



Comprehensive machine and deep learning analysis of sensor-based human activity recognition

Hossam Magdy Balaha¹ · Asmaa El-Sayed Hassan²

Received: 19 November 2022 / Accepted: 13 February 2023 / Published online: 8 March 2023
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2023

Abstract

Human Activity Recognition (HAR) is a crucial research focus in the body area networks and pervasive computing domains. The goal of HAR is to examine activities from raw sensor data, video sequences, or even images. It aims to classify input data correctly into its underlying category. In the current study, machine and deep learning approaches along with different traditional dimensionality reduction and TDA feature extraction techniques are suggested to solve the HAR problem. Two public datasets (i.e., WISDM and UCI-HAR) are used to conduct the experiments. Different data balancing techniques are utilized to deal with the problem of imbalanced data. Additionally, a sampling mechanism with two overlapping percentages (i.e., 0% and 50%) is applied to each dataset to retrieve four balanced datasets. Five traditional dimensionality reduction techniques in addition to the Topological Data Analysis (TDA) are utilized. Seven machine learning (ML) algorithms are used to perform HAR where six of them are ensemble classifiers. In addition to that, 1D-CNN, BiLSTM, and GRU deep learning approaches are utilized. Three categories of experiments (i.e., ML with traditional features, ML with TDA, and DL) are applied. For the first category experiments, the best-reported scores concerning the WISDM dataset are accuracy and WSM of 99.10% and 86.61%, respectively. When concerning the UCI-HAR dataset, the best-reported scores are accuracy and WSM of 100% and 100%, respectively. For the second category experiments, the best-reported scores concerning the WISDM dataset are accuracy and WSM of 95.34% and 89.62%, respectively. When concerning the UCI-HAR dataset, the best-reported scores are accuracy and WSM of 96.70% and 92.57%, respectively. For the third category experiments, the best-reported scores concerning the WISDM dataset are accuracy and WSM of 99.90% and 99.76%, respectively. When concerning the UCI-HAR dataset, the best-reported scores are accuracy and WSM of 100% and 100%, respectively. After concluding the final results, the suggested approach is compared with 6 related studies utilizing the same dataset(s).

Keywords Deep learning (DL) · Human activity recognition (HAR) · Machine learning (ML) · Oversampling · Topological data analysis (TDA)

Abbreviations

AI	Artificial intelligence	CNN	Convolution neural network
HAR	Human activity recognition	LSTM	Long short-term memory
SVM	Support vector machine	SMOTE	Synthetic minority oversampling technique
		SMOTEN	Synthetic minority over-sampling technique for nominal
		DBSMOTE	Density-based SMOTE
		PCA	Principal component analysis
		LDA	Linear discriminant analysis
		ICA	Independent component analysis
		RP	Random projection
		T-SVD	Truncated singular value decomposition
		TDA	Topological data analysis
		ML	Machine learning
		LGBM	Light gradient boosting machine
		XGB	XGBoost

✉ Hossam Magdy Balaha
hmbala01@louisville.edu; hossam.m.balaha@mans.edu.eg

Asmaa El-Sayed Hassan
asmaa_ehassan@mans.edu.eg

¹ Bioengineering Department, J.B. Speed School of Engineering, University of Louisville, Louisville 40292, Kentucky, USA

² Mathematics and Engineering Physics Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt

AdaBoost	Adaptive boosting
HGB	Histogram-based gradient boosting
RF	Random forest
DT	Decision tree
ETs	Extra trees
ADASYN	Adaptive synthetic
SOTA	State-of-the-art
GS	Grid search
GBDT	Gradient boosting decision tree

1 Introduction

Recently, artificial intelligence (AI) has been widely used in several applications (e.g., cancer recognition [1, 2], burnout analysis [3], exam correction [4, 5], diseases diagnosis [6–8], sign language interpretation [9], natural language processing [10], and pattern recognition [11]). AI accelerated development has paved the way for human activity recognition (HAR). It is concerned with using sensor data to recognize a specific action (or movement) of a person. It has become one of the broadest research topics because of the sensors and accelerometers availability, less power consumption, and low cost. It has been widely employed in smart home [12], medical care [13, 14], image analysis [15], video surveillance [16], military defense [17], sleep state detection [18], and behavior monitoring [19, 20]. In HAR, movements are indoors and outdoors-performed activities (e.g., talking, walking, running, sitting, and standing). Additionally, they can be more focused activities such as the activities achieved in a kitchen or on a factory floor [21]. In short, the fundamental task of HAR is to choose a suitable sensor and use it to observe and capture the activities of the user [22] as shown in Fig. 1. Recently, HAR can be classified into the sensor- and visual-based recognition [23, 24]. HAR sensor-based data

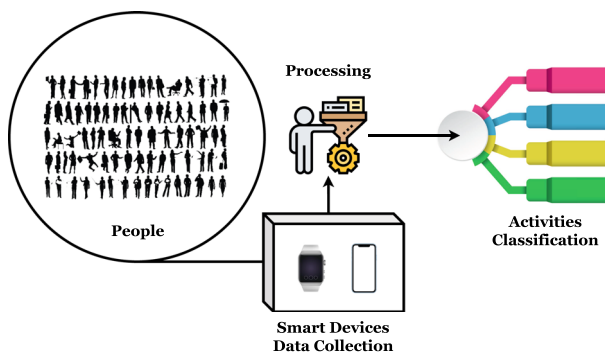


Fig. 1 The process of human activity recognition (HAR)

has become a research focusing-field because of the wide usage of wearable and portable sensors in daily life. The HAR sensors mainly include geomagnetic [25], acceleration [26, 27], and gyroscope [28].

Historically, it required custom hardware and was costly to gather and store data from sensors for activity recognition. Nowadays, smartphones, smartwatches, and other personal tracking devices, utilized for health and fitness monitoring, are inexpensive and omnipresent. As a result, sensor data collected from these devices are more common and inexpensive to collect, so, the activity recognition problem become a wide study field. Smartwatches have been beneficial in a broad range of healthcare applications, especially, the ones that are concentrating on health and fitness monitoring [29, 30]. Compared to other smart devices, smartwatches are truly wearable without interrupting the daily lives of the user [31]. The growth of smartwatches in the healthcare field has facilitated people to monitor their fitness and health [32]. Unfortunately, wearable sensors-gathered data are time-series data that are complex, noisy, and imbalanced [33]. Hence, HAR is a complex procedure, which contains the following steps: pre-process and segment the time series data, extract the data features, and then classify them by utilizing a classification algorithm.

Manual features extraction for HAR, based on classical machine learning (ML) algorithms, is required [34]. Dimensionality reduction and feature extraction methods are required for ML algorithms to achieve better performance. These methods aspire to find the most informational and compacted set of features by generating new ones from the existing features. They represent the most crucial part of classification because the performance is decreased significantly if the features are not suitable.

Creating classification models that can classify the less common activities is a significant challenge. Classification models designed to classify imbalanced data are biased to learn about the more frequently occurring classes. This type of bias happens as the models learn better from classes containing more data. Different methods were proposed to deal with the class imbalance problem and these methods can be split into two main approaches: data-level and classifier-level methods [35]. Traditional ML methods that have been used to perform the HAR task include Naive Bayes and support vector machines (SVM) [36]. Recently, the evolution of deep learning has resulted in being utilized widely in HAR [37]. It learns and extracts features automatically without the complex steps of manual feature extraction, hence, the workload of feature engineering is significantly decreased [38, 39]. Deep Neural Networks such as Recurrent Neural Networks and Convolutional Neural Networks (CNN) have gained significant performance across different applications and outperformed

many traditional methods. Lately, Long Short-Term Memory (LSTM) and CNN provide state-of-the-art results on HAR tasks with no or little feature engineering [40].

1.1 Research gap

In the HAR research field, a high-quality benchmark dataset for HAR methods is missing. Most of the publically available datasets suffer from limited or imbalanced data problems [33]. Most of the observed activities are simplistic and do not cover the entirety of human actions. Integrating deep architectures for solving HAR using context information can be difficult [41]. Although these methods deliver state-of-the-art performance on benchmark datasets, they are still overconfident in their predictions.

1.2 Research objectives

The major objective of the current study is to suggest an approach that is used for Human Activity Recognition (HAR). The objective is to suggest an analysis of machine and deep learning algorithms to deal with HAR. Additionally, extracting the most relevant features from raw data and reducing their dimensionality in an efficient manner are two challenging tasks.

1.3 Research contributions

The contributions of the current study can be summarized as follows:

- Performing human activity recognition tasks using a detailed comparative analysis of a variety of machine and deep learning algorithms to determine the optimal modal.
- Analyzing balancing and sampling techniques to deal with imbalanced data, hence, determining the best approaches.
- Reporting state-of-the-art performance metrics and comparing them with different related studies and approaches.

1.4 Paper organization

The rest of the current study is organized as follows: Sect. 2 reviews and summarizes the related literature. In Sect. 3, the background is discussed. It represents imbalanced data and oversampling techniques, features engineering and dimensionality reduction, Topological Data Analysis (TDA), feature scaling, and classification and optimization. In Sect. 4, a discussion about the methodology, datasets acquisition, data pre-processing phase, features engineering and dimensionality reduction techniques, ML classification

and optimization phase, and DL classification phase. Section 5 presents the details and discussions of the experiments and results. Section 6 presents the study limitations and finally, Sect. 7 conclude the paper and present the future work.

2 Literature review

Classical ML algorithms demand extensive domain expertise and feature engineering to transform raw sensor data into features, from which a classifier identifies activities (e.g., SVM [42] and random forest [43]). In Shi et al. [44], an algorithm based on the standard deviation trend analysis was utilized to recognize transition activity. For basic activity, SVM was mainly used for recognition. For transition activity, the standard deviation value of data was analyzed to evaluate the trend of the flow of the overall data to recognize the activity. The achieved accuracy by their proposed model was over 80% on real data.

In Garcia et al. [45], their placement-, orientation-, and subject-independent HAR dataset was introduced. An SVM algorithm was presented to perform the experiments on the dataset and accuracy of 74.39% was obtained. Their proposed model was able to tighten the gap between the real-life application and a model. Ahmed et al. [46] have proposed a hybrid method that contains a filter and a wrapper for the feature selection process. The process employed a sequential floating forward search to extract the features that would be fed to a multi-class SVM. Their model was validated on a public benchmark dataset [47] and an average accuracy of 98.13% was delivered. Their proposed system provided acceptable activity recognition and operated efficiently with limited hardware resources.

Deep learning algorithms such as recurrent neural networks [48] and convolutional neural networks [49] conduct automatic feature extraction and classification. They have delivered promising results in different sensor-based HAR scenarios [50]. Barut et al. [51] used a single wearable sensor to create a new dataset and utilized a multi-task LSTM model for intensity evaluation and activity recognition to deliver better outcomes. Accuracy of 97.76% and F1-score of 83.43% were obtained. Wang and Liu [52] suggested a Hierarchical-LSTM based on the LSTM for human activity recognition. Three public UCI datasets were used to train and evaluate their model and accuracy of 99.15% was achieved.

Furthermore, convolutional neural networks are used for HAR tasks for temporal features extraction and to produce significant performance advancement [53–55]. Zhang et al. [34] utilized the encoder and decoder operations of the U-Net architecture in creating their proposed HAR framework. Rather than sliding window labeling, dense

labeling was used to provide a single label per sample in the time series data. Additionally, to enhance the performance of the dense prediction outcome, a post-correction algorithm was utilized. Four datasets, including the WISDM dataset [56], UCI HAPT dataset (HAPT) [47], UCI OPPORTUNITY Gesture dataset (OPP Gesture) [57], and the dataset of self-collected Sanitation was used to conduct experiments. For the OPP Gesture dataset, an accuracy of 94.78% was obtained by their U-Net_PC model. Teng et al. [58] used the local loss function to achieve the layer-wise training of a convolution neural network for HAR. Their method was evaluated on five public datasets, namely UCI HAR dataset [42], Opportunity dataset [59], UniMib-SHAR dataset [60], PAMAP2 dataset [61], and WISDM dataset [36]. The reported accuracy and F1-score were 98.82% and 98.81%, respectively.

To extract powerful features from raw sensor-based data automatically and effectively, Ronao and Cho [62] proposed a model formed of alternating convolution and pooling layers. To predict human activities, the extracted features from the previous layers were passed to the fully connected and SoftMax layers. The dataset has been collected from 30 volunteer subjects and the dataset proposed in [63] was used to train and evaluate their model, reaching an overall performance of 94.79% on the dataset with raw sensor data, and 95.75% with further information of temporal fast Fourier transform of the HAR dataset. Bianchi et al. [64] suggested a CNN model formed of four convolution layers and only one fully connected layer for HAR and performed well on their small training set. Their system was designed to recognize nine different activities with an accuracy of 97%.

A different design paradigm that was very prevalent among the community of activity recognition was to create hybrid models [50, 65]. In Ordonez and Roggen [66], a DL architecture using a combination of convolutional and recurrent neural networks was proposed to conduct HAR from wearable sensors. Two public datasets, OPPORTUNITY [59] and Skoda [67] were used to evaluate their proposed approach. For the Skoda dataset, an F1-score of 95.8% was obtained. In Xia et al. [68], two LSTM layers and cascading convolutional layers were employed to extract features from time-series data. To maintain the classification performance while reducing the model parameters, a global average pooling layer was utilized rather than the fully connected layer. Three public datasets, UCI [47], WISDM [56], and OPPORTUNITY [57, 59] were used to evaluate the model performance. The overall accuracy of the model for the UCI-HAR dataset was 95.78%, for the WISDM dataset was 95.85%, and for the OPPORTUNITY dataset was 92.63%.

