filters learned in the layer;
- *Stride*: the number of pixels the filter is shifted along the spatial dimensions of the input volume. It is usually set to one but it can be set to higher values, depending on the image geometry, to achieve less redundancy in the output volume. The stride controls the spatial dimensions of the output volume;

- *Padding* or *zero-padding*: the width in pixels of a spatial region on the borders of the output that is filled with zeros. It controls the spatial dimensions of the output volume and can be used to preserve the spatial dimensions through the layer.

Finally, to ensure that each kernel is learning a single feature that has a consistent interpretation across the spatial extent of the input, all neurons in the same depth slice share the same weights and biases, irrespective of where across the extent of the input they are applied. Thus the action of each filter in the forward pass becomes a discrete convolution of a single set of kernel weights with the input.

A convolutional layer acts to encode its inputs into a latent space spanned by the features it learns. However, the autoencoder architecture we will consider in later sections also involves a transformation from a learned latent space back into the image domain. Thus, while convolutional layers typically decrease the spatial extent of their inputs, we will also need *deconvolutional layers* which increase them, recombining a potentially large number of learned features into a flat image. Mathematically both convolutional and deconvolutional layers can be summarized as

$$l^h = f\left(\sum_{i \in L} x^i \otimes w^h + b^h\right) \tag{9}$$

where l^h is the latent representation of the h th activation map of the current layer, f is the activation function, and x^i is the i th activation map of the L -feature activation of the previous layer in the network (or the l th channel of an L -channel image in the case of the first convolutional layer after the input image). w^h and b^h are, respectively, the weights and biases of the h th activation map (shared by all neurons of the map) of the current layer. Given that x^i has size $m \times m$ and the filters have size $k \times k$, a convolutional layer produces an output feature map with shape $(m - k + 1) \times (m - k + 1)$, thus reducing the size of the input. A de-convolutional layer outputs a feature map with shape $(m + k - 1) \times (m + k - 1)$, thus increasing the size of the input.

6.2 Pooling Layer

The architectural function of a Pooling Layer is to reduce the spatial size of the representation, which reduces the number of parameters, lightens the computational load, and mitigates overfitting. The pooling operation is carried independently on each input feature, leaving the number of input features unchanged. Different criteria in the literature exist to perform the pooling operation, including *max*, *average* and *L2-norm pooling*; max-pooling is the most commonly used. There are also *un-pooling* layers to desegregate and expand activation maps in transformations back towards the image domain.

A max-pooling layer pools features by computing the maximum within the feature map and outputs a feature map with reduced size, according to the chosen size of the pooling kernel. To perform a successive un-pooling, the max-pooling layer also

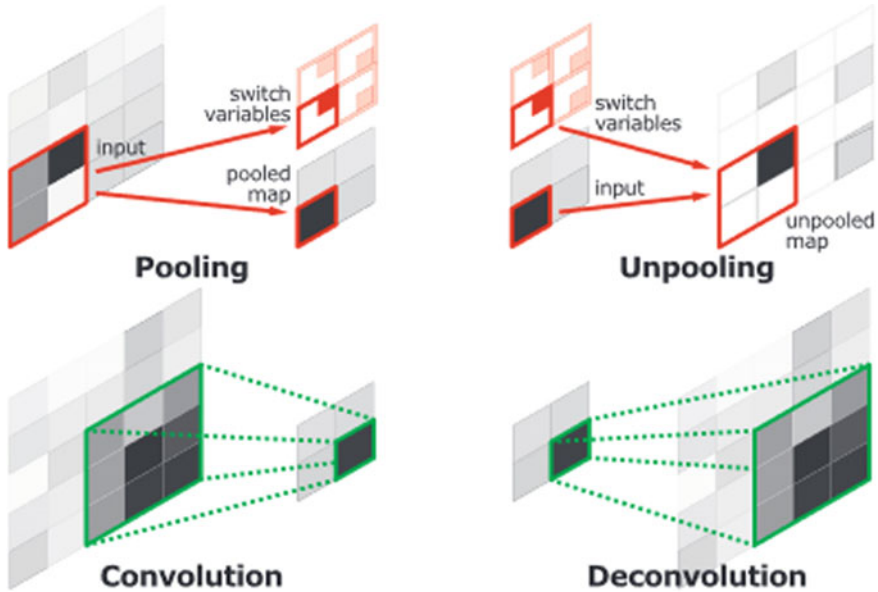


Fig. 5 Illustration of max-pooling, unpooling, convolution and deconvolution layers [29]

records a set of switch variables which describe the positional information relative to the pooled features. The un-pooling layer restores the max-pooled features into the correct position specified by the relative switch variable values. The combination of max-pooling and un-pooling layers is thus able to retain both the image magnitude (answering the “what” question) and the positional information (the “where” question). Figure 5 [29] shows some stylised representations of convolution—deconvolution and pooling—un-pooling operations.

6.3 Deep Convolutional Autoencoder

An Auto-Encoder model (AE) is a neural network composed by an encoder and decoder part; the encoder $f : X \rightarrow H$ transforms the input image into a lower-dimensional representation (the *latent space*), while the decoder $g : H \rightarrow X$ tries to reconstruct the original input image from this representation. By constraining the latent space to be of lower dimension than the original input data, we can force the autoencoder to capture the most important features of the input data in order to reproduce it successfully. This type of restriction can be used for feature extraction and for dimensionality reduction.

During the learning process, network parameters are adjusted to minimize a loss function

$$L(x, g(f(x))) \tag{10}$$

that encodes the difference between the input x and its reconstruction $g(f(x))$. As for the NNs discussed in Sect. 6, L must be smooth in order to use gradient-based minimization algorithms such as SGD. If L is chosen to be linear, the auto-encoder performs a dimensionality reduction similar to Principal Component Analysis (PCA); in fact, the latent space h ends up to be the principal subspace of the input data. If, instead, L is non-linear the auto-encoder can learn much complex representation.

Generally, autoencoders are built by two shallow fully connected NNs joined through a lower-dimensional latent space. A CAE (Convolutional AutoEncoder), instead, contains, in the encoder part, a stack of convolutional and max-pooling layers before the fully connected layer and, in the decoder part, a stack of up-sampling and de-convolutional layers after the fully connected layer. It has been shown [45] that CAE are better suited, with respect to AE, for image processing and reconstruction tasks, due to the full utilisation of the CNNs capacity to extract a hierarchical set of features from the images. These have been proven to show a better performance over shallow neural networks when working with noisy or complex images. Moreover, the combination of a convolutional and max-pooling layer allows the higher-layers representations to be invariant to small rotations and translations thus helping with the TOLIMAN satellite inevitable jitters and translations.

In recent years AEs have been applied to solve a wide range of problems in the Astrophysical context; to model the Point Spread Function of Wide Field Small Aperture Telescope [20], to uncover and separate the faint cosmological signal from the epoch of reionization [24], to classify galaxies Spectral Energy Distributions [12], to identify Strong Lenses candidates in the simulated data of the Euclid Space Telescope [5] and to solve the Star—Galaxy classification problem [18]. Moreover in the fields of Computer Vision and Image Processing, AEs have been successfully used to recover structured signals from natural images [28], for image compression [1, 10, 37, 38, 41], achieving compressing performances similar or better than the JPEG 2000.

Encouraged by the results obtained in literature in lossless image compression and signal recovery from images through AEs, we decided to develop our custom CAE architecture to recover the astrometric signal from the TOLIMAN simulation images. Sect. 7 contains an in-detail description of the architectural design, given the peculiar nature of the scientific problem.

7 Model Architecture

In this section we take implement knowledge detailed in Sect. 6 to build the actual CAE architecture that compresses the TOLIMAN simulated images into a latent space that showed a periodic trend with time. Some of the architectural choices came from our knowledge of the physical problem, some from the expected behaviour of the network, and others were discovered on a trial-and-error basis.

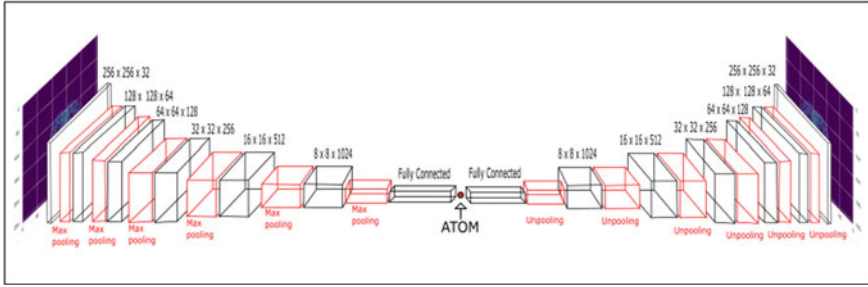


Fig. 6 Architecture of the Deep Convolutional Auto-Encoder

Fig. 7 Comparison Between Rectified Linear Unit and Exponential Linear Unit activation functions

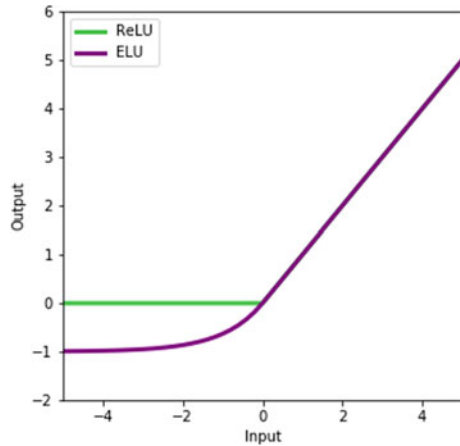


Figure 6 shows the overall architecture of the CAE. Each convolutional and deconvolutional layer is followed by an *Exponential Linear Unit* (ELU) activation function. It has been shown by [6] that this function is able to capture the degree of presence of particular phenomena (the signal) and not its absence, thus creating in the network a complex weight space, chains of connections specialised in solving particular tasks (like encoding the signal). Moreover, since ELU may have negative values, it pushes the mean of the activations closer to zero. Having mean activations closer to zero causes faster learning and convergence. Said that, ELU is very similar to RELU, except for negative input values. In fact ELU becomes smooth slowly until its output equals $-\alpha$ where RELU sharply smooths (Fig. 7).

For negative activations, RELU’s gradient will be 0 and this may prevent the network weights to be adjusted during descent. This means that all the affected neurons going into that state will stop, responding to variations in input (being the gradient 0 there is no input that can make them change, they have reached a local minima from which they are unable to escape). This is called dying RELU problem. Apart from the described computational problem, RELU is less responsive to negative activations, something that may harm the signal reconstruction in the latent space

for all images where the binaries separation is smaller than their mean separation in the training set. For all these reasons, we chose ELU as the activation function of all hidden layers.

The Network latent space was chosen to be uni-dimensional (represented by the single ATOM in Fig. 6), for the following reasons: (i) when a higher dimensional latent space was used, the Pearson correlation coefficient between the latent variables was found to be compatible with a value of 1.0; (ii) given that the separation of the star's PSFs is radial, and, given that the only varying feature in the images is the signal, it seems reasonable to think that the only information the network needs to recover from the latent space in order to decode, and thus reproduce, the images is the signal itself. The remaining constant information (pixel luminosity and image geometry) can be stored in the network weights. In literature, Deep Neural Networks tend to employ two types of loss functions: the Mean Square Error (MSE) and entropy-based loss function like cross-entropy or binary-entropy or the kullback leibler divergence. Although all types of loss functions have explicit probabilistic interpretations, MSE is estimating the mean of any distribution, while the entropy-based functions try to maximize the likelihood of a multinomial distribution, they differ in their application field. The latter type, with a logistic output, tends to heavily penalize wrong class predictions and thus are specifically suited to work in classification tasks where the decision boundary is significant. The first (MSE) is very forgiving on misclassifications but is well suited to handle regression problems, where the distance between two predicted values is small. Since our scope is dimensionality reduction, i.e. a regression problem, the MSE was chosen. To test the quality of the image reconstruction, we computed the mean MSE (MMSE) and the mean Structural Similarity Index (MSSI), [43], between all the available images and their reconstructions. the SSI models any image distortion as a combination of three factors: correlation loss, luminance and contrast distortions. When comparing two images, the estimator takes into account the mean luminance difference between the two images, the closeness of their contrast and their correlation coefficient. The number of layers, filters and other layer parameters were heuristically chosen through a trial-and-error campaign.

8 Signal Analysis

As the reader can see from the workflow figure (Fig. 1), the second step in the proposed pipeline is to perform the Signal Analysis in order to unveil periodic trends in time. For these reasons, in this section we present an overview of the chosen method, for instance the Lomb Scargle Periodogram technique, explaining the pre-processing steps performed to compare the atom time series (see Sect. 7 and VanderPlas [42] for details.)

8.1 The Lomb-Scargle Periodogram

The most commonly used tool for period searching in irregular cadence astronomical light curves is the Lomb-Scargle periodogram [LSP, 25, 33, with 4000 citations] that assumes a sinusoidal periodic behavior. It is a generalization of the Schuster periodogram in Fourier analysis

$$P_s(f) = \frac{1}{N} \left\| \sum_{n=1}^N g_n e^{-2\pi i f t_n} \right\|^2, \quad (11)$$

but for irregularly cadences. The LSP stands out as a robust procedure to build a power spectrum in order to detect periodic components in unevenly sampled datasets. In the uniform sampling regime, the Schuster periodogram encodes all of the relevant frequency information present in the data. This definition can be generalized to the non-uniform case, which is the scenario we explore here. It follows that the generalized form of the periodogram addressed by [33] takes the form:

$$P_{LS}(f) = \frac{1}{2} \left\{ \frac{\left\| \sum_{n=1}^N g_n \cos(2\pi f [t_n - \tau]) \right\|^2}{\sum_{n=1}^N \cos^2(2\pi f [t_n - \tau])} + \frac{\left\| \sum_{n=1}^N g_n \sin(2\pi f [t_n - \tau]) \right\|^2}{\sum_{n=1}^N \sin^2(2\pi f [t_n - \tau])} \right\}, \quad (12)$$

where τ is specified for each f to ensure time-shift invariance:

$$\tau = \frac{1}{4\pi f} \tan^{-1} \left(\frac{\sum_n \sin(4\pi f t_n)}{\sum_n \cos(4\pi f t_n)} \right). \quad (13)$$

This modified periodogram differs from the classical periodogram only to the extent that the denominators $\sum_n \sin^2(2\pi f t_n)$ and $\sum_n \cos^2(2\pi f t_n)$ differ from $N/2$, which is the expected value of each of these quantities in the limit of complete phase sampling at each frequency.

8.2 Atom Time Series Analysis

To estimate the period of the atom time series, we used the Lomb Scargle Periodogram and to validate the goodness of the period estimation, we employed the following metrics:

- False Alarm Probability (FAP): encodes the probability of measuring a peak of a given height (or higher) conditioned on the assumption that the data consists of Gaussian noise with no periodic component;
- Full Width at Half Maximum (FWHM): this expresses the extent of a function produced by the difference between the two extreme values of the independent

variable at which the dependent variable is equal to half of its maximum value. Treating the FWHM as an error measure, we derived an error on the period through the following expression:

$$P = \frac{1}{f(peak)} \quad (14)$$

$$\Delta P = \frac{1}{f(peak)^2} \times \left[f\left(peak + \frac{FWHM}{2}\right) - f\left(peak - \frac{FWHM}{2}\right) \right] \quad (15)$$

where *peak* stands for the peak of the power spectrum and $f(peak)$ its relative frequency.

In order to compare the atom time series and the signal, we standardized both of them, i.e. with subtracted to both time series their mean values and divided by their standard deviations. This preprocessing step was needed due to the Network inability to perfectly recover the signal amplitude in the latent space.

9 Experiments and Results

This section describes all the experiments performed with the CAE to compress the images to a lower-dimensional representation, showing a periodic trend in time, i.e. a latent space that preserved the signal, analysing the compressed representation in search of a periodic signal in time.

Before deploying the model on the simulations containing the signal with an amplitude a factor of 10^{-6} smaller than the pixel dimension (for details on the simulations see Sect. 4) and the realistic binaries PSFs flux ratio, the Network encoding capabilities were tested on images containing signals with amplitudes respectively 10^{-2} , 10^{-3} , 10^{-4} , 10^{-5} smaller than the pixel dimensions, equal flux PSFs (the binaries PSFs presented the same flux) and an image peak value of 10^9 photons and photon noise arising from the Poisson statistics. Due to the absence of any realistic noise components (jitter, rotations, aberrations etc.), each image was cropped with a 256×256 pixels window centred around the image barycenter. This preprocessing was needed in order to eliminate any spurious shift in the image pixels that could have compromised any training attempt capable of extracting the signal from the images. In fact, both the max-pooling and convolutional layers (see Sect. 6) are not shift-invariant and, as clearly shown in [44], the presence of a shift can completely change the outcome of these operations unpredictably. Each dataset thus contains 1095 single-channel centred images of which 985 were used for training and 110 for validation. The network was trained for 10, 000 epochs. The signal is sinusoidal with a period of 356 days and thus it performs three complete cycles in the 1095 images.

The final MMSE and MSSI on the validation set are found to be respectively 4.4×10^{-8} and 0.999938 and thus the images are reconstructed with a precision good enough (with respect to the accuracy needed) to recover the signal. Although

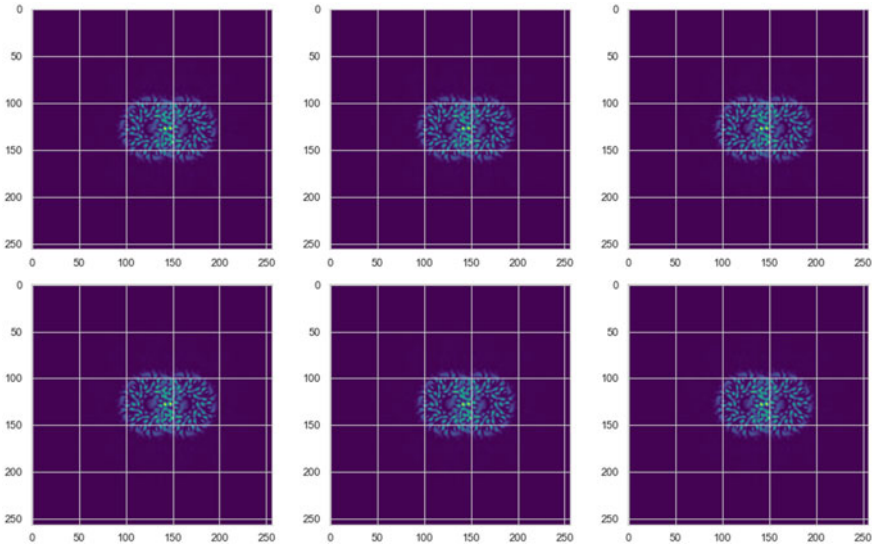


Fig. 8 Example of the signal reconstruction after training the network. The first row contains a random subset of simulated Toliman images, the second row shows their respective reconstructions produced by the trained network

the image reconstruction capability of the network directly correlates with these losses (as one should expect), we do not find any direct correlation with the signal reconstruction capabilities. Although after 1000 epochs the MSE Loss gradient flattened, for some reasons the latent space began showing an increasing sinusoidal trend in time with an increasing number of epochs. To have a loss function that correlates with the signal reconstruction in the latent space, we would need an architecture that makes use of the time dimension of the images: something not anticipated at the time of the publication of this work. For this reason, the network is encoding only the detection of the signal and not its amplitude. In order to make sure that the periodic trend observed in the latent space was coming from a signal injected in the images and not from any other periodic trend (in time) in the images or by chance, the Network was run on a blind set of simulations of which some contained a signal and some did not. The Network latent space did not show any periodic trend for all the simulations with no signal injected or, even if a period was recorded, the resulting FAP (see Sect. 8) would be extremely low (Fig. 8).

Table 1 shows the result of applying the method of compressing the signal using Deep Convolutional Auto-Encoders (CAE) and afterwards using Lomb Scargle Periodogram to analyse this compressed representation. This table shows that the proposed method is able to capture the signal with very low FAP and reasonable relative error, when compared with the error obtained by direct analysis of a perfect signal. The Signal and Atom time series, and their relative Lomb Scargle Periodogram, are

Table 1 Period found with the Lomb Scargle Periodogram and relative error and FAP

Time Series	Period	FAP
Signal	0.33 ± 0.05	0
Atom	0.33 ± 0.06	7.2×10^{-68}

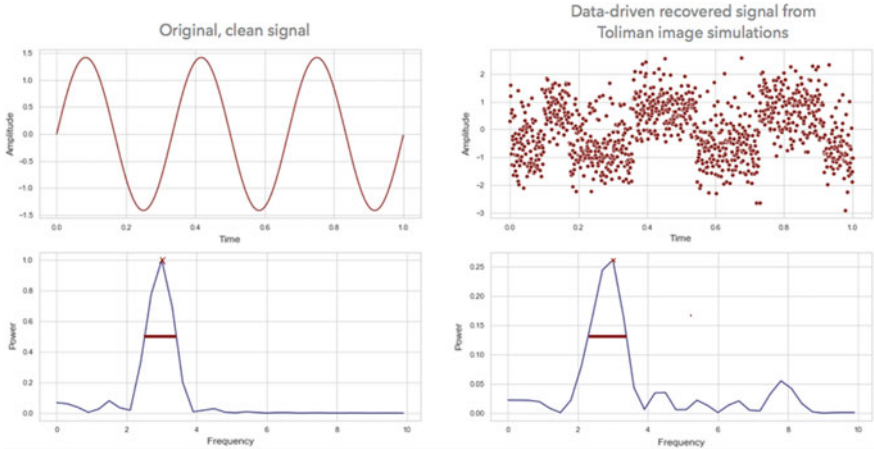


Fig. 9 In the left panels, a perfect signal is represented in the top and the relative Lomb Scargle Periodogram obtained from its analysis is represented in the bottom. In the right, a time series from the atoms obtained with the deep convolutional auto-encoder applied to TOLIMAN simulation with a 10^{-6} -level astrometric shifts is shown on the top, while its Lomb Scargle Periodogram is represented in the bottom. The power peaks and their relative FWHM are shown in red over the power spectrum

shown in Fig.9. In yellow we highlighted the power peak and the FWHM of the power spectrum around that peak.

9.1 Discussion of Results

Section 9 describes both the Network reconstruction capabilities and the analysis on the atom time series to find its periodicity. We have shown that the found periodicity is compatible with the injected astrometric signal period and thus that the architecture is able to recover the signal directly from the TOLIMAN simulation images. One of the main current issues is the lack of correlation between the Network reconstruction of the TOLIMAN images and the presence of a periodic trend in the atom time series. Since the Network is only training with spatial information and that the used loss (MSE) only takes into consideration the ability to reconstruct the input images, in reality there is no encoded reason why the latent space should present a sinusoidal trend with time. The only thing that the latent space should be encoding is “how

to reconstruct the images” and nothing else. That being said, given our knowledge of the sinusoidal nature of the astrometric signal and being the astrometric shift the only element changing through the images, we do not see any reason why the latent space should not present a sinusoidal trend with time, regardless the fact that we did not apply any constrain (on the architectural level) to force it. As seen in Fig. 9, in fact, given enough epochs, the time series actually shows a sinusoidal behaviour.

A necessary step forward in this work is to produce simulations with increasing noise realism and complexity, in order to evaluate if Deep Learning can still be used to recover the astrometric signal. It must be expected that this simple approach would fail to recover the signal if spatial transformations invariance is achieved on an architectural level.

10 Conclusions

In this work, we have shown how Deep Learning, in particular deep convolutional autoencoders (CAE) can be used to extract, in a completely unsupervised way, periodic astrometric signals with amplitudes of the order of 10^{-6} with respect to the size of a pixel. This is the magnitude of the signals that would be produced by an Earth-like planet at the habitable zone of a star in the Alpha Centauri binary system (see Sect. 2).

We presented a detailed explanation of the adopted network architecture (see Sect. 7) and of the simulations used, which were created using FFT techniques (see Sect. 4). Although the present simulations do not yet contain some realistic systematic noise components, such as telescope jitter, rotations and aberrations, they pose a significant challenge to classical unsupervised techniques, due to the small amplitude of the signal with respect to the pixel size. We have shown that, from the obtained CAE latent space, we can obtain a time-trend that can be analysed for periodicity, using any time-domain signal extraction technique. Here we used a standard Lomb Scargle technique (see Sect. 8), and were able to find a period consistent with that of the injected signal (see Sect. 9).

Finally, we note that in this work we only explored a fully unsupervised method for the compression, although semi-supervised and hybrid methods can be a natural extension, by considering that we may constrain the problem’s dimensionality—for instance, a first-order approximation of the shape of the PSF. A further step will be the generation of increasingly realistic systematic noise contributions, to design network architectures that can handle them and still allow for detection of the planetary signal. This work opens an exciting path that we believe should be further studied, towards the extraction of periodic signals of binary systems at the milliarcsecond level, directly from times series of satellite imaging data.²

²<https://cosmostatistics-initiative.org/focus/toliman1/>.

Acknowledgements This work was partially produced during the 2nd COIN-Focus: Toliman Event (COIN-Focus # 2) held in Rome, Italy, in November 2019. The COIN-Focus: Toliman participants acknowledge the fundamental support of the Breakthrough Initiatives. The Breakthrough Watch initiative and committee (notably Olivier Guyon, Pete Klupar & Pete Worden) have supported and framed the problem. We also acknowledge input and ideas from people in the wider TOLIMAN collaboration including Ben Pope, Barnaby Norris, Bryn Jeffries, Anthony Horton and others. MB acknowledges financial contributions from the agreement *ASI/INAF 2018-23-HH.0, Euclid ESA mission - Phase D* and the *INAF PRIN-SKA 2017 program 1.05.01.88.04*. EEOI acknowledges financial support from CNRS 2017 MOMENTUM grant under project *Active Learning for Large Scale Sky Surveys*. AKM acknowledges the support from the Portuguese Fundação para a Ciência e a Tecnologia (FCT) through grants SFRH/BPD/74697/2010, PTDC/FIS-AST/31546/2017 and from the Portuguese Strategic Programme UID/FIS/00099/2013 for CENTRA.

References

1. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimization of nonlinear transform codes for perceptual quality (2016)
2. Bessel, F.W.: On the variations of the proper motions of Procyon and Sirius. *Mon. Not. R. Astron. Soc.* **6**, 136–141 (1844). <https://doi.org/10.1093/mnras/6.11.136>
3. Casertano, S., Lattanzi, M.G., Sozzetti, A., Spagna, A., Jancart, S., Morbidelli, R., Pannunzio, R., Pourbaix, D., Queloz, D.: Double-blind test program for astrometric planet detection with Gaia. *Astron. Astrophys.* **482**(2), 699–729 (2008). <https://doi.org/10.1051/0004-6361/20078997>
4. Chen, D.: STEP mission: high-precision space astrometry to search for terrestrial exoplanets. *J. Instrum.* **9**(04), C04040–C04040 (2014). <https://doi.org/10.1088/1748-0221/9/04/c04040>
5. Cheng, T.-Y., Li, N., Conselice, C.J., Aragón-Salamanca, A., Dye, S., Metcalf, R.B.: Identifying strong lenses with unsupervised machine learning using convolutional autoencoder. *Mon. Not. R. Astron. Soc.* **494**(3), 3750–3765 (2020). <https://doi.org/10.1093/mnras/staa1015>
6. Clevert, D.-A., Unterthiner, T., Hochreiter S.: Fast and accurate deep network learning by exponential linear units (elus). In: 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2–4, 2016, Conference Track Proceedings (2016). [arXiv:1511.07289](https://arxiv.org/abs/1511.07289)
7. Cybenko, G.: Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **2**(4), 303–314 (1989). <https://doi.org/10.1007/BF02551274>
8. d’Aspremont, A., Ghaoui, L.E., Jordan, M.I., Lanckriet, G.R.: A direct formulation for sparse PCA using semidefinite programming. In: *Advances in Neural Information Processing Systems*, pp. 41–48 (2005)
9. Dunteman, G.H.: *Principal Components Analysis*, vol. 69. Sage, Thousand Oaks (1989)
10. Dumas, T., Roumy, A., Guillemot, C.: Autoencoder based image compression: can the learning be quantization independent? (2018)
11. ESA SP-1200: The HIPPARCOS and TYCHO catalogues. Astrometric and photometric star catalogues derived from the ESA HIPPARCOS Space Astrometry Mission. ESA Special Publication 1200, January (1997)
12. Frontera-Pons, J., Sureau, F., Bobin, J., Le Floch, E.: Unsupervised feature-learning for galaxy SEDs with denoising autoencoders. *Astron. Astrophys.* (2017). <https://doi.org/10.1051/0004-6361/201630240>
13. Gaia Collaboration, Prusti, T., de Bruijne, J.H.J., Brown, A.G.A., Vallenari, A., Babusiaux, C., et al.: The Gaia mission. *Astron. Astrophys.* **595**, A1 (2016). <https://doi.org/10.1051/0004-6361/201629272>
14. Gaia Collaboration, Brown, A.G.A., Vallenari, A., Prusti, T., de Bruijne, J.H.J., Babusiaux, C., et al.: Gaia data release 2. Summary of the contents and survey properties. *Astron. Astrophys.* **616**, A1 (2018). <https://doi.org/10.1051/0004-6361/201833011>

15. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016). <http://www.deeplearningbook.org>
16. GRAVITY Collaboration, Abuter, R., Accardo, M., Amorim, A., Anugu, N., Ávila, G., et al.: First light for gravity: phase referencing optical interferometry for the very large telescope interferometer. *A&A* **602**, A94 (2017). <https://doi.org/10.1051/0004-6361/201730838>
17. Guyon, O., Bendek, E.A., Eisner, J.A., Angel, R., Woolf, N.J., et al.: High-precision astrometry with a diffractive pupil telescope. *Astrophys. J. Suppl.* **200**(2), 11 (2012). <https://doi.org/10.1088/0067-0049/200/2/11>
18. Hao-ran, Q., Ji-ming, L., Jun-yi, W.: Stacked denoising autoencoders applied to star/galaxy classification. *Chin. Astron. Astrophys.* **41**(2), 282–292 (2017). ISSN 0275-1062. <https://doi.org/10.1016/j.chinastron.2017.04.009>, <http://www.sciencedirect.com/science/article/pii/S0275106217300656>
19. Hyvärinen, A., Oja, E.: Independent component analysis: algorithms and applications. *Neural Netw.* **13**(4–5), 411–430 (2000)
20. Jia, P., Li, X., Li, Z., Wang, W., Cai, D.: Point spread function modelling for wide-field small-aperture telescopes with a denoising autoencoder. *Mon. Not. R. Astron. Soc.* **493**(1), 651–660 (2020). ISSN 0035-8711. <https://doi.org/10.1093/mnras/staa319>
21. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015). <https://doi.org/10.1038/nature14539>
22. Lee, C.-H.: Exoplanets: past, present, and future. *Galaxies* **6**(2), 51 (2018). ISSN 2075-4434. <https://doi.org/10.3390/galaxies6020051>
23. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: *Advances in Neural Information Processing Systems*, pp. 556–562 (2001)
24. Li, W., Xu, H., Ma, Z., Zhu, R., Hu, D., Zhu, Z., Gu, J., Shan, C., Zhu, J., Wu, X.-P.: Separating the EoR signal with a convolutional denoising autoencoder: a deep-learning-based method. *Mon. Not. R. Astron. Soc.* **485**(2), 2628–2637 (2019). ISSN 0035-8711. <https://doi.org/10.1093/mnras/stz582>
25. Lomb, N.R.: Least-squares frequency analysis of unequally spaced data. *Astrophys. Space Sci.* **39**(2), 447–462 (1976). <https://doi.org/10.1007/BF00648343>
26. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 689–696 (2009)
27. Mayor, M., Queloz, D.: A Jupiter-mass companion to a solar-type star. *Nature* **378**(6555), 355–359 (1995). <https://doi.org/10.1038/378355a0>
28. Mousavi, A., Patel, A.B., Baraniuk, R.G.: A deep learning approach to structured signal recovery. In: *Proceeding of 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (2015). <https://doi.org/10.1109/ALLERTON.2015.7447163>
29. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation (2015). CoRR [arXiv:1505.04366](https://arxiv.org/abs/1505.04366)
30. Pearson, K.: On lines of closes fit to system of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **2**, 559–572 (1901)
31. Ranalli, P., Hobbs, D., Lindgren, L.: Astrometry and exoplanets in the Gaia era: a Bayesian approach to detection and parameter recovery. *Astron. Astrophys.* **614**, A30 (2018). <https://doi.org/10.1051/0004-6361/201730921>
32. Resnick, A.: Fourier optics and computational imaging, by Kedar Khare. *Contemp. Phys.* **58**(1), 102–103 (2017). <https://doi.org/10.1080/00107514.2016.1248491>
33. Scargle, J.D.: Studies in astronomical time series analysis. II. Statistical aspects of spectral analysis of unevenly spaced data. *Astrophys. J.* **263**, 835–853 (1982). <https://doi.org/10.1086/160554>
34. Shao, M., Marcy, G., Catanzarite, J.H., Edberg, S.J., Léger, A., Malbet, F., Queloz, D., Muterspaugh, M.W., Beichman, C., Fischer, D., Ford, E., Olling, R., Kulkarni, S., Unwin, S.C., Traub, W.: Astrometric detection of earthlike planets. In: *Astro2010: The Astronomy and Astrophysics Decadal Survey*, vol. 2010, p. 271, January (2009)
35. STScI Development Team: pysynphot: synthetic photometry software package (2013)

36. The Theia Collaboration, Boehm, C., Krone-Martins, A., Amorim, A., Anglada-Escude, G., Brandeker, A., et al.: Theia: Faint objects in motion or the new astrometry frontier (2017). arXiv e-prints [arXiv:1707.01348](https://arxiv.org/abs/1707.01348)
37. Theis, L., Shi, W., Cunningham, A., Huszár, F.: Lossy image compression with compressive autoencoders (2017)
38. Toderici, G., Vincent, D., Johnston, N., Jin Hwang, S., Minnen, D., Shor, J., Covell, M.: Full resolution image compression with recurrent neural networks (2016)
39. Tuthill, P., Bendek, E., Guyon, O., Horton, A., Jeffries, B., Jovanovic, N., Klupar, P., Larkin, K., Norris, B., Pope, B., Shao, M.: The TOLIMAN space telescope. In: Creech-Eakman, M.J., Tuthill, P.G., Mérand, A. (eds.) *Optical and Infrared Interferometry and Imaging VI*, vol. 10701, pp. 432–441. International Society for Optics and Photonics, SPIE (2018). <https://doi.org/10.1117/12.2313269>
40. Unwin, S.C., Shao, M., Tanner, A.M., Allen, R.J., Beichman, C.A., et al.: Taking the measure of the universe: precision astrometry with SIM PlanetQuest. *Publ. Astron. Soc. Pac.* **120**(863), 38 (2008). <https://doi.org/10.1086/525059>
41. van den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks (2016)
42. VanderPlas, J.T.: Understanding the lomb-scargle periodogram. *Astrophys. J. Suppl. Ser.* **236**(1), 16 (2018). <https://doi.org/10.3847/1538-4365/aab766>
43. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
44. Zhang, R.: Making convolutional networks shift-invariant again. In: ICML (2019)
45. Zhao, J.J., Mathieu, M., Goroshin, R., LeCun, Y.: Stacked what-where auto-encoders (2015). CoRR [arXiv:1506.02351](https://arxiv.org/abs/1506.02351), <http://dblp.uni-trier.de/db/journals/corr/corr1506.htmlZhaoMGL15>

Comparison of Outlier Detection Methods on Astronomical Image Data



Lars Doorenbos, Stefano Cavuoti, Massimo Brescia, Antonio D’Isanto,
and Giuseppe Longo

Abstract Among the many challenges posed by the huge data volumes produced by the new generation of astronomical instruments there is also the search for *rare* and *peculiar* objects. Unsupervised outlier detection algorithms may provide a viable solution. In this work we compare the performances of six methods: the Local Outlier Factor, Isolation Forest, k-means clustering, a measure of novelty, and both a normal and a convolutional autoencoder. These methods were applied to data extracted from SDSS stripe 82. After discussing the sensitivity of each method to its own set of hyperparameters, we combine the results from each method to rank the objects and produce a final list of outliers.

