CHAPTER 4

Evolving from Data to Knowledge Mining to Uncover Hidden Relationships

Konstantinos Demestichas, Konstantina Remoundou, Ioannis Loumiotis, Evgenia Adamopoulou, Wilmuth Mueller, Dirk Pallmer, Dirk Mühlenberg, Rafal Kozik, Michael Choras, David Faure, Roxana Horincar, Edward Benedict Brodie of Brodie, Charlotte Jacobe de Naurois, Krishna Chandramouli, and Alexandra Rosca

K. Demestichas (✉) · K. Remoundou · I. Loumiotis · E. Adamopoulou
Institute of Communication and Computer Systems, Athens, Greece
e-mail: cdemest@cn.ntua.gr; kremoundou@cn.ntua.gr; i_loumiotis@cn.ntua.gr; eadam@cn.ntua.gr

W. Mueller · D. Pallmer · D. Mühlenberg
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation, Karlsruhe, Germany
e-mail: wilmuth.mueller@iosb.fraunhofer.de; dirk.pallmer@iosb.fraunhofer.de; dirk.muehlenberg@iosb.fraunhofer.de

B. Akhgar et al. (eds.), *Technology Development for Security Practitioners*, Security Informatics and Law Enforcement,
https://doi.org/10.1007/978-3-030-69460-9_4

4.1 Introduction

Over the last decades, human activities have progressively moved from person-to-person to seamless interactions between the physical and the information technology (IT) worlds; crime has naturally followed the same path, with imagination as the sole limit. This, in addition to the diversity of the events that constitute a crime to be prevented, has forced the law enforcement agencies (LEAs) to research for evidence data from different data sources, such as video, audio, text/documents, social media and web data, telecom data, surveillance systems data, police databases, etc. making it a very difficult and time-consuming task to analyse them and conclude to end evidence results in order to fight terrorism and crime cases in time. So, the main challenge presented in this chapter is to assist the LEAs in their fight against crime, by resolving the problem of heterogeneity of the massive volumes of primary data collected, by fusing and analysing them in order to uncover hidden relationships among data items, compute trends for the evolution of security incidents, ultimately (and at a faster pace) reaching solid evidence that can be used in court, gaining also better awareness and understanding of current or past security-related situations.

To this end, novel approaches that will address the significant needs of LEAs in their fight against terrorism and organized crime, related to the prevention, investigation and prosecution of criminal offences, are

R. Kozik · M. Choras
ITTI SP. o.o., Poznań, Poland
e-mail: rafal.kozik@itti.com.pl; mchoras@itti.com.pl

D. Faure · R. Horincar · Edward Benedict Brodie of Brodie · C. J. de Naurois
Thales Research and Technology, Palaiseau, France
e-mail: david.faure@thalesgroup.com; roxana.horincar@thalesgroup.com;
edward-benedict.brodieofbrodie@thalesgroup.com;
charlotte.jacobedenaurois@thalesgroup.com

K. Chandramouli
Venaka Media, Kent, UK
e-mail: k.chandramouli@venaka.co.uk

A. Rosca
Siveco, Bucharest, Romania
e-mail: Alexandra.Rosca@siveco.ro

required. In the current work, the authors propose a system based on sophisticated knowledge representation, advanced semantic reasoning and augmented intelligence, well integrated in a common, modular platform with open interfaces. In order to produce court-proof evidence for the LEA's criminal investigation actions, the collection and unification of different evidence data sources is presented, as well as a common representation model for internal data representation. This unified data model is placed in the ontology that enables joint exploration and exploitation of the multiple diverse data sources, allowing the anticipation and prediction of the future trends (e.g. threats) and establishing the ground for reasoning and cognition (operational and situational awareness). According to the authors' knowledge, there are no any recent attempts to integrate all the above technologies to a simple system for the LEAs.

The chapter is structured as follows. In Sect. 4.2, the current approaches of big data collection and processing in order to fight crime and terrorism are presented. Next, in Sect. 4.3, the proposed system is explained, while in Sect. 4.4 the proposed methodology, the tools and the initial results are described in more detail. Finally, Sect. 4.5 concludes the chapter.