Ignatov et al. [69] used CNN and statistical features to extract features from sensor data. The assembled features vector was handed to the next layers to identify the activities. The proposed approach was evaluated on two commonly used datasets (i.e., WISDM [56] and UCI [47]). The reported accuracy and F1-score were 97.63% and 97.62%, respectively. The results indicated that their presented model delivered state-of-the-art performance while demanding no manual feature engineering and low computational cost. In Xu et al. [55], the inception module of GoogLeNet architecture was explored to extract spatial features from sensor data.

Furthermore, temporal features obtained using the recurrent neural network and spatial features were combined. Three benchmark datasets were used to conduct experiments, OPPORTUNITY dataset [59], PAMAP2 dataset [61], and Smartphone database [70]. For the OPPORTUNITY dataset, an F-measure of 94.6% was obtained. Khan and Ahmad [71] proposed a multi-head attention-based model for HAR. Their framework included three lightweight convolutional heads, each created using one-dimensional CNN to extract features from input sensor data. Their model was induced with attention to strengthening the representation ability of CNN. Two publicly available datasets: WISDM [56] and UCI HAR [47] were used to conduct ablation experiments and studies and evaluate the proposed model. The achieved F1-score was 97.20% for the WISDM dataset.

When dealing with HAR data, a problem of imbalanced data may exist. To tackle the imbalance problem, the most intuitive path is to re-sample the class with the largest number of samples as done in Alani et al. [33]. Experiments were done using an extensive sensor-based multi-modal dataset developed from the Sensor Platform for Healthcare in a Residential Environment [72]. The results showed that when using the SMOTE oversampling technique to correct the class imbalance, CNN-LSTM achieved the highest classification accuracy of 93.67% followed by CNN of 93.55%, and LSTM of 92.98%. Grzeszick et al. [73] used two augmentation techniques, Gaussian noises perturbation, and interpolation to solve the class imbalance problem. In their study, a CNN was utilized on the multiple inertial measurement units sequential data. A dataset introduced in [74] was used to evaluate the proposed model and classification accuracy of $73.9\% \pm 4.6\%$ was obtained.

2.1 Related studies summarization

Table 1 presents a comparison between the related studies and the current study. The related studies are ordered from the oldest to the latest.

Table 1 Comparison between the related studies and the current study

Author(s)	Year	Methodology	No. of Datasets	ML	DL	Oversampling	Undersampling	Optimization	TDA	Feature reduction	Results
Ordonez and Roggen [66]	2016	CNN + LSTM	2	X	✓	X	X	X	X	X	F1-score: 95.8%
Ronao and Cho [62]	2016	CNN	2	X	✓	X	X	X	X	X	Accuracy: 95.75%
Grzeszick et al. [73]	2017	IMU-CNN	1	X	✓	✓	X	X	X	X	Accuracy: 73.9% ± 4.6%
Ignatov et al. [69]	2018	CNN + statistical features	2	✓	✓	X	X	X	X	X	Accuracy: 97.63% and F1-score: 97.62%
Bianchi et al. [64]	2019	CNN	1	X	✓	X	X	X	X	X	Accuracy: 97%
Xu et al. [55]	2019	InnoHAR model	3	X	✓	X	X	X	X	✓	F-measure: 94.6%
Zhang et al. [34]	2019	U-Net	4	X	✓	X	X	X	X	X	Accuracy: 94.78%
Teng et al. [58]	2020	CNN	5	X	✓	X	X	X	X	X	Accuracy and F1-score are 98.82% and 98.81%
Alani et al. [33]	2020	CNN-LSTM	1	X	✓	✓	✓	X	X	X	accuracy: 93.67%
Shi et. al. [44]	2020	SVM + STD-TA	1	✓	X	X	X	X	X	✓	Accuracy: 80%
Garcia et. al. [45]	2020	SVM	1	✓	X	X	X	X	X	✓	Accuracy: 74.39%
Ahmed et. al. [46]	2020	SFFS + SVM	1	✓	X	X	X	X	X	✓	Average: 98.13%
Barut et. al. [51]	2020	Multitask LSTM	1	X	✓	X	X	X	X	X	Accuracy: 97.76% and F1-score: 83.43%
Wang and Liu [52]	2020	Hierarchical deep LSTM (H-LSTM)	3	X	✓	X	X	X	X	✓	Accuracy: 99.15%
Xia et al. [68]	2021	CNN + LSTM	3	X	✓	X	X	X	X	X	Accuracy: 95.85%
Khan and Ahmad [71]	2021	Attention induced multi-head CNN	2	X	✓	X	X	X	X	X	F1-score: 97.20%
Current Study	2022	Section 4	2	✓	✓	✓	X	✓	✓	✓	Section 5

3 Background

3.1 Imbalanced data and oversampling techniques

The imbalance problem appears when one of the target classes has a small instances number compared to other classes. Normally, a typical classifier neglects to detect a minority class due to the little number of class samples. Newly, there is a significant interest to solve the class imbalance issue. It considers as a challenging issue that requires more attention to be resolved by several researchers [75, 76]. Using resampling techniques to make the dataset balanced is considered one of the common procedures. Resampling methods can be utilized either by oversampling or undersampling the dataset [77]. Undersampling can be described as the process of decreasing the amount of majority target samples (i.e., instances) [78] such as tomesk's links [79] and cluster centroids [80]. Oversampling can be achieved by increasing the amount of minority class samples by repeating some instances or producing new instances [81].

In the current study, only the oversampling approach is used. They are summarized as follows:

- **Synthetic Minority Oversampling Technique (SMOTE):** In a classic oversampling technique, the number of data is increased, no further variation or information is given to the ML model. Chawla et al. presented SMOTE [82] that operates differently. It creates synthetic data to overcome the overfitting problem posed by random oversampling. It utilizes the K-nearest neighbor algorithm. Initially, it starts by choosing data randomly from the minority class. Then, the K-nearest neighbors are set from that data. Synthetic data would be constructed among the selected K-nearest neighbor and the random data.
- **Synthetic Minority Over-sampling Technique for Nominal (SMOTEN)** is a development of SMOTE which is used for nominal features proposed by Chawla et al. [82]. In it, the nearest neighbor is calculated using a modified version of the Value Difference Metric [83, 84].
- **Borderline-SMOTE** is a variation of the SMOTE. Unlike the SMOTE, it generates synthetic data only along the decision boundary between the two classes [85].
- **Adaptive Synthetic (ADASYN)** is a generalized form of the SMOTE algorithm. Similar to SMOTE, it aims to oversample the minority class by generating synthetic instances for it. When compared to Borderline SMOTE, it takes a different approach. As Borderline SMOTE aims to synthesize the data around the decision

boundary of the data, while ADASYN creates synthetic data considering the data density [86].

- **K-means SMOTE** is an effective and simple oversampling method that is used to solve class-imbalanced data based on SMOTE and K-means clustering oversampling. It aims to aid classification by developing minority class instances in crucial and safe areas of the input space. This method averts the noise generation and effectively overcomes the imbalances within and between classes [87].
- **Borderline SMOTE SVM (SVM SMOTE)** is another variation of the Borderline SMOTE [88]. The primary difference between this technique and the other SMOTE ones, it incorporates the SVM algorithm to identify the misclassification instead of using K-nearest neighbors. In the SVM SMOTE, the support vectors are used to approximate the borderline area after training SVMs on the original training set.

3.2 Features engineering and dimensionality reduction techniques

Feature engineering pipeline is the preprocessing step that extracts features from raw data and transforms them into formats that can be ingested by the ML algorithms [4, 89]. It helps the ML algorithms to determine patterns in data that boost their performance. Feature engineering is an important task to develop predictive solutions [90] but it is a challenging and the least well-studied topic in ML and data-mining [91]. Feature engineering is a manual problem-specific process performed by ML and domain experts [92]. In ML, feature engineering consists of four main steps: feature creation, feature transformation, feature extraction, and feature selection [93, 94].

- **Feature Creation:** This step includes specifying the variables that can be useful in the predictive model. It is a subjective process that requires human creativity and intervention. Features are incorporated by multiplication, addition, and subtraction to construct new derived features with higher predictive power [95]. Usually, ML experts combine features in a trial-and-error manner until the generated features fulfill the expectations [90, 96]. Using automated feature generation methods takes a long time to produce an outcome and is computationally expensive [97, 98].
- **Feature Transformation:** Feature transformation usually implies simpler modifications over the features [99]. Transformation involves manipulating the predictor variables to improve model performance. It is used to ensure that the variables are on the same scale, guarantee the flexibility of the model in which a variety of data can be ingested, make the model easier to

understand, avoid computational errors, and improve accuracy. Some of the standard transformations are binning, rounding, scaling, exponential transformations, logarithmic transformations, and power functions [93].

- **Feature Extraction:** These techniques aim to find a smaller set of new variables in which each is a combination of the input variables, including the same information as them. Feature extraction is used to develop new variables by extracting them from raw data. It aims to reduce the data volume into a more suitable set for modeling [100]. These methods include text analytics, cluster analysis, principal components analysis, and edge detection algorithms.
- **Feature Selection:** When applied to the original dataset, only the most relevant variables are kept. Feature selection algorithms are used to analyze, judge, and rank a subset of features from the pool of available features. Selection is employed to determine which features are irrelevant and redundant to be removed, and which are the most useful features for the model and should be prioritized [101]. These methods can be divided into four high-level categories: filter (i.e., ANOVA, Pearson correlation, variance thresholding), wrapper (i.e., forward, backward, and stepwise selection), embedded (i.e., Lasso, Ridge, Decision Tree), and (4) hybrid methods [102, 103].

3.2.1 Features extraction

In the current work, only feature extraction techniques are applied to perform dimensionality reduction. These techniques are operated to reduce model complexity, overfitting, generalization error, and increase the computation efficiency of the model [104]. They are:

- **Principal Component Analysis (PCA)** is a method of acquiring the important features from a large feature-set available in a dataset. It finds the direction of maximum variance in high-dimensional data and views the data onto a further subspace with dimensions equal to or fewer than the original ones [105].
- **Linear Discriminant Analysis (LDA)** is a supervised learning feature extraction technique that targets to decrease the spreading inside the class itself and increase the distance between the classes mean [106, 107].
- **Independent Component Analysis (ICA)** is a linear method that takes independent components mixture as input data and targets to identify each of them correctly [108].
- **Random Projection (RP)** is a technique utilized to perform feature reduction for points set that lie in the Euclidean space [109].

- **Truncated Singular Value Decomposition (T-SVD)** is a matrix factorization technique, similar to PCA, used to reduce the dimensionality of the data. Opposite to PCA, the data is not centered before calculating the singular value decomposition. This means it can be used efficiently with sparse matrices [110]. The data matrix is factorized by t-SVD where the number of columns is equal to the truncation.

3.2.2 Topological data analysis (TDA)

Topology [111] is the study of shapes and their properties. It deals with properties of the shapes (e.g., the number of components and loops in shapes). Topological Data Analysis (TDA) is an approach of datasets analysis using topological techniques [112]. It exploits the topological and geometrical properties of data such as shape and connectivity. TDA is inspired by the notion that geometry and topology deliver a robust approach to infer strong qualitative information about the data structure [113]. Datasets that are incomplete, high-dimensional, and noisy are challenging to extract information from. TDA provides a general framework to analyze such sets and provide robustness to noise and dimensionality reduction [113]. Additionally, TDA inherits functoriality (i.e., a functor is a mapping between categories), a fundamental concept of modern mathematics which allows it to adapt to new mathematical tools [114].

Persistent homology is a central tool of TDA to construct multi-scale invariants of data and represents them with barcodes or persistence diagrams [115, 116]. It considers data as a point cloud and tries to find the holes in point clouds using discretization and triangulation of the initial data space with simplicial complexes. TDA offers: (1) **A compressed mathematical representation of a dataset:** The single data point detail up to the global structure of a dataset can be studied without bearing a cognitive overload, (2) **Missing data and noise resistance:** TDA maintains important features of the data, (3) **Invariance.** The size, orientation, or skew of data does not change the data as only connectedness matters, (4) **Exploration tool of data:** Fetch answers to questions that have not been asked yet, and (5) **Data and manifolds shape study tool:** TDA inherits functoriality and has a robust theoretical foundation.

TDA creates the persistence diagram which is a 2D plot that indicates the birth and death of n-dimensional holes in the induced topological spaces. TDA provides the Mapper [117]. It is considered as a combination of clustering, dimensionality reduction, and graph networks techniques utilized to get a higher-level understanding of the data structure. It is used to: (1) Visualize the shape of data

through a particular lens, (2) Detect interesting topological structures (i.e., clusters) that cannot be found by the traditional methods, and (3) Select the best features that discriminate data and for model interpretability.