Preprint version of the manuscript to appear in the Volume “Intelligent Astrophysics” of the series “Emergence, Complexity and Computation”, Book eds. I. Zelinka, D. Baron, M. Brescia, Springer Nature Switzerland, ISSN: 2194-7287.

L. Doorenbos

Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence,
University of Groningen, Nijenborgh 9, 9747AG Groningen, The Netherlands
e-mail: larsdoorenbos@msn.com

S. Cavuoti (✉) · G. Longo

Department of Physics, University of Naples Federico II, Strada Vicinale Cupa Cintia, 21,
80126 Naples, Italy
e-mail: stefano.cavuoti@gmail.com

G. Longo

e-mail: longouniversita@gmail.com

S. Cavuoti · M. Brescia

INAF - Astronomical Observatory of Capodimonte, Salita Moiariello 16, 80131 Naples, Italy
e-mail: massimo.brescia@inaf.it

A. D’Isanto

Astroinformatics Group, Heidelberg Institute for Theoretical Studies, Schloss-Wolfsbrunnengasse
35, 69118 Heidelberg, Germany
e-mail: antonio.disanto@h-its.org

1 Introduction

As it has been amply discussed in the literature (cf. [1, 2]) the discovery of most new astronomical objects and phenomena can be interpreted in terms of either an enlargement or better sampling of the observable parameter space (OPS) or in terms of an increased capability to extract patterns and trends. An example of the first type being the discovery of quasars (due to the opening of a new dimension defined by the radio fluxes), while an example of the second type can be the discovery of the fundamental plane of elliptical galaxies [3]. So far, our capability to explore the OPS has been strongly limited by the human factor, i.e. by the difficulties encountered by the human brain in finding patterns or outliers in OPS with more than three dimensions.

The OPS defined by modern panchromatic multi-epoch surveys has a very large amount of dimensions, with more than several hundreds of parameters measured for each object. While on the one hand it poses large computational problems, on the other it offers unprecedented possibilities in moving from the so called “serendipitous discoveries” to a more systematic search for the *rare* and of the *unknown*.

This aspect, by considering also the huge amounts of data collected by modern sky surveys, implies that it is no longer feasible to go through all observations by hand to retrieve and flag the interesting objects. The unbiased and automatic data-driven capability of methods based on the paradigms of Machine Learning (ML), to explore the OPS and to extract hidden correlations among data, may achieve a realistic possibility to perform a manageable selection of peculiar objects, most likely to be interesting. In astrophysical terms, the peculiar objects, or outliers, are mostly characterized by their intrinsic singularity within a data distribution. Satellites, imaging artifacts, interacting galaxies, lensed objects or other unexpected entities are just a few of examples of outliers that could emerge from a distribution. ML methods may efficiently analyze a dataset and isolate such peculiar candidates, by proceeding in two different ways: (i) supervised, i.e. by inferring knowledge of the nature of objects from a limited number of samples during training and by generalizing the obtained insights in order to recognize outliers in new data; (ii) unsupervised, i.e. by self-organizing their internal structure, just driven by data themselves, in order to recognize commonalities among data and isolating potential outliers as single entities within the parameter space. The supervised approach has an important downside, since it limits the capability of a method to recognize only expected (hence known) peculiarities, i.e. those provided within the training set. Therefore, we preferred to proceed in an unsupervised way and to perform a totally unbiased data exploration.

Among the varieties of unsupervised methods available, by considering the known heuristic process at the base of any ML approach, we selected a set of models and performed a comparison of their performance in extracting outliers. These methods were: (a) Local Outlier Factor [4], (b) Isolation Forest [5], (c) k-means [6], (d) a recently introduced novelty measure [7] (hereinafter named modified novelty or MN), and (e) both a normal and a convolutional autoencoder [8, 9].

The comparison was driven not only in terms of the efficiency and accuracy to extract peculiar objects, but also regarding the capability to recognize common objects as potential outliers, trying to investigate the performance in case of decision discrepancies.

In this work these methods were used to detect the most abnormal points in the training dataset. These can be singular observations, or small clusters of relatively normal data, lying far away from the normal dataset distribution within the feature space. Both types were considered as outliers. Alternatively, one could consider the training dataset as “normal” and use the trained models for the detection of anomalies in a different test set by searching for the largest deviations from the observations from the training set [10]. This approach will be explored in a forthcoming project.

Most methods suggested in the literature for the detection of outliers in sky surveys, focus on spectroscopic or tabular data. Examples include Chaudhary et al. [11], who proposed a new outlier detection method, using five-dimensional SDSS data. In Fustes et al. [12] self-organizing maps were used to find outlying spectra. Giles et al. [13] used a variant of the DBSCAN clustering algorithm to detect outliers in derived light curve features. The work more pertinent to that presented here is the one proposed by Baron et al. [14], where an unsupervised Random Forest was used to detect the most outlying galaxy spectra within the SDSS survey and its results were compared with a standard Random Forest, a one-class Support Vector Machine and an Isolation Forest. The key difference is that in the present work the outliers are searched within images, instead of spectra, and including multiple kinds of objects in the dataset, instead of just galaxies.

The remainder of this work is organized as follows. We begin in Sect. 2 with an exploration of the dataset used and its preparation procedure. Then, in Sect. 3 we describe and assess the performance of the various selected methods. The results are compared and analyzed in Sect. 4, followed by a description of the model proposed to find outliers in new data in Sect. 4.1. Finally, we draw the conclusions and mention some perspectives for future work.

2 The Data

As template data set for our experiments we used a subset of data extracted from the Sloan Digital Sky Survey (SDSS, [15, 16]) Data Release 9 (hereafter SDSS-DR9). This subset was originally assembled to estimate the photometric redshifts without any pre-classification in (cf. [17]). The extraction was performed randomly, trying to maintain the object class types as much as possible balanced in terms of their quantities, thus minimizing any risk of selection effect induced by the overabundance of a particular class.

It consists of 200, 000 stars, 200, 000 galaxies and 185, 718 quasars, represented by a OPS composed by the five bands (*ugriz* filters).

2.1 Data Exploration and Preprocessing

Prior to start the experiments, we performed an analysis of the extracted data, in order to identify possible anomalies or artifacts that could affect the outlier detection performances.

A first consideration is that the spectroscopic and photometric classifications of the objects composing the SDSS dataset do not always match. For example, the SDSS object J145416.34+212953.9¹ is a galaxy according to the spectroscopy, but is listed as a star by the photometric classification. An example of the reverse is the SDSS object J170616.32+242609.2,² spectroscopically classified as a star, but resulting as a galaxy from the photometry. However, since this work is based on data for which the spectroscopy was always available, we refer to the spectroscopic classification in all experiments.

Furthermore, a number of duplicate entries in the dataset were found: a couple of identical stars, two galaxy couples and 393 couples of quasars. These duplicates were removed.

Another example of irrelevant anomalies was the presence of strong straight lines passing through the r band, as for instance caused by a satellite or an asteroid. In order to remove such events from the dataset, a heuristic procedure was applied, by setting a magnitude threshold in some band, able to reveal the occurrence of such events. By looking at all images where the maximum value in the r band was at least 1.75 times higher than the highest value in all other bands, we found 106 images affected by the straight line (considered as true positives), and 3 images with a false anomaly event (called false positives). The true positives found with this criterion were removed from the data. An example is given in Fig. 1a.

By applying the same procedure on the other bands (u, g or i) with a threshold of 2, an additional 88 images were detected. Only four of these had a line present, all in the i band, similar to those found in the r band. Such a line was visible only for one object in the object explorer, see Fig. 1b. These objects were not removed from the dataset.

Furthermore, we noticed that the five bands do not span the same range of pixel values, see Table 1. Therefore, we decided to standardize all bands, using a scaling factor, able to obtain images with zero mean and unitary standard deviation [18]. In the cases where the dataset was split into a training and testing set, the standardization was performed using the mean and standard deviation of the training set as reference, in order to align both training and test distributions.

Finally, we were interested in detecting all occurrences of double sources. This was done by looking at the center of mass within the images, averaged on all bands. Figure 2 shows a histogram of the distances between the center of mass and the center of the images. There is no obvious cut-off which could separate double sources and

¹<http://skyserver.sdss.org/dr12/en/tools/explore/summary.aspx?ra=223.568087062725&dec=21.4983280618327>.

²<http://skyserver.sdss.org/dr12/en/tools/explore/summary.aspx?ra=256.568027662234&dec=24.4358907062636>.

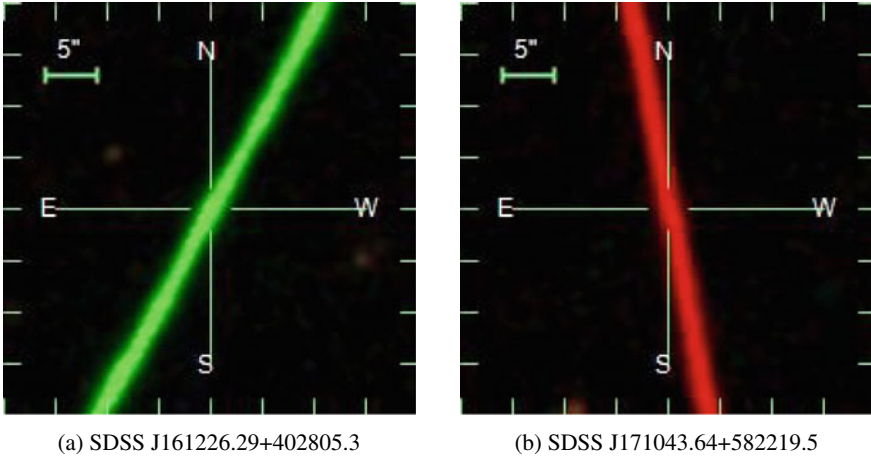


Fig. 1 Examples of straight lines within the images, mostly due to Asteroids/satellites

Table 1 Mean and standard deviations of the pixel values among the different bands

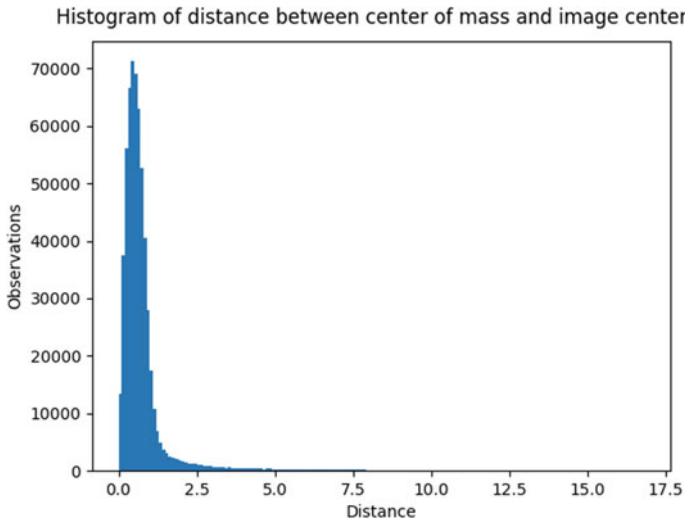
Filter	Mean	Standard deviation
u	0.028	0.387
g	0.076	0.769
r	0.133	1.290
i	0.179	1.628
z	0.224	2.183

single sources. Therefore, we performed a searching within SDSS archive, in order to find all candidate sources to be paired. We used a cross-match threshold of 15 pixels, corresponding to 0.6 arcsec. We found 48, 814 galaxies, 26, 126 quasars and 33, 130 stars with at least one more object within this distance in our dataset.

This information was not used to reject these sources from the dataset, but was taken into account in the post-processing phase in order to recognize this special category of objects.

3 Outlier Detection Methods

The outlier detection methods considered in this work can be broadly grouped in two categories: (i) those applied on a dataset whose dimensionality is pre-reduced by Principal Component Analysis (Local Outlier Factor, Isolation Forest, K-Means, Modified Novelty Measure), and (ii) deep learning models, for instance standard and convolutional autoencoders.



(a) Whole dataset

Fig. 2 Histogram of the distances between the image center of mass and image center

3.1 PCA-Based Detection Methods

Before applying the outlier detection metrics on the pixel values, we decided to preliminarily reduce the dimensionality of the images. We used the sklearn implementation³ of the widely used Principal Component Analysis (PCA) for this [19].

Two common approaches, for choosing the number of components to reduce the input data, are the elbow method [20] and explaining some set amount of variance [21].

As there is no clear elbow present in the variance graph of Fig. 3a, we chose to explain a set amount, namely a 90% of the variance, which comes down to 14 components, see Fig. 3.

As stated in Sect. 3, we used the features extracted by the PCA as input for a series of outlier detection methods, for instance the Local Outlier Factor, Isolation Forest, K-Means and Modified Novelty Measure, described in the following sections.

3.1.1 Local Outlier Factor

The Local Outlier Factor (LOF) is a measure for how isolated is an object with respect to its neighbourhood [4].

Let the k -distance kd of an object be the distance to its k th nearest neighbours, and let $N_k(p)$ be the set of those neighbours. The reachability distance of an object

³<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>.

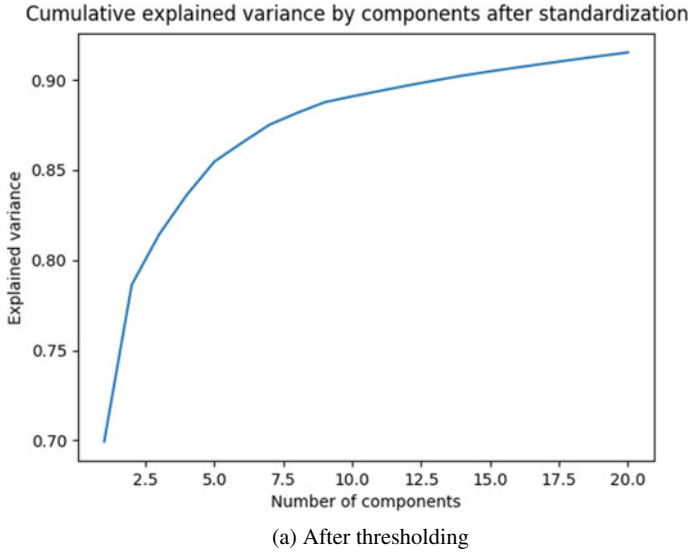


Fig. 3 Explained variance by number of PCA components, see Sect. 3

p with respect to another object o , when considering k nearest neighbours, is given by

$$rd_k(p, o) = \max\{kd(o), d(p, o)\},$$

where $d(p, o)$ is the distance between points p and o . Note that this is the actual distance between p and o , if p does not belong to the k nearest neighbours of o , and the k -distance of o if it does. Now define the local reachability density of p considering k nearest neighbours as

$$lrd_k(p) = 1 / \left(\frac{\sum_{o \in N_k(p)} rd_k(p, o)}{|N_k(p)|} \right).$$

The LOF score for an object p is then calculated by

$$LOF_k(p) = \frac{\sum_{o \in N_k(p)} lrd_k(o)}{lrd_k(p)|N_k(p)|}.$$

Intuitively, when the density around point p is lower than the average of the densities around its k neighbours, this fraction will result in a value above 1, in which case p can be considered an outlier above some threshold.

The most important hyperparameter to set when calculating the LOF is the number of neighbours used. In their paper [4], the authors recommended taking a range. The lower bound should be seen as the minimum number of samples a cluster has to

contain, such that other samples can be local outliers relative to this cluster. The upper bound can be instead considered the maximum number of close samples that can potentially be local outliers. The LOF value for each datapoint is computed for each value in this range and the maximum is taken as the final value.

The choice of the best bound in our case, by considering the reduced parameter space after having applied the PCA, is not obvious. Therefore, we refer to the range introduced by the authors of the method, used in an analogous case of a large dataset, namely 30–50.

As the current scikit learn implementation⁴ does not support a range but only a scalar value for this hyperparameter, we computed the scores for the parameter values $k = 30, 35, 40, 45, 50$ instead of at every integer value between 30 and 50 to save computing time.

The matrix shown in Fig. 4 provides information about the sensitivity to the hyperparameter setting. Each cell in this matrix shows what fraction of objects obtained by using the hyperparameter settings denoted in the column name are also present in results obtained by using the hyperparameter settings of the row name. For example, out of the 2,507 objects scoring above 5 standard deviations when using the range [30, 50], approximately 80% also score above 5 standard deviations when using the range [10, 30]. This format is used throughout this work.

There is more overlap between the outliers scoring above 5 standard deviations from the mean, for the different hyperparameter ranges used, than for the outliers scoring between 3 and 5 standard deviations. All in all, apart from the [10, 30] range, the results are quite insensitive to the parameter settings, with the overlap mostly reaching over 80% between 3 and 5 standard deviations and over 90% above 5 standard deviations.

3.1.2 Modified Novelty Measure

A modified novelty (MN) measure originally designed for collider physics proposed, in Hajer et al. [7], as an improvement to the LOF, is defined by

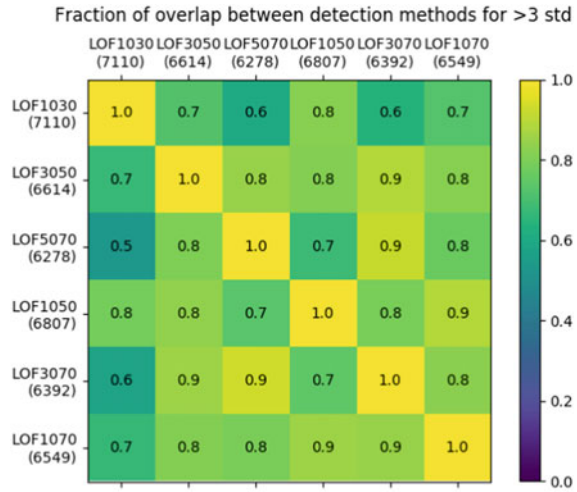
$$\frac{d_{test}^{-m} - d_{train}^{-m}}{d_{train}^{-m/2}}$$

where d_{test} denotes the average distance from a point to its k nearest neighbours in the testing set, d_{train} the average distance to its k nearest neighbours in the training set, and m the dimensionality.

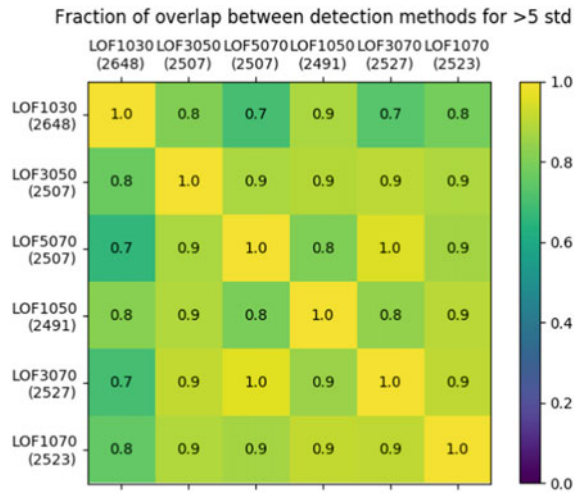
In contrast to the LOF, which only considers one point at a time, this measure takes into account the clustering of testing data [7]. While we do not split our dataset for the calculation of the LOF, and as such testing data clustering is not an issue, it

⁴<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html#sklearn.neighbors.LocalOutlierFactor>.

Fig. 4 LOF results using different parameter settings. The name indicates the hyperparameter range used, e.g. LOF1030 used a range of [10, 30]. The size of the results is shown between brackets



(a) Between 3 and 5 standard deviations from the mean



(b) Above 5 standard deviations from the mean

is still interesting to see how the results for the two measures differ. When applying the methods to novel data, this benefit will come into play.

The one hyperparameter to consider is the number of nearest neighbours used for the calculation of the metric. The authors do not provide guidelines on how to choose the hyperparameter for this method. As it serves a similar purpose to the hyperparameter used for LOF, we chose a value of 40, the average of the range of values used there.

There is no library currently implementing this method for Python. In contrast to the previous methods, this measure assigns a score very close to zero to its outliers, while the most normal points can get a very negative score. As a result it is not the outliers that score above 3 standard deviations but the most normal objects instead.

3.1.3 Isolation Forest

The Isolation Forest (IF, [5]) method is based on the idea that an outlier will be separated from normal datapoints at an early stage when randomly partitioning the dataset. This is implemented by repeatedly partitioning a sampled subset of the dataset with a random value between the minimum and maximum value for a random feature until all data points are isolated, resulting in a tree structure. This process is repeated multiple times on different samples, creating a forest, and the mean path length to a data point over all trees in the forest is the outlier score for that point.

The two most important hyperparameters of IF are the number of trees in the forest, and the number of randomly sampled data points to use for the construction of each tree. In their paper the authors suggest using 100 trees, as the path lengths usually converge well before this point, as well as using a sub-sampling size of 256 for higher dimensional datasets [5].

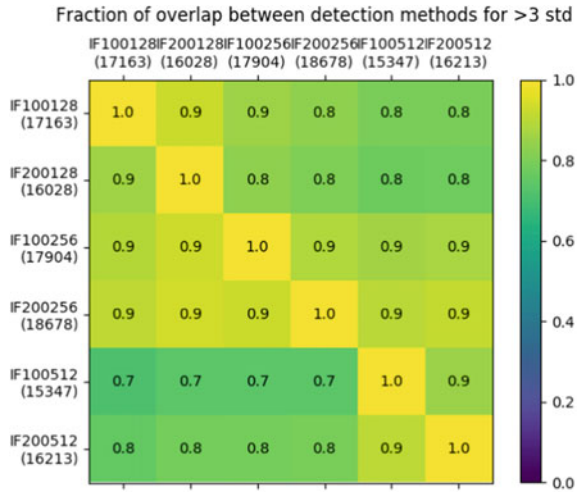
In Fig. 5 we see the sensitivity of the detections to these hyperparameters. A logical consequence of using more samples is that the size of the set of objects with higher outlier scores becomes larger, as with bigger trees there will be larger differences between the average path lengths to isolation for a normal object and an outlier. For 100 trees with a sample size of 128 the consequence is that no object scores above 5 standard deviations and 17,163 between 3 and 5. On the other hand, with 100 trees of size 512 this is shifted to 3, 288 and 15, 347. This makes it more difficult to compare the sets with different sample sizes, especially above 5 standard deviations.

Between 3 and 5 standard deviations the results are very similar for sample sizes of 128 and 256. For a sample size of 512 there is less overlap. This can be explained by the fact that, due to the larger trees, the outliers scoring between 3 and 5 standard deviations, when using the lower sample sizes, are now scoring too high to be included in this set and are now scoring above 5 standard deviations. Note that the trees are not built from the same random samples, so the small differences in outliers can be caused by which data points were used to build the trees, instead of what hyperparameter settings were chosen.

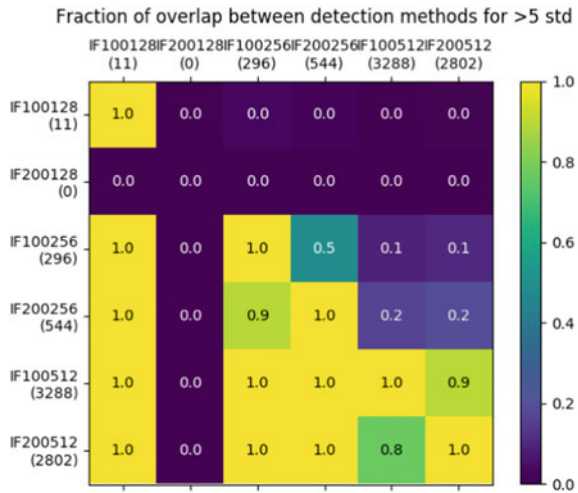
3.1.4 Clustering

Another approach to detecting outliers is by grouping the dataset into clusters and using the distance between objects and their closest cluster center as the outlier score [22]. One of the most commonly used clustering algorithms is k-means (KM, [6]). The only hyperparameter of this algorithm determines how many cluster prototypes will be generated. Points are then assigned to the closest prototype, after which the

Fig. 5 Overlap between the IF results using different parameter settings. The name indicates the hyperparameters used, e.g. IF100256 used 100 trees with 256 samples each. The size of the results is shown between brackets



(a) Between 3 and 5 standard deviations from the mean



(b) Above 5 standard deviations from the mean

location of the prototypes is recomputed as the mean of all points assigned to it. This is repeated until convergence [6].

As there are three types of objects in our dataset the logical choice for k is three. We can verify this choice by plotting the quantization error against the number of clusters. At a value of three there is a slight kink in the line, suggesting that this would be a good choice for k (Fig. 6). We noticed that the absence of a true elbow in the curve gives a first indication that our data is not partitioned into well-defined clusters using KM.

Fig. 6 Normalized quantization error by number of clusters used

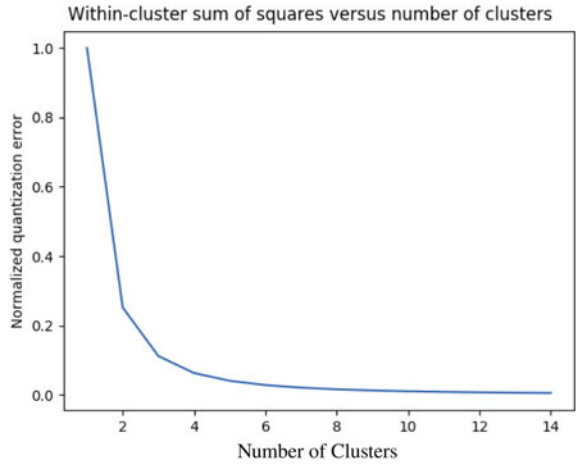


Table 2 Cluster membership with $k = 3$

	Members
0	47,883
1	517,110
2	13,097

The clusters are not equally populated, shown in Table 2, and each cluster does not represent each type of object. Visualizing the prototypes by reverting the PCA gives the results shown in Fig. 7. The defining difference between the clusters seems to be the maximum value. This explains the uneven cluster membership as the prototype with the lowest value, the second cluster, has by far the largest number of objects associated with it, and by far the largest number of objects in our dataset have a low maximum value (Fig. 12b).

The distance metric used when determining the outliers after clustering is another hyperparameter. A standard choice is the L_k norm for some value k , defined by

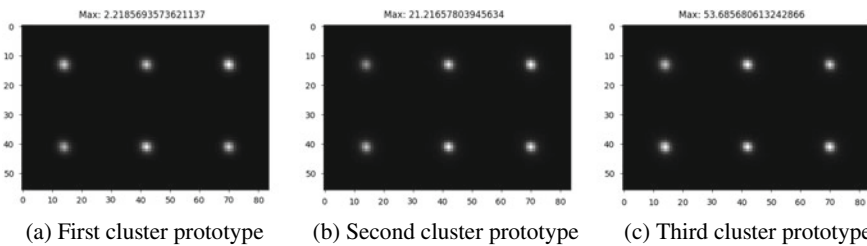


Fig. 7 Cluster prototypes when using $k = 3$ and Euclidean distance

$$L_k(x, y) = \sum_{i=1}^d (\|x^i - y^i\|^k)^{1/k}$$

When using the L_k norm for KM with high-dimensional data, often a lower k is preferred [23]. As the current scipy implementation only supports integer values for k ,⁵ we use the Manhattan distance ($k = 1$).

We can still gain insights into the sensitivity of the results to the choice of these hyperparameters by looking at the overlap matrix, found in Fig. 8. Especially the lower scoring outliers are different when using a different value for k . Where the maximal values of the prototypes for $k = 3$ have rounded values of 2, 21 and 54, for $k = 4$ we have 2, 14, 37 and 63, and for $k = 5$ they are 1, 9, 25, 49 and 74, the morphologies for the prototypes with higher k are very similar to the ones in Fig. 7. Apparently the KM is producing a clustering mostly based on the maximum value of the pixels, rather than a proper partition. They are unable to take another shape as the center of mass is almost uniformly distributed around the center, as illustrated in Fig. 9. A convolutional approach might be favorable to deal with this problem.

3.2 Autoencoders

Autoencoders are a fundamental architecture for unsupervised outlier detection using deep learning [24]. Autoencoders consist of two mirrored parts, an encoder and a decoder. The idea is that the encoder tries to learn a representation of the input in a lower dimensionality, while the decoder rebuilds the original input from the smaller representation. Ideally the autoencoder as a whole behaves the same as the identity function.

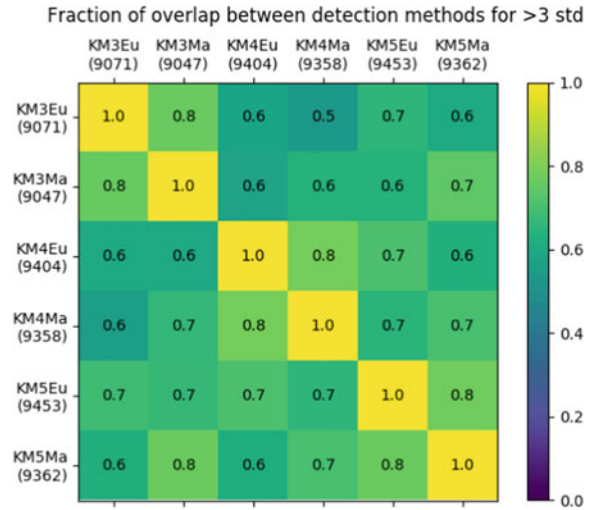
Autoencoders can be used in two ways to detect outliers. The first way is to feed an input to the trained model and look at the difference between the output and the input. This difference is called the reconstruction error. The bigger this error is, the more outlying the input is with respect to the data the autoencoder was trained with. Secondly, one can use the lower dimensional representation obtained after the encoder as a new feature vector to apply other outlier detection methods on. We use the first method in this work.

3.2.1 Standard Autoencoder

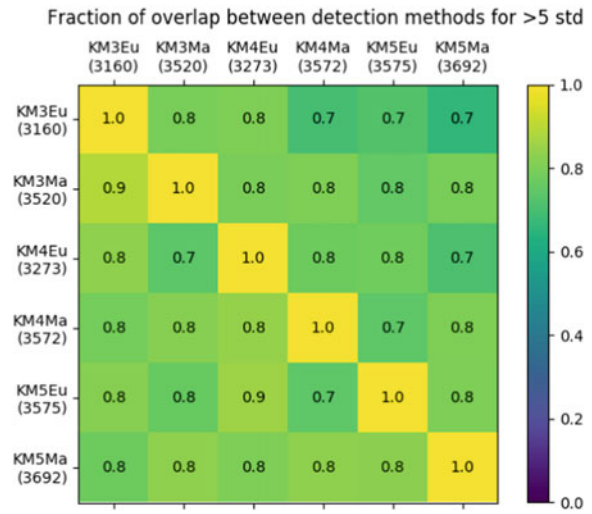
The simplest autoencoders (AE, [8]) treat the input as a one-dimensional feature vector with in our case $28 * 28 * 5 = 3920$ elements (five images, one for each band of 28×28 pixels). As we are dealing with image-like data, and tuning and training autoencoders is a time-consuming process, we focused on tuning the convolutional

⁵<https://docs.scipy.org/doc/scipy/reference/spatial.distance.html>.

Fig. 8 Overlap between the KM results using different parameter settings. The name indicates the hyperparameters used, e.g. KM3Eu used 3 clusters and measured the Euclidean distance. The size of the results is shown between brackets



(a) Between 3 and 5 standard deviations from the mean



(b) Above 5 standard deviations from the mean

autoencoder as it should be better suited to our dataset. For this reason the normal autoencoder is included in the visual comparisons in Sect. 4, to get an idea if the results are different from other variants, but the results are not used in the final comparisons.

The model we used for the comparison reduces the 3,920 dimensional input to 128 dimensions with 3 hidden layers for both the encoder and decoder part, see Table 3 for the details.

Fig. 9 Image pixels colored by center of mass occurrence

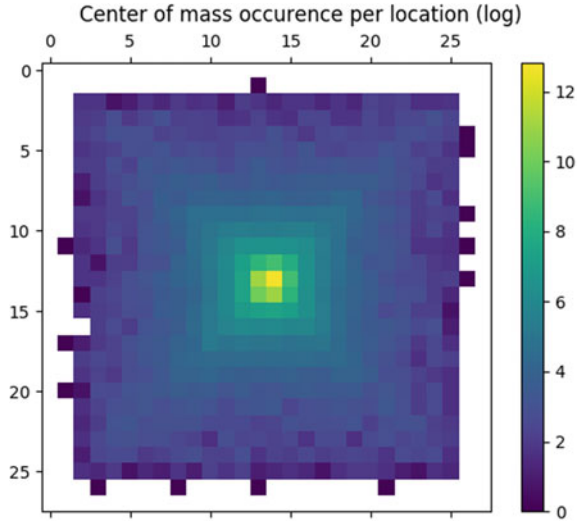


Table 3 Standard autoencoder architecture

Layer	0	1	2	3	4	5	6
Number of neurons	3,920	784	256	128	256	784	3,920

3.2.2 Convolutional Autoencoder

Convolutional autoencoders (CAE) do not treat each individual measurement as a separate entity, but instead use small filters to extract and detect local image features [9].

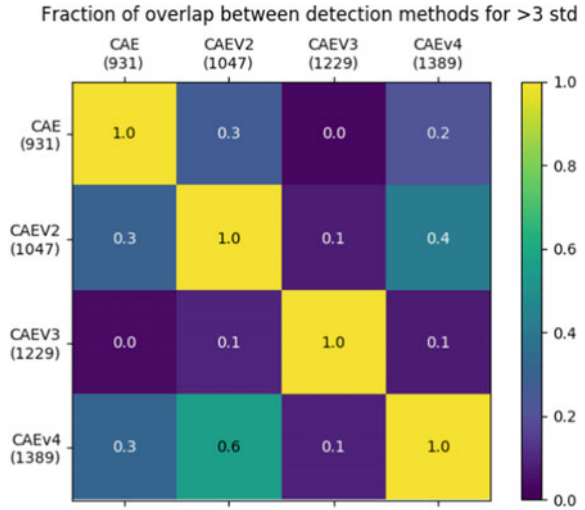
The performance of neural networks such as autoencoders can be very sensitive to the hyperparameters used. Unlike the previous PCA-based methods, the training of the CAE is a supervised procedure, enabling the use of hyperparameter optimization algorithms such as random search [25]. We used the hyperparameters of the best performing network obtained from the random search to train on the whole dataset and used this for the comparison in the next chapter. The input is reduced by 2 hidden layers to a 7×7 image. Each hidden layer consists of a convolutional layer with a filter size of 3×3 , followed by a max pooling layer for the encoder and an upsampling layer for the decoder, both of size 2×2 . A learning rate of 0.075 was used, linearly decaying to 0 over the epochs, finally we used a stride of 1×1 (Table 4).