4.2 State of the Art

During the last years, there have been many attempts towards enhancing LEAs work with modern technologies. These attempts have been organized and funded by the European Commission, including projects such as RED-ALERT [1] which aims to create data mining and predictive analytics tools for complex event processing targeting mainly the social media and LASIE [2] which scope is to assist forensic analysis with multiple sources.

Another attempt towards enhancing LEA's work is TENSOR [3] which aims to provide a terrorism intelligence platform that will allow LEAs fast and reliable planning and prevention functionalities for the early detection of terrorist organized activities, radicalization and recruitment. VICTORIA [4] is another attempt that focuses on creating a platform that accelerates the video analysis for investigating criminal and terrorism activities. Other attempts in the security domain include ROBORDER [5] which aims to create a fully functional autonomous border surveillance system with unmanned mobile robots including aerial, water surface, underwater and ground vehicles which will incorporate multimodal sensors as part of an interoperable network.

Although these attempts aim to increase the efficiency of LEAs in their daily work, they lack of semantic and fusion techniques to assist the processing of information from multiple data sources. Our current approach proposes a novel system that will provide reliable threat assessment and prediction by semantic fusion and trends analysis, that will allow the identification of correlations and hidden relationships among data.

4.3 Proposed System

The proposed system includes a set of tools that aim to facilitate LEAs in their daily work. These tools will allow the processing of data collected by different data sources and the identification of hidden patterns within the data that will allow LEAs officer to solve a case faster. Data processing will be based on an ontology model specified for this purpose. An ontology is a formal explicit specification of a shared conceptualization where, conceptualization is an abstract, simplified view of the world that describes the objects, concepts and other entities, existing in a domain along with their relationships. One of the most critical contributions of an ontology is its ability for providing the higher-level distinction of concepts which help understand systematically and consistently the lower-level details with domain concepts which is hard to obtain without ontological ways of thinking. The whole process of data processing includes five main steps:

- Data acquisition, which is the process of collecting primary data.
- Data preprocessing, which is the first step in making the input information understandable by the computer. It is different for each source of data. For example, for textual data, it is based on natural language processing techniques.
- Feature extraction is converting a set of input information into a set of numerical features.
- Feature representations. The proposed approach will rely on the use of bipartite graphs, more specifically a subset of the conceptual graphs to represent semantic information and knowledge.
- Semantic fusion and classification. Semantic information fusion typically covers two phases: (i) building the knowledge and (ii) pattern matching. The first phase incorporates the most appropriate knowledge into semantic information. Then, the second phase fuses relevant attributes and provides a semantic interpretation of the input data.

In the following sections, the reasoning, the semantic processing and the trend analysis will be presented in more detail.

4.4 Methodology and Tools

Knowledge representation [6] is the field of artificial intelligence that focuses on designing computer representations that capture information about the world that can be used to solve complex problems. It goes hand in hand with automated reasoning because one of the main purposes of explicitly representing knowledge is to be able to reason about that knowledge, to make inferences, assert new knowledge, etc. Virtually all knowledge representation languages have a reasoning or inference engine as part of the system. The advanced semantic reasoning services presented in this chapter enables a computable framework for systems to deal with knowledge in a formalized manner, allowing navigation through the different pieces of data and discovery of relations and correlations among them, thus broadening the spectrum of knowledge capabilities for the LEAs.

Reasoning is a procedure that allows the addition of rich semantics to data and helps the system to automatically gather and use deep-level new information. Specifically, by logical reasoning, the system is able to uncover derived facts that are not expressed in the knowledge base explicitly, as well as discover new knowledge of relations between different objects and items of data.

A reasoner is a piece of software that is capable of inferring logical consequences from stated facts in accordance with the ontology's axioms and of determining whether those axioms are complete and consistent. Reasoning with technologies like Resource Description Framework Schema (RDFS) and Web Ontology Language (OWL) allows adding rich semantics to data, and it helps the system to automatically gather and use deep-level new information, allowing also to derive facts that are not expressed in the knowledge base explicitly, as well as discover new knowledge of relations between different objects and items of data. In other words, reasoners are able to infer logical consequences from a set of asserted facts or axioms. In the last decade, due to growth of the Semantic Web field, some of the most popular reasoners were developed to fulfil this task on different domains and with the ability to cover use cases on different levels of complexity and expressivity. For the purposes of the proposed system, the ontology attributes have been categorized in the following types [7]:

- Reasoning characteristics: Basic features of ontology reasoners can be described by this category (e.g. methodology, sound, expressivity, incremental classification).
- Practical usability characteristics: This category describes the view angle of a developer using the OWL API support, availability and type of license.
- Performance indicators: This category addresses performance aspects such as classification performance and consistency checking performance, in particular for ontologies that are time critical or expect fast query, including reasoning and response time.