In biological research fields, several publications have successfully used TDA. These include Type-2 diabetes (T2D) subgrouping using clinical parameters [118], modeling RNA hairpin folding [117], and gene expression patterns-based breast cancer classification [119].

3.3 Feature scaling techniques

Feature scaling, also named data normalization, is an approach employed to normalize the range of data features or independent variables [120]. In this work, five of the most commonly used feature scaling techniques are utilized [121].

- **Absolute Maximum (Max-Abs) Scaling** is used to compute the absolute maximum value of the feature in the dataset and then all the values in the column are divided by that value [122]. The output values will range between 1 and -1.
- **Minimum Maximum (Min-Max) Scaling** is achieved by subtracting the minimum value with all the values in the dataset and then dividing the output value by the dataset range (i.e., maximum value - minimum value) [123].
- **Normalization:** The maximum value is used to perform normalization [124]. In the previous cases, the range of the data is changed while in normalization the distribution shape of the data is changed.
- **Standardization (Z-Score Normalization):** Z-score is calculated for each data point and replaces the data value with the calculated one [124]. As a result, all features are centered around the mean value with a standard deviation of 1.
- **Robust Scaling** is not prone to outliers. In this method, the median value is subtracted from all the data points and then the output values are divided by the Inter Quartile Range (IQR) value [125]. The IQR is the distance between the 25th and the 75th percentile points. Hence, the median value is centered at zero.

3.4 Classification, optimization, and performance evaluation

Machine learning (ML) algorithms are proved to be useful in a broad variety of applications (e.g., email filtering, computer vision, speech recognition, and medicine) in which it is infeasible to design conventional algorithms to accomplish the required tasks [126]. ML techniques are

gaining huge attention in data mining, leveraging it to recognize historic trends to deliver future models [127].

Deep learning (DL) is a subset of ML where artificial neural networks learn from large amounts of data [128, 129]. DL algorithms are used to solve complex problems where datasets are diverse, unstructured, and inter-connected. The more DL algorithms learn, the better they perform [130]. Several types of DL algorithms have existed. These include convolutional neural networks (CNNs) and recurrent neural networks (RNNs). A CNN is an algorithm designed for object detection and image processing. The convolution is a unique filtering process performed through an image to assess every element within it [131, 132]. The RNN has built-in feedback loops that allow the algorithms to remember past data points. They can use this memory to inform their understanding of current events.

3.4.1 Machine learning classifiers

Simple classifiers include Naive Bayes, Decision Tree, Logistic Regression, and K-Nearest Neighbor [133]. Then, there are ensemble classifiers which refer to algorithms that merge the predictions from two or more models. Their popularity is because of their ease in implementation and success on an expansive domain of predictive modeling problems [134]. In the current study, seven types are used where six of them are ensemble classifiers, they are:

- **Light Gradient Boosting Machine (LGBM) Classifier** is a distributed gradient boosting framework primarily created by Microsoft to be employed in ML [135, 136]. It is established over decision tree [137] algorithms and used for classification, ranking, and further tasks. The development of LightGBM concentration is on scalability and performance [138].
- **XGBoost (XGB) Classifier** is a distributed scalable gradient-boosted decision tree algorithm [139]. It is the leading ML algorithm for ranking, classification, and regression problems. It delivers parallel tree boosting, sparse optimization, multiple loss functions, regularization, bagging, and early stopping.
- **Adaptive Boosting (AdaBoost) Classifier** is a meta-estimator that starts with fitting a classifier on a set of data and then additional copies of the classifier are fitted on the same set [140].
- **Histogram-based Gradient Boosting (HGB) Classifier** is a gradient boosting method that executes this technique and customizes the training algorithm around input variables under this transform. It has native support for the missing values. The main drawback of gradient boosting is that training the model is time-

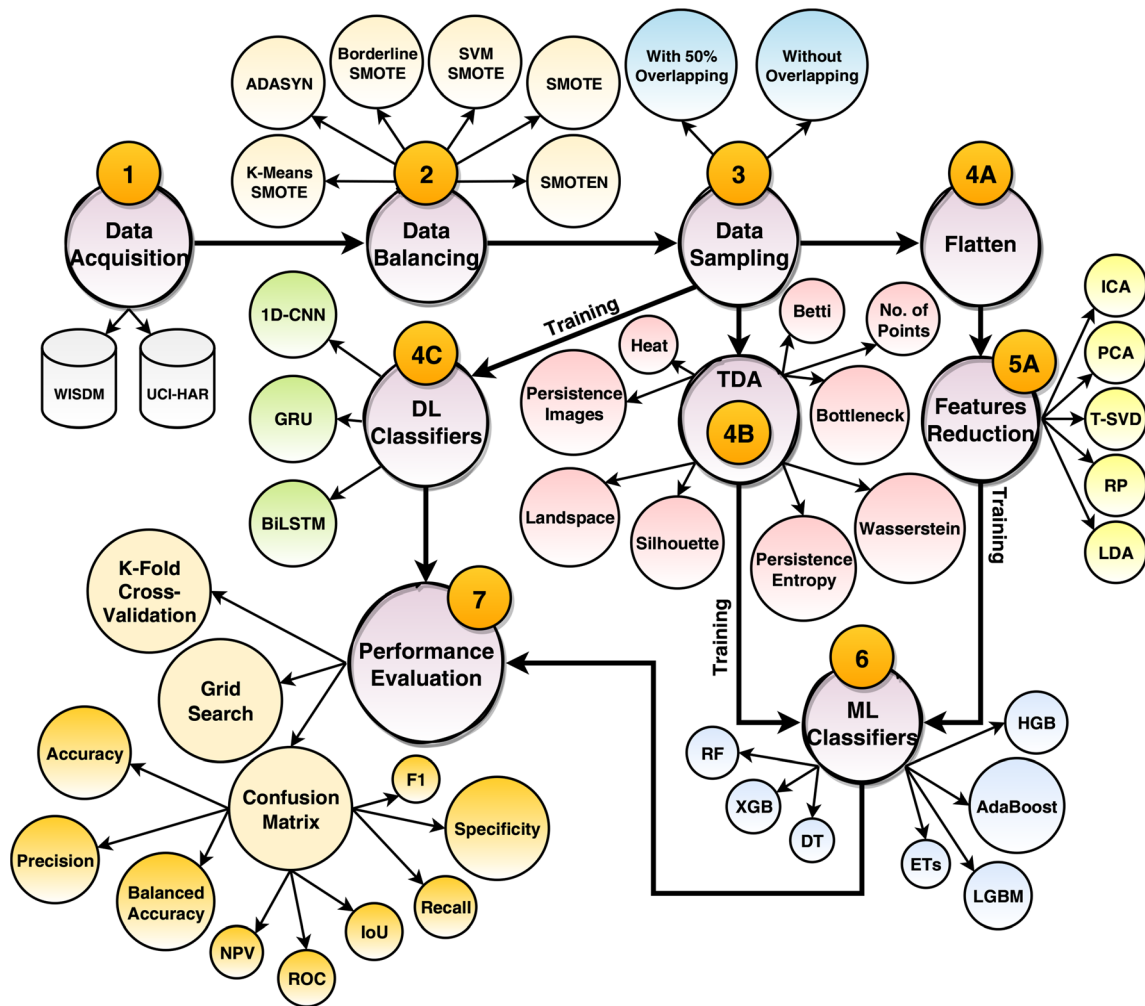


Fig. 2 Graphical presentation of the suggested methodology

consuming which appears when using the model on datasets with tens of thousands of examples.

- **Random Forest (RF) Classifier** is a method that merges a large number of independent trees trained by equally and randomly distributed subsets of the data [141]. It is one of the most utilized algorithms because of its simplicity and diversity (i.e., can be used for regression, classification, and other tasks that function by building a group of decision trees at the time of training) [142, 143].
- **The Decision Tree (DT) Classifier** is a supervised non-parametric learning algorithm used for classification and regression [144]. It is a tree-structured classifier, in which the features of a dataset are represented by internal nodes, the decision rules are represented by branches (i.e., decision nodes), and an outcome is represented by a leaf node. Hence, there are two types of nodes including decision and leaf nodes [145].
- **The Extra Trees (ETs) Classifier** is an ensemble learning method in which the results of numerous de-

correlated decision trees gathered in a forest are aggregated to output the result of the classification [146].

3.4.2 Deep learning classifiers

In DL, algorithms use the input distribution to extract features and useful data patterns during the training process. Deep learning models include several algorithms. In the current study, three types of algorithms are used. They are 1D Convolutional Neural Network (1D-CNN) [147, 148], Gated Recurrent Unit (GRU) [149], and Bi-directional Long Short-Term Memory Network (BiLSTM) [48]. They are discussed as follows:

- **1D Convolutional Neural Network (1D-CNN) Classifier** is a modified version of CNN that has been recently developed. In 1D-CNN, the computational complexity is quite lower than the CNN [148]. Hence,

Table 2 Summary of the datasets used in the current study

Dataset	Classes #	Classes	Records #	Features #	Is balanced?
Wireless Sensor Data Mining (WISDM) v1.1 [56]	6	“Walking”, “Jogging”, “Sitting”, “Standing”, “Upstairs”, and “Downstairs”	1,098,208	3	No
Human Activity Recognition Using Smartphones Data Set v1.0 (UCI-HAR) [70]	6	“WALKING”, “WALKING_UPSTAIRS”, “WALKING_DOWNSTAIRS”, “SITTING”, “STANDING”, and “LAYING”	10,299	561	No

Table 3 Comparison between the oversampling techniques on the “WISDM” dataset

Technique	Crash?	Average time	Balanced?
Synthetic minority over-sampling technique (SMOTE)	✗	11.90 seconds	✓
Synthetic minority over-sampling technique for nominal (SMOTEN)	✓	N/A	N/A
Borderline SMOTE	✗	26.95 seconds	✓
Adaptive synthetic (ADASYN)	✗	30.27 seconds	✗
K-Means clustering with SMOTE	✓	N/A	N/A
Support vector machine (SVM) with SMOTE	✗	Too long time	N/A

1D-CNNs are suitable for real-time and low-cost applications.

- **Gated Recurrent Unit (GRU) Classifier** is a gating mechanism in RNNs. It is similar to an LSTM with a forget gate [150] but with fewer parameters, as the output gate does not exist [151]. GRUs have been delivered better performance on less frequent and smaller datasets [152].
- **Bi-Directional Long Short Term Memory Network (BiLSTM) Classifier** is a sequence processing model that includes two LSTMs: the first processes the input in a forward direction, and the other in a backward direction [153]. It increases the available amount of information to the network effectively, enhancing the context available to the algorithm.

3.4.3 K-fold cross-validation

Cross-validation [154] is a resampling technique employed to assess ML models on a limited dataset. Cross-validation is used to detect overfitting (i.e., failing to generalize a model) [155]. This approach has only one parameter called “K” where the input data is split into K-fold (i.e., subsets of data).

3.4.4 Grid search hyperparameter optimization

Hyperparameter optimization (i.e., tuning) is the process of finding the most suitable values of the hyperparameters and it is one of the most important parts of ML [156, 157]. A model with poor performance and wrong results may result from a wrong choice of the values of the hyperparameters [158]. Hyperparameters are the parameters of a model whose values are specified before training and their values influence the model behavior. For example, the number of trees in a random forest is a hyperparameter as its value is set before training. As the optimal values of the hyperparameters are unknown, hyperparameters optimization algorithms are required. The current study utilizes the grid search, also referred to as full factorial design [159]. For the model with several hyperparameters, the best combination of hyperparameters values needed to be found by searching in a multi-dimensional space.

4 Methodology

The suggested methodology is presented and summarized graphically in Fig. 2 and discussed in detail in the next subsections.