In Fig. 10 we look at the overlap in outliers when training a CAE with different values for the number of hidden layers, the number of convolutions in each layer, the learning rate and the batch size. The first 3 versions are trained with hyperparameters tested in the random search. The 4th version uses the same settings as version 2 but

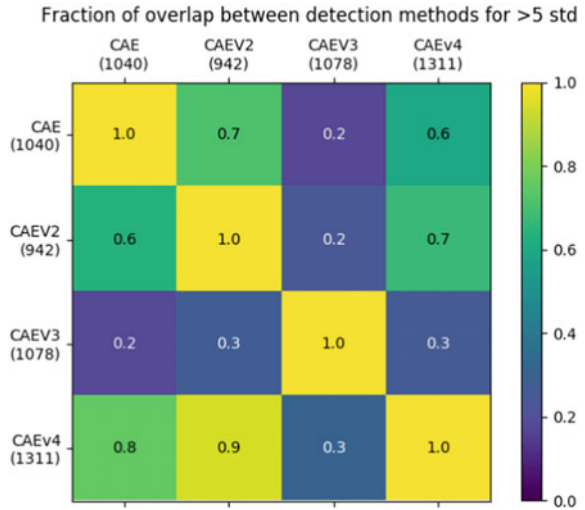
Table 4 CAE architecture

Layer	0	1	2	3
Number of convolutions	48	24	24	48

Fig. 10 Overlap between the CAE results using different parameter settings. The size of the results is shown between brackets



(a) Between 3 and 5 standard deviations from the mean



(b) Above 5 standard deviations from the mean

is trained on the whole dataset and is the one used for our comparisons. As the CAEs were trained using a different training/testing split, at least part of the difference will be because of testing data being inherently more likely to have a higher reconstruction error than the training data.

The third iteration had a total reconstruction error more than 6 times higher than the first two (± 0.033 versus ± 0.20). Its outliers are also very different from the others, never reaching an overlap above 30%. While the first two had a similar total reconstruction error, the outliers differ, especially between 3 and 5 standard deviations, but it is difficult to determine how much of this can be attributed to the training/testing split.

4 Discussion

Through an exploration of the input dataset, composed by the image cutouts extracted from the SDSS, an analysis at pixel level of images was performed. By projecting on a histogram the distribution of pixels with the maximum value among all involved bands (shown in Fig. 12b), it appeared a peak at ~ 148 . Looking at the sources around the peak, these were recognized as objects located nearby to a very bright star. Two examples of such objects are given in Fig. 11.

When we look at the maximum value of pixel for each object (over all bands), we get the histogram shown in Fig. 12b.

Since such kind of objects were affected by the presence of the much brighter second source, they were considered as unreliable and removed from the dataset. We decided to cut the images with a maximal value higher than 100 from the dataset.

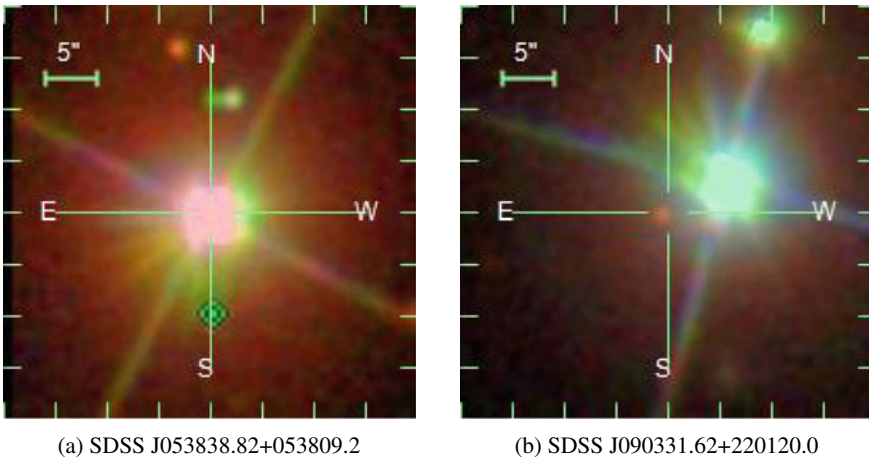


Fig. 11 Two examples of images showing the highest values in terms of pixel's intensity among all SDSS bands. It is possible to see a set of sources close to a very bright star nearby

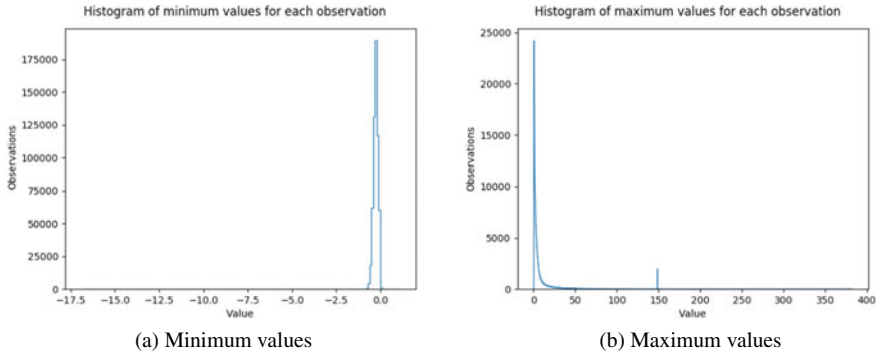


Fig. 12 Histogram the distribution of pixels with the maximum value among all SDSS bands

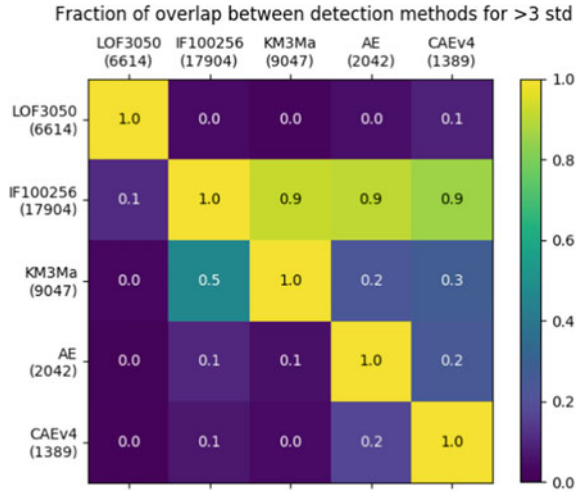
This threshold was chosen because, when running the outlier detection methods on the whole dataset, only five outliers were detected with a maximum value below 50, none with a maximum value between 50 and 100 and the rest with a maximum value over 100. Of the latter, all objects are child sources close to another, very bright object. Applying this threshold we removed 300 galaxies, 97 quasars and 7, 232 stars, leaving a total of 578, 090 objects. The histogram of minimal values shown in Fig. 12a does not show any unexpected peak.

In terms of outlier detection, we decided to compare the results from different methods, by considering the overlap of objects scoring above 5 standard deviations from the mean and between 3 and 5 standard deviations from the mean for the different outlier detection methods. This is shown in Fig. 13. Note that the modified novelty measure is not included in this comparison here, due to the absence of outliers scoring above 3 standard deviations for this method.

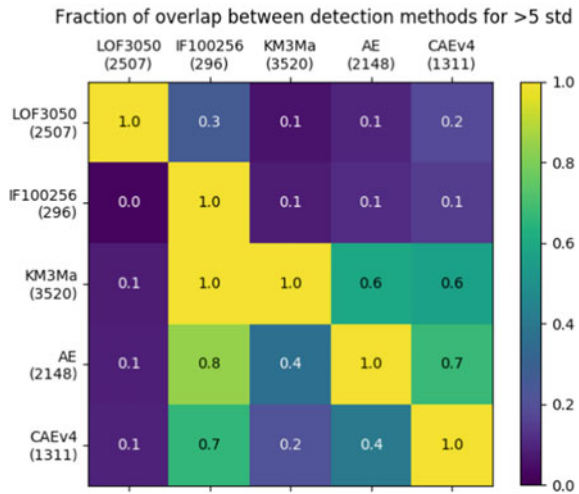
We explored the possibility to visualize the results by reducing the dimensionality with two different methods, PCA and t-SNE [26] but without noticing any clear clusters. What we saw from both the visualization and the correlation matrices is that the LOF gives very different results than the other methods. The other PCA-based methods and the normal autoencoder produce more similar results, especially when looking at the visualization. The CAE, while not as different as the LOF, does have less overlap with the other methods.

We found 63 objects that score above 5 standard deviations from the combination of LOF, IF, KM and CAE methods. They consist of 6 galaxies, no quasars and 57 stars. Of the 57 stars, 35 are observations with 2 bright stars present in the frame. While these are definitely outliers from a data perspective, as they have two relatively bright objects, they are not the most interesting from an astronomical point of view. There are 4 detections with triple stars that, although rarer, are equally uninteresting. The rest are either big, blue single stars or objects labeled as galaxies very close to bright stars. The 6 galaxies vary from the reflection of a bright star in the telescope to irregular starburst regions, but do not share common traits. Four of the weirdest objects can be found in Fig. 14d.

Fig. 13 Overlap between the different outlier detection methods. The size of the results is shown within brackets



(a) Between 3 and 5 standard deviations from the mean



(b) Above 5 standard deviations from the mean

Out of the 63 objects, 12 have no other sources in the image within 0.6 arcsec from the object center, respectively, 9 stars and 3 galaxies. One galaxy is an artifact caused by a reflection in the telescope, while the other 2 are either close by or interacting with a nearby galaxy further away than the threshold. Out of the 9 stars, for 2 of them the bands are moved, 1 is close to an artifact, while the other 6 are relatively big, blue stars.

When looking at objects that score over 5 standard deviations for some methods and between 3 and 5 for others, the number of objects found does not increase as

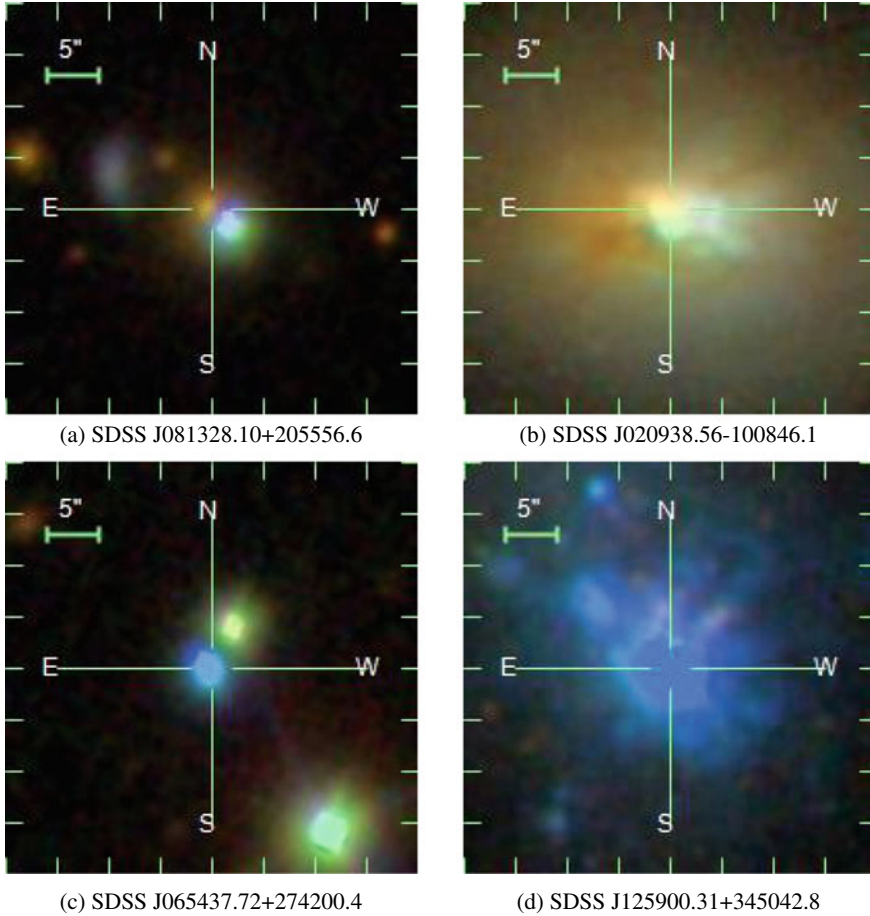


Fig. 14 4 objects scoring above 5 standard deviations for LOF, IF, KM and CAE

expected. This is again caused by the LOF way of finding mostly different outliers, compared to the other 3 methods, as shown in Table 5. None of the filtered asteroids showed up in the intersections of the combination of the 4 methods.

Splitting the results into the three classes of objects (Stars, Galaxies and QSOs) present and looking at their overlaps, could give us an idea if there is one method that filtered the outlying galaxies and quasars out of the intersection. The correlation matrices are found in Fig. 15 for galaxies, Fig. 16 for quasars and Fig. 17 for stars. The CAE outliers consist of around 80% stars in both categories, followed by around 10% galaxies and slightly less quasars. The other methods except LOF follow the same ordering. The LOF finds more quasars than galaxies, and both categories are better represented in the results, with stars only making up around 50% of the objects scoring between 3 and 5 standard deviations.

Table 5 Number of objects that score above 5 for some models and between 3 and 5 standard deviations for other models when considering those given in the row name. For example, there are 153 objects that, when taking into account all four models, score above 5 standard deviations for two of them, and between 3 and 5 standard deviations for the others. The number of single source objects is reported between brackets

Number of methods with $score > 5\sigma$	4	3	2	1	0
Number of methods with $5 \geq score > 3\sigma$	0	1	2	3	4
Methods	Number of objects				
LOF/IF/KM/CAE	63 (12)	134 (39)	153 (35)	112 (24)	44 (6)
IF/KM/CAE	–	194	593	890	382
LOF/IF/KM	–	75	195	350	225

The fact that all these stars are detected as outliers again indicates that an high maximum value of pixels, among all SDSS bands, make for strong outliers. In Fig. 18 we see that stars are the brightest class present in the dataset, followed by galaxies, while quasars have the lowest intensity. This is in line with the objects with high outlier scores for all methods except LOF.

Similar to using the SDSS database to find double objects, this split into the object classes cannot be performed on new datasets and as such is only used to understand the results.

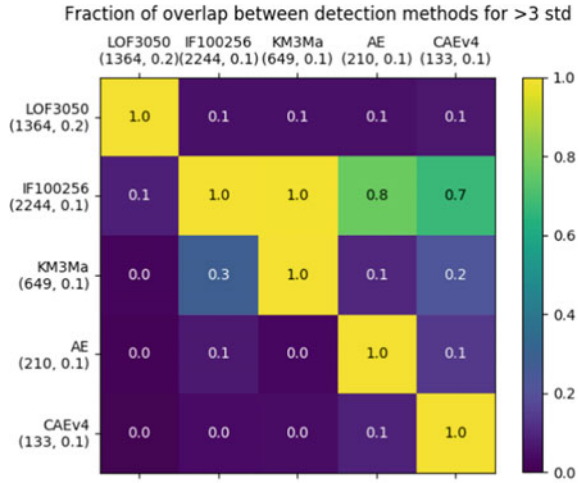
By looking at the strongest scoring objects per method we can get a general idea of what kind of objects the methods consider especially outlying.

The highest scoring objects for the LOF are the most varied. The top 250 objects include 19 galaxies and 50 quasars. The lowest maximal image value is 0.15 and the highest 94.33, and there are 108 single sources. There are 24 asteroids in the top 250 that are filtered from the results as discussed in Sect. 2, while none of the others have a single one of these in their 250 highest scoring objects. There are 8 objects with an entirely flat band in this list, whereas none of the others score such an object this highly.

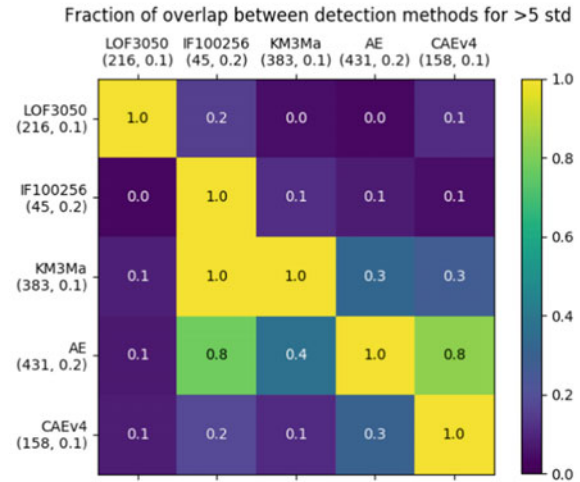
The strongest IF outliers are mostly very bright objects. The lowest maximal intensity is 4.10, which is an entirely blue field due the reflection of a star, but where the LOF has 27 objects with a maximal value below 1, the IF has 27 objects below 50. There are 10 objects with a maximum value of exactly 50.93, all big blue stars. The IF is the strongest filter of quasars, with only 3 quasars scoring in the top 250.

The KM results on the other hand contain the most single sources, 187 out of 250. It also has the largest number of stars, namely 210. A large number of these are big, blue stars instead of the double sources we see more often for the other methods. 15 of these have a maximal value of exactly 50.93, of the in total 17 of these that exist in our dataset. While not as extreme as IF, the KM results are also mostly very bright objects. with only 3 of them having a maximal intensity below 25.

Fig. 15 Overlap between the different outlier detection methods on galaxies. The size of the results and the fraction of the total results this constitutes is shown within brackets



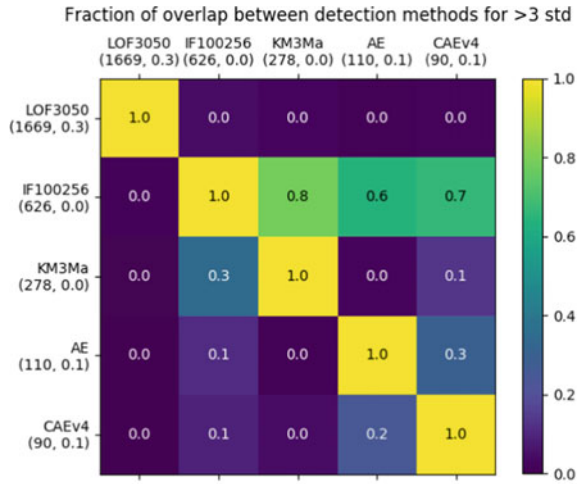
(a) Between 3 and 5 standard deviations from the mean



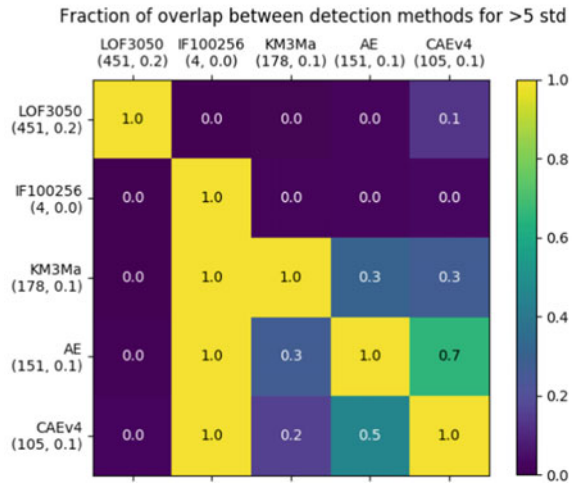
(b) Above 5 standard deviations from the mean

Finally, for the CAE we again have a more diverse list with 4 objects having a maximal value below 1 and the different types of objects better represented, even though not as varied as the LOF results. While the other 3 methods do not have a single object scoring in the top 250 with a center of mass over 8 pixels away from the image center, there are 58 objects that do so for the CAE. These are all relatively faint objects with a very bright star at the image edges. This method thus seems to consider double sources as the most outlying.

Fig. 16 Overlap between the different outlier detection methods on quasars. The size of the results and the fraction of the total results this constitutes is shown within brackets



(a) Between 3 and 5 standard deviations from the mean

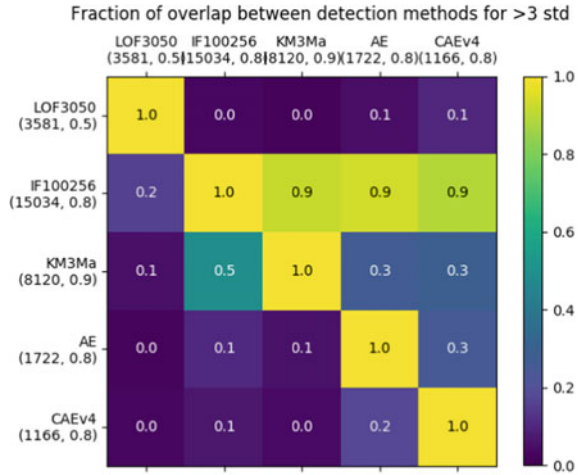


(b) Above 5 standard deviations from the mean

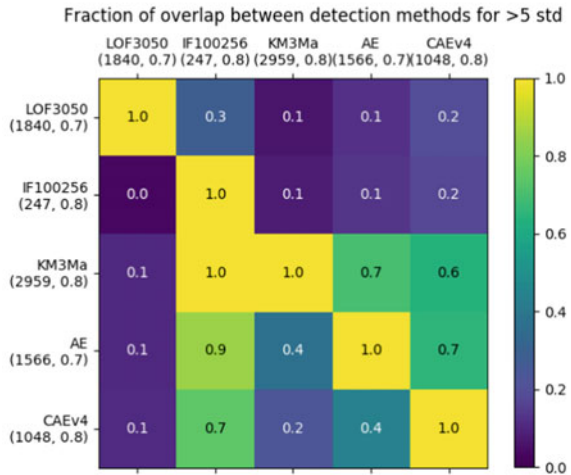
While for the method detecting the least single sources still over half of the strongest 250 objects contain a single source, the intersection between the methods contains comparatively a very small number of single sources, see Table 5.

Taking the above observations into consideration, we can look at the objects scoring above 5 standard deviations for 3 methods and between 3 and 5 for one method to see if they match our expectations. We found that 86 out of the 135 objects have a lower score for IF, most of which will be caused by the small number of objects scoring above 5 standard deviations for IF. There are no objects only scoring between 3 and 5 standard deviations for KM. The 7 objects that do so for the

Fig. 17 Overlap between the different outlier detection methods on stars. The size of the results and the fraction of the total results this constitutes is shown between brackets



(a) Between 3 and 5 standard deviations from the mean



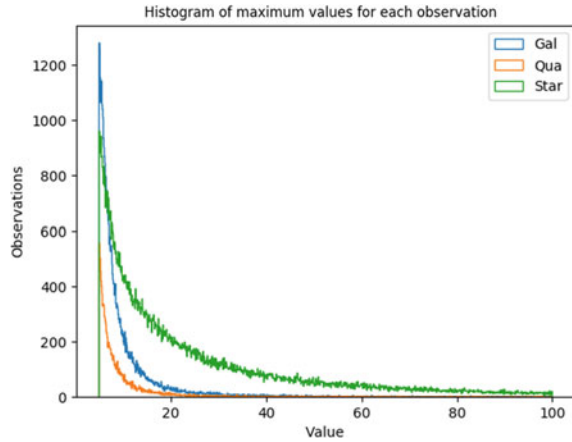
(b) Above 5 standard deviations from the mean

CAE seem representative of the objects scoring high for all methods and do not share a common trait. The 42 objects scoring lower for the LOF all have a high intensity, the lowest being 50.93, which is in line with the expectations.

We tried a number of ideas, such as normalizing each object independently, but the resulting outliers are simply noisy images as these tend to have strange intensity distributions.

Secondly, as we know that each image is centered on the object, we tried segmenting this object and using it as a mask on the original observation. The distance between masked background and the lowest intensity value within the object, as

Fig. 18 Histogram of the maximal value of pixels, among all SDSS bands, for Galaxies, Stars and QSOs



well as the degree to which the intra-object values are scaled, can be considered hyperparameters that influence the trade-off in importance between morphology and intensity. Segmenting the object with a threshold, followed by morphological closing, however, makes too many assumptions about the underlying object.

Therefore, taking into account the four methods used in our comparison, the strongest outliers are dominated by images with multiple, relatively bright sources. Clearly, some pre-processing step is needed to get rid of multiple sources as well as enabling low intensity objects to reach a higher score. Deciding which method outperforms the others, if any, is not possible without a domain expert, analyzing the most outlying results one by one. We can conclude that the CAE and especially the LOF find mostly different outliers than the other more similar methods discussed in this work, and that taking the intersection of the highest scoring objects for each method does result in a list of manageable size for further inspection.

In conclusion, despite at this stage we did not find very interesting objects, the proposed strategy is actually capable to identify outliers, like satellites, asteroids or problematic images and we think that, when applied to data of a higher quality, for instance with a deeper and higher quality photometry, possibly extending to IR bands, it could lead to more interesting results.

4.1 Future Work

There is a variety of possible improvements for the methods applied in this work.

The first could be, instead of using PCA for the dimensionality reduction, to use more involved options, such as the features extracted by a CNN or the lower dimensional representation of the CAE. They could be used as the input to the outlier detection methods.

Using all integers for the LOF as hyperparameter values in the chosen range, instead of using intervals of 5, is in line with the recommendation of the authors. This can efficiently be computed by calculating the required values for the upper bound number of neighbours, and using these pre-computed values to quickly calculate the LOF values in the whole range [4].

As for the IF, a variant, such as the Extended Isolation Forest (EIF), can be used. The EIF claims to increase the reliability and consistency of the outlier scores compared to those of vanilla IF, which can produce artifacts as its decision boundaries are either vertical or horizontal [27]. When applied to spectra, IF is unable to find more subtle outliers, which are often more interesting from an astronomical standpoint. Restricting the value at which to split at a certain node, to be for example between the 10th and 90th percentile of the feature value, improved the performance [28].

Variants of KM such as fuzzy k-means can be tried out, as well as many other different clustering algorithms.

For the CAE, further testing with a wider range of hyperparameters and their values can be performed to improve its performance. Whether this also improves the quality of the outliers found can also be investigated.

Creating the mask of the center object could be done by more sophisticated methods than thresholding, such as using an object detection and segmentation tool like MTOBjects to detect the shape of the object [29]. These could also be used to detect observations with multiple sources present without the need for a database.

For the LOF, KM and MN methods, a decision about which distance metric to use has to be made. The effect of different values for k in the L_k norm or using a different distance metric altogether, such as the cosine distance, remains to be investigated.

Finally, an adaptation of the unsupervised random forest algorithm, which outperformed IF and some other outlier detection methods on the SDSS galaxy spectra, can be tried out on our dataset [14].

Acknowledgements The authors wish to thank Prof. Michael Biehl for useful discussion and contribution which improved the quality of the work. All tables and graphs were made using Matplotlib [30]. The programs themselves make heavy use of the Scipy library [31]. The autoencoders are made using Keras [32]. SC acknowledges the financial contribution from FFABR 2017. MBr acknowledges financial contributions from the agreement *ASI/INAF 2018-23-HH.0, Euclid ESA mission - Phase D* and the *INAF PRIN-SKA 2017 program 1.05.01.88.04*.

References

1. Harwit, M.: *Phys. Today* **56**(11), 38 (2003)
2. Harwit, M.: *Cosmic Discovery*. Cambridge University Press, Cambridge (2019)
3. Djorgovski, S., Davis, M.: *Astrophys. J.* **313**, 59 (1987)
4. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. Association for Computing Machinery, New York, NY, USA, 2000, SIGMOD'00, pp. 93–104. <https://doi.org/10.1145/342009.335388>
5. Liu, F.T., Ting, K.M., Zhou, Z.H.: In: *2008 8th IEEE International Conference on Data Mining*, pp. 413–422. IEEE (2008)

6. Lloyd, S.: IEEE Trans. Inf. Theory **28**(2), 129 (1982)
7. Hajer, J., Li, Y.Y., Liu, T., Wang, H.: arXiv preprint [arXiv:1807.10261](https://arxiv.org/abs/1807.10261) (2018)
8. Lyudchik, O.: Outlier detection using autoencoders. Technical report, CERN (2016). <https://cds.cern.ch/record/2209085>
9. Ribeiro, M., Lazzaretti, A.E., Lopes, H.S.: Pattern Recognit. Lett. **105**, 13 (2018)
10. Pimentel, M.A., Clifton, D.A., Clifton, L., Tarassenko, L.: Signal Process. **99**, 215 (2014)
11. Chaudhary, A., Szalay, A.S., Moore, A.W.: In: DMKD (2002)
12. Fustes, D., Manteiga, M., Dafonte, C., Arcay, B., Ulla, A., Smith, K., Borrachero, R., Sordo, R.: Astron. Astrophys. **559**, A7 (2013)
13. Giles, D., Walkowicz, L.: Mon. Not. R. Astron. Soc. **484**(1), 834 (2018)
14. Baron, D., Poznanski, D.: Mon. Not. R. Astron. Soc. **465**(4), 4530 (2016)
15. York, D.G., Adelman, J., Anderson Jr., J.E., Anderson, S.F., Annis, J., Bahcall, N.A., Bakken, J., Barkhouser, R., Bastian, S., Berman, E., et al.: Astron. J. **120**(3), 1579 (2000)
16. Abazajian, K.N., Adelman-McCarthy, J.K., Agüeros, M.A., Allam, S.S., Allende Prieto, C., An, D., Anderson, K.S.J., Anderson, S.F., Annis, J., Bahcall, N.A., et al.: ApJS **182**, 543–558 (2009). <https://doi.org/10.1088/0067-0049/182/2/543>
17. D’Isanto, A., Polsterer, K.L.: A&A **609**, A111 (2018). <https://doi.org/10.1051/0004-6361/201731326>
18. Kreyszig, E., Kreyszig, H., Norminton, E.J.: Advanced Engineering Mathematics, 10th edn. Wiley, Hoboken (2011)
19. Abdi, H., Williams, L.J.: Wiley Interdiscip. Rev.: Comput. Stat. **2**(4), 433 (2010)
20. Thorndike, R.L.: Psychometrika **18**(4), 267 (1953). <https://doi.org/10.1007/BF02289263>
21. Rea, A., Rea, W.: arXiv e-prints [arXiv:1610.03588](https://arxiv.org/abs/1610.03588) (2016)
22. Wei, Y., Jang-Jaccard, J., Sabrina, F., McIntosh, T.: arXiv e-prints [arXiv:1910.06588](https://arxiv.org/abs/1910.06588) (2019)
23. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: In: International Conference on Database Theory, pp. 420–434. Springer (2001)
24. Chalapathy, R., Chawla, S.: arXiv preprint [arXiv:1901.03407](https://arxiv.org/abs/1901.03407) (2019)
25. Bergstra, J., Bengio, Y.: J. Mach. Learn. Res. **13**, 281 (2012)
26. Maaten, L.v.d., Hinton, G.: J. Mach. Learn. Res. **9**, 2579 (2008)
27. Hariri, S., Kind, M.C., Brunner, R.J.: arXiv preprint [arXiv:1811.02141](https://arxiv.org/abs/1811.02141) (2018)
28. Baron, D.: arXiv preprint [arXiv:1904.07248](https://arxiv.org/abs/1904.07248) (2019)
29. Teeninga, P., Moschini, U., Trager, S.C., Wilkinson, M.H.: Math. Morphol.-Theory Appl. **1**, 1 (2016)
30. Hunter, J.D.: Comput. Sci. Eng. **9**(3), 90 (2007). <https://doi.org/10.1109/MCSE.2007.55>
31. Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: open source scientific tools for Python (2001–). <http://www.scipy.org/>
32. Chollet, F. et al.: Keras. <https://keras.io> (2015)

Anomaly Detection in Astrophysics: A Comparison Between Unsupervised Deep and Machine Learning on KiDS Data



Maurizio D'Addona, Giuseppe Riccio, Stefano Cavuoti, Crescenzo Tortora, and Massimo Brescia

Abstract Every field of Science is undergoing unprecedented changes in the discovery process, and Astronomy has been a main player in this transition since the beginning. The ongoing and future large and complex multi-messenger sky surveys impose a wide exploiting of robust and efficient automated methods to classify the observed structures and to detect and characterize peculiar and unexpected sources. We performed a preliminary experiment on KiDS DR4 data, by applying to the problem of anomaly detection two different unsupervised machine learning algorithms, considered as potentially promising methods to detect peculiar sources, a Disentangled Convolutional Autoencoder and an Unsupervised Random Forest. The former method, working directly on images, is considered potentially able to identify peculiar objects like interacting galaxies and gravitational lenses. The latter instead, working on catalogue data, could identify objects with unusual values of magnitudes and colours, which in turn could indicate the presence of singularities.

M. D'Addona (✉) · S. Cavuoti
Department of Physics, University of Naples Federico II, Strada Vicinale Cupa Cintia, 21,
80126 Naples, Italy
e-mail: mauritiussdadd@gmail.com

S. Cavuoti
e-mail: stefano.cavuoti@gmail.com

G. Riccio · S. Cavuoti · C. Tortora · M. Brescia
INAF - Astronomical Observatory of Capodimonte, Salita Moiariello 16, 80131 Naples, Italy
e-mail: giuseppe.riccio@inaf.it

C. Tortora
e-mail: crescenzo.tortora@inaf.it

M. Brescia
e-mail: massimo.brescia@inaf.it

C. Tortora
INAF - Osservatorio Astrofisico di Arcetri, Largo Enrico Fermi 5, 50125 Firenze, Italy

1 Introduction

Due to the rapid growth in volume and complexity of astronomical datasets, Machine Learning (ML) paradigms are gaining a key role within the data exploration and analysis. They are demonstrated as valid mechanisms to find hidden correlations among data and to discover rare and unexpected structures that do not fit those relations [1–3]. The latter, considered as outliers of a data distribution, can be of various nature and may have different degrees of scientific relevance: they can be artifacts produced by anomalies in the data processing pipelines or in the observing conditions, as well as peculiar objects underlining special and rare astronomical events, whose detection may improve the scientific knowledge of relevant physical phenomena.

Machine learning paradigms are divided into two main classes, respectively, supervised and unsupervised methods. While in the supervised case, an a priori *Knowledge Base* is needed to train the algorithms, unsupervised methods can learn the complex relationships among data, without inferring any known information and with a minimum of human supervision. Therefore, it is evident that unsupervised methods are the most suitable to detect anomalies. In particular, we focus on two specific models: an unsupervised variant of random forests (*Unsupervised Random Forest* or *URF*) [4] and a hybrid type of autoencoder (*Disentangled Convolutional Autoencoder* or *DCA*), which exploits the disentangling property of a variational autoencoder [5] but preserving the structure of a standard convolutional autoencoder [6].

In recent years both methods have successfully been used in the astrophysical context. For example, Tuccillo et al. [7] validated the former method on both analytic profiles and real galaxy images. Baron et al. [1] used a *URF* on galaxy spectra from the *Sloan Digital Sky Survey* (*SDSS*), finding objects with extreme emission line ratios, abnormally strong absorption lines, extremely reddened galaxies and other peculiar objects. Reis et al. [8] applied this method to infrared spectra of stars, showing that the metric defined in this algorithm traced the physical properties of the stars. Finally, Reis et al. [9] also discovered 31 new redshifted broad absorption line quasars within *SDSS* spectral data. Concerning the *DCA* model, a very similar architecture was successfully applied to radio data to disentangle noise signal contamination, revealing emissions from air showers, thus enabling accurate measurements of cosmic particle kinematics and identity [10]. More in general, such models are faster, compared to other traditional profile fitting methods, can be easily adapted to more simple/complex models and could be used to detect peculiar substructures, such as strong gravitational lenses and galaxy mergers.