In order for a reasoner to infer new axioms from the ontology's asserted axioms, a set of rules should be provided to the reasoner. Rules are of the form of an implication between an antecedent (body) and consequent (head). The intended meaning can be read as: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold.

The antecedent is the precondition that has to be fulfilled that the rule will be applied; the consequent is the result of the rule that will be true in this case. An empty antecedent is treated as trivially true, so the consequent must also be satisfied by every interpretation; an empty consequent is treated as trivially false, so the antecedent must also not be satisfied by any interpretation [8].

For the reasoning process, the Pellet [9] reasoner has been employed, which can support the Semantic Web Rule Language (SWRL) rules. Furthermore, in order to allow the reasoner to infer the new axioms, the OWL Application Programming Interface (API) and the SWRL API have been also used.

4.4.1 Semantic Fusion Tools

The proposed system described in this chapter will benefit from the heterogeneous information collected by the different data sources, and it will combine the collected information in order to enable actions and decisions that would be more accurate than those that were produced by a single data source. However, the data from the different data sources might contain instances related to the same persons and/or events that cannot be easily identified. Data fusion [10] is the integration of multiple information and knowledge about the same object in order to obtain a

more accurate description. The goal of data fusion is to improve data quality of that object, which can be achieved if the information is stored separately obtaining synergy, which can be defined as the representation of a whole is better than the representation of the individual components.

The scope of the fusion tools is to utilize the redundancy of the information collected by the different data sources in order to eliminate duplicate instances regarding the person and event identities and increasing the credibility of the collected information. Four different fusion tools will be available in the system: person fusion, event fusion, trajectory fusion and graph-based semantic information fusion. The person fusion tool will be responsible for finding different person instances in the knowledge graph that refer to the same person and fuse these instances by checking the similarities between the different person instances in the knowledge graph and calculating the probability of these different person instances to refer to the same physical person. It is based on a variation of the k-nearest neighbours algorithm [11], and it becomes apparent that one of the key decisions in the implementation of the algorithm is the distance metric that will be used for calculating the distance between different observations. Based on the data type of the features that will be either numeric or symbolic, the appropriate distance metrics will be used. These includes the Euclidean distance for numeric features [12] and the Jaro distance [13] for the symbolic/alphanumeric features.

On the other hand, regarding the event fusion tool, the definition of an event can be very diverse inside an ontology. An event is represented by many types, such as crime, incident, killing, robbery, riot, burglary, etc., using also the moment and the place where the event took place. Similarity functions are defined to compare the concept instances involved in the definition of an event, such as the ones used in the person fusion tool. Furthermore, if it is decided that two concept instances are to be fused, the fusion is done following the defined fusion operators.

In addition, trajectory fusion tool refers to a time-ordered sequence of geographical positions; trajectory captures the movement of an agent that corresponds to a moving object whose position can change over time, represented in the ontology by concepts such as person or vehicle. A trajectory concept has at least two different location items representing the origin and the final destination of the movement and several optional intermediate ones. Several possible fusion strategies can be defined in order to fuse two trajectories. A necessary condition for two trajectories to be fused is for them to have the same starting and ending points, defined

by a place and time pair. Furthermore, more constraints can be added that refer to the intermediate positions, defined as place and time pairs. Finally, if it is decided that two trajectory instances are to be fused, the fusion is done following the defined fusion operators.

Lastly, graph-based semantic information fusion module provides a high-level semantic information fusion tool based on graph data and algorithms. The goal of this tool is to fuse any two pieces of semantic information that come from different data sources. In particular, it can also help deduplicate the ontology and increase its quality level. It relies on the use of bipartite graphs, more specifically a subset of the conceptual graphs [14, 15], to represent semantic information and knowledge. The result of the semantic fusion may be further used by other information processing tools implemented in the system. The application of conceptual graphs on trajectories is particularly motivated by the similarity of trajectory points, including both time and place. The implementation of the similarity functions may go from the simplest, that is based on an exact match, to more complex ones based on distance measurements.