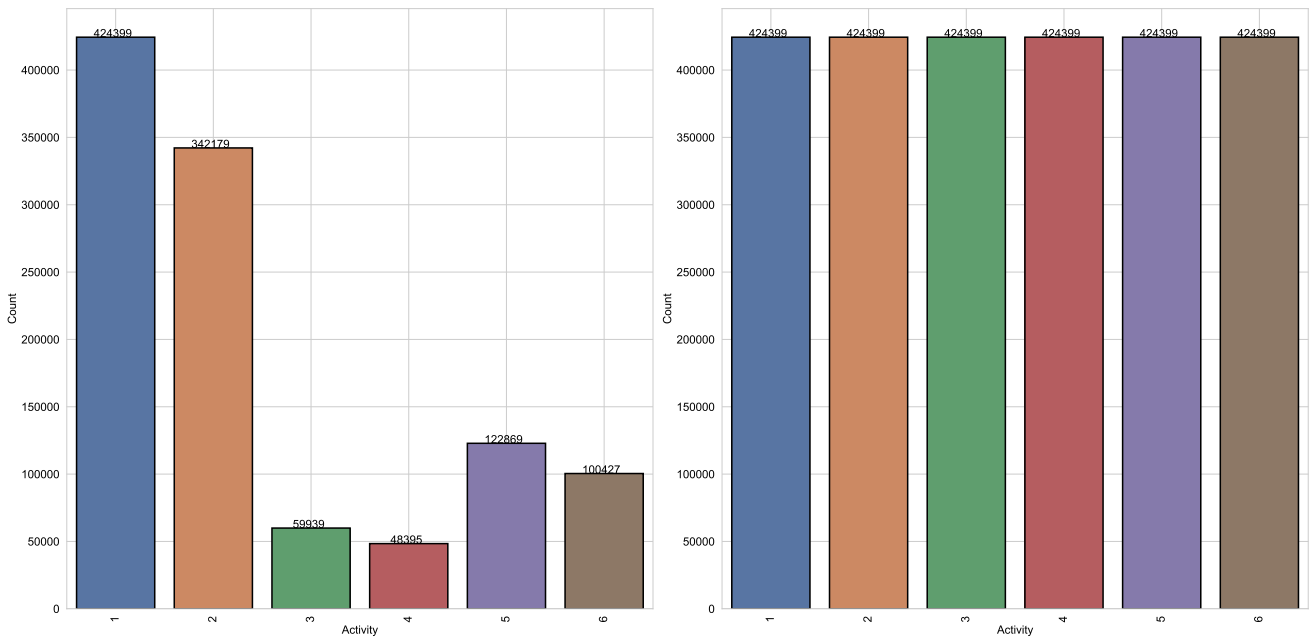


Fig. 3 The distribution of each category of the “WISDM” dataset before (left plot) and after (right plot) SMOTE oversampling

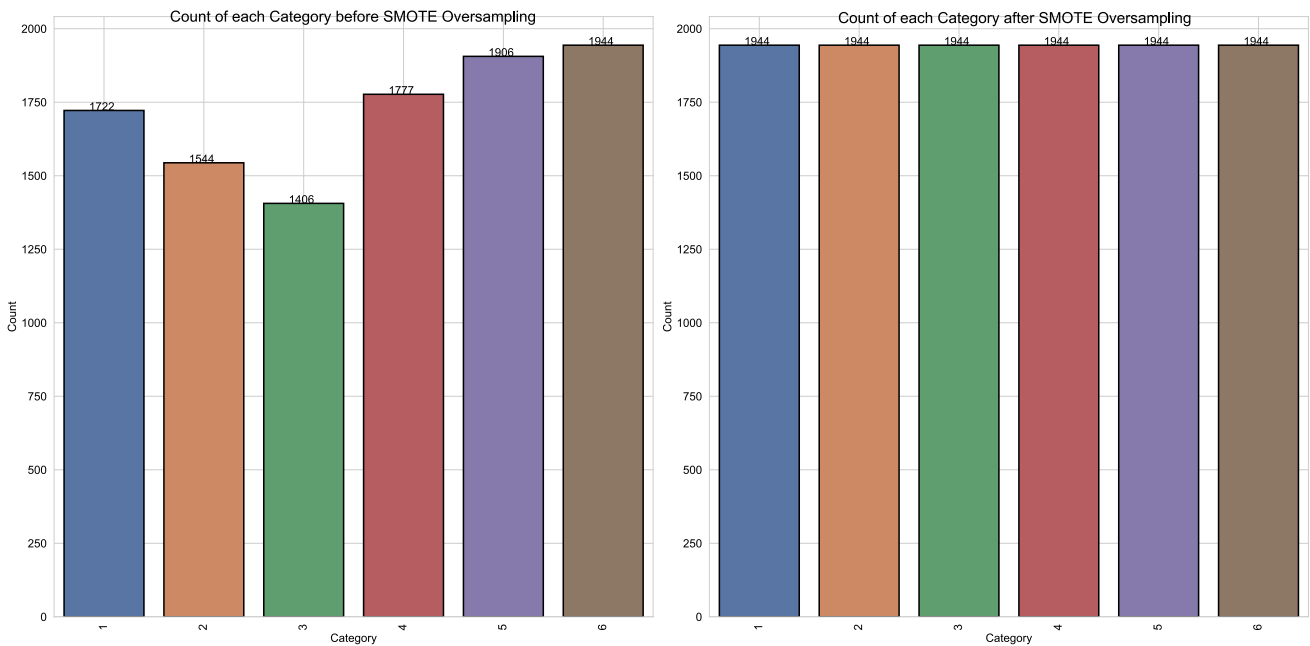


Fig. 4 The distribution of each category of the “Human Activity Recognition Using Smartphones Data Set v1.0 (UCI-HAR)” dataset before (left plot) and after (right plot) SMOTE oversampling

4.1 Data acquisition phase

The datasets used in the current study are retrieved from different public sources. They are:

- **Wireless Sensor Data Mining (WISDM) v1.1:** It consists of 6 columns and 1,098,208 rows (i.e., records). The columns are “user”, “activity”,

“timestamp”, “x-acceleration”, “y-acceleration”, and “z-acceleration”. There are 6 activities (i.e., “Walking,” “Jogging,” “Sitting,” “Standing,” “Upstairs,” and “Downstairs”). The data were sampled with a sampling rate of 20Hz (i.e., one sample per 50ms). The “user” field is from 1 to 36. The “x-acceleration,” “y-acceleration,” and “z-acceleration” fields are from -20 to 20 and are measured by the Android phone’s

Table 4 The data sampling criteria

Dataset	Sample size	Step size	Overlapping	Record # (imbalanced)	Record # (balanced)	Output shape (balanced and sampled)
Wireless Sensor Data Mining (WISDM) v1.1 [56]	100	50	50%	1,098,208	2,546,394	(50,926 × 100 × 3)
	100	100	0%	1,098,208	2,546,394	(25,463 × 100 × 3)
Human Activity Recognition Using Smartphones Data Set v1.0 (UCI-HAR) [70]	100	50	50%	10,299	11,664	(232 × 100 × 561)
	100	100	0%	10,299	11,664	(116 × 100 × 561)

accelerometer [56]. It can be retrieved from <https://www.cis.fordham.edu/wisdm/dataset.php>.

- **Human Activity Recognition Using Smartphones Data Set v1.0 (UCI-HAR):** It is applied on 30 volunteers in the age ranged from 19 to 48 years using “Samsung Galaxy S II” Android device. There are 6 activities (i.e., “WALKING,” “WALKING_UP-STAIRS,” “WALKING_DOWNSTAIRS,” “SITTING,” “STANDING,” and “LAYING”). The dataset is partitioned into two subsets where 70% for training (i.e., “train” folder) and 30% for testing (i.e., “test” folder). The number of training data is 7,352 while the number of testing data is 2,947. The features are 561 and their names are defined in “features.txt” [70]. It can be retrieved from <https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones>. The training and testing subsets are merged in the current study and partitioned in a later process.

It is worth noting that the datasets combine static and dynamic activities. Static activities include sitting, standing, and lying while dynamic activities include walking, walking downstairs, and walking upstairs. Table 2 presents a summary of the datasets used in our study.

Table 5 The different used hyperparameters of the classifiers

Classifier	Hyperparameter	Alternatives
LGBM	Learning rate	[0.01, 0.1, 1.0]
XGB	Learning rate	[0.01, 0.1, 1.0]
AdaBoost	Learning rate	[0.01, 0.1, 1.0]
DT	Criterion	“Gini” and “Entropy”
	Splitter	“Best” and “Random”
ETs	Criterion	“Gini” and “Entropy”
RF	Criterion	“Gini” and “Entropy”
HGB	–	–

4.2 Data pre-processing phase

The datasets are pre-processed before being used in the later phases. Data balancing and sampling are applied.

4.2.1 Data balancing

The datasets are not balanced and this can lead to overfitting or misclassification issues [160]. The current study utilized the techniques discussed in Sect. 3.1 to determine the most suitable technique to use in the experiments. The techniques are executed for 10 runs on the same dataset to determine the average time between them. Table 3 shows if the technique crashes or not, the average time and if the technique produces balanced data. They are applied to the “WISDM” dataset as it contains a large volume of records to check if the technique is scalable or not. If the technique crashes then it is not a scalable technique.

Table 3 shows that the SMOTE method outperforms other methods as it does not crash, produces balanced datasets, and consumes less time. Hence, synthetic oversampling is applied in the current study on the used datasets using the SMOTE technique [82]. Figure 3 shows the distribution of each category of the “WISDM” dataset before and after SMOTE oversampling. It reached 2,546,394 records after the SMOTE balancing process. Figure 4 shows the distribution of each category of the “Human Activity Recognition Using Smartphones Data Set v1.0 (UCI-HAR)” dataset before and after SMOTE oversampling. It reached 11,664 records after the SMOTE balancing process. In both figures, the x-axis is the categories and the y-axis is the count of each category.

4.2.2 Data sampling

The datasets are time-series data and instead of using row-by-row in the classification phase, a sampling mechanism is applied. The sampling is performed vertically which means that the records are stacked row-by-row after the sampling process. It is applied using different configurations as shown in Table 4. The second column is the sampling size, the third column is the step size, the fourth

Table 6 Summary of the feature scaling equations

Technique	Equation	Specifications
Standardization	$X_{\text{scaled}} = \frac{X_{\text{input}} - \mu}{\sigma}$	Zero mean and standard deviation of 1
Normalization	$X_{\text{scaled}} = \frac{X_{\text{input}}}{\max(X)}$	–
Min-Max	$X_{\text{scaled}} = \frac{X_{\text{input}} - \min(X)}{\max(X) - \min(X)}$	Output range $\in [0, 1]$
Max-Abs	$X_{\text{scaled}} = \frac{X_{\text{input}}}{ \max(X) }$	Output range $\in [-1, 1]$
Robust	$X_{\text{scaled}} = \frac{X_{\text{input}} - \text{median}(X)}{IQR}$	–

column is the overlapping percentage, the fifth column is the number of records of the imbalanced dataset, the sixth column is the number of records of the balanced dataset, and the seventh column is the output shape of the balanced dataset after sampling.

4.3 Features engineering and dimensionality reduction techniques

4.3.1 Features extraction using TDA

The TDA is used to extract the features from the datasets in nine flavors. They are: (1) **Persistence entropy** in two views: normalized and non-normalized, (2) **Number of Points**, (3) **Bottleneck**, (4) **Wasserstein** where the p in L^p is set to 2.0, (5) **Betti** where the p in L^p is set to 2.0 and number of bins is set to 100, (6) **Landspace** where the p in L^p is set to 2.0, the number of bins is set to 100, and the number of to consider in the persistence landscape is set to three values: 1, 2, and 3, (7) **Persistence Image** where the p in L^p is set to 2.0, sigma value is set to 0.1, and the number of bins is set to 100, (8) **Heat** where the p in L^p is set to 2.0, sigma value is set to 0.1, and the number of bins is set to 100, and (9) **Silhouette** where the p in L^p is set to 2.0, the number of bins is set to 100, and the power is set to two values: 1.0 and 2.0.

The pipeline of extracting the features consists of three stages: (1) **Cubical Persistence** resulted from the filtered cubical complexes, (2) **Bottleneck scaler** as it makes the lifetime of the most persistent point across all diagrams and homology dimensions equal to two, and (3) a **filter** to discard the points that have a lifetime less than or equal to a cutoff value (set to 0.01 in the current study). From that, the number of extracted features per row is 26.

4.3.2 Dimensionality reduction

The current study utilized five features reduction techniques. They are PCA, LDA, ICA, RP, and T-SVD, as discussed in Sect. 3.2. The features are reduced to 3 for the “WISDM” dataset. The reason behind this is that the

original dataset consists of 3 columns. Also, the features are reduced to 100 for the “UCI-HAR” dataset.

4.4 ML classification and optimization phase

The current study uses different classification algorithms to achieve the state-of-the-art (SOTA) performance metrics. They are the (1) LGBM, (2) XGB, (3) AdaBoost, (4) HGB, (5) ETs, (6) DT, and (7) RF classifiers.