In this preliminary work, we first use *DCA* on synthetic images in order to evaluate its theoretical performance, then we apply both methods on real image cutouts and catalogue counterparts. In particular, in Sect. 3 we describe the use of a *DCA* to perform an outlier detection using images extracted from the 4th Data Release of the European Southern Observatory (ESO) Kilo Degree Survey (*KiDS*) [11]. Then, for the same purpose, in Sect. 4 we describe the use of an *URF* on the same subset of

objects but using photometric data, always extracted from the KiDS DR4. Finally in Sects. 5 and 6 we discuss the results and compare the performance of the two methods.

2 Data Preparation

In order to validate the *DCA* model and assess its performance we generated three sets of 20,000 synthetic images of 64×64 pixels, using three different models of surface brightness profile of galaxies that are further described in Sect. 3.1. These images have a dynamic range between 0 and 1. A Gaussian noise, drawn from a folded normal distribution with a standard deviation of $\sigma_{noise} = 0.09$, has also been added to each image and the value of the standard deviation has been chosen to maintain the $\sim 99\%$ of the values within the 30% of the dynamic range. The generated noise has a mean value $\mu_{noise} \approx 5 \cdot 10^{-2}$ that corresponds to the 5% of the maximum value of the dynamic range of the image.

The real data selected to perform our tests on both methods are extracted from the KIDS Data Release 4 [11]. In particular, we randomly extracted a subset of object cutouts from the tiles that are in common with the DR3 data release [12] and using the DR4 photometry in the related catalogue.

For the photometry we used the Gaussian Aperture and PSF (GAaP) magnitudes in the four bands u, g, r, i , with the minimum aperture of 1.0 arcsec and the corresponding automatic minimal aperture magnitudes $u_{auto}, g_{auto}, r_{auto}, i_{auto}$, which are also corrected for the galactic extinction. In addition to these features, we also included all colours and magnitude ratios [13], derived from all the above magnitudes, resulting in a total of 36 photometric features. From this dataset we excluded all objects with missing data in any of the photometric bands. We also applied a minimum set of magnitude cuts, in order to remove the objects lying in the tails of the distributions: $16 < i < 22$ and $16 < r < 22$. The result is a dataset of 400,000 objects.

For each object, a cutout of 32×32 pixels (corresponding to $\sim 6.7 \times 6.7$ arcsec) has been extracted from the corresponding photometrically and astrometrically calibrated r band coadded tiles. The size of the cutouts has been chosen so that almost all of them contain only the central object, while preserving a sufficient amount of surrounding pixels and angular size. All pixel values of the cutouts have been normalised between 0 and 1.

About the 90% of these objects was also present in the candidate quasars (QSOs) catalogue, produced for the 3rd Data Release of KiDS, containing a mixed set of stars, QSOs and galaxies, classified with Machine Learning [14]. The two catalogues were cross-matched, resulting in a subset of ~ 1100 QSOs and ~ 260 stars with a reliable classification, considered useful information to take into account in the evaluation of the anomaly detection experiment results.

3 Disentangled Convolutional Autoencoders

Autoencoders are a particular type of neural network used to learn data codings by efficiently mapping high-dimensional inputs into low-dimensional encoded vectors and reconstructing the input data from the encoded vector only [15]. By forcing the low-dimensional representation, or *latent space*, to have less dimensions than the input data, the network is forced to learn useful features from the data and, through the use of the backpropagation algorithm, in combination with a smooth loss function, the content of the latent space is iteratively adapted, in order to achieve a good reconstruction performance. For such reasons, the autoencoder is able to perform the feature extraction and dimensionality reduction tasks in a completely unsupervised fashion. The basic structure of an autoencoder consists of two sections (Fig. 1):

- An **encoder** that maps the input data into *semantic code vectors* that live in a so called *latent space*.
- A **decoder** that learns to decompress the *semantic code vectors* from the *latent space* back to the input space, producing a reconstructed representation of the input.

In traditional convolutional autoencoders, both the encoder and the decoder are *Convolutional Neural Networks (CNN)* [16]. The convolution operations allow to identify the key features in an image, thus making them well suited for classification, denoising and image compression tasks. However, since there is no direct control on how the input space is mapped to the *latent space*, it is difficult to extract specific and valuable information from the encoded *semantic code vectors*.

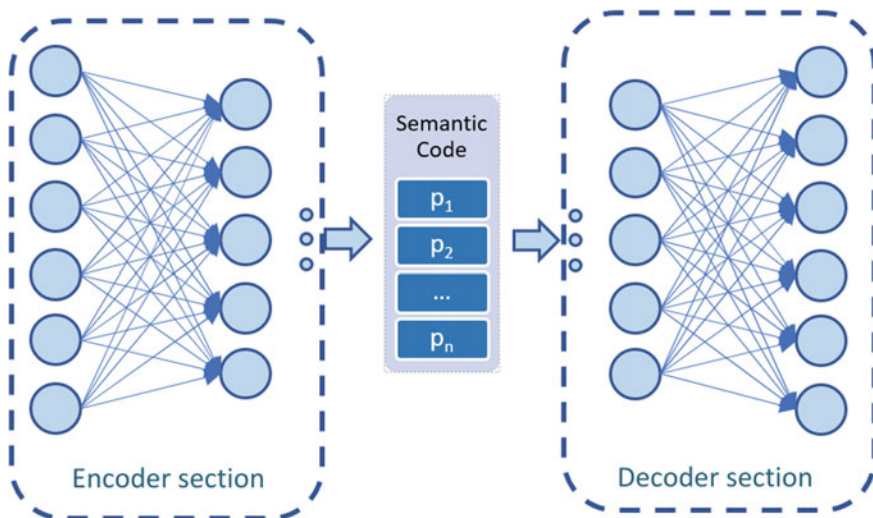


Fig. 1 Representation of the basic structure of an autoencoder

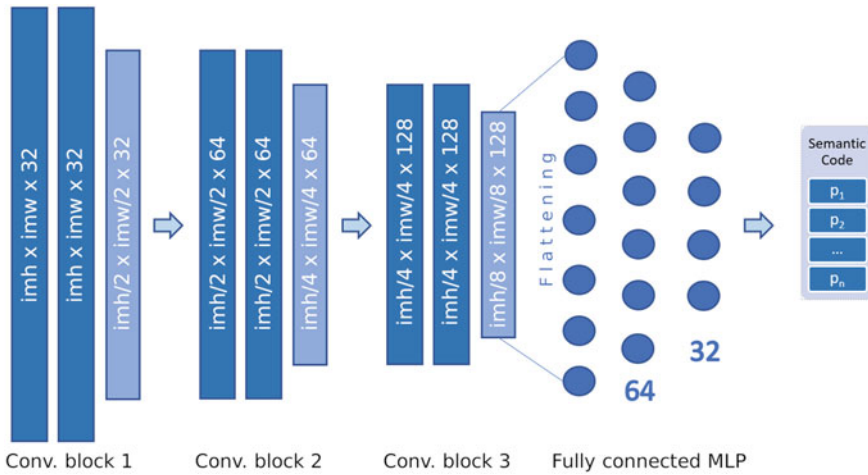


Fig. 2 The structure of the encoder used in our experiments: there are three convolutional blocks followed by a fully connected MLP with two hidden layers. Each block has two convolutional layers followed by a 2×2 max-pooling operation. imw and imh are respectively the width and height of the input images

One way to overcome this limitation is to replace the decoder *CNN* with a given function that produces a synthetic model of the input data, as already proposed by Aragon-Calvo [17]. In this way, after a successful training, the *latent space* is forced to coincide with the domain of the model function and each parameter of the *semantic code* controls a different characteristic of the generated model, thus the name *Disentangled Convolutional Autoencoder*. An interesting feature of this type of autoencoders, implicitly deriving from its construction, is that they can successfully represent only objects compatible with the model assumed. Identifying those objects means to detect artifacts, images containing wrong data, but also interesting outliers.

In our experiments we developed a multi-GPU *DCA*, using the Python bindings of *TensorFlow* [18] and its built-in Keras module [19]. The encoder part is made by three convolutional blocks, each one containing two convolution layers, using a ReLU activation function and followed by a 2×2 max-pooling layers (Fig. 2). The convolutional layers in the three blocks have respectively 32, 64 and 128 kernels of size 4×4 . The last max-pooling layer has 128 channels of size $\frac{imh}{8} \times \frac{imw}{8}$, where imh and imw are, respectively, the height and width of the input images. This hierarchical module is then flattened and fed to a fully connected *Multi-Layer Perceptron (MLP)* [20], with two hidden layers of 64 and 32 neurons, respectively. The output layer of the MLP section has as many neurons as many parameters there are in the model used by the decoder.

The decoder is a custom TensorFlow layer object that encapsulates a given model and passes to it the encoder output p along with a pair of coordinate meshes X and

Y . If needed, the model may also take care of applying some appropriate constraints to the parameters. The coordinate meshes have the same size of the input images and contain, respectively, the x and y pixel coordinates.

3.1 Validation with Synthetic Data

In order to evaluate the autoencoder performance, we have first created two simple models of galaxy surface brightness: an exponential and a Sérsic profile model. Then we added a third more complex Bulge/Disk model, which is a linear combination of these two. For each profile, we generate a set of synthetic images as described in Sect. 2 and used half of them as train set, while the rest as a blind test set.

3.1.1 Exponential Profile of Galaxy Surface Brightness

The exponential profile usually well describes the light distribution of the disk of a galaxy as a function of the distance from its centre [21]. The model we implemented has five parameters:

- x_0 : the x coordinate of the center of the galaxy;
- y_0 : the y coordinate of the center of the galaxy;
- a : the size of the semi-major axis in pixels;
- q : the ratio between the minor and major axis;
- θ : the rotation angle, defined as the angle that the major axis forms with the x axis of the image.

Using these parameters, we first apply a coordinate transformation to take into account the translation and rotation of the galaxy (Eq. 1).

$$\begin{aligned} x'(x, y) &= (x - x_0) \cdot \cos(\theta) - (y - y_0) \cdot \sin(\theta) \\ y'(x, y) &= (x - x_0) \cdot \sin(\theta) + (y - y_0) \cdot \cos(\theta) \end{aligned} \quad (1)$$

Using the transformed coordinates we then compute the radius value for a give pixel coordinate (x, y) with the Eq. 2.

$$r'(x, y) = \frac{1}{a} \cdot \sqrt{x'(x, y)^2 + \left(\frac{y'(x, y)}{q}\right)^2} \quad (2)$$

And finally we compute the exponential brightness profile (Eq. 3).

$$f_{exp}(x, y) = \exp(-r'(x, y)) \quad (3)$$

This profile is normalised so that the maximum value is 1 at $(x = x_0, y = y_0)$ and the minimum value is zero. With this profile and using random parameters, we generated 20,000 synthetic images of 64×64 pixels, according to the procedure described in Sect. 2. We then split the images into a train set and test set of 10,000 images each. We run the autoencoder on the train set using different optimizers and loss functions. We obtained the best results using the *Adam* optimizer [22] with a learning rate of $lr = 1e - 4$, a batch size of 128 images and a maximum number of 2000 training epochs. We also used a custom loss function defined as follows (Eq. 4):

$$\text{loss}_{mael} = \frac{1}{N \cdot W \cdot H} \sum_{j=0}^N \sum_{x=0, y=0}^{W, H} |\ln(1 + f_j(x, y)) - \ln(1 + I_j(x, y))| \quad (4)$$

where W and H are, respectively, the width and height of the input images I_j ; f_j is the output image generated by the autoencoder for the corresponding input image and N is the total number of the images. The logarithmic transformations in Eq. (4) give more weight to the fainter regions of the galaxies that are also the parts more difficult to fit. Using a higher learning rate, the training time decreases, but it also increases the chances that the algorithm will not converge to an optimal solution. Other optimizers like *Adadelata* [23] or *Stochastic gradient descent (SGD)* [24, 25] very often did not converge to an optimal solution, even using different learning rates. Using these training parameters we performed 25 executions and selected the trained model that provided the minimum *mean absolute error (MAE)* between the input and the output images and ran it on the test set (Fig. 3).

As described in Sect. 3, the output of the autoencoder is a reconstruction of the input images, based on the parameters of the model. Therefore, to assess the goodness of the reconstructed image and in turn of the parameters, we computed the *MAE* and the *normalised median absolute deviation (NMAD)* of the residuals for each pair of

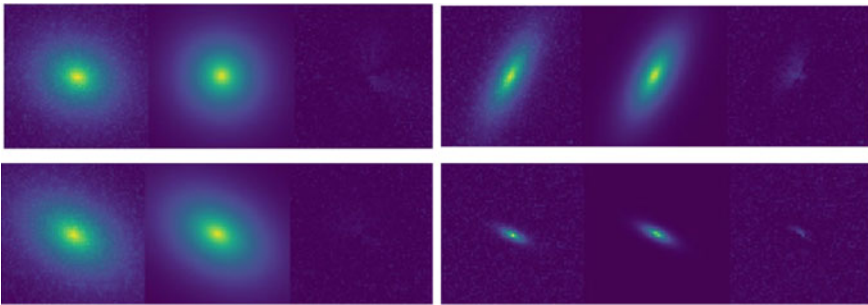


Fig. 3 Comparison between the autoencoder input and output images. The colour-map was chosen for better visualization, so that the brightest pixels are yellow and the darkest ones are in dark blue. The two top images are training examples, while the two on the bottom are examples from the test set. For each image, from left to right panels respectively, there are the original synthetic image with noise, the output of the autoencoder and the residual of the two images

Table 1 Statistical estimators for the true versus predicted values for each parameter of the exponential galaxy profile model. Note that, although the uncertainty on the size of the galaxy is relatively larger than other parameters, the uncertainty on the axis ratio is small

Parameter	x_0	y_0	a	q	θ
NMAE	0.16	0.16	0.32	0.02	0.01
NMAD	0.02	0.02	0.03	0.01	0.01

input-output images, finding an average $\overline{MAE} = 0.07 \pm 0.02$, which is compatible with the mean noise level and an average $\overline{NMAD} = 0.03 \pm 0.01$, from which we can compute the equivalent standard deviation $\overline{\sigma_{NMAD}} \approx 1.5 \cdot \overline{NMAD} = 0.05 \pm 0.02$, which is compatible with the standard deviation of the noise. In Table 1 the normalised *MAE* and *NMAD* for each parameter of the model are also reported, computed using the true parameter values and the ones predicted by the trained encoder. The small values of these statistical indicators show that the autoencoder was able to train the model successfully.

3.1.2 Sérsic Profile of Galaxy Surface Brightness

The Sérsic profile usually describes well the light distribution of the bulge of a galaxy and elliptical galaxies in general [26, 27]. This profile is a generalization of the exponential profile, obtained by introducing a parameter n , called *Sérsic index*, that controls how the light is distributed across the galaxy. The inverse of this parameter $\beta = 1/n$ is used as exponent of the radius in the surface brightness profile equation. For $n = 1$ the Sérsic profile coincides with the exponential one. The model we implemented has six parameters:

- x_0 : the x coordinate of the center of the galaxy;
- y_0 : the y coordinate of the center of the galaxy;
- a : the size of the semi-major axis in pixels;
- c : the ratio between the minor and major axis;
- θ : the rotation angle defined as the angle that the major axis forms with the x axis of the image;
- β : the inverse of the Sérsic index.

The only difference with the exponential model is in the profile function of Eq. (3) that becomes as specified in Eq. (5).

$$f_{\text{Sérsic}}(x, y) = \exp(-r'(x, y)^\beta) \quad (5)$$

As in the previous profile case, by using random parameters we generate 20,000 synthetic images, divided into a training and a test set, each one composed by 10,000 images. Also in this case we obtained the best training results using the *Adam* optimizer with the custom loss function (4), a learning rate of $lr = 1e - 4$, a batch size

Table 2 Statistical estimators for the true versus predicted values, reported for each parameter of the Sérsic galaxy profile model

Parameter	x_0	y_0	a	q	θ	β
NMAE	0.17	0.17	0.35	0.02	0.02	0.33
NMAD	0.02	0.02	0.04	0.01	0.01	0.08

of 128 images, and a maximum number of training epochs of 2000. We computed the *MAE* and the *NMAD* of the residuals for each pair of input-output images, finding an average $\overline{MAE} = 0.06 \pm 0.03$ that is compatible with the mean noise level and an average $\overline{NMAD} = 0.03 \pm 0.01$. In Table 2 the normalised *MAE* and *NMAD* for each parameter of the model are reported, computed using the true parameter values and the ones predicted by the trained encoder. The small values of these statistical estimators show that the autoencoder was able to train the model successfully.

3.1.3 Bulge/Disk Profile of Galaxy Surface Brightness

This model is a linear combination of the Exponential and Sérsic profiles, used to mimic a combination of bulge and disk components as well as a uniform background. We also introduced a constant background level to take into account the sky background present in almost all real images. It has eleven parameters:

- x_0 : the x coordinate of the center of the galaxy;
- y_0 : the y coordinate of the center of the galaxy;
- a_{disk} : the size of the semi major axis of the disk component in pixels;
- c_{disk} : the ratio between the minor and major axis of the disk component;
- θ_{disk} : the rotation angle of the disk component, defined as the angle that the major axis forms with the x axis of the image;
- α : the fractional ratio between the central brightness of the bulge and the central brightness of the disk;
- a_{bulge} : the size of the semi major axis of the bulge component in pixels;
- c_{bulge} : the ratio between the minor and major axis of the bulge component;
- θ_{bulge} : the rotation angle of the bulge component, defined as the angle that the major axis forms with the x axis of the image;
- β : the inverse of the bulge Sérsic index;
- k : the background level expressed as the fractional ratio between the brightness of the background and the maximum brightness of bulge+disk.

The profile function of this model is shown in the Eq. (6).

$$f_{bd}(x, y) = (1 - k) \cdot (\alpha \cdot f_{sersic}(x, y) + (1 - \alpha) \cdot f_{exp}(x, y)) + k \quad (6)$$

As done in the previous tests, by using random parameters, we generate 20,000 synthetic images, divided into a training and a test set, each one containing 10,000

images. Also in this case we obtained the best training results using the Adam optimizer with the custom loss function (4), a learning rate of $lr = 1e - 4$, a batch size of 128 images, and a maximum number of 2000 training epochs. The results of the test and the training were similar to those found in the previous test, finding an average $\overline{MAE} = 0.07 \pm 0.04$ that is compatible with the mean noise level and an average $\overline{NMAD} = 0.03 \pm 0.02$.

3.2 Application to KiDS Data

After having validated the autoencoder model on synthetic data, we tried to apply the *Bulge/Disk* profile model on real data. As already introduced in Sect. 2, the images used in this experiment are cutouts taken from the r band tiles of the KIDS DR4. We divided them into a training set of 30, 000 images and a set of 370, 000 images used to detect potentially interesting outliers.

We trained the autoencoder by using the *Bulge/Disk* profile model, the optimizer and training parameters validated with synthetic data. 25 training runs were performed, selecting the trained model with the lowest *MAE*. Finally, we ran the best trained model on the image test set.

3.2.1 Anomaly Detection with DCA

As we said above, if the autoencoder is correctly trained and the chosen model is a valid representation of the input objects, then the residual images—obtained by subtracting the output of the decoder from the corresponding input—should contain only residual noise. Therefore, it is clear that the statistical estimators computed on the residual images have a key role in detecting anomalies that the model is not able to describe. We used the following statistical estimators:

- **MAD:** since it is not very influenced by extreme values, the median of the pixel values in the residual image corresponds approximately to the mean background value. Thus, the *Median Absolute Deviation* is a valid measure of how broadly the residuals are distributed around the background. A high value could indicate the presence of substructures or artifacts.
- **Skewness:** unusually high or low values of this statistical moment could indicate that there is something odd in the image.
- **Maximum:** hot pixels, artifacts but also other objects in the whole image produce very bright pixels in the residual images.

The outliers were selected using the following automated procedure: as first step, the average maximum value of the residuals \overline{max} was computed along with the respective standard deviation σ_{max} and all objects for which $max > \overline{max} + 3\sigma_{max}$ were marked as outliers. Then for each unique pair of statistical estimators, the

average number density of the objects, $\overline{\rho_n}$, and the corresponding standard deviation σ_n were computed; then this two-dimensional space was divided into 400 tiles of equal size. The local number density ρ_n was computed in each tile and the resulting density map was smoothed with a gaussian kernel. Finally, each object falling in a sub-region with a density $\rho < \overline{\rho_n} - 2\sigma_\rho$ was marked as an outlier, as it can be seen in Fig. 4. In Fig. 5 the percentage of objects detected as outliers is reported as a function of the detection threshold previously defined. We note the robustness of the detection that remains approximately constant above the value of $2\sigma_\rho$. About the 93% of the objects are concentrated in a quite continuous region with an average of $\overline{MAD} = 0.011 \pm 0.005$, $\overline{skewness} = 2.3 \pm 1.7$ and $\overline{max} = 0.12 \pm 0.03$. These are objects that the autoencoder was able to fit with the model. The low values of the \overline{MAD} and \overline{max} indicate that the reconstructions of the autoencoder describe very well the original images and that the residuals contain basically only background noise. This is also confirmed by the value of $\overline{skewness}$ greater than zero, which is typical of Poissonian distributions, characterized by a low value of the mean, as in the case of the shot noise that affects digital images. We identified few objects

Fig. 4 From top to bottom: scatter plots of the skewness versus maximum, skewness versus MAD and MAD versus maximum. Axes are in logarithmic scale and the colour indicates the logarithm of the local number density of the points, where a lighter colour means a denser region. Objects identified as outliers are highlighted in green

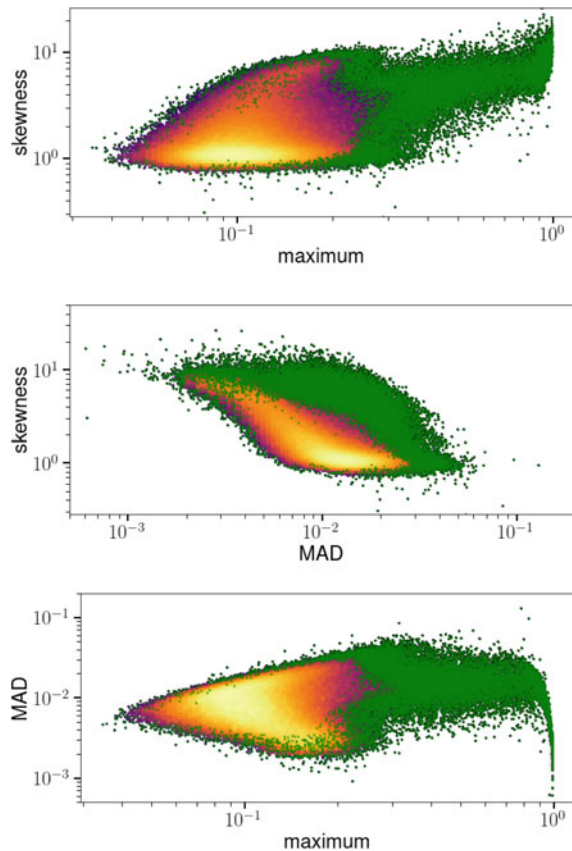
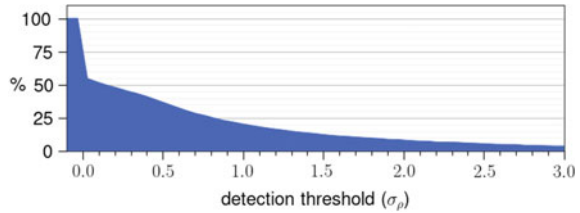


Fig. 5 Percentages of objects classified as outliers by the DCA, as a function of the detection threshold expressed in units of σ_ρ



having a very low skewness, which usually indicates a Gaussian-like distribution of the residuals and thus the presence of something else beyond the pure Poissonian background noise.

Some of these objects were bigger than the cutout area (Fig. 6), which the autoencoder was less able to fit properly, while others showed traces of substructures in the residual image, which were hidden by the galaxy light. There was also a small clump of objects, less than 2% of the total amount, having a very high maximum value: these were very faint sources or objects with a very bright companion (Fig. 7).

A small set of objects has also a very low *MAD*. Although a low value of this statistical estimator could imply a low dispersion of the residuals, an unusual low value means that most of the pixels in the residual image have the same value, which in turn could indicate some sort of corruption. In fact, most of these objects were located on the edge of the tile, thus resulting in partially corrupted cutouts (Fig. 8). Finally, there was a subset of objects, approximately the 5% of the total amount, having a fairly average of *MAD* and *skewness* values, but with a quite large maximum value ranging from 0.3 to 0.8. Almost all of these objects have one or more than one faint companion, as shown in the examples of Fig. 9.

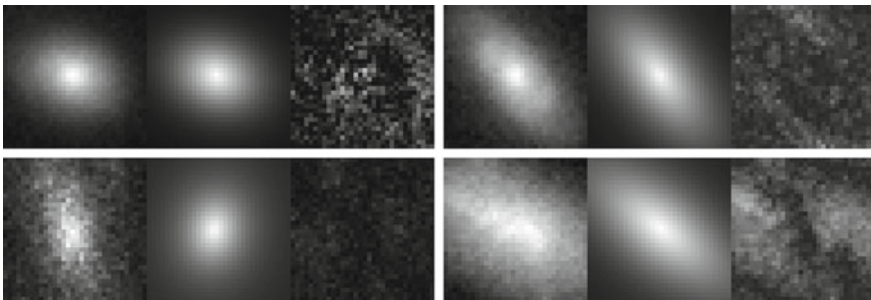


Fig. 6 Some of these objects show the presence of substructures that were hidden by the light of the galaxy (upper left and upper right), while in other cases the autoencoder failed to fit the surface brightness profile because the objects were bigger than the size of the cutout. For each image, from left to right panels, there is the original image, the images produced by the autoencoder and the residual image, re-scaled to highlight the presence of substructures

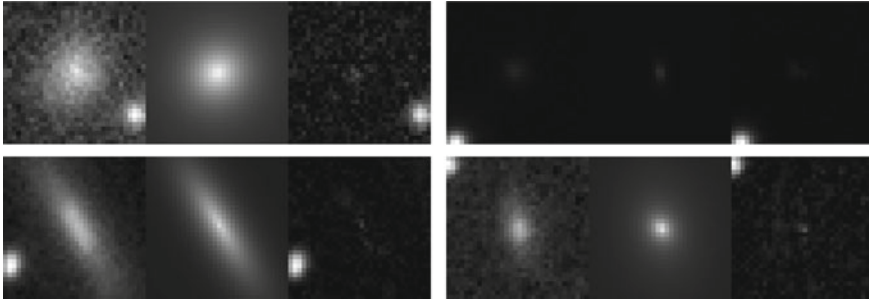


Fig. 7 Examples of KiDS galaxy that are very faint or have a very bright close companion or present artifacts like hot-pixels. For each image, from left to right panels, there is the original image, the images produced by the autoencoder and the residual image

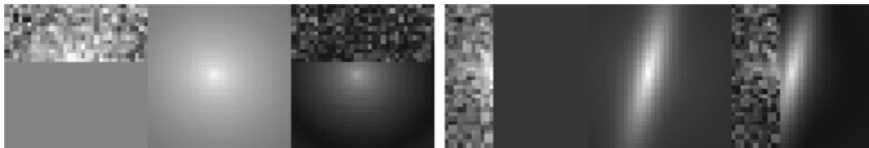


Fig. 8 Examples of objects that are just on the border of the tile from where the cutouts have been extracted. For each image, from left to right panels, there is the original image, the images produced by the autoencoder and the residual image

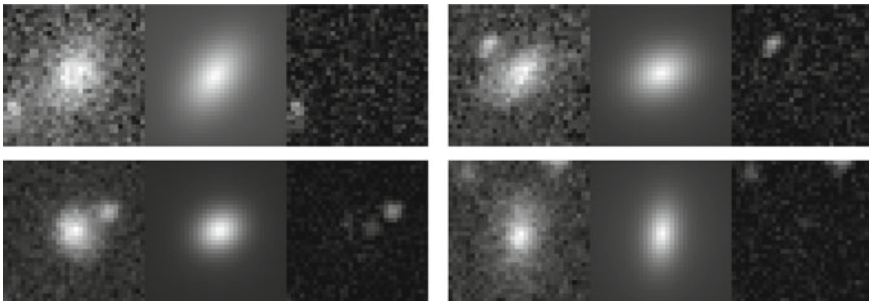


Fig. 9 Examples of objects showing the presence of a faint quite close companion. For each image, from left to right panels, there is the original image, the images produced by the autoencoder and the residual image

4 Unsupervised Random Forests

Random Forests are an ensemble of several independently grown decision tree classifiers, where each tree is a non-parametric model organized in a top-bottom tree-like structure and is grown using a random subset of the features of the training dataset [28]. They are usually used to classify objects for which a training set of labelled objects exists so that each tree in the forest learns to map the input features

to the corresponding correct label. When an object identified by a set of feature is passed to the forest, each tree votes for its belonging to one of the given classes, identified by the labels, and the resulting class is usually determined by majority voting.

For the problem of outliers detection, where obviously a labelled training set is not available, random forests can also be used in an unsupervised configuration.

A simple but efficient way to use Random Forest as an unsupervised method is to generate a synthetic dataset from the original one, with the same size and the same marginal distribution in all its features, but without the covariance among objects. Then the Random Forest is trained on both datasets to learn to recognize their similarity, thus isolating the outliers. By defining a *similarity index* $S_{i,j}$ between any two objects as the number of common “real” leaves of the trees, divided by the total number of trees in the forest, a *weirdness score* can be introduced, which describes how distant is, on average, from all the others. This score can assume any value between 0 and 1, but the distribution of its values mostly depends on the specific dataset involved. Therefore, a reasonable way to use it is to impose a certain threshold, based on the distribution of its values for all the objects in the dataset and then to consider as outliers all objects with a weirdness value greater of such threshold. Baron and Poznanski [1] proposed this method that was able to find some galaxies with peculiar spectra in the 12th data release of the Sloan Digital Sky Survey [29].

4.1 Anomaly Detection in KiDS Data Based on the URF

For this experiment we used the photometric catalogue containing the counterparts of the image cutouts, organized as described in Sect. 2. According to what described in Sect. 4, we then created a synthetic dataset of the same size of the real one and with objects drawn randomly from the same marginal distribution of each feature (Fig. 10).

We then merged the two datasets into a single one, labelling the objects depending on whether they were real or synthetic, and used for training and testing a *Random Forest Classifier* containing 800 trees, built using the Python package *scikit-learn* [30]. We remark that in the case of the *URF* model, both training and testing sets coincide, since the same data are used to perform the anomaly detection experiment along the construction of the random forest trees process. We then divided the original dataset into batches of 6000 objects and computed the *weirdness* index for each batch. The size was limited by the amount of memory necessary to compute the *weirdness*. The whole process has been repeated four times and the *weirdness* values have been averaged for each object. The objects show a distribution centred on an average value of *weirdness* of $\bar{w} = 0.83$ with a standard deviation of $\sigma_w = 0.06$, while the number of objects decreases as the *weirdness* value increases (Fig. 11). In analogy to what done in the case of the *DCA* model (Sect. 3.2.1), and to perform a direct comparison between the two models, we imposed a detection threshold of 2σ and considered as outliers all objects for which $w > \bar{w} + 2\sigma_w = 0.95$.

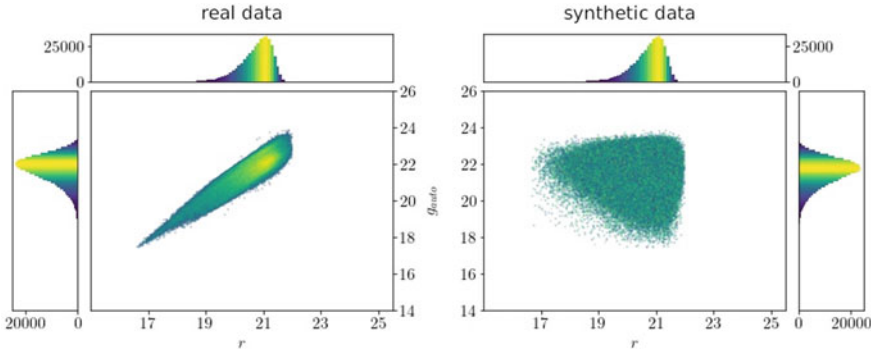
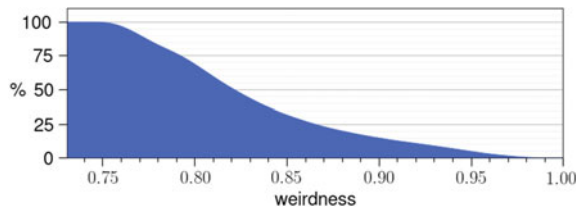


Fig. 10 Projection on two features (magnitudes r and g) of the density distributions of the KiDS sample (left panel) and synthetic data (right panel). The latter was generated from the same marginal distributions of the real one, by removing the covariance among original data

Fig. 11 Percentages of objects classified as outliers by the URF , as a function of the weirdness



5 Discussion

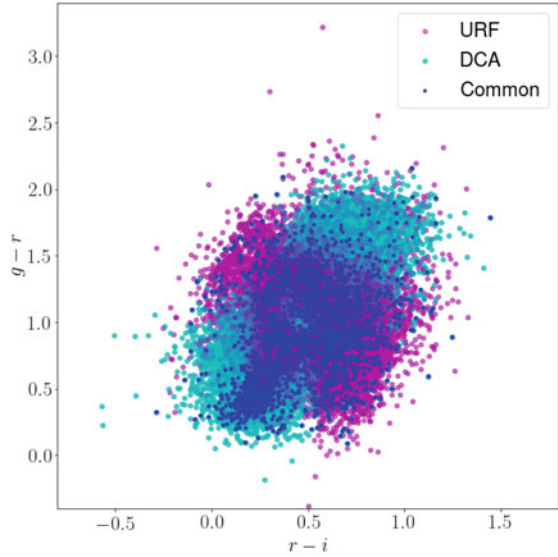
Both chosen algorithms, DCA and URF , were tested on a subset of 400, 000 objects extracted from the KiDS survey Data Release 4. The DCA was used directly on image cutouts extracted from the r band coadded tiles, while the URF was used on the catalogue of counterparts, made by magnitudes in the bands $ugri$, their derived colours and ratios. The model DCA required only a minimum of human supervision during the training, just to check the convergence of the algorithm to an optimal solution. It performed very well in both terms of memory requirements and computing time and was able to pinpoint some peculiar sources, about the 5% of the sample, showing substructures that were hidden by the close galaxy light, as well as objects with very small and/or faint close companions.

Since the URF is based on the computation of a *similarity matrix*, whose size increases as the square of the number of the objects, the dataset has to be analysed in batches and a supplementary amount of human intervention was required in order to determine the optimal batch size.