In the example presented in Fig. 4.1, there are two trajectories, one done by a "Mr. Blue" and the other one by a "Mrs. Red". After fusing the common trajectory point A, the original two conceptual graphs depicting the two initial trajectories are connected into a single one. This highlights that both Mr. Blue and Mrs. Red have passed by the same place in the same time, and therefore their trajectories can be compared. In the case when it is decided that two different trajectories have some common trajectory points, the two trajectories may be fused into a single one or kept separate. Once the trajectory fusion step is done, several types of requests may be done on top of the fused information, serving various applications.

4.4.2 Trend Prediction

Following the fusion of the data, next step in the system is the trend identification and prediction through data clustering, classification and regression analysis.

Data classification can be used for use cases such as detection of a specific behaviour and sentiment analysis. This process is typically trying to identify a specific topic from a natural language source of a large volume of data. Text classification techniques help doing that as they divide the data classification process in three parts: model training, model verification and prediction. For model training and prediction of a behaviour,

Fig. 4.1 Example of two trajectories with a common trajectory point

clustering algorithms, decision trees and random forests algorithms are used, as well as regression analysis and neural models for identifying and predict hidden patterns and suspicious actions.

K-means clustering [16] is one of the most popular unsupervised algorithms for data clustering, which is used when we have unlabelled data without defined categories or groups. This is usually used when some expected behaviour is studied and when the number of groups under study is previously known. It is an iterative algorithm that assigns the data points to a specific – from the k known – cluster based on the distance from the arbitrary cluster centroid. During the first iteration, the centroids are randomly defined, and the data points are assigned to the cluster based on the least distance from the centroid. Once the data points are allocated, within the subsequent iterations, the centroids are realigned to the mean

of the data points, and the data points are once again added to the clusters based on the least vicinity from the centroids. These steps are iterated to the point where the centroids do not change more than the set threshold.

Another algorithm, for behavioural identification is the decision or regression trees [16] that can be used for classification or regression predictive modelling problems. Regression or prediction trees use the tree to represent the recursive partition. Each of the terminal nodes, or leaves, of the tree represents a cell of the partition and has attached to it a simple model which applies in that cell only. To figure out which cell we are in, we start at the root node of the tree and ask a sequence of questions about the features. The interior nodes are labelled with questions, and the edges or branches between them are labelled by the answers. In the classic version, each question refers to only a single attribute, a single input variable (x) and a split point on that variable which has a yes or no answer. In order to make a prediction for a given observation, we typically use the mean of the training data in the region to which it belongs.

A greedy approach is used to divide the space called recursive binary splitting, a numerical procedure where all the values are lined up and different split points are tried and tested using a cost function. The split with the best cost is selected, and the procedure stops when a predefined stopping criterion is used, such as a minimum number of training instances assigned to each leaf node of the tree. If the count is less than some minimum, then the split is not accepted and the node is taken as a final leaf node. The stopping criterion is important as it strongly influences the performance of your tree.

Figure 4.2 illustrates an example of a decision tree applied to a dataset regarding financial data records (FDR) based on the purposes of the proposed system.

A method that is similar to decision trees is the random forest [17] which can be also used for both classification and regression problems. A decision tree gives the set of rules that are used in building models, which can be executed against a test dataset for the prediction. In decision trees, first step is to calculate the root node. Similarly, in Random Forest, each tree will predict a different target variable that we will sum with respect to a key. The key with the highest count, predicted by the maximum number of trees, is the final. There are several advantages to this method, such as making fast predictions and easy implementation. Also, there might be no capability to go all the way down the tree, if some of the data is missing, but a prediction still can be made by averaging all the leaves in the sub-tree.