4.4.1 Hyperparameters optimization using grid search (GS)

There are different hyperparameters for each used machine algorithm and hence the GS optimization approach is implemented to find the best combinations for each classifier. Table 5 summarizes the different used hyperparameters of each classifier to select from. The used machine learning algorithms and their hyperparameters:

- **LGBM Classifier:** The boosting type is set to Gradient Boosting Decision Tree (GBDT). The max depth is the maximum tree depth for the base learners and is set to “None” which means that there are no limitations. The

Table 7 Summary of the performance evaluation metrics

Metric	Equation
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Balanced accuracy	$0.5 \times (\text{Recall} + \text{Specificity})$
Precision = PPV	$\frac{TP}{TP+FP}$
Specificity	$\frac{TN}{TN+FP}$
Recall = Sensitivity = Hit Rate = TPR	$\frac{TP}{TP+FN}$
Dice = F1 = Overlap Index	$\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
IoU = Jaccard Index	$\frac{TP}{TP+FP+FN}$
NPV	$\frac{TN}{TN+FN}$
ROC	$\frac{\sqrt{\text{Sensitivity}^2 + \text{Specificity}^2}}{\sqrt{2}}$

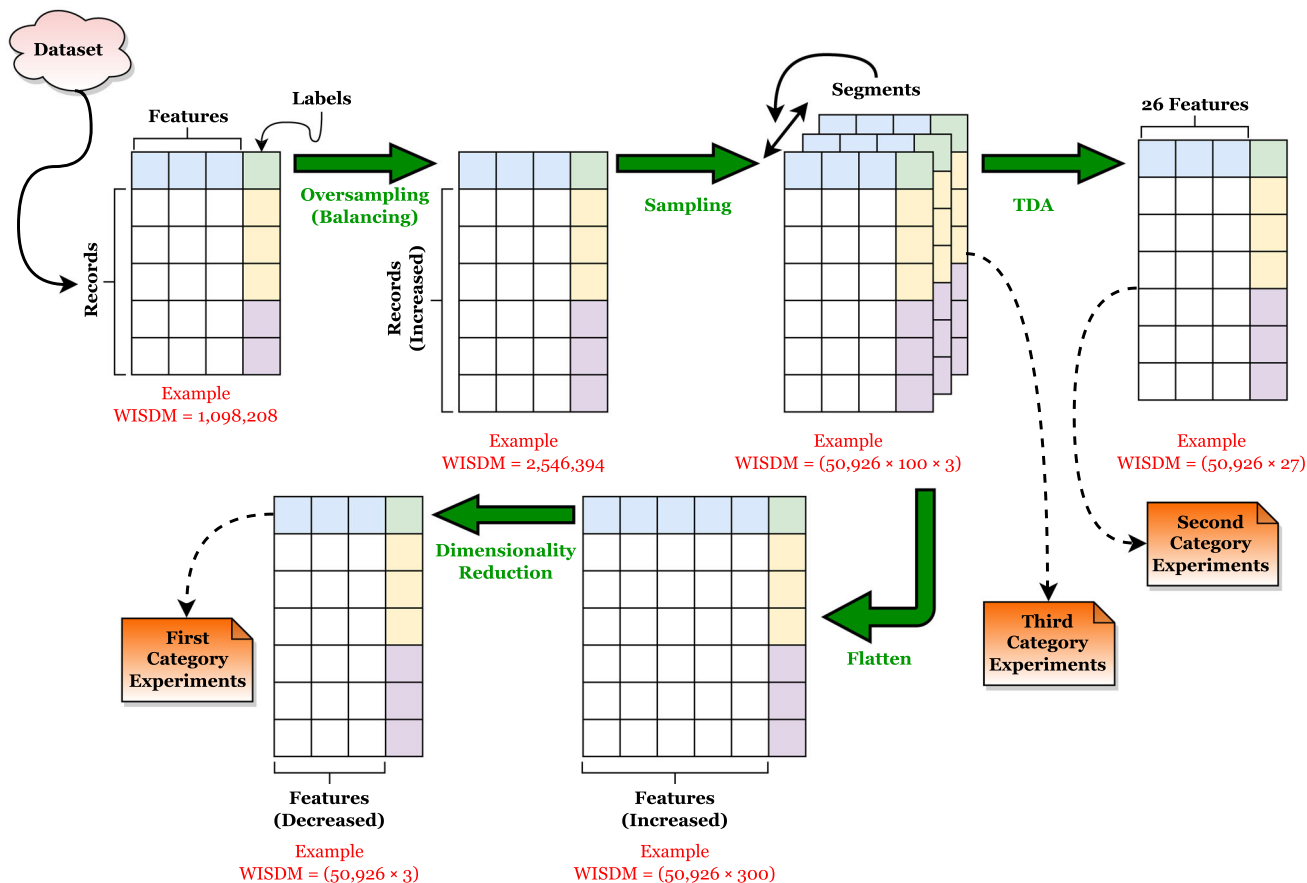


Fig. 5 The flow of the numerical data and the corresponding experiments categories numbers

Table 8 The experiments configurations

Configuration	Details
Datasets	Presented in Table 2
Datasets sampling criteria	Presented in Table 4
Features scaling techniques	Presented in Table 6
Features reduction techniques	PCA, LDA, ICA, RP, and T-SVD
Oversampling technique	SMOTE
ML classifiers	LGBM, XGB, AdaBoost, DT, ETs, RF, and HGB
Classifiers hyperparameters	Presented in Table 5
Hyperparameters optimization approach	Grid Search (GS)
K-fold cross-validation	fiveolds
DL classifiers	1D-CNN, BiLSTM, and GRU
Number of DL epochs	64
Performance metrics	Presented in Table 7
Scripting language	Python
Python packages	NumPy, Pandas, SciPy, Giotto-TDA, Scikit-Learn, and Matplotlib

learning rate is the boosting learning rate. The number of estimators is the number of boosted trees to fit and is set to 300.

- **XGB Classifier:** The boosting type is set to GBDT. The number of estimators is set to 300. The max depth is set to “None”.
- **AdaBoost Classifier:** The number of estimators is set to 300.
- **DT Classifier:** The combinations are applied between criteria to measure the quality of a split and the splitting mechanism. The max depth is set to “None.”
- **ETs Classifier:** The combinations are applied on the criterion. The max depth is set to “None.” The number of estimators is set to 300.
- **RF Classifier:** The combinations are applied on the criterion and class weight. The number of estimators is set to 300. The max depth is set to “None.”
- **HGB Classifier:** The max iteration is set to 100 and the learning rate is set to 0.1.

Table 9 The best combinations on the reduced “WISDM” dataset (50, 926 × 3) using each classifier

Feature reduction technique	Classifier	Scaling technique	Criterion	Splitter	Max depth	Estimators #	Class weight	Learning rate
PCA	DT	MinMaxScaler	Gini	Best	None	–	–	–
PCA	AdaBoost	MinMaxScaler	–	–	–	300	–	0.10
PCA	RFC	RobustScaler	Gini	–	None	300	Balanced	–
PCA	ETC	MaxAbsScaler	Entropy	–	None	300	–	–
PCA	HGB	MinMaxScaler	–	–	–	–	–	–
PCA	XGB	MinMaxScaler	–	–	None	300	–	0.10
PCA	LGBM	MinMaxScaler	–	–	None	300	–	0.10
LDA	DT	StandardScaler	Entropy	Best	None	–	–	–
LDA	AdaBoost	Normalizer	–	–	–	300	–	0.01
LDA	RFC	StandardScaler	Entropy	–	None	300	None	–
LDA	ETC	MinMaxScaler	Gini	–	None	300	–	–
LDA	HGB	StandardScaler	–	–	–	–	–	–
LDA	XGB	MinMaxScaler	–	–	None	300	–	0.10
LDA	LGBM	StandardScaler	–	–	None	300	–	0.10
ICA	DT	StandardScaler	Entropy	Best	None	–	–	–
ICA	AdaBoost	Normalizer	–	–	–	300	–	0.01
ICA	RFC	RobustScaler	Gini	–	None	300	Balanced Subsample	–
ICA	ETC	MaxAbsScaler	Gini	–	None	300	–	–
ICA	HGB	RobustScaler	–	–	–	–	–	–
ICA	XGB	RobustScaler	–	–	None	300	–	0.10
ICA	LGBM	RobustScaler	–	–	None	300	–	0.10
T-SVD	DT	MaxAbsScaler	Entropy	Best	None	–	–	–
T-SVD	AdaBoost	StandardScaler	–	–	–	300	–	0.10
T-SVD	RFC	MinMaxScaler	Entropy	–	None	300	Balanced	–
T-SVD	ETC	RobustScaler	Entropy	–	None	300	–	–
T-SVD	HGB	RobustScaler	–	–	–	–	–	–
T-SVD	XGB	StandardScaler	–	–	None	300	–	0.10
T-SVD	LGBM	RobustScaler	–	–	None	300	–	0.10
RP	DT	StandardScaler	Entropy	Best	None	–	–	–
RP	AdaBoost	StandardScaler	–	–	–	300	–	0.10
RP	RFC	RobustScaler	Entropy	–	None	300	Balanced	–
RP	ETC	MinMaxScaler	Gini	–	None	300	–	–
RP	HGB	RobustScaler	–	–	–	–	–	–
RP	XGB	MinMaxScaler	–	–	None	300	–	0.01
RP	LGBM	RobustScaler	–	–	None	300	–	0.01

4.4.2 Features scaling

Standardization, normalization, min-max scaling, max-absolute scaling, and robust scaling are used with the grid search to find the best scaler technique. Table 6 summarizes the used feature scaling equations.

where X_{scaled} is the scaled output vector while X_{input} is the input vector, IQR is the interquartile range, μ is the mean value, and σ is the standard deviation value.

4.4.3 Performance improvement

The train-to-test splitting and K-fold cross-validation are used to improve the estimated performance of the

Table 10 The corresponding performance metrics using each classifier applied on the reduced “WISDM” dataset (50,926 × 3)

Feature Reduction Technique	Classifier	Elapsed ttime (s)	Accuracy (%)	Balanced accuracy (%)	Precision (%)	Specificity (%)	Recall (%)	F1 (%)	IoU (%)	ROC (%)	NPV (%)
PCA	DT	20,795	91.97	85.54	75.91	95.18	75.91	75.91	61.17	86.08	95.18
PCA	AdaBoost	716,806	88.09	78.56	64.27	92.85	64.27	64.27	47.35	79.85	92.85
PCA	RFC	3832763	93.81	88.85	81.42	96.28	81.42	81.42	68.67	89.16	96.28
PCA	ETC	451,381	93.75	88.75	81.26	96.25	81.26	81.26	68.43	89.07	96.25
PCA	HGB	178,572	93.87	88.96	81.60	96.32	81.60	81.60	68.91	89.26	96.32
PCA	XGB	4,930,503	93.83	88.90	81.50	96.30	81.50	81.50	68.78	89.21	96.30
PCA	LGBM	268,647	93.83	88.90	81.50	96.30	81.50	81.50	68.77	89.21	96.30
LDA	DT	21,295	89.97	81.95	69.91	93.98	69.91	69.91	53.74	82.83	93.98
LDA	AdaBoost	723,000	86.70	76.05	60.09	92.02	60.09	60.09	42.95	77.71	92.02
LDA	RFC	4,087,621	92.21	85.97	76.62	95.32	76.62	76.62	62.10	86.48	95.32
LDA	ETC	480,904	92.14	85.85	76.41	95.28	76.41	76.41	61.82	86.36	95.28
LDA	HGB	163,050	92.44	86.39	77.32	95.46	77.32	77.32	63.03	86.87	95.46
LDA	XGB	4,960,229	92.42	86.35	77.26	95.45	77.26	77.26	62.94	86.83	95.45
LDA	LGBM	274,763	92.33	86.20	76.99	95.40	76.99	76.99	62.59	86.69	95.40
ICA	DT	20,186	91.92	85.46	75.77	95.15	75.77	75.77	61.00	86.01	95.15
ICA	AdaBoost	735,040	89.57	81.23	68.72	93.74	68.72	68.72	52.35	82.19	93.74
ICA	RFC	3,920,886	93.82	88.88	81.46	96.29	81.46	81.46	68.72	89.19	96.29
ICA	ETC	451,598	93.80	88.84	81.40	96.28	81.40	81.40	68.64	89.15	96.28
ICA	HGB	183,653	93.88	88.99	81.64	96.33	81.64	81.64	68.98	89.29	96.33
ICA	XGB	4,933,501	93.87	88.97	81.61	96.32	81.61	81.61	68.94	89.27	96.32
ICA	LGBM	267,109	93.86	88.95	81.59	96.32	81.59	81.59	68.91	89.26	96.32
T-SVD	DT	28,717	79.06	62.31	37.19	87.44	37.19	37.19	22.84	67.19	87.44
T-SVD	AdaBoost	744089	81.82	67.28	45.47	89.09	45.47	45.47	29.42	70.73	89.09
T-SVD	RFC	5,391,457	81.88	67.39	45.65	89.13	45.65	45.65	29.58	70.81	89.13
T-SVD	ETC	694,845	81.82	67.27	45.46	89.09	45.46	45.46	29.41	70.72	89.09
T-SVD	HGB	96,698	82.65	68.78	47.96	89.59	47.96	47.96	31.54	71.86	89.59
T-SVD	XGB	5,195,648	82.70	68.85	48.09	89.62	48.09	48.09	31.65	71.92	89.62
T-SVD	LGBM	309,397	82.76	68.97	48.29	89.66	48.29	48.29	31.83	72.01	89.66
RP	DT	20,546	91.99	85.58	75.96	95.19	75.96	75.96	61.24	86.12	95.19
RP	AdaBoost	739,715	89.01	80.22	67.04	93.41	67.04	67.04	50.42	81.30	93.41
RP	RFC	3,888,200	93.84	88.91	81.52	96.30	81.52	81.52	68.81	89.22	96.30
RP	ETC	458,880	93.80	88.84	81.40	96.28	81.40	81.40	68.63	89.15	96.28
RP	HGB	187,493	93.87	88.97	81.61	96.32	81.61	81.61	68.94	89.27	96.32
RP	XGB	4,899,129	93.92	89.06	81.76	96.35	81.76	81.76	69.15	89.36	96.35
RP	LGBM	276611.3	93.91	89.04	81.74	96.35	81.74	81.74	69.12	89.34	96.35

classifiers. The train-to-test splitting partitions the datasets into train and test subsets. The current study uses 85% to 15% for the train and test subsets, respectively, after shuffling them. The current study uses fivefold (i.e., $K = 5$).

4.5 DL classification phase

The current study utilized four DL approaches as discussed in Sect. 3.4.2. The first is the GRU that consists of nine

layers: (1) input layer, (2) three cascaded GRU layers with 32 kernels, 20% dropout, and 20% recurrent dropout, (3) dense layer with 64 units and LeakyReLU activation function, (4) dropout layer with a dropout ratio of 50%, (5) another dense layer with 32 units and LeakyReLU activation function, (6) another dropout layer with a dropout ratio of 50%, (7) output layer with a SoftMax activation function.