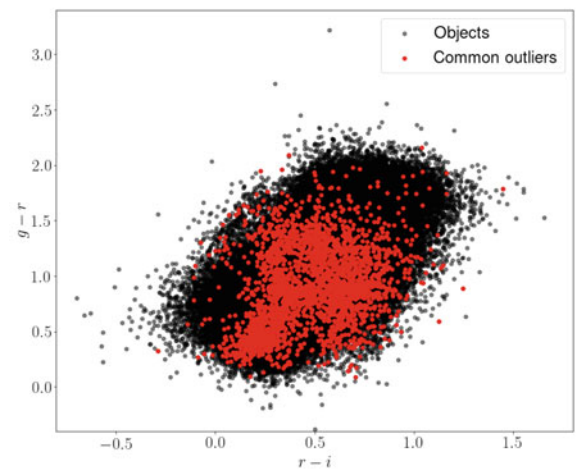
To perform a comparative analysis of the results obtained by the two methods, we imposed a similar criterion to extract candidate outliers, for instance, a common value of 2σ with $w > \bar{w} + 2\sigma_w = 0.95$ in terms of weirdness w for the URF and object density $\rho < \bar{\rho}_n - 2\sigma_\rho$ in the case of DCA . With such outlier detection thresholds,

both methods found a comparable amount of peculiar objects, $\sim 7\%$ of the test set for *DCA* and ~ 5 for *URF*. Among the objects considered as an anomaly by at least one of the two methods, the $\sim 7\%$ were detected as peculiar objects by both of them. The distributions of the outliers (Fig. 12) shows that most of the peculiar objects found by the two models cover a wide and uncorrelated area of the parameter space, with a limited overlapping region in which most of the common outliers lay. This seems to suggest a certain amount of complementarity of the two methods in

Fig. 12 Upper panel: colour-colour diagram of all the candidate outliers detected by *URF* (in magenta) and *DCA* (in cyan). Common outliers found by both methods are coloured in blue. Lower panel: colour-colour diagram of the outliers detected by both *DCA* and *URF* (in red), plotted against all the objects in the dataset (in black)



(a)



(b)

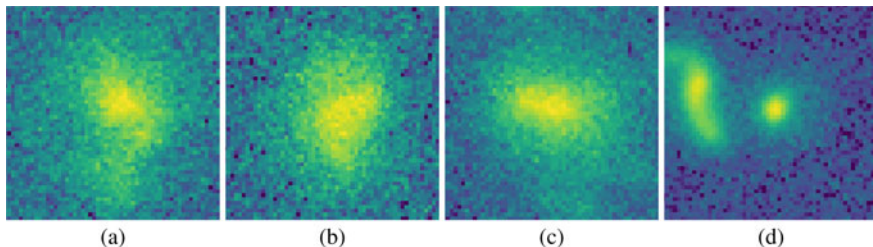


Fig. 13 Examples of cutouts including irregular galaxies (a, b and c) and interacting galaxies (d), detected as anomalies by the two methods

detecting peculiarities, according to a similar behaviour found in [31], concerning the analysis of outliers identified from a distribution of photometric redshifts, estimated by different methods, however no any particular evidence of interesting peculiarity seems to emerge.

By analyzing the detected peculiar objects having a class label provided in [14], only about the $\sim 27\%$ of stars and QSOs were detected as anomalies. Most of these objects, in fact, were not confirmed as peculiar by *DCA* and appear uniformly distributed with respect to the different thresholds of weirdness calculated by the *URF*. This behaviour was expected for *DCA* because no any limitation was imposed on the value of the Sérsic index nor on the galaxy size, thus the model should be able to fit also star-like objects. Through visual inspection of the cutouts for the peculiar objects detected, we observe that both methods tend to assign as peculiar the irregular and interacting galaxies (see examples in Fig. 13), as well as objects that are in more crowded fields, like the ones showed in Fig. 14.

6 Conclusions

The identification of anomalies in Astronomy has always played a major role in making new scientific discoveries. Nowadays, the shift to more large and complex surveys makes essential the use of robust and efficient automated algorithms to identify peculiar patterns. In this context, we performed a preliminary set of anomaly detection experiments, by testing two different unsupervised machine learning algorithms, a *Disentangled Convolutional Autoencoder* and an *Unsupervised Random Forest*, using the former on real image cutouts and the latter on the catalogue of their counterparts, which includes measured magnitudes, derived colours and magnitude ratios, both extracted from the 4th KiDS Data Release.

We performed a comparative analysis of the peculiar objects detected by both methods, by analyzing their colour distribution in the parameter space and their capability to disentangle the presence of QSOs and stars from galaxies within a mixed dataset. The results of this preliminary experiment revealed that most of the

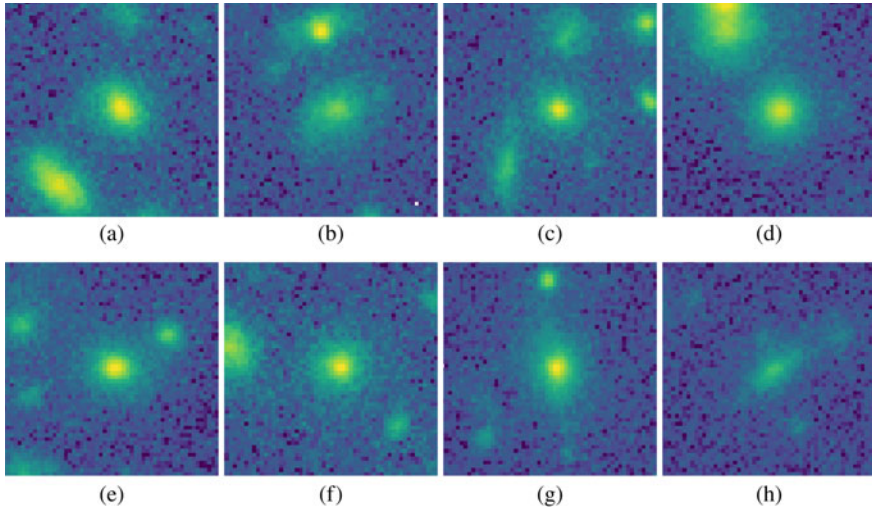


Fig. 14 Examples of cutouts for sources within crowded fields, detected as anomalies by the two methods

anomalies detected by both methods involve irregular and interacting galaxies and sources located in more crowded fields. Further experiments are then required on these models, especially in terms of their setup and configuration, to investigate their real capability to isolate peculiar types of sources. In particular, since the *DCA* is mainly a method to estimate the goodness of a fit to the data, it may result affected by the presence of nearby objects, not taken into account by the model. Improving the detection criteria for *DCA* is thus one of the future enhancements of this method, as well as to take into account the PSF and seeing in the Bulge/Disk model, which should achieve a more accurate estimation of the structural parameters. Regarding *URF*, on the other hand, a further step is the introduction of the infrared bands in the photometric dataset, as well as the search for spectroscopic counterparts, which can improve the classification accuracy and the validation of the method.

Acknowledgements Based on observations made with ESO Telescopes at the La Silla Paranal Observatory under programme IDs 177.A-3016, 177.A-3017, 177.A-3018 and 179.A-2004, and on data products produced by the KiDS consortium. The KiDS production team acknowledges support from: Deutsche Forschungsgemeinschaft, ERC, NOVA and NWO-M grants; Target; the University of Padova, and the University Federico II (Naples). MB acknowledges financial contributions from the agreement ASI/INAF 2018-23-HH.0, *Euclid* ESA mission - Phase D. MB and CT acknowledge the INAF PRIN-SKA 2017 program 1.05.01.88.04. SC acknowledges the financial contribution from FFABR 2017.

References

1. Baron, D., Poznanski, D.: *Mon. Not. R. Astron. Soc.* **465**(4), 4530 (2017). <https://doi.org/10.1093/mnras/stw3021>, <https://academic.oup.com/mnras/article-lookup/doi/10.1093/mnras/stw3021>
2. Brescia, M., Cavuoti, S., Amaro, V., Riccio, G., Angora, G., Vellucci, C., Longo, G.: Data Analytics and Management in Data Intensive Domains. In: Kalinichenko, L., Manolopoulos, Y., Malkov, O., Skvortsov, N., Stupnikov, S., Sukhomlin, V. (eds.) *Communications in Computer and Information Science*, vol. 822, pp. 61–72. Springer International Publishing, Berlin (2018)
3. Fluke, C.J., Jacobs, C.: arXiv e-prints [arXiv:1912.02934](https://arxiv.org/abs/1912.02934) (2019)
4. Shi, T., Horvath, S.: *J. Comput. Graph. Stat.* (2006). <https://doi.org/10.1198/106186006X94072>
5. Chen, R.T.Q., Li, X., Grosse, R.B., Duvenaud, D.K.: Advances in Neural Information Processing Systems 31. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.), pp. 2610–2620. Curran Associates Inc., New York (2018). <http://papers.nips.cc/paper/7527-isolating-sources-of-disentanglement-in-variational-autoencoders.pdf>
6. Guo, X., Liu, X., Zhu, E., Yin, J.: *Neural Information Processing*. In: Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy, E.S.M. (eds.), pp. 373–382. Springer International Publishing, Cham (2017)
7. Tuccillo, D., Huertas-Company, M., Decencière, E., Velasco-Forero, S., Domínguez Sánchez, H., Dimauro, P.: *Mon. Not. R. Astron. Soc.* **475**(1), 894 (2018). <https://doi.org/10.1093/mnras/stx3186>
8. Reis, I., Baron, D., Shahaf, S.: *Astron. J.* (2018). <https://doi.org/10.3847/1538-3881/aaf101>
9. Reis, I., Poznanski, D., Hall, P.B.: *MNRAS* **480**(3), 3889 (2018). <https://doi.org/10.1093/mnras/sty2127>
10. Erdmann, M., Schlüter, F., Šmída, R.: *J. Instrum.* **14**(4), P04005 (2019). <https://doi.org/10.1088/1748-0221/14/04/P04005>
11. Kuijken, K., Heymans, C., Dvornik, A., Hildebrandt, H., De Jong, J.T., et al.: *Astron. Astrophys.* (2019). <https://doi.org/10.1051/0004-6361/201834918>
12. de Jong, J.T.A., Verdoes Kleijn, G.A., Erben, T., Hildebrandt, H., Kuijken, K., et al.: *A&A* **604**, A134 (2017). <https://doi.org/10.1051/0004-6361/201730747>
13. D’Isanto, A., Cavuoti, S., Gieseke, F., Polsterer, K. L.: *A&A* **616** (2018). <https://doi.org/10.1051/0004-6361/201833103>, [arXiv:1904.07248](https://arxiv.org/abs/1904.07248)
14. Nakoneczny, S., Bilicki, M., Solarz, A., Pollo, A., Maddox, N., Spiniello, C., Brescia, M., Napolitano, N.R.: *Astron. Astrophys.* (2019). <https://doi.org/10.1051/0004-6361/201834794>
15. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press, Cambridge (2016). <http://www.deeplearningbook.org>
16. Fukushima, K.: *Biol. Cybern.* (1980). <https://doi.org/10.1007/BF00344251>
17. Aragon-Calvo, M.A.: arXiv e-prints [arXiv:1907.03957](https://arxiv.org/abs/1907.03957) (2019)
18. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., et al.: *TensorFlow: large-scale machine learning on heterogeneous systems* (2015). Software available from <https://www.tensorflow.org/>
19. Chollet, F., et al.: *Keras*. <https://keras.io> (2015)
20. Van Der Malsburg, C.: *Brain Theory*, pp. 245–248. Springer, Berlin (1986). https://doi.org/10.1007/978-3-642-70911-1_20, http://link.springer.com/10.1007/978-3-642-70911-1_20
21. Binney, J., Tremaine, S.: *Galactic Dynamics*, 2nd edn. Princeton Series in Astrophysics. Princeton University Press, Princeton (2008). <http://gen.lib.rus.ec/book/index.php?md5=cd0fd2e719d8966f78eee1f04eee540e>
22. Kingma, D.P., Ba, J.L.: In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings (2015)
23. Zeiler, M.D.: arXiv e-prints [arXiv:1212.5701](https://arxiv.org/abs/1212.5701) (2012)
24. Kiefer, J., Wolfowitz, J.: *Ann. Math. Stat.* **23**(3), 462 (1952). <https://doi.org/10.1214/AOMS/1177729392>
25. Robbins, H., Monro, S.: *Ann. Math. Stat.* **22**(3), 400 (1951). <https://doi.org/10.1214/AOMS/1177729586>

26. Graham, A.W., Driver, S.P.: *Publ. Astron. Soc. Aust.* (2005). <https://doi.org/10.1071/AS05001>
27. Roy, N., Napolitano, N.R., La Barbera, F., Tortora, C., Getman, F., et al.: *Mon. Not. R. Astron. Soc.* (2018). <https://doi.org/10.1093/mnras/sty1917>
28. Breiman, L.: *Mach. Learn.* **45**(1), 5 (2001). <https://doi.org/10.1023/A:1010933404324>
29. Alam, S., Albareti, F.D., Allende Prieto, C., Anders, F., Anderson, S.F., et al.: *The Eleventh and Twelfth Data Releases of the Sloan Digital Sky Survey: Final Data from SDSS-III* (2015). <https://doi.org/10.1088/0067-0049/219/1/12>
30. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., et al.: *J. Mach. Learn. Res.* **12**, 2825 (2011)
31. Brescia, M., Salvato, M., Cavuoti, S., Ananna, T.T., Riccio, G., LaMassa, S.M., Urry, C.M., Longo, G.: *MNRAS* **489**(1), 663 (2019). <https://doi.org/10.1093/mnras/stz2159>

Rejection Criteria Based on Outliers in the KiDS Photometric Redshifts and PDF Distributions Derived by Machine Learning



Valeria Amaro, Stefano Cavuoti, Massimo Brescia, Giuseppe Riccio, Crescenzo Tortora, Maurizio D'Addona, Michele Delli Veneri, Nicola R. Napolitano, Mario Radovich, and Giuseppe Longo

Abstract The Probability Density Function (PDF) provides an estimate of the photometric redshift (z_{phot}) prediction error. It is crucial for current and future sky surveys, characterized by strict requirements on the z_{phot} precision, reliability and completeness. The present work stands on the assumption that properly defined rejection criteria, capable of identifying and rejecting potential outliers, can increase the precision of z_{phot} estimates and of their cumulative PDF, without sacrificing much in terms of completeness of the sample. We provide a way to assess rejection through proper cuts on the shape descriptors of a PDF, such as the width and the height of the maximum PDF's peak. In this work we tested these rejection criteria to galaxies with photometry extracted from the Kilo Degree Survey (KiDS) ESO Data Release 4, proving that such approach could lead to significant improvements to the z_{phot} quality: e.g., for the clipped sample showing the best trade-off between precision and completeness, we achieve a reduction in outliers fraction of $\simeq 75\%$ and an improvement of $\simeq 6\%$ for NMAD, with respect to the original data set, preserving the $\simeq 93\%$ of its content.

V. Amaro (✉) · N. R. Napolitano
School of Physics and Astronomy, Sun Yat-sen University, Zhuhai Campus, Guangzhou 519082, People's Republic of China
e-mail: valeriaa@mail.sysu.edu.cn

S. Cavuoti · M. Brescia · G. Riccio · C. Tortora
INAF - Astronomical Observatory of Capodimonte, Salita Moiariello 16, 80131 Napoli, Italy
e-mail: stefano.cavuoti@gmail.com

M. D'Addona · G. Longo
Department of Physics, University of Naples Federico II, Strada Vicinale Cupa Cintia, 21, 80126 Napoli, Italy
e-mail: longo@na.infn.it

M. D. Veneri
DIETI, University of Naples Federico II, via Claudio 21, 80125 Napoli, Italy
e-mail: micheledelliveneri@gmail.com

M. Radovich
INAF - Osservatorio Astronomico di Padova, Vicolo Osservatorio 5, 35122 Padova, Italy
e-mail: mario.radovich@inaf.it

1 Introduction

Photometric redshifts (z_{phot}) are crucial for modern cosmology surveys, since they provide the only viable approach to determine the distances of large samples of galaxies. Over the years they have been used to constrain the dark matter and dark energy contents of the Universe through weak gravitational lensing [1–3], to reconstruct the cosmic Large Scale Structure [4], to identify galaxy clusters and groups [5–7], to disentangle the nature of astronomical sources [8, 9]; to map the galaxy colour-redshift relationships [10] and to measure the baryonic acoustic oscillations spectrum [11, 12].

Baum [13] first noticed that the stretching of a galaxy spectrum due to the redshift affects the observed colours and hence, if the correlation between photometry and redshift can be uncovered, multi-band photometry could become a powerful tool to estimate redshifts.

However it became immediately apparent that such correlation is highly non-linear and too complex to be derived analytically [14] and that the derivation of z_{phot} required alternative, interpolative approaches.

Nowadays, it is common praxis to divide these methods into two broad classes: the Spectral Energy Distribution (SED) template fitting methods (e.g., [15–17]) and the empirical (or interpolative) methods (e.g., [18–28]), both characterized by their pros and cons.

SED methods rely on fitting the multi-wavelength photometric observations of the objects to a library of synthetic or observed template SEDs, shifted to create synthetic magnitudes for each galaxy template as a function of the redshift. SED fitting methods, while relying on many assumptions, allow pushing z_{phot} beyond the spectroscopic limit.

Empirical methods use instead an a priori spectroscopic knowledge (z_{spec}) for a subsample of objects to infer the complicated relationship existing between the photometric data (i.e. magnitudes and or derived colours, in some cases complemented by morphological information) and the redshift. Among these methods, a critical role is played by machine learning (ML). Among the many ML models applied to the z_{phot} estimation, we quote just a few: neural networks, boosted decision trees, random forests, self-organized maps, convolutional neural networks (see [29] and references therein). A primary advantage of ML is the high accuracy of predicted z_{phot} within limits imposed by the spectroscopic knowledge base (KB). On the other hand, ML methods have an inferior capability to extrapolate information outside the regions of the parameter space properly sampled by the training data and, for instance, they cannot be used to estimate the redshift of objects fainter than those present in the spectroscopic sample.

Extensive reviews of both approaches can be found in [30–32]. An additional difference between the two approaches is that SED fitting methods allow obtaining, at once, the z_{phot} , the spectral type of the objects and the Probability Density Function (hereafter, PDF) of the predicted z_{phot} . In contrast, ML-based methods do not naturally provide a PDF unless special procedures are implemented.

The essential complementarity of the two methodologies was proven to be the most reliable and efficient way to produce a high-quality z_{phot} catalogue [33], particularly

suitable for extensive surveys, like Euclid (Desprez et al., in prep.) and VRST [34]. In this work, we explore the possibility to improve the quality of zphot predictions by excluding from the data potential outliers without loosing much in completeness.

The work is structured as follows: in Sect. 2, we introduce some aspects of PDF evaluation; in Sect. 3, we describe the photometry and spectroscopy used for the analysis herein; in Sect. 4, we give a description of the method for the rejection criteria identification as well as of the statistical estimators used to quantify the performance of both zphot and cumulative PDF statistics. In Sect. 5, we show all the results, and, finally, in Sect. 6, we draw the conclusions.

2 Probability Density Function

In general terms, a PDF is a way to parametrise the uncertainty on the zphot prediction and to provide a robust estimate of the reliability of any individual redshift. From a rigorous statistical point of view, however, a PDF is an intrinsic property of a particular phenomenon, regardless of the measurement methods that allow quantifying the phenomenon itself [35].

Unfortunately, in the zphot context, the PDF depends both on the measurement methods (and chosen internal parameters of the methods themselves) as well as on the underlying physical assumptions. In this sense, the definition of a PDF in the context of zphot estimation needs to be taken with some caution [36].

The factors affecting the reliability of zphot PDFs are: photometric errors, intrinsic errors of the methods and statistical biases. The PDF becomes, therefore, just a way to somehow compress the information contained in a single error estimate. In other words, the parametrization of a single error, through a probability, allows to cover an entire redshift range (with the chosen bin accuracy), thus leading to an increase of the information rate in order to match the precision required by a specific scientific goal (cf. for instance, the cases of the determination of cosmological parameters [37], and weak lensing measurements [38]).

Therefore, over the last few years, much attention has been paid to develop methods able to compute a full zphot PDF for both individual sources and entire galaxy samples [39–45].

The study of the PDFs and their properties (see Sect. 4) represents a useful tool to test zphot reliability. In fact, rejection criteria, aimed at removing unreliable zphot and PDFs estimates, as long as they are reproducible in the photometric space, can improve the precision of the results in terms of both NMAD and fraction of outliers for zphot estimates, as well as the quality of individual PDFs and their cumulative performances described by the statistical indicators discussed in Sect. 4. Of course, this comes at the cost of a decreased completeness.

3 Data

As spectroscopic knowledge base we used zspec for 136,057 galaxies extracted from the fourth Data Release (DR) of the ESO Public Kilo-Degree Survey (hereafter, KiDS-ESO-DR4, [46]) which combines data from KiDS and the VISTA Kilo degree Infrared Galaxy survey (VIKING; [47]).

KiDS is an optical survey of about 1350 deg^2 carried in 4 bands (ugri) with limiting magnitude $r=25.0 \text{ AB}$ (5σ in $2''$), i.e. 2.5 magnitudes deeper than the Sloan Digital Sky Survey (SDSS), in good seeing conditions ($\sim 0.7''$ median full width at half-maximum, FWHM, in the r band). KiDS has been complemented with the NIR photometry in the five bands Z, Y, J, H and Ks from the VIKING survey.

The survey is complemented by a set of spectroscopic observations available within the KiDS collaboration as well as from other surveys: COSMOS [48], zCOSMOS [49], CDFS [50], DEEP2 [51] and GAMA DR2 and DR3 [52, 53] fields public data.

The photometry used in this work consists of the 9 Gaussian Aperture and PSF (GAaP) magnitudes (u, g, r, i, Z, Y, J, H, Ks), corrected for extinction and zero-point offsets, and 8 derived colours, for a total of 17 photometric parameters for each object. After investigating the photometric parameter distribution for the sample, the data was cleaned by cutting the tails of the magnitude distributions in order to ensure a homogeneous distribution of training points in the parameter space.

The spectroscopic data set has been randomly shuffled and split in a 70% training set and a 30% test set (95, 261 and 40, 796 sources, respectively). We stress that objects in the test set used to evaluate and validate the trained model are not used during the training phase of the methods (see Sect. 4). In what follows, we used PDFs obtained using METAPHOR (Machine-learning Estimation Tool for Accurate PHOTometric Redshifts [54]) on KiDS-ESO-DR4 data (cf. [36] for details).

METAPHOR is a modular workflow designed to produce both *zphot* and related PDFs. The internal *zphot* estimation engine is MLPQNA (Multi-Layer Perceptron trained with Quasi-Newton Algorithm; [55, 56]) while for the *zphot* we used the *best-estimates* as defined in [36].

We just recall that the selected binning step in *zphot* used to estimate individual PDFs is 0.02 and the spectroscopic depth available is equal to $z_{\text{spec}}=7.01$. The z_{spec} distribution for the whole test set is shown in Fig. 1.

4 Methods

The *zphot* statistics are calculated on the residuals:

$$\Delta z = (z_{\text{spec}} - z_{\text{phot}})/(1 + z_{\text{spec}}) \quad (1)$$

using as *zphot* the *best-estimate* referenced in Sect. 3.

We use as accuracy estimators the mean (or bias), the fraction of catastrophic outliers, defined as those objects for which $|\Delta z| > 0.15$, and the normalized median absolute deviation (NMAD), defined as:

$$NMAD = 1.4826 \times \text{median}(|\Delta z - \text{median}(\Delta z)|) \quad (2)$$

The shapes of the PDFs, and therefore their intrinsic quality, can be characterized in terms of:

- *pdfWidth*: the width of the PDF in terms of redshift;

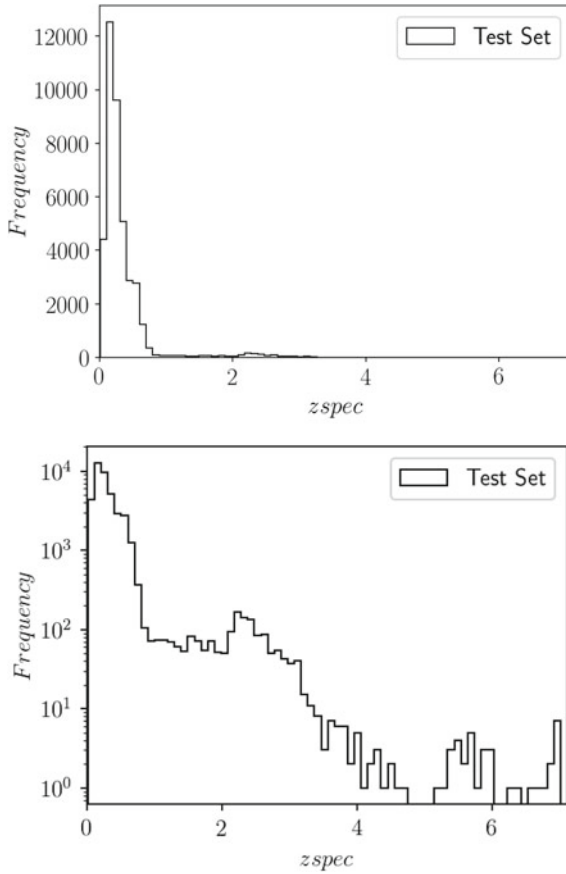


Fig. 1 The zspec distribution of the initial test set (40, 796 sources). The bottom diagram is the same top plot but in a logarithmic scale

- *pdfNBins*: the total number of bins of chosen amplitude (which defines the accuracy of the PDF itself), in which the PDF is different from 0;
- *pdfPeakHeight*: the amplitude of the peak of the PDF, i.e. the value of the maximum probability of the PDF.

The cumulative performance of the stacked PDF on the entire sample is instead evaluated by means of the following three estimators:

- $f_{0.05}$: the percentage of residuals Δz within ± 0.05 ;
- $f_{0.15}$: the percentage of residuals Δz within ± 0.15 ;
- $\langle \Delta z \rangle$: the average of all the residuals Δz of the stacked PDFs.

where by stacked PDFs we mean the individual zphot PDFs transformed into the PDFs of scaled residuals Δz and then stacked for the entire sample.

Furthermore, the quality of the individual PDFs is evaluated against the single corresponding *zspec* in the test set, by defining five types of occurrences:

- *zspecClass* = 0: the *zspec* is within the *bin* containing the peak of the PDF;
- *zspecClass* = 1: the *zspec* falls in one bin from the peak of the PDF;
- *zspecClass* = 2: the *zspec* falls into the PDF, e.g. in a bin in which the PDF is different from zero;
- *zspecClass* = 3: the *zspec* falls in the first bin outside the limits of the PDF;
- *zspecClass* = 4: the *zspec* falls out of the first bin outside the limits of the PDF.

Finally, we use two additional diagnostics to analyze the *cumulative* performance of the PDFs: the credibility analysis presented in [57] and the Probability Integral Transform (hereafter PIT), described in [58].

The credibility test should assess if PDFs have the correct *width* or, in other words, it is a test of the *overconfidence* of any method used to calculate the PDFs. In particular, the method is considered overconfident if the produced PDFs result too narrow, i.e. too sharply peaked, underconfident otherwise. The implementation of the credibility method is straightforward and is reached by computing the threshold credibility C_i for the i -th galaxy with

$$C_i = \sum_{z \in p_i \geq p_i(z_{\text{spec},i})} p_i(z) \quad (3)$$

where p_i is the normalized PDF for the i -th galaxy. The credibility is then tested by calculating the cumulative distribution $F(C)$, which should be equal to C . $F(C)$ is a q-q plot, (a typical quantile-quantile plot used to compare two distributions), in which F is expected to match C , i.e. it follows the bisector in the F and C ranges equal to $[0,1]$. Therefore, the *overconfidence* corresponds to $F(c)$ falling below the bisector (implying that too few galaxies have *zspec* with a given credibility interval), otherwise, the *underconfidence* occurs. In both cases, this method indicates the inaccuracy of the error budget [57]. *Overconfidence* and *underconfidence* are plotted in Fig. 2.

The PIT analysis measures how consistent are the predicted *zphot* and the true redshift (*zspec*) distributions, by calculating the histogram for the following probabilities:

$$p_i = F_i(x_i) \quad (4)$$

F_i in Eq. 4 is the cumulative distribution function (CDF) of the i -th object and $x_i = z_{\text{spec},i}$. The closer the histogram is to a uniform distribution, the better is the calibration between *zphot* and *zspec* distributions. A strongly U-shaped PIT histogram denotes a highly *underdispersive* character of the *zphot* distribution.

The visual inspection of a PIT can, therefore, shed light on the consistency between the *zspec* and *zphot* distributions. In particular, if the PDFs are too broad, then the relative PIT histogram appears overdispersed, that is with a peak in the centre of the histogram itself. In contrast, if the PDFs are too narrow, then the PIT is U-shaped and it results underdispersed. Finally, only when the widths of the PDFs agree with the

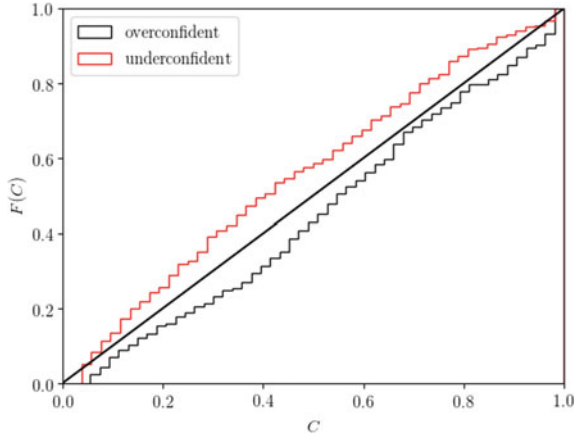


Fig. 2 Credibility analysis examples of *overconfidence*, represented by the black curve below the bisector of the plot $F(C)$ vs C , and of *underconfidence*, indicated by the red curve above the same bisector

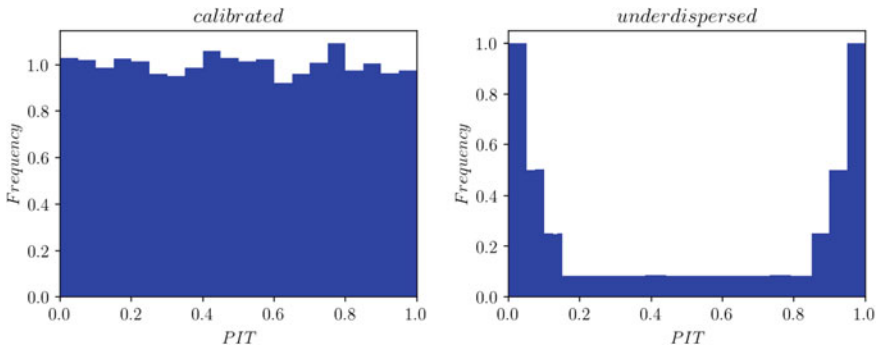


Fig. 3 Examples of well-calibrated (left) and underdispersed U-shaped PIT (right). The overdispersion is simply the opposite of the right panel, with a very broad PDF distribution, having a high peak in the centre of the diagram

discrepancies between *zphot* and *zspec*, then a uniformly distributed PIT histogram is produced. In Fig. 3 an example of well-calibrated and *underdispersed* PIT histograms is shown.

Credibility and PIT tests are complementary since both the underdispersion and the overconfidence are related to the narrowness of the PDFs. The narrower the PDFs are, the more the PIT histogram is underdispersed and the results of credibility are overconfident.

5 Results

The initial test dataset was composed by 40, 796 *zphot* estimates and relative individual PDFs. Among these, we have 970 outlier sources (2.4%), and 39, 826 non-outliers

(97.6%): outliers were singled out as explained in Sect. 4. We then proceeded with the visual inspection of the individual PDFs, i.e. their width, number of bins, the height of the maximum peak. To do so, we first derived a set of statistical descriptors, such as mean, standard deviation and the minimum and maximum values of the PDF shape properties, dividing the sample into outliers and non-outliers. These values are given in Tables 1 and 2, respectively. By comparing the mean values of such descriptors, the expected differences for the two populations become apparent: outliers have wider PDFs, with a higher number of bins (intervals of amplitude $\Delta z=0.02$), in which the PDF is not null and lower peaks with respect to those for non-outliers samples.

Table 1 Statistics of the three descriptors of the PDF shape: width, number of bins, and maximum peak height, defined in Sect. 4, for the outliers in the test set

PDF feature	Mean	σ	Min	Max
<i>PdfWidth</i>	3.94	2.33	0.1	7.02
<i>PdfNBins</i>	124.90	71.74	5	277
<i>PdfPeakHeight</i>	0.067	0.072	0.011	0.66

Table 2 Statistics of the three descriptors of the PDF shape: width, number of bins, and maximum peak height, defined in Sect. 4, for the non-outliers in the test set

PDF feature	Mean	σ	Min	Max
<i>PdfWidth</i>	1.23	1.82	0.040	7.02
<i>PdfNBins</i>	25.01	29.91	2	305
<i>PdfPeakHeight</i>	0.28	0.15	0.012	0.99

For the test set data, in Figs. 4, 5 and 6 we plot, respectively: (i) the histogram of the *pdfWidth* distribution, (ii) the scatter plot of *PdfPeakHeight* against *PdfWidth*, and (iii) *PdfNBins* versus *PdfWidth*, distinguishing outliers and non-outliers populations.

The inspection of these plots led us to define four data sets:

- In Fig. 4, we can see that a cut of objects with *PdfWidth* > 4 can remove a fraction of outliers $\simeq 1.1\%$ from the test sample. We, therefore, define a first (*Cut-1*) data set of objects with reliable PDF widths, using the condition:

$$pdfWidth < 4 \quad (5)$$

with which we come out with 36, 170 sources (of which 518 are outliers, and 35, 652 non-outliers, respectively, 1.4% and 98.6% of the total number of objects in the sample).

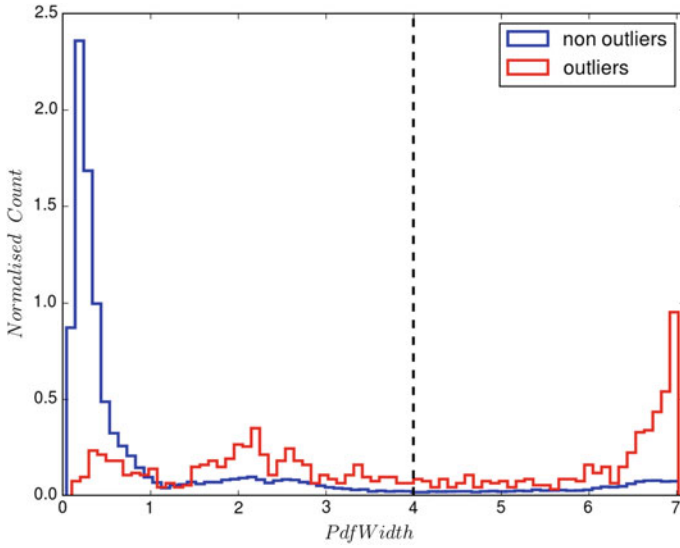


Fig. 4 PdfWidth normalized counts for outliers and non-outliers populations. Dashed vertical line: PdfWidth value equal to 4, identified as threshold for clipping outliers that populate the region with widths larger than 4 (*Cut-1*)

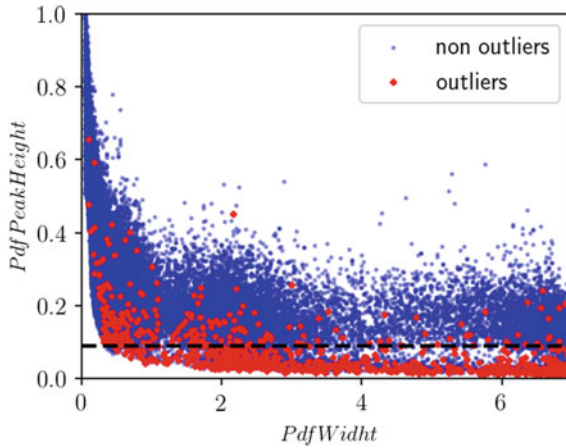


Fig. 5 Scatter plot of PdfPeakHeight versus PdfWidth. The dashed horizontal line indicates the PdfPeakHeight value, equal to 0.09, identified as threshold for removing outliers laying under the line (see *Cut-2* in the text)

- In Fig. 5, we show the PdfPeakHeight versus the PdfWidth. Outliers lay at the bottom of the scatter plot, thus allowing to define a second data set (*Cut-2*) via the condition:

$$PdfPeakHeight > 0.09 \tag{6}$$

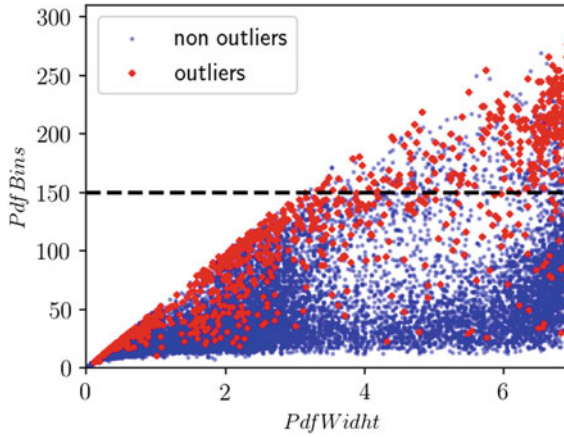


Fig. 6 Scatter plot $PdfNBins$ versus $PdfWidth$. The dashed horizontal line identifies the value of $PdfNBins$ equal to 150, useful to clip the outliers populating the region above this threshold (see *Cut-3* in the text)

The *Cut-2* data set contains a total of 38, 107 sources: 226 outliers (0.6%), and 37, 881 non-outliers (99.4%).