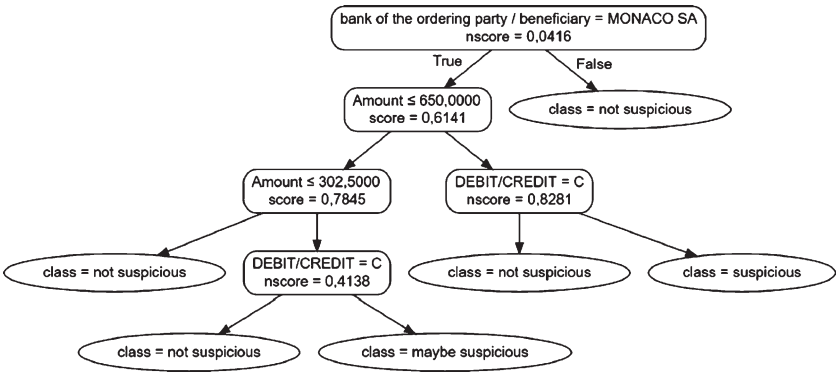

Fig. 4.2 Decision tree of the FDR dataset

The regression methods that are used for creating models which can identify and predict hidden patterns are statistical methods that examines the relationship between two or more variables of interest. The goal of the regression analysis is to predict the value of one or more target or response variables given the value of a vector of input or explanatory variables. In the simplest approach [17], this can be done by directly constructing an appropriate function y whose values for new inputs $\boldsymbol{x}$ constitute the predictions for the corresponding values of y, such as:

$$y = \beta_0 + \beta_1 * x_1 + \ldots + \beta_n * x_n$$

which is a linear regression model that combines a specific set of input values, the solution to which is the predicted output for that set of input values.

Learning a linear regression model means estimating the values of the coefficients used in the representation of the data. Below, there are some methods usually used for estimating the coefficients $\{\beta_0, \ldots \beta_n\}$ such as ordinary least squares [17], where we seek to minimize the sum of the squared residuals. This means that given a regression line through the data, the distance from each data point to the regression line is calculated, squared and summed. Another method is the gradient descent method [18], which is very useful in large datasets, works by starting with random values for each coefficient and the sum of the squared errors is calculated for each pair of input and output values. A learning rate (a) must be

selected that determines the size of the improvement step to take on each iteration of the procedure as a scale factor, and the coefficients are updated in the direction towards minimizing the error. The process is repeated until a minimum sum squared error is achieved or no further improvement is possible. Finally, the regularization method [18] which seeks to both minimize the sum of the squared error of the model on the training data (using ordinary least squares) and also reduce the complexity of the model (like the number or absolute size of the sum of all coefficients in the model). This way also manages to avoid the problem of overfitting the data which can lead to model inaccuracy. Like before, the coefficients are chosen, such that they minimize the loss function through regularization procedures for linear regression such as the Ridge regression and the Lasso regression, whose main difference is the penalty of the high coefficients they use in order to minimize the loss function.

For non-linear problems, artificial neural networks and general regression neural networks are used. More specifically, ANNs [19] provide a general practical method for learning real-valued, discrete-valued and vector-valued functions from examples, and, thus, they can be used in any regression problem. They are based on simple units called perceptron that takes a vector of real-valued inputs x, calculates a linear combination of these inputs by using appropriate weights for each input and provide an output based on an activation function φ. Perceptron, as a simple unit, can only express linear decision surfaces; however, by using multiple perceptrons, we can build multilayer networks that can express a rich variety of non-linear decision surfaces. The most common type is the feedforward neural network where the perceptrons are fully connected and there is only one direction in the information flow in the network.

Usage of ANNs requires two phases: the training phase and the evaluation phase. In the training phase, the network tries to learn the weights of the neurons that better fit the desired output. For this purpose, the common backpropagation algorithm is employed. On the other hand, in the evaluation phase, the network is tested under unseen data that have been excluded from the training in order to evaluate its performance. Depending on how the data is split into the training set and the unknown dataset, the following techniques are commonly used: N-fold cross-validation, leave-one-out cross-validation and repeated random test-train splits [20].

Moreover, general regression neural networks (GRNN) [21] is a one-pass algorithm that provides estimates of variables and converges to the

underlying (linear or nonlinear) regression surface. Assume that $f(x, y)$ represents the known joint probability density function of a vector random variable, x, and a scalar random variable, y. Let X be a particular measured value of the random variable x. The estimation of Y given x is:

$$\widehat{Y(x)} = \frac{\sum_{i=1}^{n} Y^i * expexp\left(-\frac{D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{n} expexp\left(-\frac{D_i^2}{2\sigma^2}\right)}$$

where $D_i^2 = \left(X - X^i\right)^T \left(X - X^i\right)$, and X is a specific value of the random variable x, n is the number of sample observations, X_i *and* Y_i are the sample values of the random variables x and y and σ is the smoothing parameter.