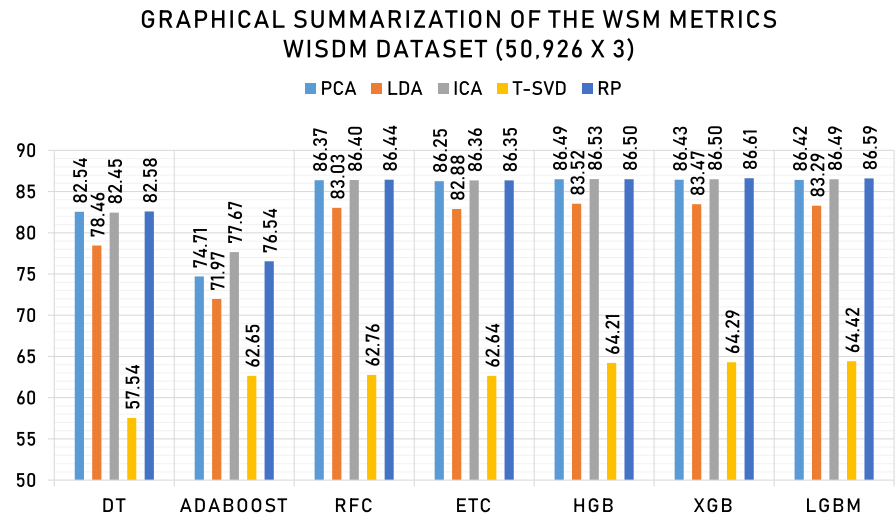
The second is the 1D-CNN that consists of 14 layers: (1) input layer, (2) four cascaded 1D convolutional layers with

Table 11 Tabular summarization of the WSM metrics using the “WISDM” dataset (50,926 × 3)

	AdaBoost (%)	DT (%)	ETC (%)	HGB (%)	LGBM (%)	RFC (%)	XGB (%)	Max value (%)
PCA	82.54	74.71	86.37	86.25	86.49	86.43	86.42	86.49
LDA	78.46	71.97	83.03	82.88	83.52	83.47	83.29	83.52
ICA	82.45	77.67	86.40	86.36	86.53	86.50	86.49	86.53
T-SVD	57.54	62.65	62.76	62.64	64.21	64.29	64.42	64.42
RP	82.58	76.54	86.44	86.35	86.50	86.61	86.59	86.61
Max Value	82.58	77.67	86.44	86.36	86.53	86.61	86.59	86.61

All the bold refers to the highest WSM score acquired

Fig. 6 Graphical summarization of the WSM metrics using the “WISDM” dataset (50,926 × 3)



32, 64, 128, and 256 filters (i.e., kernels), respectively, kernel size of 3, same padding, and LeakyReLU activation function, (3) 1D max-pooling layer with pooling size of 3, a stride of 2, and same padding, (4) dropout layer with a dropout ratio of 50%, (5) dense layer with 256 units and LeakyReLU activation function, (6) dropout layer with a dropout ratio of 50%, (7) flatten layer, (8) another dense layer with 512 units and LeakyReLU activation function, (9) another dropout layer with a dropout ratio of 50%, (10) a third dense layer with 1024 units and LeakyReLU activation function, and (11) output layer with a SoftMax activation function.

The third is the BiLSTM that consists of 3 layers: (1) input layer, (2) bi-directional LSTM with 4 units and LeakyReLU activation function, and (3) output layer with a SoftMax activation function. All of the DL models used the Adam parameters optimizer, Categorical Crossentropy loss function, and 64 epochs. The models’ architecture came out after a set of try-and-error trials as there are no specific rules to define the architectures because of the dataset dependency.

4.6 Performance evaluation

The confusion matrix is reported with its TP (i.e., True Positive), TN (i.e., True Negative), FP (i.e., False Positive), FN (i.e., False Negative) values. Eqs. 1, 2, 3, and 4 show how to calculate the values of TP, FP, FN, and TN, respectively, for multi-class problems.

$$TP_i = C_{(i,i)} \tag{1}$$

$$FP_i = \sum_{l=1}^n C_{(l,i)} - TP_i \tag{2}$$

$$FN_i = \sum_{l=1}^n C_{(i,l)} - TP_i \tag{3}$$

$$TN_i = \sum_{l=1}^n \sum_{k=1}^n C_{(l,k)} - TP_i - FP_i - FN_i \tag{4}$$

where C is the confusion matrix, n is the number of classes, and i is the class number.

Different performance metrics are calculated from them. They are (1) Accuracy, (2) Balanced Accuracy, (3) Precision (i.e., PPV), (4) Recall (i.e., Sensitivity, Hit Rate, and

Table 12 The best combinations applied on the reduced “WISDM” dataset (25,463 × 3) using each classifier

Feature Reduction Technique	Classifier	Scaling Technique	Criterion	Splitter	Max Depth	Estimators #	Class Weight	Learning Rate
PCA	DT	RobustScaler	Entropy	Best	None	–	–	–
PCA	AdaBoost	StandardScaler	–	–	–	300	–	0.10
PCA	RFC	MinMaxScaler	Entropy	–	None	300	Balanced	–
PCA	ETC	MinMaxScaler	Entropy	–	None	300	–	–
PCA	HGB	MaxAbsScaler	–	–	–	–	–	–
PCA	XGB	StandardScaler	–	–	None	300	–	0.10
PCA	LGBM	StandardScaler	–	–	None	300	–	0.10
LDA	DT	MaxAbsScaler	Entropy	Best	None	–	–	–
LDA	AdaBoost	Normalizer	–	–	–	300	–	0.01
LDA	RFC	StandardScaler	Entropy	–	None	300	None	–
LDA	ETC	MinMaxScaler	Gini	–	None	300	–	–
LDA	HGB	RobustScaler	–	–	–	–	–	–
LDA	XGB	StandardScaler	–	–	None	300	–	0.10
LDA	LGBM	MaxAbsScaler	–	–	None	300	–	0.01
ICA	DT	RobustScaler	Gini	Best	None	–	–	–
ICA	AdaBoost	Normalizer	–	–	–	300	–	0.01
ICA	RFC	RobustScaler	Gini	–	None	300	None	–
ICA	ETC	RobustScaler	Gini	–	None	300	–	–
ICA	HGB	MinMaxScaler	–	–	–	–	–	–
ICA	XGB	MinMaxScaler	–	–	None	300	–	0.10
ICA	LGBM	MaxAbsScaler	–	–	None	300	–	0.10
T-SVD	DT	RobustScaler	Entropy	Best	None	–	–	–
T-SVD	AdaBoost	StandardScaler	–	–	–	300	–	0.10
T-SVD	RFC	MaxAbsScaler	Gini	–	None	300	Balanced Subsample	–
T-SVD	ETC	MaxAbsScaler	Entropy	–	None	300	–	–
T-SVD	HGB	StandardScaler	–	–	–	–	–	–
T-SVD	XGB	StandardScaler	–	–	None	300	–	0.10
T-SVD	LGBM	MaxAbsScaler	–	–	None	300	–	0.10
RP	DT	MaxAbsScaler	Gini	Best	None	–	–	–
RP	AdaBoost	StandardScaler	–	–	–	300	–	0.10
RP	RFC	MaxAbsScaler	Entropy	–	None	300	Balanced	–
RP	ETC	StandardScaler	Entropy	–	None	300	–	–
RP	HGB	StandardScaler	–	–	–	–	–	–
RP	XGB	MaxAbsScaler	–	–	None	300	–	0.01
RP	LGBM	RobustScaler	–	–	None	300	–	0.01

TPR), (5) Specificity (i.e., TNR), (6) F1-score (i.e., Dice coef. and Overlap Index), (7) IoU (i.e., Jaccard Index), (8) NPV (Negative Predictive Value), and (10) ROC (i.e., Receiver Operating Characteristic). The corresponding equations for them are presented in Table 7.

To engage all of the calculated metrics together, the weighted sum metric is calculated as shown in Eq. 5.

$$\text{WSM} = \frac{1}{9} \times (\text{Accuracy} + \text{Balanced Accuracy} + \text{Precision} + \text{Specificity} + \text{Recall} + \text{F1} + \text{IoU} + \text{ROC} + \text{NPV}) \quad (5)$$

4.6.1 Multi-class averaging

During the process of evaluating the performance of multi-class dataset implementations, it is preferred to use an

Table 13 The corresponding performance metrics using each classifier applied on the reduced “WISDM” dataset (25,463 × 3)

Feature Reduction Technique	Classifier	Elapsed time (s)	Accuracy (%)	Balanced accuracy (%)	Precision (%)	Specificity (%)	Recall (%)	F1 (%)	IoU (%)	ROC (%)	NPV (%)
PCA	DT	9.8	91.57	84.83	74.71	94.94	74.71	74.71	59.63	85.43	94.94
PCA	AdaBoost	363.8	89.36	80.86	68.09	93.62	68.09	68.09	51.62	81.86	93.62
PCA	RFC	1,809	93.49	88.28	80.46	96.09	80.46	80.46	67.31	88.62	96.09
PCA	ETC	227.3	93.45	88.20	80.34	96.07	80.34	80.34	67.14	88.55	96.07
PCA	HGB	107.6	93.53	88.35	80.58	96.12	80.58	80.58	67.48	88.69	96.12
PCA	XGB	2,440	93.52	88.33	80.55	96.11	80.55	80.55	67.43	88.67	96.11
PCA	LGBM	176.5	93.40	88.13	80.21	96.04	80.21	80.21	66.97	88.48	96.04
LDA	DT	10.8	88.88	79.99	66.65	93.33	66.65	66.65	49.98	81.09	93.33
LDA	AdaBoost	384.5	84.72	72.50	54.17	90.83	54.17	54.17	37.15	74.79	90.83
LDA	RFC	1,922	91.26	84.27	73.78	94.76	73.78	73.78	58.45	84.92	94.76
LDA	ETC	247.7	91.21	84.17	73.62	94.72	73.62	73.62	58.26	84.83	94.72
LDA	HGB	91.5	91.62	84.91	74.85	94.97	74.85	74.85	59.80	85.50	94.97
LDA	XGB	2,490	91.53	84.75	74.59	94.92	74.59	74.59	59.47	85.36	94.92
LDA	LGBM	197.8	91.58	84.85	74.75	94.95	74.75	74.75	59.68	85.45	94.95
ICA	DT	9.4	91.51	84.71	74.52	94.90	74.52	74.52	59.38	85.32	94.90
ICA	AdaBoost	363.2	85.06	73.10	55.17	91.03	55.17	55.17	38.10	75.27	91.03
ICA	RFC	1,775	93.43	88.17	80.29	96.06	80.29	80.29	67.07	88.53	96.06
ICA	ETC	225.7	93.44	88.19	80.32	96.06	80.32	80.32	67.12	88.55	96.06
ICA	HGB	105.8	93.51	88.31	80.52	96.10	80.52	80.52	67.39	88.66	96.10
ICA	XGB	2,452	93.49	88.28	80.47	96.09	80.47	80.47	67.32	88.63	96.09
ICA	LGBM	192.1	93.41	88.14	80.23	96.05	80.23	80.23	66.99	88.49	96.05
T-SVD	DT	9.6	91.60	84.88	74.80	94.96	74.80	74.80	59.75	85.48	94.96
T-SVD	AdaBoost	358.9	88.95	80.11	66.85	93.37	66.85	66.85	50.20	81.20	93.37
T-SVD	RFC	1,796	93.48	88.26	80.43	96.09	80.43	80.43	67.26	88.60	96.09
T-SVD	ETC	228.3	93.45	88.22	80.36	96.07	80.36	80.36	67.17	88.57	96.07
T-SVD	HGB	114.1	93.55	88.39	80.65	96.13	80.65	80.65	67.57	88.73	96.13
T-SVD	XGB	2,415	93.53	88.35	80.58	96.12	80.58	80.58	67.48	88.69	96.12
T-SVD	LGBM	187.6	93.41	88.14	80.23	96.05	80.23	80.23	66.99	88.49	96.05
RP	DT	13.1	79.13	62.44	37.40	87.48	37.40	37.40	23.00	67.28	87.48
RP	AdaBoost	372.9	81.65	66.98	44.96	88.99	44.96	44.96	29.00	70.50	88.99
RP	RFC	2,484	81.79	67.22	45.36	89.07	45.36	45.36	29.34	70.68	89.07
RP	ETC	343.6	81.77	67.19	45.31	89.06	45.31	45.31	29.29	70.66	89.06
RP	HGB	70.5	82.67	68.81	48.02	89.60	48.02	48.02	31.60	71.88	89.60
RP	XGB	2,591	82.71	68.87	48.12	89.62	48.12	48.12	31.69	71.93	89.62
RP	LGBM	207.5	82.78	69.00	48.34	89.67	48.34	48.34	31.87	72.03	89.67

averaging method. It is used to average the scores to acquire a single number describing the overall performance instead of having multiple scores per class. This includes micro-, macro-, and macro-weighted averaging methods. They are discussed as follows:

- **Micro Averaging.** For each class, the metric will be computed independently and the average will be taken hence all classes are treated equally. In the Micro-average method, the true positives, false positives, and

false negatives are summed individually for different sets and apply them to get the statistics. For a balanced dataset, micro averaging is preferred when an understandable metric for overall performance regardless of the class is required. As the more the number of samples, the more impact the corresponding class has on the final score, thus favoring majority classes.