- In Fig. 6, we show the distribution of the descriptors $PdfNBins$ vs $PdfWidth$. As expected, the majority of the outliers rests in the region with higher values of PDF width and number of bins. This allows defining a third data set *Cut-3*, by rejecting objects with $PdfNBins > 150$. This third data set consists of 39, 905 sources, of which 591 are outliers (1, 5%) and 39, 314 non-outliers (98.5%).
- Finally, we derived a fourth data set (*Cut-4*) via the combination of *Cut-1* and *Cut-2*. This last data set contains 34, 802 sources, of which 196 (0.6%) are outliers and 34, 606 (99.4%) non-outliers.

Some additional tests showed that making more severe cuts would result in an uncomfortable loss in completeness. For instance, by producing an additional data set *Cut-5*, selecting objects with $PdfWidth < 1$, having less than 150 bins and a maximum peak height of at least 0.15, we reduced the test set by 34% (27, 297 out of 40, 796 sources).

5.1 *Zphot and Stacked PDF Statistics*

The results in terms of both *zphot* statistics and of *cumulative* PDF performance are reported in Table 3, for the test set and the four data sets corresponding to the different cuts. As we can see, the NMAD statistics is not different for the four adopted cuts, and there is only a slight improvement with respect to the whole data set. It has to be noted, however, that all cuts prove quite effective in reducing the fraction of outliers,

with an improvement of 39.8%, 75.0%, 37.7%, and 76.3% (for *Cut-1*, *Cut-2*, *Cut-3*, and *Cut-4*, respectively) with respect to the test set.

In the case of the data set extracted from *Cut-5*, we were left with a fraction of outliers $\simeq 0.2\%$. The *NMAD* is reduced to 0.013, and the fractions of residuals for the stacked PDF, $f_{0.05}$ and, $f_{0.15}$, increase to, respectively, 85.9% and 99.3%. This was expected, since the role played by *NMAD* and fraction of outliers for zphot point estimates, is analogous to the one of, respectively, $f_{0.05}$ and $f_{0.15}$ for the PDF. Another estimator that allows quantifying the reliability of the estimated PDF is the *zspecClass* flag as defined in Sect. 4. The results for *zspecClass* are reported in Table 4. As it could be expected, the best results in terms of fractions of *zspecClass* equal to 0 and 1, occur for the data sets *Cut-2* and *Cut-4*, which include the best scores in terms of zphot point estimates and of *cumulative* PDF performances (Table 3).

For the *Cut-5* data set, we achieve for classes 0 and 1 the scores of 31.0% and 42.8% respectively, and a smaller fraction of objects of class 3 (only 0.1%) with respect to the other data sets. Classes 3 and 4 quantify the number of objects falling outside the PDF. The distinction between the two classes gives the supplementary information about how far from the PDFs is their *zspec*.

In Fig. 7 we present the scatter plots of the zphot *best-estimates* as a function of the spectroscopic redshifts, for the test set and the four probed cut data sets. The mean and standard deviation of zphot are also plotted, in 40 evenly spaced *zspec* bins in a whole range of [0, 4.0]. Not all bins are populated, due to the reduction of the amount of samples resulting from the application of the rejection criteria, and the σ value in each bin increases in under-sampled bins.

Table 3 Statistics of the zphot and stacked PDFs for the whole test set sample and the four pruned data sets performed

Estimator	Test set	Cut 1	Cut 2	Cut 3	Cut 4
<i>bias</i>	-0.003	-0.004	-0.002	-0.003	-0.002
<i>NMAD</i>	0.016	0.015	0.015	0.015	0.015
<i>outliers</i>	2.4%	1.4%	0.6%	1.5%	0.6%
$f_{0.05}$	76.1%	79.2%	79.8%	77.5%	81.2%
$f_{0.15}$	94.6%	96.6%	97.4%	95.9%	98.0%
$\langle \Delta z \rangle$	-0.025	-0.016	-0.018	-0.008	-0.013

Table 4 *zspecClass* fractions for the whole test set and the four cuts.

<i>zspecClass</i>	Test set		Cut 1		Cut 2		Cut 3		Cut 4	
0	10652	(26.1%)	9930	(27.5%)	10535	(27.6%)	10631	(26.6%)	9834	(28.3%)
1	15476	(37.9%)	14224	(39.3%)	15214	(39.9%)	15430	(38.7%)	14084	(40.5%)
2	13893	(34.0%)	11353	(31.4%)	11727	(30.8%)	13115	(32.9%)	10271	(29.5%)
3	156	(0.4%)	90	(0.2%)	79	(0.2%)	115	(0.3%)	64	(0.2%)
4	619	(1.5%)	600	(1.7%)	552	(1.4%)	614	(1.5%)	540	(1.6%)

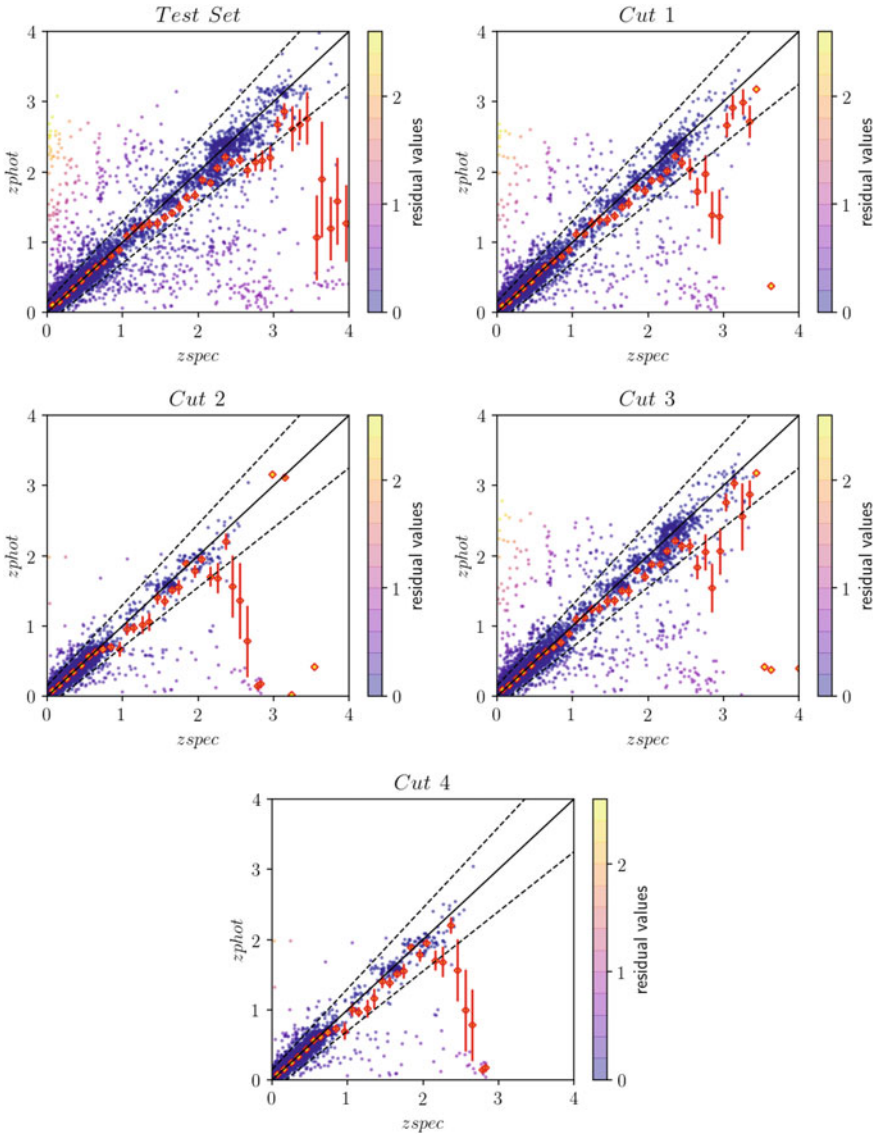


Fig. 7 Scatter plots of photometric redshift *best-estimates* as a function of spectroscopic redshifts. Red diamonds: mean of the z_{phot} *best-estimates* in 40 evenly spaced z_{spec} bins. Red bars: standard deviation of z_{phot} values populating the bins. Outliers and non-outliers are identified by the colour bar, showing the absolute values of the residuals (see Eq. 1)

It is interesting to notice the similar trends for data sets deriving from cuts 1 and 3, and cuts 2 and 4. This is expected for cut data sets 1 and 3 since, as we mentioned, PDF width and the number of bins in which PDF differ from 0, are highly correlated. In the case of cut data sets 2 and 4, being the cut data set 4 obtained by the joint application of cuts 1 and 2 conditions, the similar performance can shed light on the cut which drives the statistical performance outcome.

Of course, we should favour those rejection criteria that, leading to similar statistical performance, remove a smaller number of sources from the original data set. In other words, we should adopt rejection criteria corresponding to the best trade-off between precision and completeness.

Among the tested cuts, the best results in terms of precision and completeness are achieved for the data set *Cut-2*, which while showing comparable results (cf. Tables 3 and 4) to data set *Cut-4*, contains $\simeq 9\%$ more sources.

Finally, in Fig. 8, we show the stacked PDF for the test set and two cut data sets (*Cut-1* and *Cut-4*), along with the spectro-photometric redshift distributions of the test set. Note that the redshift range has been cut at $z = 1$, due to the low amount of objects in the test set above such value.

Besides the good agreement with the zphot distribution for the stacked PDF of the test set, we can see the effect of the rejection for the clipped data sets, which leads to a lower amount of sources at high redshift, and a more substantial amount at low redshift.

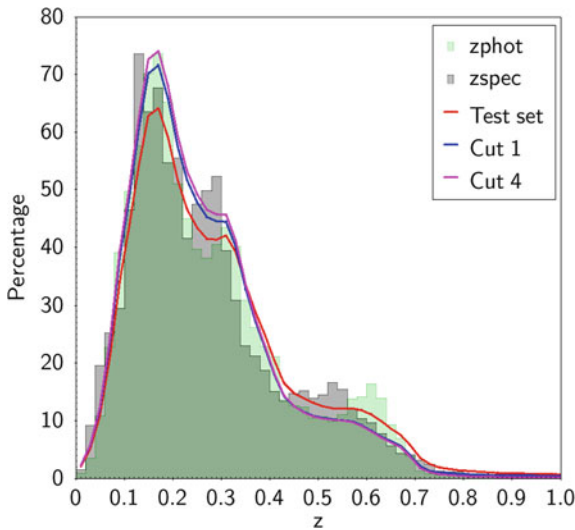


Fig. 8 Superposition of the stacked PDF (percentage) of the test set (red), for the *cut 1* (blue) and *cut 4* (magenta) data sets, and the *zphot best-estimates* (in green) distributions obtained by METAPHOR applied to the *zspec* distribution (in black), for the test set, limited to $z = 1$, due to the very few objects over such value

This is highlighted in Fig. 9, where the distribution of z_{spec} for the whole test set and the two clipped data sets *Cut-1* and *Cut-4* are shown. Moreover, in Fig. 13 in the Appendix, it is reported the z_{spec} distribution for the whole test set against the z_{spec} distributions for the tested four clipped data sets. As we can see more clearly from this figure, rejection is successful in removing outliers at higher redshift. Comparing the distributions for the *Cut-2* and *Cut-4* data sets, we can notice that the range of z_{spec} between 3 and 4 is more populated in the case of *Cut-2* data set with respect to *Cut-4* data set. This leads to the conclusion that, as anticipated above, *Cut-2* achieves a better trade-off between completeness and precision with respect to *Cut-4*. This is a clue of the effectiveness of the rejection in removing most of outliers at high redshift, i.e. in a region of the parameter space where the density of the training points is lower.

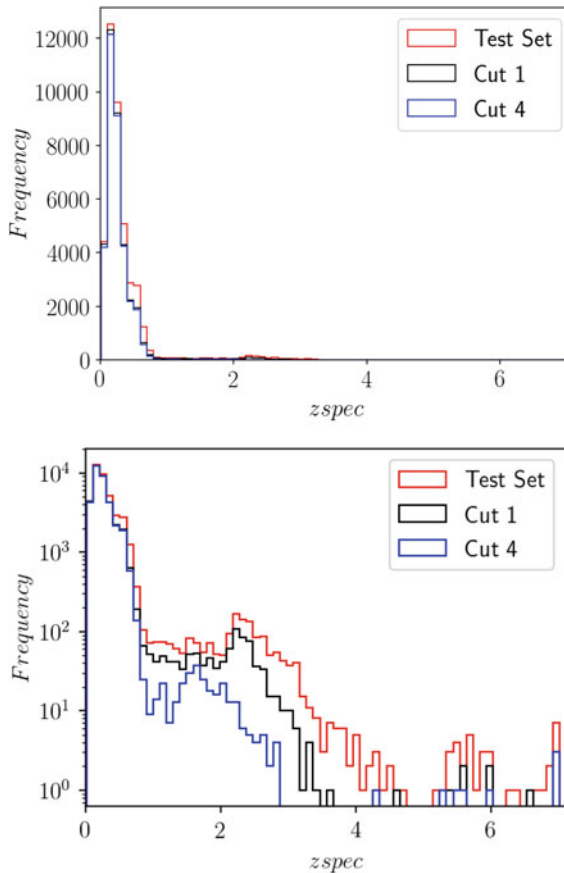


Fig. 9 *Top panel:* z_{spec} distribution for the whole test set (red), and the cut data sets *Cut 1* (black) and *Cut 4* (blue). *Bottom panel:* the same of the top panel in a logarithmic scale

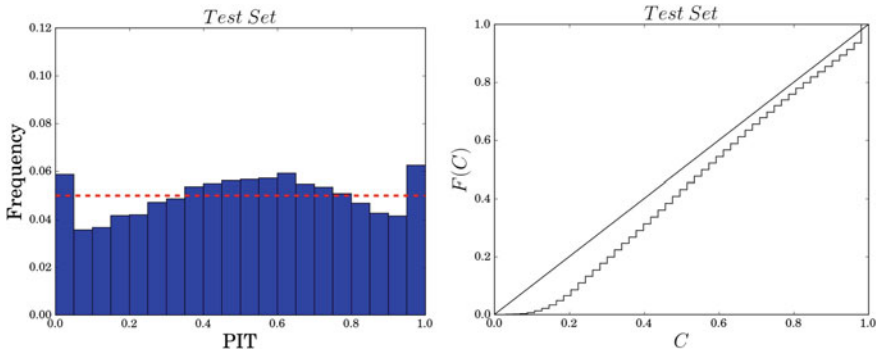


Fig. 10 PIT histogram (left panel). The red dashed line identifies the ideal PIT value, which represents the best calibration between the true *zspec* and the reconstructed *zphot* distributions. Credibility analysis (right panel) for the test set. The solid black line represents the best credibility for which the two distributions $F(C)$ and C are indistinguishable (see Sect. 4)

5.2 PIT and Credibility Analysis

PIT and credibility analysis for the test set are shown in Fig. 10. The PIT histogram shows a certain degree of *underdispersion* of the *zphot* distribution and the credibility plot stresses the *overconfidence* of the PDFs. The complementary information carried by these two *visual* diagnostics (see Sect. 4) is therefore confirmed.

In the plots of Figs. 11 and 12, we show, respectively, the credibility analysis for the four clipped data sets against the credibility of the test set, and the comparison of PIT and credibility for the data set *Cut-5*. Data sets obtained from cuts 1 and 4 show a slightly higher degree of *overconfidence* with respect to the test set, while cut data sets 2 and 3 show an indistinguishable credibility trend.

We stress that the PIT histogram fails to reveal differences between the four clipped data sets probed, with respect to the test set: for this reason, we do not show the relative plots. However, in the case of *Cut-5*, PIT shows a more significant degree of bias with respect to the test set, whereas the credibility shows a narrower shape, resulting in a larger *overconfidence* with respect to the whole test set.

6 Conclusions

In this work, we presented a method for defining low-quality *zphot* rejection criteria through the characterization of outliers, using the descriptors of the PDF shape (e.g. the width, the value of the maximum peak, etc.). The first step was, therefore, to compare the PDF descriptors for the two populations of outliers and non-outliers. Outliers appear to be characterized by wider PDFs with small maximum probability, as well as by a more significant number of bins in which the PDF differs from zero.

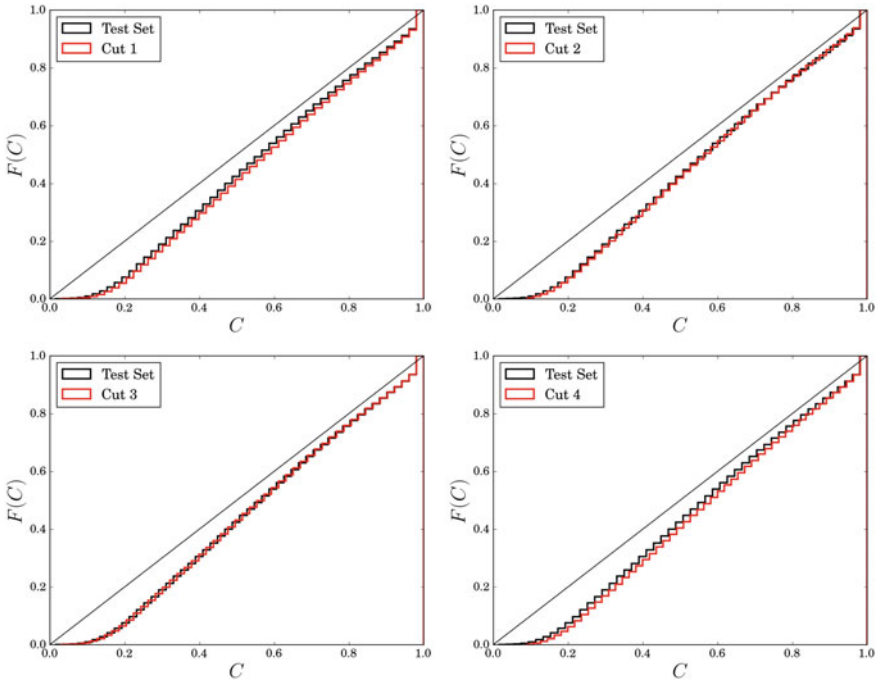


Fig. 11 Credibility analysis for the four tested cut data sets (red) against the whole test set credibility (black). In the top panels: *Cut-1* (left) and *Cut-2* (right). In the bottom panels: *Cut-3* (left) and *Cut-4* (right)

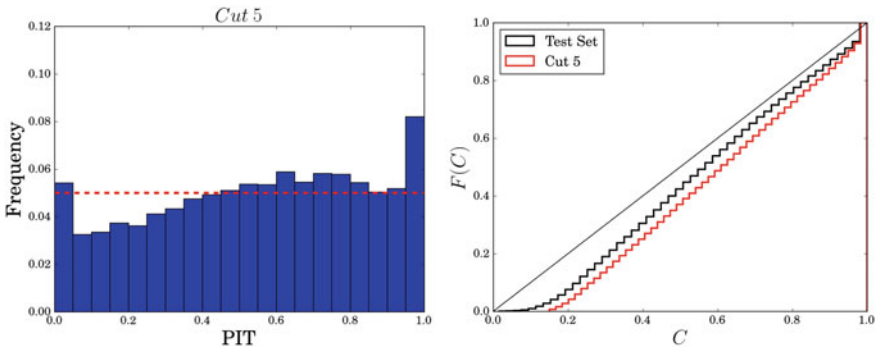


Fig. 12 PIT histogram (left panel) and credibility analysis (right panel) for the *Cut-5* data set, plotted against that of the whole test set

Zphot outliers tend to populate particular regions of the photometric parameter space and of the one defined by the PDF characteristics. Most outliers populate the top right part of a plot *PdfNBins* vs *PdfWidth*, where both the quantities are larger. Furthermore, the PDFs of the outliers have low maximum peaks, and populate a stripe

at low values of *PdfPeakHeight*, in a plane *PdfPeakHeight* vs *PdfWidth*. This allows the identification of cuts suitable to remove outliers, thus improving the precision on the clipped data sets.

We detailed the results for four different cut data sets obtained by applying rejection through the PDF width, the height of the maximum peak, and the number of bins in which the PDF is not null for, respectively *Cut-1*, *2*, *3* data sets. A further clipped data set (*Cut-4*) was created by removing outliers through the application of both the cuts used to generate *Cut-1* and *Cut-2* data sets. The best precision and completeness results were achieved for the data set *Cut-2*. This data set, in fact, from the one hand, shows comparable results to data set *Cut-4*, in terms of both zphot point estimate and *cumulative* PDF statistical performances. On the other hand, *Cut-2* data set contains $\simeq 9\%$ more sources than *Cut-4*, which mostly populate the spectroscopic region in the range [3, 4], as it is visible in Fig. 13.

Finally, we tested many others more strict rejection criteria, all of them leading to a severe loss of completeness with respect to the original data set. We reported for one of these pruned data set (*Cut-5*) the results throughout the Sect. 5, also showing the more biased PIT histogram and more *overconfident* credibility diagram with respect to the other four pruned data sets (see Fig. 12).

Although still not fully automated, the rejection approach is very general in its applicability, since it does not depend on the particular method used to calculate the PDFs. On the other hand, the overall quality of PDFs depends strictly on the particular method used to derive them. This last aspect is not discussed in this paper. However, we deem particularly useful a future comparison of rejections applied to PDFs obtained by different approaches (e.g. SED and ML methods referenced in Sect. 2). This with the final goal of further increasing the precision of the measurements.

As mentioned in Sect. 1, precision and completeness are both relevant quantities for matching the requirements of ongoing as well as future cosmological sky surveys, since the accuracy of the cosmological parameters strongly depends on an optimal trade-off between these two properties. The systematic study and automatic implementation of rejection can help to improve the precision, keeping a congruous number of non-outliers objects, thus preserving the completeness.

Acknowledgements Based on observations made with ESO Telescopes at the La Silla Paranal Observatory under programme IDs 177.A-3016, 177.A-3017, 177.A-3018 and 179.A-2004, and on data products produced by the KiDS consortium. The KiDS production team acknowledges support from: Deutsche Forschungsgemeinschaft, ERC, NOVA and NWO-M grants; Target; the University of Padova, and the University Federico II (Naples). SC acknowledges the financial contribution from FFABR 2017. GL acknowledges partial financial support from the EU ITN SUNDIAL. MB acknowledges financial contributions from the agreement *ASI/INAF 2018-23-HH.0*, *Euclid ESA mission - Phase D*. MB and CT acknowledged the *INAF PRIN-SKA 2017 program 1.05.01.88.04*.

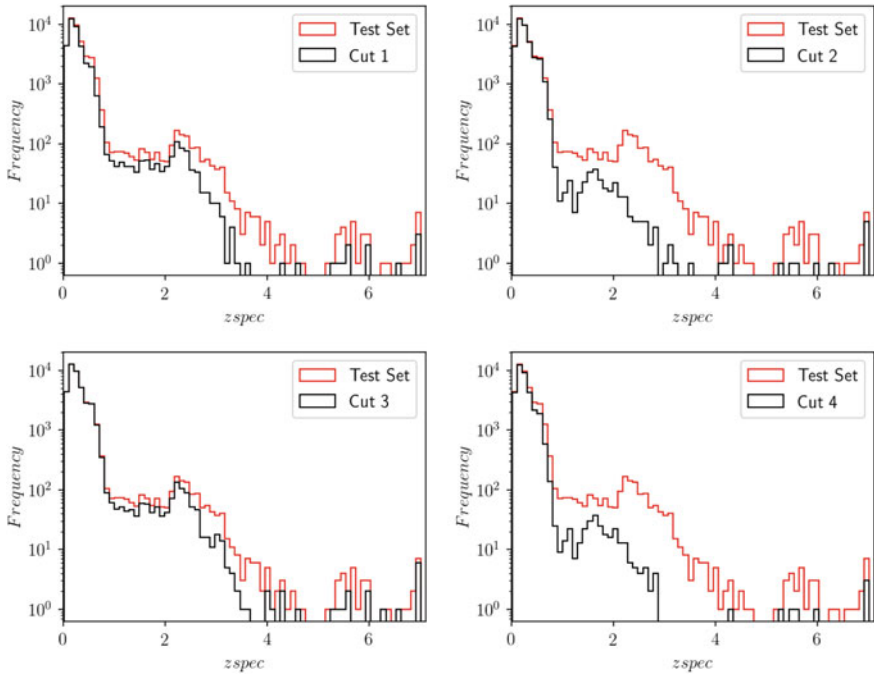


Fig. 13 z_{spec} distribution for the whole test set (red) and the four tested cut data sets (black), in a logarithmic scale. *Top panels:* *Cut-1* (left) and *Cut-2* (right). *Bottom panels:* *Cut-3* (left) and *Cut-4* (right)

Appendix

See Fig. 13

References

1. Serjeant, S.: *ApJ* **793**(1), L10 (2014). <https://doi.org/10.1088/2041-8205/793/1/L10>
2. Hildebrandt, H., Viola, M., Heymans, C., Joudaki, S., Kuijken, K., et al.: *MNRAS* **465**(2), 1454 (2017). <https://doi.org/10.1093/mnras/stw2805>
3. Fu, L., Liu, D., Radovich, M., Liu, X., Pan, C., et al.: *Mon. Not. R. Astron. Soc.* **479**(3), 3858 (2018). <https://doi.org/10.1093/mnras/sty1579>
4. Aragon-Calvo, M.A., Weygaert, R.v.d., Jones, B.J.T., Mobasher, B.: *Mon. Not. R. Astron. Soc.* **454**(1), 463 (2015). <https://doi.org/10.1093/mnras/stv1903>
5. Capozzi, D., de Filippis, E., Paolillo, M., D'Abrusco, R., Longo, G.: *MNRAS* **396**(2), 900 (2009). <https://doi.org/10.1111/j.1365-2966.2009.14738.x>
6. Annunziatella, M., Mercurio, A., Biviano, A., Girardi, M., Nonino, M., et al.: *A&A* **585**, A160 (2016). <https://doi.org/10.1051/0004-6361/201527399>

7. Radovich, M., Puddu, E., Bellagamba, F., Roncarelli, M., Moscardini, L., et al.: *A&A* **598**, A107 (2017). <https://doi.org/10.1051/0004-6361/201629353>
8. Brescia, M., Cavuoti, S., Paolillo, M., Longo, G., Puzia, T.: *Mon. Not. R. Astron. Soc.* **421**(2), 1155 (2012). <https://doi.org/10.1111/j.1365-2966.2011.20375.x>
9. Tortora, C., La Barbera, F., Napolitano, N.R., Roy, N., Radovich, M., et al.: *Mon. Not. R. Astron. Soc.* **457**(3), 2845 (2016). <https://doi.org/10.1093/mnras/stw184>
10. Masters, D., Capak, P., Stern, D., Ilbert, O., Salvato, M., et al.: *ApJ* **813**(1), 53 (2015). <https://doi.org/10.1088/0004-637X/813/1/53>
11. Gorecki, A., Abate, A., Ansari, R., Barrau, A., Baumont, S., Moniez, M., Ricol, J.S.: *A&A* **561**, A128 (2014). <https://doi.org/10.1051/0004-6361/201321102>
12. Ross, A.J., Banik, N., Avila, S., Percival, W.J., Dodelson, S., et al.: *Mon. Not. R. Astron. Soc.* **472**(4), 4456 (2017). <https://doi.org/10.1093/mnras/stx2120>
13. Baum, W.A.: In: *Problems of Extra-Galactic Research*, IAU Symposium, vol. 15, ed. by G.C. McVittie, p. 390 (1962)
14. Connolly, A.J., Csabai, I., Szalay, A.S., Koo, D.C., Kron, R.G., Munn, J.A.: *AJ* **110**, 2655 (1995). <https://doi.org/10.1086/117720>
15. Bolzonella, M., Miralles, J.M., Pelló, R.: *A&A* **363**, 476 (2000)
16. Arnouts, S., Cristiani, S., Moscardini, L., Matarrese, S., Lucchin, F., Fontana, A., Giallongo, E.: *MNRAS* **310**(2), 540 (1999). <https://doi.org/10.1046/j.1365-8711.1999.02978.x>
17. Tanaka, M.: *Astrophys. J.* **801**(1), 20 (2015). <https://doi.org/10.1088/0004-637x/801/1/20>
18. Tagliaferri, R., Longo, G., Andreon, S., Capozziello, S., Donalek, C., Giordano, G.: [arXiv:astro-ph/0203445](https://arxiv.org/abs/astro-ph/0203445), https://doi.org/10.1007/978-3-540-45216-4_26 (2000)
19. Firth, A.E., Lahav, O., Somerville, R.S.: *Mon. Not. R. Astron. Soc.* **339**(4), 1195 (2003). <https://doi.org/10.1046/j.1365-8711.2003.06271.x>
20. Ball, N.M., Brunner, R.J., Myers, A.D., Strand, N.E., Alberts, S.L., Tchenguiz, D.: *ApJ* **683**(1), 12 (2008). <https://doi.org/10.1086/589646>
21. Carrasco Kind, M., Brunner, R.J.: *Astronomical data analysis software and systems XXII*. In: D.N. Friedel (ed.) *Astronomical Society of the Pacific Conference Series*, vol. 475, p. 69. Astronomical Society of the Pacific, San Francisco (2013)
22. Brescia, M., Cavuoti, S., Longo, G., De Stefano, V.: *A&A* **568**, A126 (2014). <https://doi.org/10.1051/0004-6361/201424383>
23. Graff, P., Feroz, F., Hobson, M.P., Lasenby, A.: *MNRAS* **441**(2), 1741 (2014). <https://doi.org/10.1093/mnras/stu642>
24. Cavuoti, S., Brescia, M., Tortora, C., Longo, G., Napolitano, N.R., et al.: *MNRAS* **452**(3), 3100 (2015). <https://doi.org/10.1093/mnras/stv1496>
25. Cavuoti, S., Brescia, M., De Stefano, V., Longo, G.: *Experim. Astron.* **39**(1), 45 (2015). <https://doi.org/10.1007/s10686-015-9443-4>
26. Sadeh, I., Abdalla, F.B., Lahav, O.: *PASP* **128**(968), 104502 (2016). <https://doi.org/10.1088/1538-3873/128/968/104502>
27. Soo, J.Y.H., Moraes, B., Joachimi, B., Hartley, W., Lahav, O., et al.: *MNRAS* **475**(3), 3613 (2018). <https://doi.org/10.1093/mnras/stx3201>
28. D'Isanto, A., Cavuoti, S., Gieseke, F., Polsterer, K.L.: *A&A* **616**, A97 (2018). <https://doi.org/10.1051/0004-6361/201833103>
29. Fluke, C.J., Jacobs, C.: *WIREs Data Min. Knowl. Discov.* **10**(2), e1349 (2020). <https://doi.org/10.1002/widm.1349>
30. Hildebrandt, H., Arnouts, S., Capak, P., Moustakas, L.A., Wolf, C., et al.: *A&A* **523**, A31 (2010). <https://doi.org/10.1051/0004-6361/201014885>
31. Abdalla, F.B., Banerji, M., Lahav, O., Rashkov, V.: *Mon. Not. R. Astron. Soc.* **417**(3), 1891 (2011). <https://doi.org/10.1111/j.1365-2966.2011.19375.x>
32. Sánchez, C., Carrasco Kind, M., Lin, H., Miquel, R., Abdalla, F.B., et al.: *MNRAS* **445**(2), 1482 (2014). <https://doi.org/10.1093/mnras/stu1836>
33. Cavuoti, S., Tortora, C., Brescia, M., Longo, G., Radovich, M., et al.: *MNRAS* **466**(2), 2039 (2017). <https://doi.org/10.1093/mnras/stw3208>

34. Schmidt, S.J., Malz, A.I., Soo, J.Y.H., Almosallam, I.A., Brescia, M., et al.: [arXiv:2001.03621](https://arxiv.org/abs/2001.03621) (2020)
35. Brescia, M., Cavuoti, S., Amaro, V., Riccio, G., Angora, G., Vellucci, C., Longo, G.: [arXiv:1802.07683](https://arxiv.org/abs/1802.07683) (2018)
36. Amaro, V., Cavuoti, S., Brescia, M., Vellucci, C., Longo, G., et al.: *MNRAS* **482**(3), 3116 (2019). <https://doi.org/10.1093/mnras/sty2922>
37. Mandelbaum, R., Seljak, U., Hirata, C.M., Bardelli, S., Bolzonella, M., et al.: *MNRAS* **386**(2), 781 (2008). <https://doi.org/10.1111/j.1365-2966.2008.12947.x>
38. Viola, M., Cacciato, M., Brouwer, M., Kuijken, K., Hoekstra, H., et al.: *MNRAS* **452**(4), 3529 (2015). <https://doi.org/10.1093/mnras/stv1447>
39. Brammer, G.B., van Dokkum, P.G., Coppi, P.: *ApJ* **686**(2), 1503 (2008). <https://doi.org/10.1086/591786>
40. Ilbert, O., Arnouts, S., McCracken, H.J., Bolzonella, M., Bertin, E., et al.: *A&A* **457**(3), 841 (2006). <https://doi.org/10.1051/0004-6361/20065138>
41. Benítez, N.: *ApJ* **536**(2), 571 (2000). <https://doi.org/10.1086/308947>
42. Bonnett, C.: *MNRAS* **449**(1), 1043 (2015). <https://doi.org/10.1093/mnras/stv230>
43. Carrasco Kind, M., Brunner, R.J.: *MNRAS* **432**(2), 1483 (2013). <https://doi.org/10.1093/mnras/stt574>
44. Carrasco Kind, M., Brunner, R.J.: *MNRAS* **438**(4), 3409 (2014). <https://doi.org/10.1093/mnras/stt2456>
45. Carrasco Kind, M., Brunner, R.J.: *MNRAS* **442**(4), 3380 (2014). <https://doi.org/10.1093/mnras/stu1098>
46. Kuijken, K., Heymans, C., Dvornik, A., Hildebrandt, H., de Jong, J.T.A., et al.: *A&A* **625**, A2 (2019). <https://doi.org/10.1051/0004-6361/201834918>
47. Edge, A., Sutherland, W., Kuijken, K., Driver, S., McMahon, R., Eales, S., Emerson, J.P.: *Messenger* **154**, 32 (2013)
48. Davies, L.J.M., Robotham, A.S.G., Driver, S.P., Alpaslan, M., Baldry, I.K., et al.: *MNRAS* **452**(1), 616 (2015). <https://doi.org/10.1093/mnras/stv1241>
49. Lilly, S.J., Brun, V.L., Maier, C., Mainieri, V., Mignoli, M., et al.: *Astrophys. J. Suppl. Ser.* **184**(2), 218 (2009). <https://doi.org/10.1088/0067-0049/184/2/218>
50. Cooper, M.C., Yan, R., Dickinson, M., Juneau, S., Lotz, J.M., et al.: *Mon. Not. R. Astron. Soc.* **425**(3), 2116 (2012). <https://doi.org/10.1111/j.1365-2966.2012.21524.x>
51. Newman, J.A., Cooper, M.C., Davis, M., Faber, S.M., Coil, A.L., et al.: *ApJS* **208**(1), 5 (2013). <https://doi.org/10.1088/0067-0049/208/1/5>
52. Liske, J., Baldry, I.K., Driver, S.P., Tuffs, R.J., Alpaslan, M., et al.: *MNRAS* **452**(2), 2087 (2015). <https://doi.org/10.1093/mnras/stv1436>
53. Baldry, I.K., Liske, J., Brown, M.J.I., Robotham, A.S.G., Driver, S.P., et al.: *MNRAS* **474**(3), 3875 (2018). <https://doi.org/10.1093/mnras/stx3042>
54. Cavuoti, S., Amaro, V., Brescia, M., Vellucci, C., Tortora, C., Longo, G.: *MNRAS* **465**(2), 1959 (2017). <https://doi.org/10.1093/mnras/stw2930>
55. Brescia, M., Cavuoti, S., D'Abrusco, R., Longo, G., Mercurio, A.: *ApJ* **772**(2), 140 (2013). <https://doi.org/10.1088/0004-637X/772/2/140>
56. Brescia, M., Cavuoti, S., Longo, G., Nocella, A., Garofalo, M., et al.: *PASP* **126**(942), 783 (2014). <https://doi.org/10.1086/677725>
57. Wittman, D., Bhaskar, R., Tobin, R.: *MNRAS* **457**(4), 4005 (2016). <https://doi.org/10.1093/mnras/stw261>
58. Gneiting, T., Balabdaoui, F., Raftery, A.E.: *J. R. Stat. Soc. Ser. B* **69**(2), 243 (2007). <https://EconPapers.repec.org/RePEc:bla:jorssb:v:69:y:2007:i:2:p:243-268>

Large Astronomical Time Series Pre-processing for Classification Using Artificial Neural Networks



David Andrešič, Petr Šaloun, and Bronislava Pečková

Abstract During last years, several successful algorithms emerged for classification of time series from various areas of real world. But astronomical time series (a.k.a. light curves containing usually flux or magnitude on one axis and Julian date on the other axis) are a bit more challenging to classify. As they comes from multiple observational devices and observatories (designed for e.g. variable stars detection, stellar system analysis or extra-solar planets discoveries) that are usually located in outer space, they greatly vary in lengths, periods, noisiness and do not have clear borders between classes. Finding periods in these time series is therefore crucial for further research in this area. As these instruments produces huge amount of data (even Petabytes per observing night), we are facing big data issues and the analysis of the data requires an automated solution. In this chapter, we depict these issues on two publicly available data sets from BRITE and Kepler K2 projects and several well-performing algorithms, such as Convolutional networks, Long Short-term Memory and other Recurrent neural networks etc. We compare these approaches with various data pre-processing methods including e.g. statistic markers, periodograms or Fourier transformation for feature extraction as well as different means of data normalization and balancing. We also compare the affect of various activation functions used in the classification model. At the end, we present our own approach that includes the use of artificial neural networks (Multi-layer perceptron and Convolutional Neural Network) enhanced by genetic algorithm to find and learn the best performing classification model for pre-processed light curves with extracted features. Our approach is able to challenge the results of related work that includes these data sets.