The selection of the smoothing parameter σ is an important issue in the creation of the GRNN network because it determines how closely the GRNN network matches to the prediction result with the training set data. A useful method of selecting σ is the holdout method. The main advantages of a GRNN network are their ability to learn fast and converge on the optimal regression surface as the number of samples increases. They are best used in cases where data is sparse, making it ideal in real-time scenarios, because the regression surface is directly defined throughout the space, even in the case of a single sample.

For the purposes of the proposed system, several machine learning approaches have been tested using a dataset regarding call data records (CDR). Specifically, in Table 4.1, the results of the GRNN compared to the other regression algorithms are presented regarding the estimated call duration.

Table 4.1 Results of regression models applied in CDR dataset

Model	*Median absolute error (min)*
Linear regression	6.03
Ridge regression	4.03
Lasso regression	4.42
Regression trees	5.89
Artificial neural networks (ANN) (4-5-2-1)	4.42
General regression neural networks (GRNN) ($\sigma = 0.09$)	3.63

4.5 Conclusions

In the current work, a novel approach based on state-of-the-art techniques in semantic processing and machine learning that aims to facilitate the work of the LEAs is proposed. Specifically, the proposed system employs semantic information fusion in order to infer assertions based on stated facts and rules provided by LEAs and fuses them in order to increase the reliability of the system. Then, machine learning techniques are applied in order to identify the trends and the patterns in the collected data and predict abnormal behaviour. The system is designed in order to provide evidence and back traceability that will allow the derived results to be used in court. The methodology and the initial results are presented in the current chapter, and it is expected that they will allow LEA's officers to solve crimes in less time.

Acknowledgement

This work has been performed under the H2020 786629 project MAGNETO, which has received funding from the European Union's Horizon 2020 Programme. This chapter reflects only the authors' view, and the European Commission is not liable to any use that may be made of the information contained therein.

Bibliography

1. https://redalertproject.eu/
2. http://www.lasie-project.eu/
3. https://tensor-project.eu/
4. https://www.victoria-project.eu/
5. https://roborder.eu/
6. Markman, A. B. (2013). *Knowledge representation*. Cambridge: Psychology Press.
7. Abburu, S. (2012). A survey on ontology reasoners and comparison. *International Journal of Computer Applications, 57*(17), 33–39.
8. Lloyd, J. W. (1987). *Foundations of logic programming (second, extended edition). Springer series in symbolic computation*. New York: Springer Verlag.
9. Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). Pellet: A practical OWL-DL reasoned. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 5*, 51–53.

10. Dinca, L. M., & Hancke, G. P. (2017). The fall of one, the rise of many: A survey on multi-biometric fusion methods. *IEEE Access, 5*, 6247–6289. https://doi.org/10.1109/ACCESS.2017.2694050.
11. Hechenbichler, K., & Schliep, K. (2004). *Weighted k nearest neighbor techniques and ordinal classification.* Retrieved from http://epub.ub.uni-muenchen.de/
12. Chomboon, K., Chujai, P., Teerarassamee, P., Kerdprasop, K., & Kerdprasop, N. (2015). An empirical study of distance metrics for k nearest neighbor algorithm. In *Proceedings of 3rd International Conference on Industrial Application Engineering. Japan*.
13. Porter, E. H., & Winkler, W. E. (1997). *Advanced record linkage system.* U.S. Bureau of the Census, Research Report.
14. Sowa, J.-F. (1984). *Conceptual structures. Information processing in mind and machine.* Amsterdam: Addison-Wesley.
15. Chein, M., & Mugnier, M.-L. (2008). *Graph-based knowledge representation: Computational foundations of conceptual graphs.* London: Springer.
16. Deshpande, A. (2018). *Artificial intelligence for big data*. Birmingham: Packt Publishing.
17. Draper, N., & Smith, H. (1998). *Applied regression analysis.* New York: Wiley.
18. Bishop, C. (2011). *Pattern recognition and machine learning.* Cham: Springer.
19. Mitchell, T. (1997). *Machine learning* (1st ed.). Berlin: McGraw Hill.
20. Arlot, S., & Celisse, A. (2010). A survey of cross validation procedures for model selection. *Statistics Survey, 4*, 40–79.
21. Specht, D. (1991). A general regression neural network. *IEEE Transactions on Neural Networks, 2*(6), 568–576.