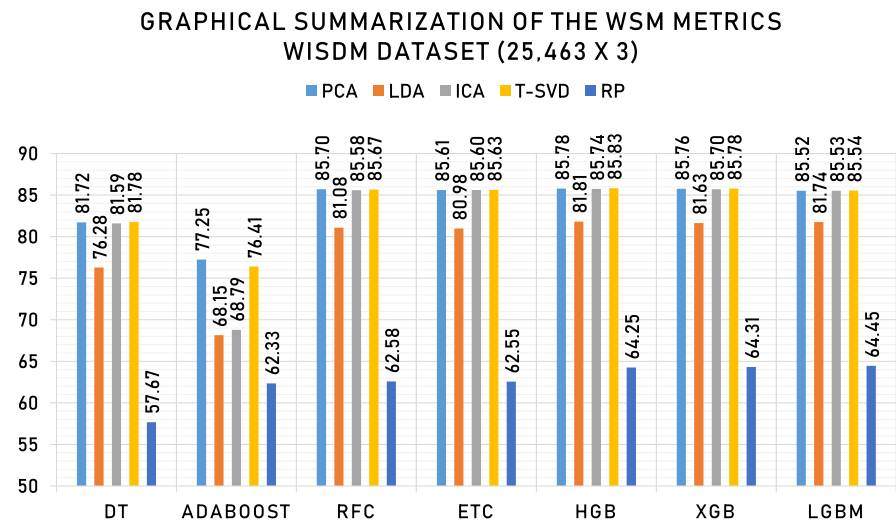
- **Macro Averaging** is straightforward as it is aggregating the contributions of all classes to calculate the average metric. Hence, the statistics of the smaller

Table 14 Tabular summarization of the WSM metrics using the “WISDM” dataset (25,463 × 3)

	AdaBoost (%)	DT (%)	ETC (%)	HGB (%)	LGBM (%)	RFC (%)	XGB (%)	Max value (%)
PCA	81.72	77.25	85.70	85.61	85.78	85.76	85.52	85.78
LDA	76.28	68.15	81.08	80.98	81.81	81.63	81.74	81.81
ICA	81.59	68.79	85.58	85.60	85.74	85.70	85.53	85.74
T-SVD	81.78	76.41	85.67	85.63	85.83	85.78	85.54	85.83
RP	57.67	62.33	62.58	62.55	64.25	64.31	64.45	64.45
Max value	81.78	77.25	85.70	85.63	85.83	85.78	85.54	85.83

All the bold refers to the highest WSM score acquired

Fig. 7 Graphical summarization of the WSM metrics using the “WISDM” dataset (25,463 × 3)



classes are reflected. It is appropriate when the performance of all classes is important equally.

- **Macro-Weighted Averaging** is calculated by weighting the score of each class label by the number of true instances. It is applied in the situation of an imbalanced dataset when assigning greater contributions to the majority classes. It is worth mentioning that using this type of averaging with balanced data will yield the same result as macro-averaging.

In the current study, since the datasets used in the evaluating phase are balanced, the micro-averaging method is utilized.

5 Experiments and discussion

Figure 5 summarizes the flow of the numerical data and the corresponding experiments categories numbers. This will facilitate for the reader to trace the experiments categories.

5.1 Experiments configurations, constrains, and assumptions

Table 8 summarizes the experiments configurations.

In the current study, the constraints and assumptions applied during the sampling and feature reduction processes are: (1) the target is to choose a number of features equal to or less than the initial number of entry features for each dataset, (2) constructing features that aware of the time-series data, and (3) the time-complexity is taken into account. From that, the current study fixed the number of the output features to be 3 for the WISDM and 100 for the UCI-HAR datasets. In the reported results tables, the “None” in the “Max Depth” column means that there is no limitation in the max tree depth, the “None” in the “Class weight” column means that there are no weights defined to the classes.

5.2 First category experiments

The current Section presents the experiments applied on both datasets after the dimensionality reduction step as

Table 15 The best combinations on the reduced “UCI-HAR” dataset (232 × 100) using each classifier

Feature Reduction Technique	Classifier	Scaling Technique	Criterion	Splitter	Max Depth	Estimators #	Class Weight	Learning Rate
PCA	DT	RobustScaler	Entropy	Best	None	–	–	–
PCA	AdaBoost	Normalizer	–	–	–	300	–	0.10
PCA	RFC	Normalizer	Gini	–	None	300	Balanced	–
PCA	ETC	MaxAbsScaler	Gini	–	None	300	–	–
PCA	HGB	Normalizer	–	–	–	–	–	–
PCA	XGB	Normalizer	–	–	None	300	–	0.10
PCA	LGBM	StandardScaler	–	–	None	300	–	0.10
LDA	DT	Normalizer	Gini	Best	None	–	–	–
LDA	AdaBoost	Normalizer	–	–	–	300	–	0.10
LDA	RFC	Normalizer	Gini	–	None	300	Balanced	–
LDA	ETC	Normalizer	Gini	–	None	300	–	–
LDA	HGB	StandardScaler	–	–	–	–	–	–
LDA	XGB	Normalizer	–	–	None	300	–	0.10
LDA	LGBM	Normalizer	–	–	None	300	–	–
ICA	DT	StandardScaler	Gini	Best	None	–	–	–
ICA	AdaBoost	StandardScaler	–	–	–	300	–	0.10
ICA	RFC	Normalizer	Gini	–	None	300	None	–
ICA	ETC	StandardScaler	Gini	–	None	300	–	–
ICA	HGB	Normalizer	–	–	–	–	–	–
ICA	XGB	Normalizer	–	–	None	300	–	0.10
ICA	LGBM	Normalizer	–	–	None	300	–	0.10
T-SVD	DT	Normalizer	Entropy	Best	None	–	–	–
T-SVD	AdaBoost	StandardScaler	–	–	–	300	–	0.10
T-SVD	RFC	Normalizer	Gini	–	None	300	–	–
T-SVD	ETC	RobustScaler	Entropy	–	None	300	–	–
T-SVD	HGB	StandardScaler	–	–	–	–	–	–
T-SVD	XGB	Normalizer	–	–	None	300	–	0.10
T-SVD	LGBM	MinMaxScaler	–	–	None	300	–	0.10
RP	DT	StandardScaler	Entropy	Best	None	–	–	–
RP	AdaBoost	Normalizer	–	–	–	300	–	0.01
RP	RFC	Normalizer	Gini	–	None	300	None	–
RP	ETC	MaxAbsScaler	Gini	–	None	300	–	–
RP	HGB	StandardScaler	–	–	–	–	–	–
RP	XGB	StandardScaler	–	–	None	300	–	0.10
RP	LGBM	MinMaxScaler	–	–	None	300	–	0.10

presented in Fig. 5. The “WISDM” dataset is sampled and five features reduction techniques are applied to reduce the number of features to 3 as described earlier. Table 9 shows the best combinations and Table 10 shows the corresponding performance metrics applied on the reduced “WISDM” data when applying 50% overlapping (*i.e.*, 50,926 × 3) using the different mentioned classifiers. It shows the best of each classifier after the grid searching process.

Table 10 reports that the best-reported performance metrics. It shows that the best-reported accuracy, balanced accuracy, precision, specificity, recall, F1-score, IoU,

ROC, and NPV are 93.92%, 89.06%, 81.76%, 96.35%, 81.76%, 81.76%, 69.15%, 89.36%, and 96.35%, respectively, by the XGB classifier and the RP feature reduction technique. In terms of the elapsed time, the best-reported classifier is DT with ICA as a feature reduction technique with 20,186 seconds. Table 11 and Fig. 6 summarize the WSM metrics using the “WISDM” dataset when applying 50% overlapping. It shows that the highest WSM value is 86.61% that is produced by the RFC classifier with the RP feature reduction method.

Table 16 The corresponding performance metrics using each classifier applied on the reduced “UCI-HAR” dataset (232 × 100)

Feature Reduction Technique	Classifier	Elapsed time (s)	Accuracy (%)	Balanced accuracy (%)	Precision (%)	Specificity (%)	Recall (%)	F1 (%)	IoU (%)	ROC (%)	NPV (%)
PCA	DT	1.4	99.28	98.71	97.84	99.57	97.84	97.84	95.78	98.71	99.57
PCA	AdaBoost	73.6	99.14	98.45	97.41	99.48	97.41	97.41	94.96	98.45	99.48
PCA	RFC	107.2	99.14	98.45	97.41	99.48	97.41	97.41	94.96	98.45	99.48
PCA	ETC	18.6	99.28	98.71	97.84	99.57	97.84	97.84	95.78	98.71	99.57
PCA	HGB	55.6	99.43	98.97	98.28	99.66	98.28	98.28	96.61	98.97	99.66
PCA	XGB	96.3	99.14	98.45	97.41	99.48	97.41	97.41	94.96	98.45	99.48
PCA	LGBM	38.1	99.28	98.71	97.84	99.57	97.84	97.84	95.78	98.71	99.57
LDA	DT	0.6	97.70	95.86	93.10	98.62	93.10	93.10	87.10	95.90	98.62
LDA	AdaBoost	34.8	86.49	75.69	59.48	91.90	59.48	59.48	42.33	77.41	91.90
LDA	RFC	71.9	97.99	96.38	93.97	98.79	93.97	93.97	88.62	96.41	98.79
LDA	ETC	15.4	97.99	96.38	93.97	98.79	93.97	93.97	88.62	96.41	98.79
LDA	HGB	27.2	96.98	94.57	90.95	98.19	90.95	90.95	83.40	94.64	98.19
LDA	XGB	57.6	97.56	95.60	92.67	98.53	92.67	92.67	86.35	95.65	98.53
LDA	LGBM	18.8	97.70	95.86	93.10	98.62	93.10	93.10	87.10	95.90	98.62
ICA	DT	2.1	78.88	61.98	36.64	87.33	36.64	36.64	22.43	66.96	87.33
ICA	AdaBoost	73.3	81.32	66.38	43.97	88.79	43.97	43.97	28.18	70.06	88.79
ICA	RFC	121.0	88.22	78.79	64.66	92.93	64.66	64.66	47.77	80.05	92.93
ICA	ETC	20.2	90.66	83.19	71.98	94.40	71.98	71.98	56.23	83.94	94.40
ICA	HGB	77.8	86.78	76.21	60.34	92.07	60.34	60.34	43.21	77.84	92.07
ICA	XGB	169.9	86.93	76.47	60.78	92.16	60.78	60.78	43.65	78.06	92.16
ICA	LGBM	61.1	87.79	78.02	63.36	92.67	63.36	63.36	46.37	79.38	92.67
T-SVD	DT	1.5	99.28	98.71	97.84	99.57	97.84	97.84	95.78	98.71	99.57
T-SVD	AdaBoost	76.7	97.70	95.86	93.10	98.62	93.10	93.10	87.10	95.90	98.62
T-SVD	RFC	107.9	99.14	98.45	97.41	99.48	97.41	97.41	94.96	98.45	99.48
T-SVD	ETC	18.3	99.28	98.71	97.84	99.57	97.84	97.84	95.78	98.71	99.57
T-SVD	HGB	55.3	98.56	97.41	95.69	99.14	95.69	95.69	91.74	97.43	99.14
T-SVD	XGB	97.0	98.56	97.41	95.69	99.14	95.69	95.69	91.74	97.43	99.14
T-SVD	LGBM	37.6	99.28	98.71	97.84	99.57	97.84	97.84	95.78	98.71	99.57
RP	DT	1.6	91.09	83.97	73.28	94.66	73.28	73.28	57.82	84.64	94.66
RP	AdaBoost	73.5	82.47	68.45	47.41	89.48	47.41	47.41	31.07	71.61	89.48
RP	RFC	100.4	97.99	96.38	93.97	98.79	93.97	93.97	88.62	96.41	98.79
RP	ETC	18.6	98.42	97.16	95.26	99.05	95.26	95.26	90.95	97.17	99.05
RP	HGB	70.4	96.41	93.53	89.22	97.84	89.22	89.22	80.54	93.63	97.84
RP	XGB	115.2	95.26	91.47	85.78	97.16	85.78	85.78	75.09	91.64	97.16
RP	LGBM	35.6	96.84	94.31	90.52	98.10	90.52	90.52	82.68	94.39	98.10

Table 12 shows the best combinations and Table 13 shows the corresponding performance metrics applied on the reduced “WISDM” data when applying 0% overlapping (*i.e.*, 25,463 × 3) using the different mentioned classifiers. It shows the best of each classifier after the grid searching process.