D. Andrešič (✉)

VŠB - Technical University of Ostrava, 17. listopadu 2172/15, 708 00 Ostrava-Poruba,
Czech Republic

e-mail: david.andresic@vsb.cz

P. Šaloun

Palacky University Olomouc, Krizkovskeho 511/8, 771 47 Olomouc, Czech Republic

e-mail: petr.saloun@upol.cz

B. Pečková

Slovak university of technology in Bratislava, Ilkovičova 2, 842 16 Bratislava, Slovakia

e-mail: xsuchanovab@stuba.sk

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

I. Zelinka et al. (eds.), *Intelligent Astrophysics*, Emergence, Complexity
and Computation 39, https://doi.org/10.1007/978-3-030-65867-0_12

1 Introduction

Time series classification in astronomy can be useful in many areas. In this chapter, we focus on so-called light curves: flux or magnitude values measured in some (often irregular) period of time. This can be used to detect variable stars (of many physical classes), extra-solar planets or maybe (in future with more powerful astronomical hardware) extra-solar moons orbiting planets, stellar systems analysis and other physical phenomena.

Since these data comes from various automatized astronomical instruments that produce huge amount of data (even petabytes per night [40]), we are facing big data issues. To handle classification of this amount of data, an automated solution is crucial. In this chapter, we present an overview of current algorithms used for time series classification in general as well as different variations of algorithms based on the artificial neural network as a mean of machine learning to test how they perform with astronomical light curves. We also propose a network enhanced by genetic algorithm capable to challenge current top results on similar data sets. We also compare the tested approaches with various feature extraction techniques such as statistics markers or e.g. Fourier transformation.

1.1 Types of Stars Variability

In the Universe, most of the stars are actually variable in their luminosity. Describing physical reasons for this variability is beyond the scope of this chapter, so we offer at least a brief overview of classes used in our training data:

- *Delta Scuti*—young pulsating stars that are similarly as *cepheids* used as a so-called *standard candle* to measure distances between galaxies.
- *Detached Eclipsing Binary*—when both companions in a binary star system have no significant gravitational effect to each other.
- *Semi-Detached/Contact Eclipsing Binary*—when one of the companions in a binary star system is affected by the gravitational pull of the other one (but not vice versa) which actually transfers its gas to itself (accretion).
- *Gamma Dor*—young pulsating stars with a not very clear physical cause.
- *RR Lyrae ab*—pulsating stars typically with half of mass of our Sun used as *standard candles*.
- *Other Periodic/Quasi-Periodic*.

In our training data, other stars are labelled as noise. An example of how such time series for these classes looks like is depicted on Fig. 1. On x -axis, there is always time (usually some variation of *Julian Date*). On y -axis, there is usually a flux defined as the total amount of energy that crosses a unit area per unit time¹ (see Eq. 1).

¹According to *COSMOS—The SAO Encyclopedia of Astronomy*: <http://astronomy.swin.edu.au/cosmos/F/Flux>.

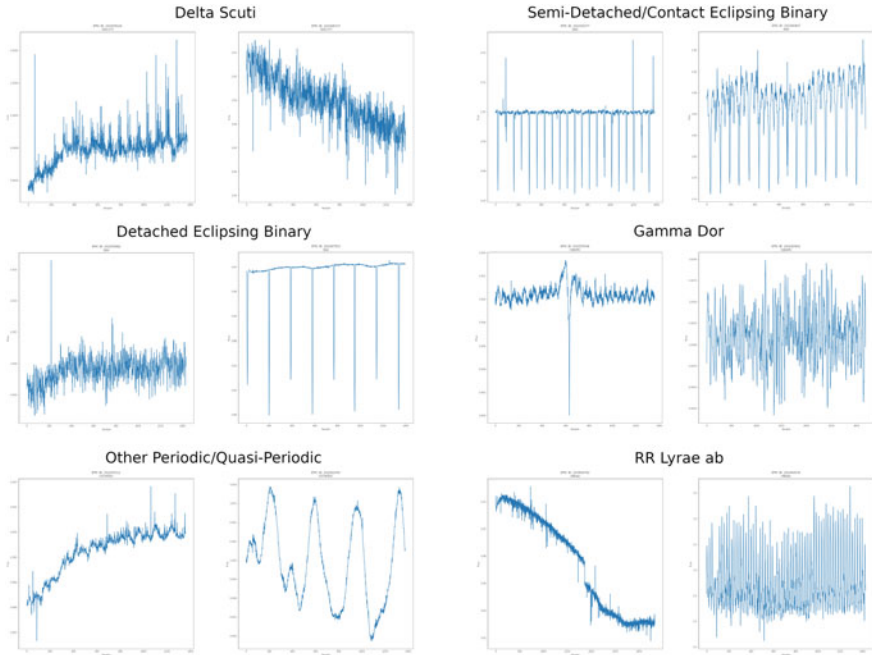


Fig. 1 Example light curves from Kepler K2 data set. Left one from the pair is always with low confidence in class (under 50%), right ones have high confidence in class (over 90%)

$$F = \frac{L}{4\pi r^2} \quad (1)$$

Where F is the flux at distance r , L is the luminosity of the source star, and r is the distance between Earth and the source star.

2 State of the Art

Time series classification is a bit specific because the classifier works with sorted data where even in the order of the data significant markers for some class can be hidden. Most literature on time series classification assumes following [1, 23]:

- copious amounts of perfectly aligned atomic patterns can be obtained,
- the patterns are all of the equal length,
- every item that we attempt to classify belongs to exactly one of our well-defined classes.

For such time series, the classification using Nearest Neighbour algorithm with the relatively expensive Dynamic Time Warping as a distance measure function usually performs best as a machine learning approach [32]. But these assumptions are challenging in astronomical time series since they greatly vary in lengths, periods, noisiness and are without clear borders between classes. In [26], authors also concludes that Long Short-Term Memory is another state-of-the-art technique for this task.

2.1 Time Series Classification Methods

During the last years, several successful machine learning methods emerged for time series classification. They are often bench-marked using UCR Time Series Archive [12] made in 2002 by the University of California. It is a set of data sets from different domains that is continuously extended, and today it contains 128 data sets. Over 1000 papers were published using this archive until now, but it may not be so conclusive since it heavily depends on experiment configuration details [5]. Nevertheless, authors in [14] shown that on this data set COTE and HIVE-COTE performs best (followed by ResNet and FCN).

2.1.1 Traditional and Other Machine Learning Methods

Although we focused on artificial neural networks, we also come with a brief overview of current methods and algorithms without them to summarize current state-of-the-art approaches.

Dynamic Time Warping (DTW)

This algorithm was introduced in 1978 [35] for speech recognition. It defines a distance between two time series, which is a metric that can be used with K-nearest neighbour. Multiple variations of this algorithms have emerged since then, but with an only marginal improvement of its original results [38]. The greatest advantage of DTW is that in opposite to a traditional Euclidean distance metric, it uses a mapping between two structurally similar points which makes this metric independent [13]. It should therefore be able to identify stars of the same variability class even in case when the stars were not in the same time in the same phase. Another advantage should be the ability to discover a similarity even with different densities of measurements (or speed of change) which means that it could discover a similarity of time series of two variable stars with different gaps between measurements (“sample rate”). It also means that it should be able to discover a similarity even in case of different periods. Authors in [5] compare similar methods of time series classification, and it turned out that they did not achieve significantly better results than DTW.

Since we have labelled data, we prefer supervised learning, which is why we did not consider this method. The metric itself could be useful for data pre-processing

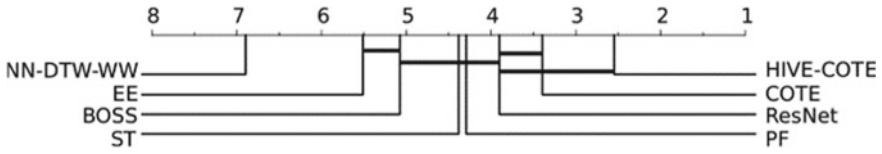


Fig. 2 Critical difference diagram comparing eight classifiers over UCR and UEA archives. *Source* [14]

(e.g. for establishing an average distance of variable/non-variable stars as one of extracted attributes).

Back of SFA Symbols (BOSS)

Introduced in 2015 by Patrick Schäfer [38]. The algorithm is based on a comparison of patterns extracted from the time series and contains three steps:

1. separation of time series to blocks of the same length (sliding windows),
2. transformation of each block using Symbolic Fourier Transformation (SFA),
 - a. a discrete Fourier transform is applied to each block,
 - b. then a symbolic representation of each block is created (SFA word),
3. a construction of the BOSS histogram for identifying structural similarities in time series.

BOSS algorithm includes a noise reduction. It is also fast and successful in classification. The author claims that it is able to supersede the best classification algorithms of the time and that one of the modifications could reach the top results on UCR data. But in experiments in [5] it performed a little bit worse.

Transformation of time series could be useful for us as well as a part of data pre-processing. We could also use SFA words themselves or BOSS histograms as inputs for the artificial neural network.

Collective of Transform-based Ensembles (COTE)

First introduced in [4] where authors designed a method that collects several time series classifiers in various domains, related transformations to these domains and metrics that evaluates individual classifiers outputs.

Authors in [4] claim that COTE reaches significantly higher accuracy than other known algorithms (those based on artificial neural networks were not covered). This statement can be supported by [5]. COTE was also a part of experiments in [14] where it was compared with seven other algorithms. Results are depicted on Fig. 2 and COTE took second place with statistically insignificant difference from the winner.

2.1.2 Methods with the Use of Artificial Neural Networks

Methods that utilize artificial neural networks were the main aim of this work. We bring a brief overview of those that we considered as they are usually performing

best for time series from various areas. In [13] authors also proved that artificial neural networks could supersede other time series classification algorithms. We can name for example Convolutional network [28], Fully Convolutional Network [13, 26], Residual Network [26], Multi-scalable Convolutional Network [11, 13] or Long Short-term Memory (Recurrent) Network [13, 26].

Multi-layer Perceptron

Consists of interconnected layers containing artificial neurons. Neurons in two layers are connected by weighted connection. There is no connection between neurons of the same layer. The output of each neuron consists of a sum of bias and weighted inputs processed by an activation function. It is trained by a traditional backpropagation algorithm. A more detailed description of MLP can be found in e.g. [33].

In [11] authors attempts to find how each training parameter affects the classifier performance, especially in variable stars classification. According to them, the training speed has no significant effect on the accuracy, but the size of internal layers has. They also experimented with layers from 4 to 18 neurons, and it turned out that those with 4 and 8 neurons performed best. But the main aim of their work was to compare MLP, KNN, SVM and RF in a field of variable stars classification. They attempted to establish a specific classifier for each class, and the final class was assigned as a combination of all classifiers outputs. From these, MLP turned out to be somewhere in the middle.

In [5] authors compares algorithms for time series classification that emerged after 2010. It also contains MLP and Naive Bayes (NB), logistic regression, SVM with linear (SVML), quadratic kernel (SVMQ), random forest (RandF) and rotation forest (RotF), 1-nearest neighbour with Euclidean distance (ED) and WEKA C4.5 (C45). From these, MLP achieved better results than DTW, RotF and RandF while it achieved worse results than BN, NB, C45 and logistic regression.

Convolutional Network

Although first convolutional networks were designed for image recognition [15, 29, 30], today they are commonly used in other domains such as speech recognition and processing [6, 42] or time series analysis [16].

A typical convolutional network consists of convolutional, pooling and fully-connected layers (as depicted on Fig. 3). A convolutional layer consists of several

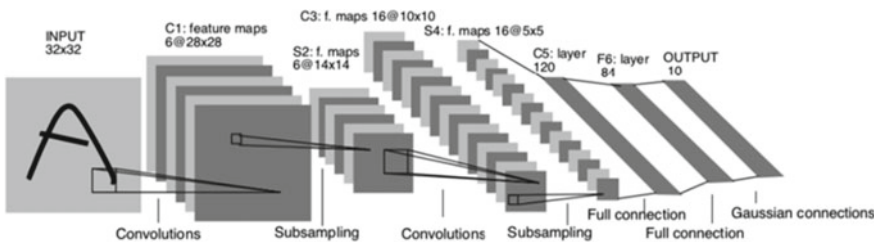


Fig. 3 Convolutional neural network architecture. *Source* [30]

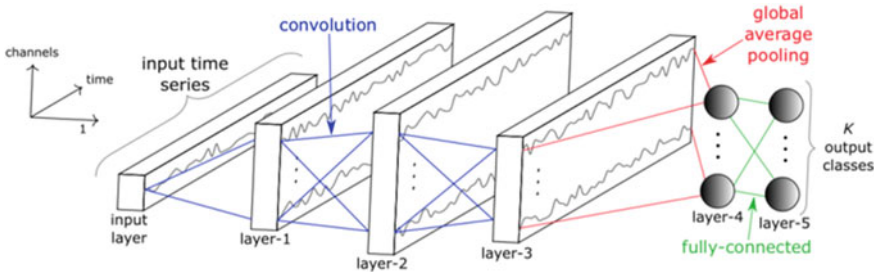


Fig. 4 Convolutional neural network architecture for time series classification. *Source* [14]

convolutional filters attempting to detect patterns (e.g. shapes, gradients or even more complex structures like hairs in the image processing domain). The filter is implemented as a matrix used for multiplication of input image. Outputs of the convolutional layer are then accumulated in pooling layer made for simplification of inputs their dimensionality reduction. At the end of the network, there is a fully-connected network similar to MLP.

In time series domain the convolution can take place in the application of its filter on time series areas. The difference from image processing is that it is applied to one-dimensional inputs. This operation can also be understood as a non-linear transformation of the time series. For example, in case of convolution of time series by filter $[\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}]$ we talk about moving average with window of 4. A visualization of a convolutional network for time series classification is depicted on Fig. 4.

In case of traditional algorithms, it is usually necessary to take care of intense data pre-processing [28]. The great advantage of a convolutional network is that it requires only minimal modifications of input data for the classifier. This is due to convolutional filters that do the pre-processing on their own. According to [14] the CNN is the most used type of artificial neural network for time series classification, most probably for its robustness a relatively good training speed in comparison to MLP and RNN.

Fully Convolutional Network

The architecture of Fully Convolutional Network (FCN) is similar to CNN. The only difference is that in the last layer, there is (instead of fully connected layer) Global Average Pooling (GAP) layer and that it does not contain local pooling layers, so the time series length remains the same during the processing by the entire network. Using GAP significantly reduces the number of parameters for the network. GAP performs dimensionality reduction of a tensor with dimensions $h \times w \times d$ to $1 \times 1 \times d$ by reducing each $h \times w$ matrix to the arithmetic average of all of its values. Please note that while in image recognition we work with $h, w > 1$, in time series domain one of these parameters is equal to one prior the reduction.

The advantage of FCN is that it does not require any significant data pre-processing or features extraction [45] and it can even be used as a mean of pre-processing [26].

FCN achieves great results in the domain of semantic segmentation of images [45] and also in the field of time series classification [18, 26]. The first use of FCN for time series classification is described in [45]. Authors compared FCN, MLP and ResNet with DTW, COTE, BOSS and several other algorithms. They used UCR archive for comparison (with 44 data sets in that time). Data pre-processing contained only z-normalization, and for FCN and ResNet training the Adam Optimizer was used. Four metrics were used to evaluate results: the arithmetic average of errors across all datasets, geometric average of errors across all data sets, number of data sets where the given algorithm performed with the lowest error and *Mean Per-Class Error (MPCE)*:

$$MPCE = \frac{1}{K} \sum PCE_k \quad (2)$$

where $PCE_k = \frac{e_k}{c_k}$ where e_k is classification error on k th data set, c_k is the number of classes in k th data set and K is number of data sets. FCN performed best using these metrics (in 18 of 44 data sets in reached lowest classification error).

Authors in [14] compared current time series classification algorithms. They experimented with 5-layered FCN with ADAM optimizer with learning factor 0.001 and Entropy cost function. Details of parameters and results can be found in [14], but we can conclude that FCN performs worse than ResNet and better than other seven classifiers.

FCN was also included in experiments in [41] although this work mostly aims to recurrent neural networks in time series classification domain. FCN with MLP were included in these experiments just to compare the accuracy. It turned out that FCN performs much better than all recurrent networks in these experiments (simple recurrent network and LSTM).

Residual Network

In general, adding more neurons and layers should increase the approximation accuracy. The problem is that when the network architecture grows, then the backpropagation algorithm faces the vanishing (or exploding) gradient issues. This problem was identified in [20]. Removing this weakness was the major aim of Residual network (ResNet) in order to allow effective network training for deep learning [18]. Another great advantage is (similarly to CNN and FCN) the need for very little data pre-processing [26].

ResNet performs very well in image recognition domain. In [18], authors designed a contest-winning network that on ImageNet data set [34] reached only 3.5%. It also turned out that ResNet can be successfully applied to time series classification. In [45], authors experimented with various types of networks and data sets, and ResNet achieved the lowest classification error in 8 of 44 data sets. In arithmetic average, it took fourth place and in geometric average, it took fifth place (of 11 algorithms).

Elman's Network

Elman's network is one of the simplest recurrent neural networks (RNN—they store an information about their previous activations in internal memory so the information

can also spread among neurons in the same layer; successfully used for sequence data processing [13]). The main difference between MLP and Elman's network is that Elman's network is enhanced with context layer that stores information about activation of neurons from the previous iteration. These activations then affect the output of the layer using recurrent links. These links exist only between i th neuron of the context layer and i th neuron of the internal layer and have weight equal to 1. The network is trained using real-time recurrent learning method [46].

In [41] authors attempts to use recurrent neural networks in time series classification domain. They compare different neural networks using *UCR Time Series Archive* [12]: MLP, FCN, LSTM and several other types of recurrent networks. They divided recurrent networks into two groups: those with a dense layer on the output and those with a recurrent layer. Their results for recurrent networks are not very conclusive because they did not reach a significantly higher accuracy than random classifier. They conclude that according the Wilcoxon signed-rank test:

- replacing recurrent layer by LSTM layer leads to better accuracy,
- adding the third layer leads to no significant improvement of accuracy,
- replacing dense layer by the recurrent layer has no significant impact on the accuracy,
- an increasing number of neurons in the one-dimensional recurrent network from 128 to 256 leads to worse classification accuracy.

Long Short-Term Memory

The main motivation to create LSTM [21] was to be able to effectively model dependencies in large time series and also to eliminate problems with vanishing and exploding gradients [26]. The strength of LSTM comes from the regulation of spreading of the activation by gates. These gates regulate input, output and the internal state of the LSTM cell. Each LSTM cell consists of three gates: input, forget and output. The forget gate decides which information will be removed from the internal memory. Input gate filters cell inputs and output gate decides what input values and values from internal layer will be sent to the output of the cell.

LSTM was also a subject of experiments in [41] on *UCR Time Series Archive* [12] where it achieves better accuracy than standard RNN, but worse than FCN. Authors also concluded that increasing the number of neurons in a layer from 128 to 256 had no significant effect on classification accuracy. Authors in [19] attempted to classify astronomical time series using LSTM and other methods. Their experiments were evaluated using *Balanced Accuracy*:

$$\frac{\frac{TP}{P} + \frac{TN}{N}}{2} \tag{3}$$

where TP is a count of correctly classified positive samples and TN is a count of correctly classified negative samples. P is a count of positive samples and N a count of negative samples. Authors were surprised that LSTM performed badly in their experiments using these metrics. They conclude that the *Balanced Accuracy* for LSTM was only 52%.

It seems that noisy time series are quite a challenge for LSTM and RNN in general. A possible solution can be inspired by [17], where the authors used a conversion into a symbolic representation with a self-organizing map.

LSTM Fully Convolutional Neural Network

Introduced in [26] and designed to extend FCNN by LSTM module. Authors state that their solution significantly improves the performance of FCNN with only a small increase of the model. They also state that their solution requires only minimal data pre-processing. They also conclude that their method super-seeds other state-of-the-art methods.

2.2 Other Related Work

In [36] authors performed multi-class classification (instead of our binary one) for the data from the original Kepler mission (“K1”) that are very similar to ours. Their original results were similarly poor as ours and among others concludes that LSTM is not suitable for Kepler data. They achieved the best results with feature extraction. The results are very similar to those described in [7] (again experimented on original Kepler data).

There is also a Kaggle² competition aiming at original Kepler data, but although it promises high accuracy, it uses highly unbalanced data set and achieved results are therefore not very informative.

In [22] authors works with a totally different light curves data set, but conclude that feature extraction (in their case a set of seven statistical markers) is a must-have for astronomical time series. They also work purely with time series without any additional meta-data describing the stars themselves. Based on just this information, they conclude that it is not possible to perform good-performing multi-class classification for most of the variability classes. Using Random Forest, Decision Trees and kNN algorithm, they reached up to 70% accuracy, which they suspect is caused by an imbalanced data set.

3 Data Sets

In this section, we briefly describe data sets we used for experiments with various classification algorithms. From publicly available, real-world data sets such as *All Sky Automated Survey for Supernovae (ASAS-SN)*, *MACHO Project*, *Microvariability & Oscillations of Stars (MOST)* we eventually chose *BRITE* (enhanced by variability data from *GCVS* catalogue) and NASA’s *Kepler* (mission *K2*) as they provide data sets in a shape suitable for machine learning (ML).

²Mystery Planet (99.8%, CNN): <https://www.kaggle.com/toregil/mystery-planet-99-8-cnn>.

Real light curves are quite challenging for ML classification. They are very often noisy due to various physical reasons or due to contamination by other signals. In [19], authors for example filtered out those light curves with large contamination from the neighbouring stars or those with total measured flux or a flux yield significantly lower than the object's total flux. The sampling rate varies, and especially in case of e.g. BRITE data set, we can see that whole parts of the time series are missing. Such sparseness is suspected to be one of the reasons why classification using ML does not work very well [7]. Another suspect for ML classification poor results is the density of the data [7], which is why we selected in case of K2 data an original pre-processed data set with 'smoothed' light curves. This data set also removes incorrect values from the light curves caused by instrumentation orientation change performed by on-board thrusters fired during the exposure time. This provides more consistent light curves, but on the other hand, it brings in another issue which is sparseness. Kepler data are also a bit specific as they contain time series measured in 2 so called cadences where each has a different sampling rate that affects the density of the data. Some authors overcome this by simply ignoring them [19]. As we can see, there are many challenges, which is why there are very often used additional metadata like those available in K2 FITS files that come with photometric data and physical properties of each measured star [7, 19]. This—side by side with feature extraction—is the usual ML framework for Kepler data set. In our work, we focused only on raw light curves and labels established by a respected third party without these meta data.

3.1 BRITE

Data from BRITE project—a group of nanosatellites on lower orbit launched in 2013 and 2014. This data set is made of tabular ASCII files containing (among others) Heliocentric Julian Date and Flux (ADU/s). There are 1119 light curves within this data set. According to GCVS catalogue [37], 601 of them are of variable stars, 279 not variable and 239 cannot be identified by cross-matching in GCVS catalogue.

Besides the fact that variability data comes from a 3rd party archive, the greatest disadvantage of this data set is its variability in sampling rate (in some cases, intervals between samples are less than 1 ms on one side and approximately 60 days on the other side). Another disadvantage is that number of samples in each light curve varies (some light curves have less than 10 measurements, while others have tens of thousands of measurements). The average length of the light curve is 12642 of samples [1].

3.1.1 General Catalogue of Variable Stars (GCVS)

GCVS [37] is a list of variable stars. Its first version contained 10820 stars and was released in 1948. Since then it was updated several times and today it contains

52011 variable stars (version 5.1 released in 2015). This catalogue allows a cross identification of stars based on their IDs in various catalogues.

Since BRITe data set does not come with information about star variability, we used this catalogue to add information about variability by cross-matching its ID. By this, we were able to obtain a label (7 classes of star variability) for 1119 light curves.

3.2 *Kepler K2*

NASA's mission to search Earth-like extra-solar planets in our Galaxy. Data from the K2 mission that are subject of this work, are publicly available [2, 3] and well documented [10, 25]. In this work we are interested in Kepler K2 light curves containing flux of individual objects in time. For these data, Kepler K2 also provides official catalogue of confirmed variable objects that we can utilize. Based on sampling frequency, we distinguish two cadency groups: long with 1765.5 s (29.4 min) and short with 58.89 s. On each Thursday, more than 160000 objects with long cadency and 512 objects with short cadency were measured and archived. The minimal length of measurement was 1/4 of year for long cadency and one month for short cadency (with the exception of Q4 where module 3 objects were lost due to hardware failure). Light curve file is in the form of time series where all undefined values are represented as NaN (not a number). As a result, we can obtain about 40000 light curves with a length up to 1300 measurements [1].

As mentioned before, we used original but corrected K2 data that excludes observations during thruster firings [44]. The resulting data are more smooth and without irrelevant samples (although with some sparseness). The difference between raw and corrected data is depicted on Fig. 5.

3.2.1 Similarity with Kepler "K1" Data

Kepler mission "K2" followed the original NASA's Kepler mission with the same purpose after the instrument stabilization failure. For this reason, these data sets are very close to each other in terms of the content. Original Kepler data also contains light curves containing flux values with two cadences: long with 29.4 min and short with 58.89 s sampling frequency.

Currently, more research seems to be done on original ("K1") Kepler data, but due to the data similarity, we include their results in our work as well. In [19], authors performed multi-class classification but with poor results (balanced accuracy 52%). They confirm that LSTM does not perform very well for Kepler data (they discuss that it was either due to the limited positive sample size within our data or the sparseness and/or noisiness of real data) and they achieved best results with a use of significant attributes extraction (with a balanced accuracy 74.7%). This conclusion was basically confirmed by experiments performed in [7] where authors performed also multi-class

classification of 150000 objects into 14 variable star classes reaching up to 65–70% accuracy. Authors also mention previous research on Kepler data achieving up to 55% accuracy.

4 Artificial Neural Networks Approaches

We decided to compare several methods of time series classification that utilizes artificial neural networks. Our primary goal is to find the best method that will classify time series (light curves) at least binary in the sense of object variability: the light curve contains some period (and is therefore of a variable object) or whether there is no period found (an object is not variable) [1] as much as the data will allow us. We started with the following approaches:

- multi-layer perceptron classification with own activation function,
- the recurrent neural network of type LSTM,
- multi-layer perceptron with own activation function in combination with time series pre-processing using Fourier transformation,
- the recurrent neural network of type LSTM in combination with time series pre-processing using Fourier transformation,
- multi-layer perceptron classification with own activation function in combination with some method of significant attribute extraction,

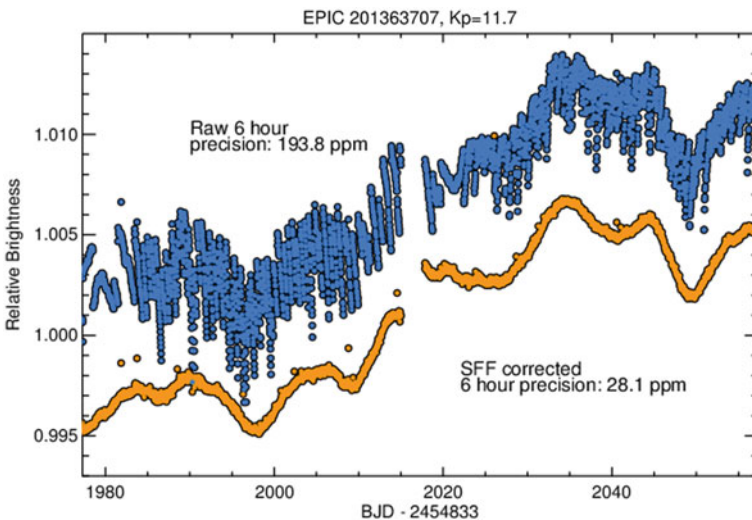


Fig. 5 An example of the uncorrected fluxes from K2 (blue) and the corrected K2SFF version (orange). *Source* [44]

- the recurrent neural network of type LSTM in combination with some method of significant attribute extraction (a.k.a. feature extraction),
- a convolutional neural network with sigmoid activation function,
- other, not so successful (in terms of our results) artificial neural networks such as fully-convolutional neural networks, MCDCNN and ResNet.

After these, we attempted (using genetic algorithm) to engineer ANN specifically for our binary classification task and established a framework that can match the classification accuracy of related work.

4.1 Data Pre-processing

We pre-processed both BRITE and Kepler K2 data sets and transformed the raw data (light curves with flux) into a common form digestible by the ANN:

- Balancing data sets, so it contains the same number of variables and not variables.
- Cutting light curves in order to equal their length.
- Mix the data.
- Generate periodogram.
- Perform Fourier transformation.
- Significant attributes extraction in order to reduce the dimensionality of time series data (different techniques).
- Data normalization into interval relevant to the selected activation function.
- Splitting the data set to training and test set.

4.1.1 Significant Attributes Extraction and Visualization

During our experiments described later we discovered that both original data sets may not provide clear examples of time series of variable and non-variable stars. This led to poor accuracy, and we were, therefore, looking for a way how to distinguish these time series by means of significant attributes extraction. We tested the usability of extracted attributes by visualization using Sammon mapping [36]. Sammon mapping attempts to find a low-dimensionality representation of objects in high-dimensional space with as much respect to their original geometric distances as possible. We used it to convert extracted significant attributes to 2D and visualize, hoping to see clear clusters with variable and non-variable time series. Such a set of attributes could be then used for further classification using ANN.

5 Experiments and Results

As we achieved poor initial results with BRITE data set (as described in Sect. 5.1, not all experiments covers it. We were looking at results with ten following activation functions: exponential, sigmoid, hyperbolic tangent, relu, elu, selu, soft plus, softsign, harp sigmoid, linear. Experiments were performed with an artificial neural network containing three hidden layers: 16 neurons in input, 34 neurons in first hidden, 16 in second hidden and 64 in third hidden layer [1].

5.1 BRITE Data Set

We had started with a more problematic BRITE data set. We attempted to classify time series in several ways, but eventually with poor results.

5.1.1 Multi-layer Perceptron

The sigmoid activation function was used in the output layer. The training was stopped after 3000 epochs. The learning rate was set to 0.005. For each activation function, we trained the MLP 10-times, and based on validation data, we selected the best model. Its accuracy was then tested on testing data. Results can be seen in Table 1.

Bad results are probably caused by a small data set. Only 538 light curves came out from pre-processing, these were then divided into the training and test set in 70:30 ratio, 10% of training set was used for validation. For the training, only 338 light curves remained. Another issue was the irregular interval between individual

Table 1 Results of MLP classification for BRITE data set

Act. function	Precision	Recall	Accuracy
Exponential	0	N/A	0.64
Sigmoid	0	N/A	0.64
Hyperb. tan.	0.03	0.18	0.59
Relu	0	0.60	0.64
Elu	0	N/A	0.64
Selu	0	N/A	0.64
Soft plus	0	N/A	0.64
Soft sign	0	0	0.62
Harp sigmoid	0	N/A	0.64
Linear	0	0	0.61

measurements within the time series. Based on these results and results with LSTM, we decided to continue only with Kepler K2 data set.

5.1.2 Long Short-Term Memory

In this case, the process was a bit different. We used 900 raw time series (as LSTM is supposed to handle it) cross-matched with GCVS catalogue. We divided them into training and test set in 70:30 ratio. Each time series had up to 66500 measurements (“feature vector”). Shorter time series were padded with -1 to this length and all data normalized. With these settings, we achieved the accuracy of 60%.

5.2 Kepler K2 Data Set

After attempts with BRITE data set, we switched to a more promising Kepler K2 data set. Configuration was the same as in the case of BRITE.

5.2.1 Multi-layer Perceptron

The results with the same configuration as in case of BRITE can be seen in Table 2. Unfortunately, there is just minimal improvement. The best activation function turned out to be Selu that achieved accuracy 0.66. The recall is also an interesting metric because it is not such an issue if some non-variable object is classified as variable, but it is important to minimize the number of undetected variables. From this point of view, the Elu function performed best. Accuracy and loss function of best models is depicted on Figs. 6 and 7.

Table 2 Results of MLP classification for Kepler K2 data set

Act. function	Precision	Recall	Accuracy
Sigmoid	0.98	0.53	0.56
Hyperbolic tangent	0.72	0.61	0.64
Relu	0.67	0.60	0.61
Elu	0.64	0.63	0.63
Selu	0.78	0.62	0.66
Soft plus	0.82	0.58	0.62
Soft sign	0.64	0.58	0.60
Harp sigmoid	0.98	0.49	0.48
Linear	0.73	0.61	0.63

Then we attempted to improve the accuracy by generating so-called periodograms created by conventional statistical analysis. The classifier then attempted to classify these periodograms instead of light curves. Results can be seen in Table 3 and shows no significant improvement in accuracy. Nevertheless, hyperbolic tangent performed best. We also attempted to establish some custom activation functions listed in Table 4 (raw light curves were used). As the last experiment with MLP, we attempted to use our own activation functions with Kepler K2 light curves processed by Fourier transformation. Results are listed in Table 5. We achieved similar results with Cosine transformation.

5.2.2 Convolutional Network

We decided to compare CNN with MLP. All data from K2 data set were normalized by min-max normalization, we run 5000 epochs with learning rate 0.005 and following network configuration: two convolutional layers with the sigmoid act. function,

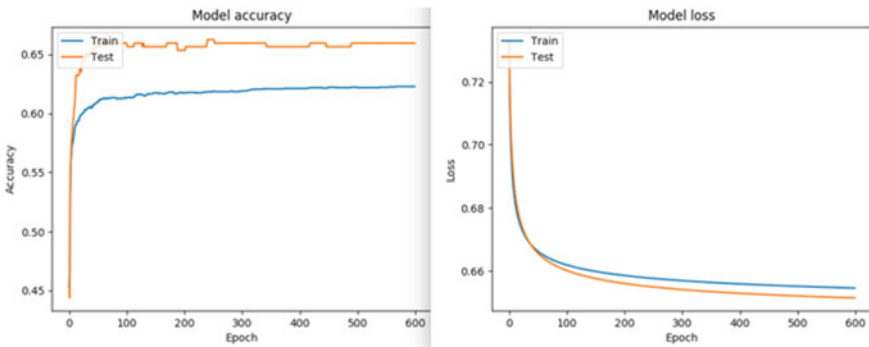


Fig. 6 Results of MLP training with Elu activation function on the K2 data set. Source [1]

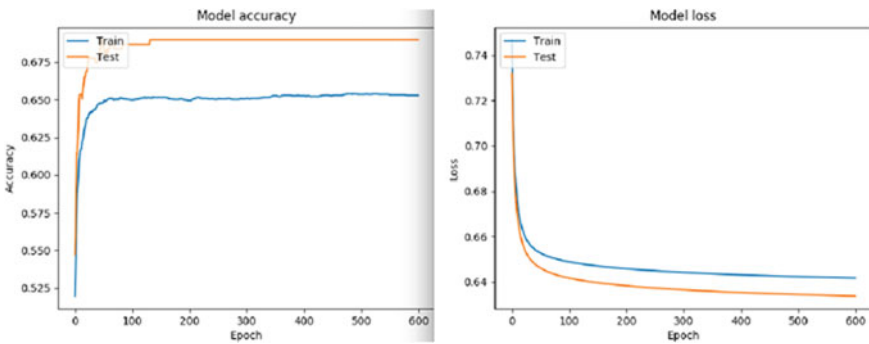


Fig. 7 Results of MLP training with Selu activation function on the K2 data set. Source [1]

Table 3 MLP classification for K2 data converted to periodograms

Act. function	Precision	Recall	Accuracy
Sigmoid	1.00	0.49	0.49
Hyperbolic tangent	0.51	0.71	0.66
Relu	0.54	0.70	0.65
Elu	0.63	0.65	0.65
Selu	0.65	0.61	0.62
Soft plus	0.80	0.61	0.65
Soft sign	0.56	0.67	0.64
Harp sigmoid	0.00	0.00	0.50
Linear	0.61	0.65	0.64

Table 4 MLP classification for K2 data using own act. functions

Act. function	Precision	Recall	Accuracy
$\tanh(0.1*x)$	0	0	0.45
$\tanh(0.3*x)$	0.05	0.53	0.45
$\tanh(0.5*x)$	0.52	0.66	0.59
$\tanh(x)$	0.72	0.68	0.66
$\tanh(1.5*x)$	0.68	0.68	0.65
$\tanh(2*x)$	0.71	0.67	0.65

Table 5 MLP classification for K2, own act. functions, FT

Act. function	Precision	Recall	Accuracy
$\tanh(1.1*x)$	0.68	0.64	0.64
$\tanh(1.4*x)$	0.75	0.65	0.67
$\tanh(1.5*x)$	0.74	0.65	0.66
$\tanh(1.7*x)$	0.68	0.61	0.61
$\tanh(2*x)$	0.72	0.63	0.63

window size of 7 and 6 (or 12) filters, each followed by pooling layer and with output layer with a sigmoid activation function. Results are in Table 6 and on Fig. 8.

5.2.3 Other Artificial Neural Networks

We have been experimenting with several other ANNs including ResNet, Fully-convolutional network, MCDCNN and other configurations of MLP and CNN with even less success than as described above. Their results are, therefore omitted from this paper.

Table 6 Results of CNN classification for Kepler K2 data with different count of light curves and measurements in each time series (cutted to this length)

Light curves	Length	Precision	Acc.	Recall
500	800	0.65	0.65	0.65
7502	1300	0.65	0.64	0.64
500	400	0.64	0.62	0.63

5.2.4 Other Significant Attributes Extraction Methods

As mentioned before, we attempted to extract the most significant attributes from K2 data set in order to distinguish variable and non-variable objects. To verify this, Sammon projection was used (see Figs. 9 and 10). The experiment confirmed that the original data does not contain clusters, and we need to focus on domain-specific details of the data.

5.2.5 Improvement of Accuracy Using Genetic Algorithm

We eventually focused on the most-promising feature extraction approach and certain domain knowledge. We used Kepler K2 variability metadata that also provides a confidence level of each class label (in the form of probabilistic distribution over all possible classes). The histogram of such label confidence is depicted on Fig. 11. For the training purposes, we fine-selected only those time series that has confidence level over 80% and thus are more “clean” for the feature extraction (in other cases there is a significant probability of a bad label)—an experimentally established set of statistical markers (calculated using *tsfresh*³ framework):

- The *absolute sum of changes* of the time series x : $\sum_{i=1}^{n-1} |x_{i+1} - x_i|$.
- *Aggregated auto-correlation*: $f_{agg} = (R(1), \dots, R(m))$ for $m = \max(n, maxlag)$ where n is the length of the time series X , $maxlag$ is the maximal number of lags

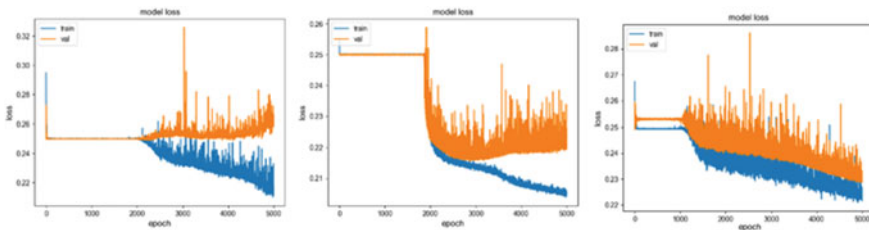


Fig. 8 CNN experiment loss function chart during training phase (from left: experiment #1, #2, #3) [1]

³tsfresh: <https://tsfresh.readthedocs.io>.

to consider, f_{agg} is mean, variance and standard deviation in our case and $R(l)$ is the autocorrelation for lag l : $R(l) = \frac{1}{(n-l)\sigma^2} \sum_{t=1}^{n-l} (X_t - \mu)(X_{t+l} - \mu)$ with σ^2 being variance and μ mean.

- *Change quantiles* with lower quantile being 0.5, higher quantile being 0.7, using absolute differences and variance as the aggregation function applied to a corridor established by quantiles.
- *CID*—an efficient complexity-invariant distance for time series x attempting to estimate its complexity specified by more peaks, valleys etc. [8]: $\sqrt{\sum_{i=0}^{n-2lag} (x_i - x_{i+1})^2}$.
- *A count above/below mean* returning the number of values in time series x that are above/below its mean.
- *Energy ratio by chunks*—sum of squares of chunk i out of N chunks expressed as a ratio with the sum of squares over the whole series (we used ten chunks).
- *Fast Fourier coefficients* $A_k = \sum_{m=0}^{n-1} a_m e^{-2\pi i \frac{mk}{n}}$ for $k = 0, \dots, n - 1$ where A_k is the k th Fourier (complex) coefficient, n is the length of the time series and a_m is the m th value of time series. We used the first 3 Fourier coefficients: their real, imaginary, absolute and angle value.
- *Aggregated Fast Fourier Transformation*—spectral centroid (mean), variance, skew, and kurtosis of the absolute Fourier transform spectrum.

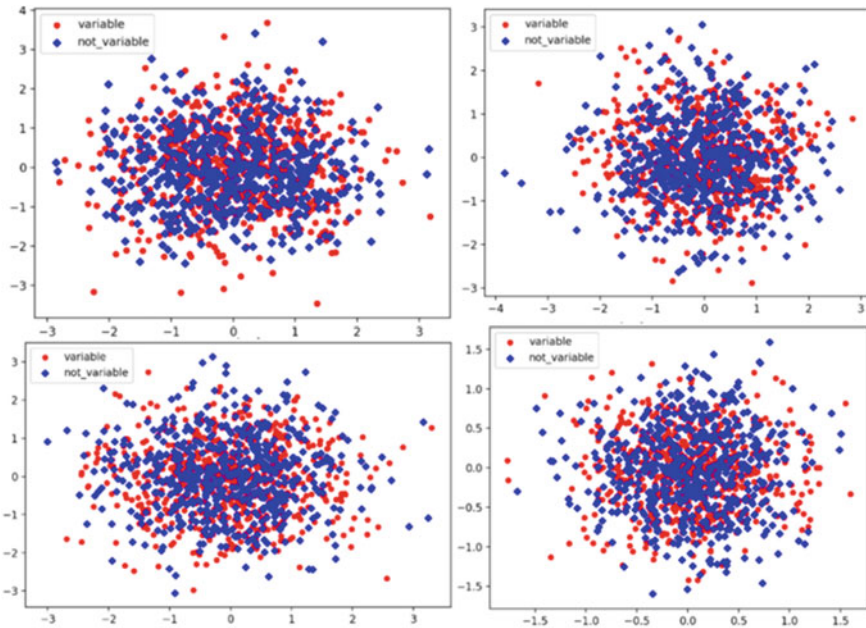


Fig. 9 Sammon projection, random init.: 500 epochs, different time series length (1200–1225 measurements), extracted attributes or just FT or min-max norm [1]

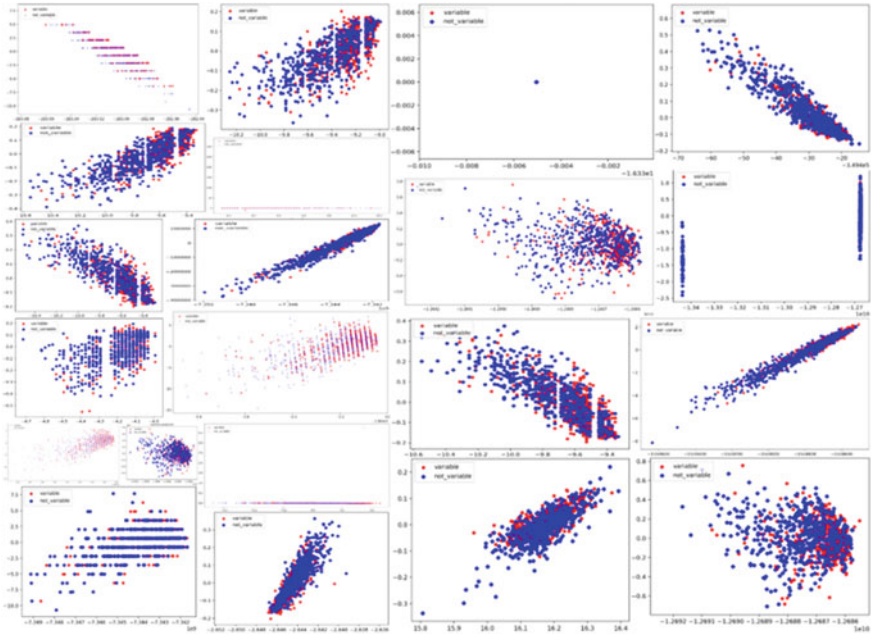


Fig. 10 Sammon mapping: 500 epochs, time series length: 1200–1225. Attr.: stand. dev., variance, min, max, mean, sum val., median, abs. sum of changes, agg. autocorr., arithmetic coeff., binned entropy, energy ratio, agg. FFT val., first loc. of min/max, mult. max values, index mass quant., linear trend etc. Or processed by FT or min-max normalization [1]

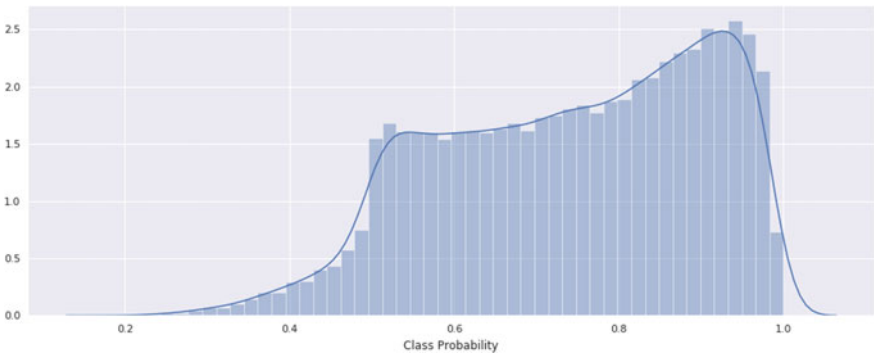


Fig. 11 Histogram of class probabilities (label confidence) for Kepler K2 data

- *C3* measuring non-linearity in time series x [39] by computing $\frac{1}{n-2lag} \sum_{i=0}^{n-2lag} x_{i+2lag}^2 \cdot x_{i+lag} \cdot x_i$ where we used $lag = 350$.
- *Mean value of a central approximation of the second derivative*: $\frac{1}{n} \sum_{i=1}^{n-1} \frac{1}{2}(x_{i+2} - 2x_{i+1} + x_i)$ where n is the length of time series x .
- *Partial autocorrelation* at lag $k = 2$ of time series x [9].
- *Quantile* $q = 0.5$.
- *Range count* of observed values within the interval $[0, 100)$.
- *A ratio beyond $r\sigma$* —ratio of values that are more than $r \cdot \sigma(x)$ away from the mean of time series x where σ is standard deviation a $r = 100$ in our case.
- *The ratio of a count of unique values to count of all values* of the give time series.
- *Skewness*.
- *The power spectrum of the different frequencies* of the given time series.
- *Standard deviation*.
- *Variance*.

On Fig. 12 we depict our pre-processed data set. Visualization in 3D was done using a non-linear projection called t-SNE [31]. We can see that variable and non-variable stars are now much more distinguishable. In the bottom part where most of the non-variable stars are located, we can see a significant number variable stars as well which suggests that there will be a need of a rather higher number of layers and neurons in them in order to “bend” the space around them and separate them in high-dimensional space.

We also introduced batch normalization [24] for each hidden layer. The best experimentally found performing network was MLP containing three dense layers (64, 128 and 256 neurons) and reaching accuracy over 70%.

We then decided to optimize the ANN hyperparameters using a simple genetic algorithm. We used an initial population of 10 individuals initialized with random values or values established by our previous experience and results. Our genetic algorithm identifies four best individuals of each iteration, and using them, it generates six new individuals. Together they represent a new population. In each iteration, genomes of 2 randomly chosen individuals are mutated.

Multi-layer Perceptron

We started with MLP with up to 1000 neurons represented as a gene of the genome. Other genes covered activation function, metrics and fitness function. As it is depicted in Table 7, after 12 iterations the number of neurons stopped at 1000. We can speculate that if we increased the upper limit for the number of neurons, the genetic algorithm would continue to this limit as well. After 12 iterations, we also got final values for activation function and metrics. The algorithm suggests sigmoid activation function an accuracy as a metric. Based on our results, we can also say that categorical cross-entropy and Kullback–Leibler divergence are equal in the sense of the accuracy.

Convolutional Neural Network

We extended the genome by several more attributes (genes) mostly for data pre-processing and convolutional layers:

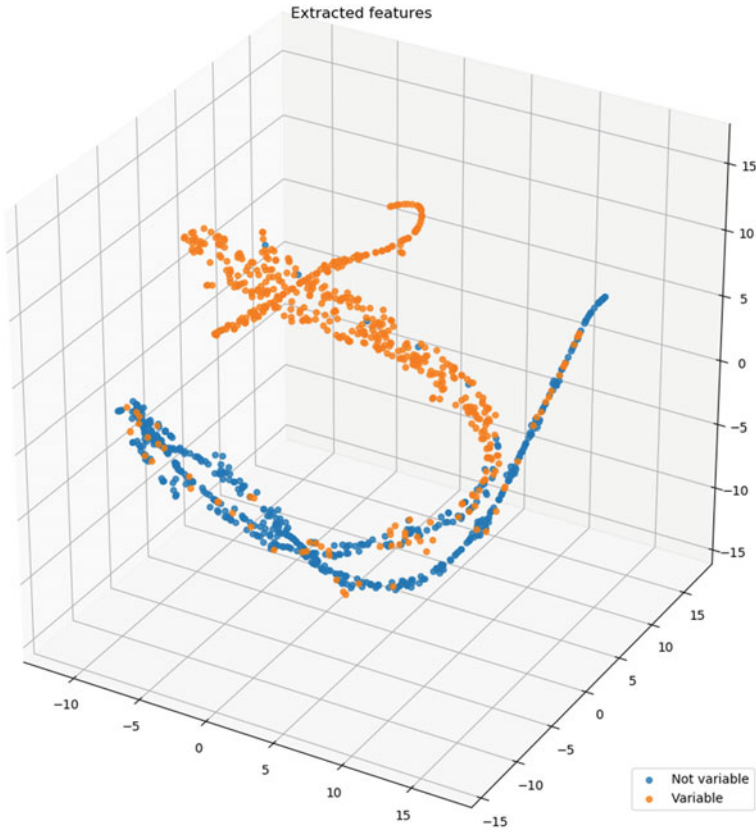


Fig. 12 Best describing extracted features for most reliable (in a sense of class confidence) time series

- time series length,
- data normalization method,
- number of epochs,
- number of neurons,
- sliding window size,
- activation function,
- cost function.

In Table 8 we can see the best individuals from each iteration. We can see that even in the first iteration, a great individual has emerged, meaning that the genetic algorithm had limited space to push the classification F1 score further. Yet we can see that even better individual was found: 0.98%. These results was achieved using the following parameters for data pre-processing and network topology:

Table 7 Results of genetic algorithm optimizing the MLP network

Number of neurons	Activation function	Cost function	F1 score
200	Relu	Categorical cross-entropy	0.8542
200	Relu	Categorical cross-entropy	0.8709
200	Relu	Categorical cross-entropy	0.8647
200	Relu	Kullback–Leibler divergence	0.8732
300	Relu	Categorical cross-entropy	0.8818
250	Relu	Kullback–Leibler divergence	0.8741
250	Relu	Kullback–Leibler divergence	0.8747
300	Relu	Kullback–Leibler divergence	0.8765
300	Sigmoid	Categorical cross-entropy	0.8861
300	Sigmoid	Categorical cross-entropy	0.8871
300	Sigmoid	Categorical cross-entropy	0.8921
1000	Sigmoid	Categorical cross-entropy	0.9176
1000	Sigmoid	Categorical cross-entropy	0.9204
1000	Sigmoid	Categorical cross-entropy	0.9165
1000	Sigmoid	Kullback–Leibler divergence	0.9216
1000	Sigmoid	Kullback–Leibler divergence	0.9177
1000	Sigmoid	Kullback–Leibler divergence	0.9181
1000	Sigmoid	Kullback–Leibler divergence	0.9159
1000	Sigmoid	Kullback–Leibler divergence	0.9199
1000	Sigmoid	Categorical cross-entropy	0.9201

Table 8 Results of genetic algorithm optimizing the CNN network

Time series length	Data normalization method	Number of epochs	Number of neurons	Sliding window size	Activation function	Cost function	F1 score
750	Mean	456	30	3	Relu	Mean squared error	0.9522
750	Mean	456	30	3	Relu	Mean squared error	0.9522
850	Mean	456	15	3	Relu	Mean squared error	0.9513
850	Mean	456	15	3	Relu	Poisson	0.9539
850	Mean	456	15	3	Relu	Poisson	0.9539
850	Mean	456	70	13	Relu	Mean squared error	0.9625
850	Mean	456	70	13	Relu	Mean squared error	0.9673
850	Mean	456	70	11	Relu	Poisson	0.9790
850	Mean	456	70	11	Relu	Poisson	0.9790
850	Mean	456	70	11	Relu	Poisson	0.9790

- time series length: 850,
- data normalization method: mean,
- number of epochs: 456,
- number of neurons: 70,
- sliding window size: 11,
- activation function: relu,
- cost function: Poisson.

Fully-convolutional Neural Network

In Table 9 we can best individuals of 10 iterations performed with the same genome as in the previous case, just using FCN. In this case, the best individual found in the first iteration reached even higher F1 score, yet there was found a slightly better one in later iterations reaching 97%. The best individual had the following parameters:

- time series length: 800,
- data normalization method: mean,
- number of epochs: 100,
- number of neurons: 50,

Table 9 Results of genetic algorithm optimizing the FCN network

Time series length	Data normalization method	Number of epochs	Number of neurons	Sliding window size	Activation function	Cost function	F1 score
400	Mean	100	10	20	relu	Mean squared error	0.9613
300	min_max	456	15	50	relu	Mean squared error	0.9432
800	Mean	100	50	50	relu	Mean squared error	0.9716
800	Mean	100	50	50	relu	Mean squared error	0.9716
800	Mean	100	15	50	relu	Mean squared error	0.9707
800	Mean	100	15	50	relu	Mean squared error	0.9622
700	Mean	100	50	50	relu	Mean absolute percentage error	0.9645
700	Mean	1000	50	50	relu	Mean absolute percentage error	0.9656
700	min_max	350	30	50	relu	Mean absolute percentage error	0.9539
700	Mean	350	30	50	relu	Mean squared error	0.9666

- sliding window size: 50,
- activation function: relu,
- cost function: mean squared error.

5.2.6 Results Summary

Similarly to [19] we achieved poor results with LSTM and best results using feature extraction. Although in [7, 19] authors worked with original Kepler mission data and multi-class classification (instead of our binary one), we were able to reach higher accuracy on a very similar data Kepler K2 data set and binary classification. It seems that the key lies in data pre-processing and selection of correct training data as the confidence level for the label data is often deeply under 50%. Comparison with Kaggle competition results is also problematic not just for the fact that it also uses the original Kepler mission data, but also for using a highly imbalanced data set.

6 Conclusions

Astronomical time series (a.k.a. light curves) are very specific in comparison to other ones—they are noisy, variable in length, and observational intervals and with overlapping classes. Traditional methods for their classification seems to not perform very well with them. Significant attributes extraction that includes various statistical markers and FFT coefficients seems to be necessary to achieve good results and the classification accuracy can be highly improved by optimizing the hyper-parameters and topology with genetic algorithm. It also seems that a successful classification model must contain a large amount of neurons, high-length samples and a high number of epochs during training. It also seems that mean normalization of data performs better.

In the end it can reach high accuracy confirmed by validation and test set to prevent over-fitting. So far, we are not aware of a related work reaching similar accuracy on Kepler data set. Similar work usually performs multi-class classification with the help of photometric meta-data as additional features while we rely just on the time series themselves (although we select only those with high confidence in the label). No other work using Kepler data that we are aware of also mentions what kind of or how much data they used (that is, whether the data set was balanced, what was the number of samples used or what was the minimum or mean of the label confidence).

We expect that by introducing photometric meta-data we can lower the confidence in label while keeping similar accuracy in the future. We would also like to confirm this approach on different astronomical data sets and possibly on multi-class classification (as samples of individual classes may differ too much from each other to be united as one ‘variable’ class).

Our future work will include testing various ANN topology and hyperparameters optimization using e.g. evolutionary algorithms. We are also aware of papers that attempt to speed up the convergence of hyperparameters optimization using e.g. Decision trees or Bayesian optimization. A nice overview of optimizing (deep) neural networks also offers e.g. [43]. Another possibility for binary classification could be for example a solution inspired by anomaly detection where the anomaly could be anything but noise (where the noise simply masks a non-variable star, and everything else would be considered variable).

References

1. Andrešič, D., Šaloun, P., Suchánová, B.: Large astronomical time series pre-processing and visualization for classification using artificial neural networks. In: 2019 IEEE 15th International Scientific Conference on Informatics, pp. 000311–000316 (2019)
2. Armstrong, D.J., et al.: K2 variable catalogue I: a catalogue of variable stars from K2 field 0 (2014). [arXiv:1411.6830](https://arxiv.org/abs/1411.6830) [astro-ph.SR]
3. Armstrong, D.J., et al.: K2 variable catalogue: variable stars and eclipsing binaries in K2 campaigns 1 and 0. *Astron. & Astrophys.* **579**, A19 (2015). ISSN: 1432-0746. <https://doi.org/10.1051/0004-6361/201525889>
4. Bagnall, A., et al.: Time-series classification with COTE: the collective of transformation-based ensembles. *IEEE Trans. Knowl. Data Eng.* **27**(9), 2522–2535 (2015). ISSN: 2326-3865. <https://doi.org/10.1109/TKDE.2015.2416723>
5. Bagnall, A., et al.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Disc.* **31**(3), 606–660 (2016). <https://doi.org/10.1007/s10618-016-0483-9>
6. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate (2014). [arXiv:1409.0473](https://arxiv.org/abs/1409.0473) [cs.CL]
7. Bass, G., Borne, K.: Supervised ensemble classification of Kepler variable stars. *Mon. Notices R. Astron. Soc.* **459**, stw810 (2016). <https://doi.org/10.1093/mnras/stw810>
8. Batista, G.E.A.P.A., et al.: CID: an efficient complexity-invariant distance for time series. *Data Min. Knowl. Disc.* **28**(3), 634–669 (2013). <https://doi.org/10.1007/s10618-013-0312-3>
9. Box, G.E.P., Jenkins, G.M., Reinsel, G.C.: *Time Series Analysis*. Wiley, New York (2008). <https://doi.org/10.1002/9781118619193>
10. van Cleve, J.E., et al.: Kepler: a search for terrestrial planets - Kepler data characterization Handbook (2016)
11. Cui, Z., Chen, W., Chen, Y.: Multi-scale convolutional neural networks for time series classification (2016). [arXiv:1603.06995](https://arxiv.org/abs/1603.06995) [cs.CV]
12. Dau, H.A., et al.: The UCR time series archive (2018). [arXiv:1810.07758](https://arxiv.org/abs/1810.07758) [cs.LG]
13. Elman, J.L.: Finding structure in time. *Cognit. Sci.* **14**(2), 179–211 (1990). https://doi.org/10.1207/s15516709cog1402_1
14. Fawaz, H.I., et al.: Deep learning for time series classification: a review. *Data Min. Knowl. Discov.* **33**(4), 917–963 (2019). <https://doi.org/10.1007/s10618-019-00619-1>
15. Fukushima, K.: Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biolog. Cybern.* **36**(4), 193–202 (1980). <https://doi.org/10.1007/bf00344251>
16. Gamboa, J.C.B.: Deep learning for time-series analysis (2017). [arXiv:1701.01887](https://arxiv.org/abs/1701.01887) [cs.LG]
17. Lee Giles, C., Lawrence, S., Chung Tsoi, A.: *Mach. Learn.* **44**(1/2), 161–183 (2001). <https://doi.org/10.1023/a:1010884214864>
18. He, K., et al.: Deep residual learning for image recognition (2015). [arXiv:1512.03385](https://arxiv.org/abs/1512.03385) [cs.CV]
19. Hinners, T.A., Tat, K., Thorp, R.: Machine learning techniques for stellar light curve classification. *Astron. J.* **156**(1), 7 (2018). ISSN: 1538-3881. <https://doi.org/10.3847/1538-3881/aac16d>
20. Hochreiter, S.: Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München (1991)
21. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–780 (1997). <https://doi.org/10.1162/neco.1997.9.8.1735>
22. Hosenie, Z., et al.: Comparing multiclass, binary, and hierarchical machine learning classification schemes for variable stars. *Mon. Notices R. Astron. Soc.* **488**(4), 4858–4872 (2019). ISSN: 1365-2966. <https://doi.org/10.1093/mnras/stz1999>
23. Hu, B., Chen, Y., Keogh, E.J.: Time series classification under more realistic assumptions. In: *SDM* (2013)
24. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift (2015). [arXiv:1502.03167](https://arxiv.org/abs/1502.03167) [cs.LG]

25. Jenkins, J.M.: Kepler data processing handbook: overview of the science operations center. Kepler Science Document (2017)
26. Karim, F., et al.: LSTM fully convolutional networks for time series classification. *IEEE Access* **6**, 1662–1669 (2018). ISSN: 2169-3536. <https://doi.org/10.1109/access.2017.2779939>
27. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995). <https://doi.org/10.1109/ICNN.1995.488968>
28. Lecun, Y., Bengio, Y.: Convolutional networks for images, speech, and time-series (1995)
29. LeCun, Y., et al.: Handwritten digit recognition with a back-propagation network. In: D.S. Touretzky (ed.) *Advances in Neural Information Processing Systems*, vol. 2, pp. 396–404. Morgan-Kaufmann, Burlington. <http://papers.nips.cc/paper/293-handwritten-digit-recognition-with-a-back-propagation-network.pdf> (1990)
30. LeCun, Y., et al.: Object recognition with gradient-based learning. In: *Shape, Contour and Grouping in Computer Vision*, pp. 319–345. Springer, Berlin (1999). https://doi.org/10.1007/3-540-46805-6_19
31. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605. <http://www.jmlr.org/papers/v9/vandermaaten08a.html> (2008)
32. Petitjean, F., et al.: Dynamic time warping averaging of time series allows faster and more accurate classification. In: 2014 IEEE International Conference on Data Mining, vol. 27, pp. 470–479 (2014). <https://doi.org/10.1109/ICDM>
33. Rumelhart, D.E.: Chapter parallel distributed processing, exploration in the microstructure of cognition (1986)
34. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge (2014). [arXiv:1409.0575](https://arxiv.org/abs/1409.0575) [cs.CV]
35. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoustics Speech Signal Process.* **26**(1), 43–49 (1978). ISSN: 0096-3518. <https://doi.org/10.1109/TASSP.1978.1163055>
36. Sammon, J.W.: A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.* **C-18**(5), 401–409 (1969). <https://doi.org/10.1109/t-c.1969.222678>
37. Samus, N.N., et al.: General catalogue of variable stars: version GCVS 5.1. *Astron. Repor.* **61**(1), 80–88 (2017). <https://doi.org/10.1134/s1063772917010085>
38. Schäfer, P.: The BOSS is concerned with time series classification in the presence of noise. *Data Mining Knowl. Discov.* **29**(6), 1505–1530 (2014). <https://doi.org/10.1007/s10618-014-0377-7>
39. Schreiber, T., Schmitz, A.: Discrimination power of measures for nonlinearity in a time series. *Phys. Rev. E* **55**(5), 5443–5447 (1997). <https://doi.org/10.1103/physreve.55.5443>
40. Skoda, P.: Optical spectroscopy with the technology of virtual observatory. *Baltic Astron.* **20** (2011). <https://doi.org/10.1515/astro-2017-0332>
41. Smirnov, D., Nguifo, E.M.: Time series classification with recurrent neural networks (2018)
42. Sutskever, I., Vinyals, O., Le, Q.: Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* **4** (2014)
43. Talbi, E.-G.: Optimization of deep neural networks: a survey and unified taxonomy. In: Working Paper or Preprint, <https://hal.inria.fr/hal-02570804> (2020)
44. Vanderburg, A.: K2 Extracted Lightcurves (“K2SFF”). <http://archive.stsci.edu/doi/resolve/resolve.html?doi=10.17909/T9BC75> (2015). <https://doi.org/10.17909/T9BC75>
45. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: a strong baseline (2016). [arXiv:1611.06455](https://arxiv.org/abs/1611.06455) [cs.LG]
46. Williams, R.J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.* **1**(2), 270–280 (1989). ISSN: 0899-7667. <https://doi.org/10.1162/neco.1989.1.2.270>

Frontiers

In this book, which you have just read, the issue of data processing in the field of astrophysics using Machine learning methods or, if you want more generally, using artificial intelligence methods, was discussed. This area, if we look at the history of its development, is relatively young and has less than a hundred years of development, if we take into account the first neuron model of the 1920s as the first technical algorithm in the field of artificial intelligence. Over the last 80 years, the development of computational methods has undergone relatively fundamental changes, both at the hardware level and at the software level. Your mobile phone can now handle mainframe computers that used to match the power of a small power plant. In the field of computer science, significant changes can also be observed, especially in the area of the already mentioned artificial intelligence. In the last 40 years, artificial intelligence algorithms have evolved a lot, and the class of these algorithms is wealthy today. We can encounter neural networks, fuzzy logic, evolutionary algorithms, flock intelligence, and many other algorithms, including exotic ones as cell automata or fractal geometry. These are no longer officially part of artificial intelligence, but can also be used to solve very complex tasks. Based on all this, it is challenging to predict further developments, because recent developments have proven to be very surprising. Developments in data processing technologies can, therefore, only be speculated. Today, artificial intelligence is used for many tasks such as speech analysis, language structure analysis, for modelling extremely complex processes and processes, for analyzing and simplifying data from very complex processes and many other exciting tasks. It is logical that astrophysics that generates a large amount of data using modern technologies has become an object of interest in the application of these methods and an object of interest of the community that deals with these methods. Thus, an apparent increase in the application of artificial intelligence methods to astrophysical data can be expected in the near future, not only for some simple noise cleaning or interpretation but also for creating very complex models, estimating missing data based on the existing data set context, analysis of signals coming from space, whether from natural or possibly other sources, and many others.

It can also be expected that artificial intelligence will help us to find an explanation of new phenomena, or their prediction, as was the case with the use of evolutionary techniques in the discovery of a new quantum phenomenon, which would be appreciated at the GECCO conference. This recorded case clearly shows the potential of artificial intelligence and its ability to estimate incomplete data the correct result or predict the correct behaviour of the model. Or the existence of unknown phenomena can be successfully estimated in cooperation with other methods, such as the already mentioned fractal geometry, which is reflected both in the time series of various complex systems and in the structures of natural systems. Thus, it can be expected that artificial intelligence, together with other exotic methods, will bring us many exciting results in the astrophysics.