Table 13 reports that the best reported performance metrics. It shows that the best reported accuracy, balanced accuracy, precision, specificity, recall, F1-score, IoU, ROC, and NPV are 93.53%, 88.39%, 80.65%, 96.13%,

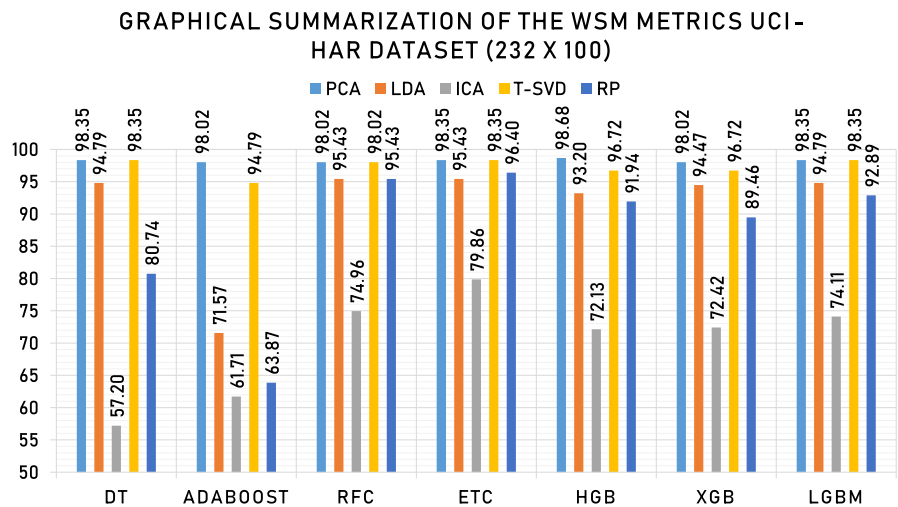
80.65%, 80.65%, 67.57%, 88.73%, and 96.13%, respectively, by the HGB classifier and T-SVD feature reduction technique. In terms of the elapsed time, the best reported classifier is DT and ICA as feature reduction technique with 9.4 seconds. Table 14 and Fig. 7 summarize the WSM metrics using the “WISDM” dataset when applying 0% overlapping. It shows that the highest WSM value is 85.83% that is produced by the LGBM classifier with the T-SVD feature reduction technique.

Table 17 Tabular summarization of the WSM metrics using the “UCI-HAR” dataset (232 × 100)

	DT	AdaBoost (%)	RFC (%)	ETC (%)	HGB (%)	XGB (%)	LGBM (%)	Max value (%)
PCA	98.35	98.02	98.02	98.35	98.68	98.02	98.35	98.68
LDA	94.79	71.57	95.43	95.43	93.20	94.47	94.79	95.43
ICA	57.20	61.71	74.96	79.86	72.13	72.42	74.11	79.86
T-SVD	98.35	94.79	98.02	98.35	96.72	96.72	98.35	98.35
RP	80.74	63.87	95.43	96.40	91.94	89.46	92.89	96.40
Max value	98.35	98.02	98.02	98.35	98.68	98.02	98.35	98.68

All the bold refers to the highest WSM score acquired

Fig. 8 Graphical summarization of the WSM metrics using the “UCI-HAR” dataset (232 × 100)



The “UCI-HAR” dataset is sampled and five features reduction techniques are applied to reduce the number of features to 100 as described earlier. Table 15 shows the best combinations and Table 16 shows the corresponding performance metrics applied on the reduced “UCI-HAR” data when applying 50% overlapping (*i.e.*, 232 × 100) using the different mentioned classifiers. It shows the best of each classifier after the grid searching process.

Table 16 reports that the best reported performance metrics. It shows that the best reported accuracy, balanced accuracy, precision, specificity, recall, F1-score, IoU, ROC, and NPV are 99.43%, 98.97%, 98.28%, 99.66%, 98.28%, 98.28%, 96.61%, 98.97%, and 99.66%, respectively, by HGB classifier with the PCA feature reduction technique. In terms of the elapsed time, the best reported classifier is DT with LDA as feature reduction technique with 0.6 seconds. Table 17 and Fig. 8 summarize the WSM metrics using the “UCI-HAR” dataset when applying 50% overlapping. It shows that the highest WSM value is 98.68% that is produced by the HGB with the PCA feature reduction technique.

Table 18 shows the best combinations and Table 19 shows the corresponding performance metrics applied on

the reduced “UCI-HAR” data when applying 0% overlapping (*i.e.*, 116 × 100) using the different mentioned classifiers. It shows the best of each classifier after the grid searching process.

Table 19 reports that the best reported performance metrics. It shows that the best reported accuracy, balanced accuracy, precision, specificity, recall, F1-score, IoU, ROC, and NPV are 100% by the LGBM classifier with the T-SVD feature reduction technique. In terms of the elapsed time, the best reported classifier is DT with LDA as feature reduction technique with 0.5 seconds. Table 20 and Fig. 9 summarize the WSM metrics using the “UCI-HAR” dataset when applying 0% overlapping. It shows that the highest WSM value is 100% that is produced by the LGBM classifier with the T-SVD feature reduction technique.

5.2.1 First category experiments remarks

Does the overlapping appliance during the dataset sampling process affect the performance?

According to Tables 11 and 14, applying 50% overlapping increases the best-reported WSM by 0.78%; however, the increase is not

Table 18 The best combinations applied on the reduced “UCI-HAR” dataset (116×100) using each classifier

Feature reduction technique	Classifier	Scaling technique	Criterion	Splitter	Max Depth	Estimators #	Class weight	Learning rate
PCA	DT	Normalizer	Entropy	Best	None	–	–	–
PCA	AdaBoost	Normalizer	–	–	–	300	–	0.01
PCA	RFC	RobustScaler	Gini	–	None	300	Balanced	–
PCA	ETC	StandardScaler	Gini	–	None	300	–	–
PCA	HGB	StandardScaler	–	–	–	–	–	–
PCA	XGB	Normalizer	–	–	None	300	–	0.10
PCA	LGBM	Normalizer	–	–	None	300	–	0.10
LDA	DT	MaxAbsScaler	Gini	Best	None	–	–	–
LDA	AdaBoost	StandardScaler	–	–	–	300	–	0.10
LDA	RFC	Normalizer	Entropy	–	None	300	Balanced	–
LDA	ETC	MinMaxScaler	Gini	–	None	300	–	–
LDA	HGB	Normalizer	–	–	–	–	–	–
LDA	XGB	Normalizer	–	–	None	300	–	0.01
LDA	LGBM	MinMaxScaler	–	–	None	300	–	0.01
ICA	DT	MinMaxScaler	Gini	Random	None	–	–	–
ICA	AdaBoost	RobustScaler	–	–	–	300	–	1.00
ICA	RFC	MinMaxScaler	Gini	–	None	300	None	–
ICA	ETC	StandardScaler	Gini	–	None	300	–	–
ICA	HGB	Normalizer	–	–	–	–	–	–
ICA	XGB	StandardScaler	–	–	None	300	–	1.00
ICA	LGBM	StandardScaler	–	–	None	300	–	1.00
T-SVD	DT	MinMaxScaler	Entropy	Best	None	–	–	–
T-SVD	AdaBoost	MinMaxScaler	–	–	–	300	–	0.01
T-SVD	RFC	StandardScaler	Entropy	–	None	300	None	–
T-SVD	ETC	MaxAbsScaler	Gini	–	None	300	–	–
T-SVD	HGB	StandardScaler	–	–	–	–	–	–
T-SVD	XGB	StandardScaler	–	–	None	300	–	0.01
T-SVD	LGBM	MinMaxScaler	–	–	None	300	–	1.00
RP	DT	RobustScaler	Entropy	Random	None	–	–	–
RP	AdaBoost	StandardScaler	–	–	–	300	–	0.01
RP	RFC	RobustScaler	Gini	–	None	300	None	–
RP	ETC	MinMaxScaler	Entropy	–	None	300	–	–
RP	HGB	Normalizer	–	–	–	–	–	–
RP	XGB	Normalizer	–	–	None	300	–	0.10
RP	LGBM	MaxAbsScaler	–	–	None	300	–	0.10

considerable. **Does the overlapping appliance during the dataset sampling process affect the performance?**

According to Tables 17 and 20, applying 0% overlapping increases the best-reported WSM by 1.32%. **According to Table 20, is it reasonable to obtain a WSM value of 100%?** The answer can be “YES” as this happens because of the high model complexity and the number of records is relatively low (i.e., 116).

5.3 Second category experiments

The current Section presents the experiments applied on both datasets after the TDA feature extraction step as presented in Fig. 5. Table 21 shows the best combinations and Table 22 shows the corresponding performance metrics applied on the four sampled datasets after TDA using the

Table 19 The corresponding performance metrics using each classifier applied on the reduced “UCI-HAR” dataset (116 × 100)

Feature Reduction Technique	Classifier	Elapsed time (s)	Accuracy (%)	Balanced Accuracy (%)	Precision (%)	Specificity (%)	Recall ()	F1 (\$)	IoU (%)	ROC (%)	NPV (%)
PCA	DT	1.2	99.14	98.45	97.41	99.48	97.41	97.41	94.96	98.45	99.48
PCA	AdaBoost	51.2	99.14	98.45	97.41	99.48	97.41	97.41	94.96	98.45	99.48
PCA	RFC	82.2	99.14	98.45	97.41	99.48	97.41	97.41	94.96	98.45	99.48
PCA	ETC	16.6	99.43	98.97	98.28	99.66	98.28	98.28	96.61	98.97	99.66
PCA	HGB	34.3	99.43	98.97	98.28	99.66	98.28	98.28	96.61	98.97	99.66
PCA	XGB	72.2	99.14	98.45	97.41	99.48	97.41	97.41	94.96	98.45	99.48
PCA	LGBM	18.4	99.14	98.45	97.41	99.48	97.41	97.41	94.96	98.45	99.48
LDA	DT	0.5	93.97	89.14	81.90	96.38	81.90	81.90	69.34	89.43	96.38
LDA	AdaBoost	33.0	83.91	71.03	51.72	90.34	51.72	51.72	34.88	73.61	90.34
LDA	RFC	66.9	96.55	93.79	89.66	97.93	89.66	89.66	81.25	93.88	97.93
LDA	ETC	14.8	95.69	92.24	87.07	97.41	87.07	87.07	77.10	92.39	97.41
LDA	HGB	17.5	93.10	87.59	79.31	95.86	79.31	79.31	65.71	87.98	95.86
LDA	XGB	52.0	95.69	92.24	87.07	97.41	87.07	87.07	77.10	92.39	97.41
LDA	LGBM	13.9	95.98	92.76	87.93	97.59	87.93	87.93	78.46	92.88	97.59
ICA	DT	1.3	79.60	63.28	38.79	87.76	38.79	38.79	24.06	67.85	87.76
ICA	AdaBoost	51.0	75.29	55.52	25.86	85.17	25.86	25.86	14.85	62.94	85.17
ICA	RFC	84.4	77.01	58.62	31.03	86.21	31.03	31.03	18.37	64.79	86.21
ICA	ETC	17.1	80.17	64.31	40.52	88.10	40.52	40.52	25.41	68.57	88.10
ICA	HGB	38.9	76.15	57.07	28.45	85.69	28.45	28.45	16.58	63.84	85.69
ICA	XGB	88.1	75.57	56.03	26.72	85.34	26.72	26.72	15.42	63.24	85.34
ICA	LGBM	26.3	76.44	57.59	29.31	85.86	29.31	29.31	17.17	64.15	85.86
T-SVD	DT	1.2	98.85	97.93	96.55	99.31	96.55	96.55	93.33	97.94	99.31
T-SVD	AdaBoost	51.9	99.14	98.45	97.41	99.48	97.41	97.41	94.96	98.45	99.48
T-SVD	RFC	81.8	98.85	97.93	96.55	99.31	96.55	96.55	93.33	97.94	99.31
T-SVD	ETC	16.4	99.43	98.97	98.28	99.66	98.28	98.28	96.61	98.97	99.66
T-SVD	HGB	34.7	99.14	98.45	97.41	99.48	97.41	97.41	94.96	98.45	99.48
T-SVD	XGB	70.9	98.56	97.41	95.69	99.14	95.69	95.69	91.74	97.43	99.14
T-SVD	LGBM	21.4	100	100	100	100	100	100	100	100	100
RP	DT	1.2	85.34	73.62	56.03	91.21	56.03	56.03	38.92	75.69	91.21
RP	AdaBoost	51.9	77.87	60.17	33.62	86.72	33.62	33.62	20.21	65.77	86.72
RP	RFC	78.9	97.41	95.34	92.24	98.45	92.24	92.24	85.60	95.40	98.45
RP	ETC	16.2	97.99	96.38	93.97	98.79	93.97	93.97	88.62	96.41	98.79
RP	HGB	38.3	93.68	88.62	81.03	96.21	81.03	81.03	68.12	88.94	96.21
RP	XGB	77.3	93.39	88.10	80.17	96.03	80.17	80.17	66.91	88.46	96.03
RP	LGBM	19.4	95.40	91.72	86.21	97.24	86.21	86.21	75.76	91.89	97.24

different mentioned classifiers. It shows the best of each classifier after the grid searching process.

Table 22 reports that the best reported performance metrics. It shows that for the UCI-HAR dataset with 0% overlap, the best reported accuracy, balanced accuracy, precision, specificity, recall, F1-score, IoU, ROC, and NPV are 95.69%, 92.24%, 87.07%, 97.41%, 87.07%, 87.07%, 77.10%, 92.39%, and 97.41%, respectively, by the LGBM classifier. In terms of the elapsed time, the best reported

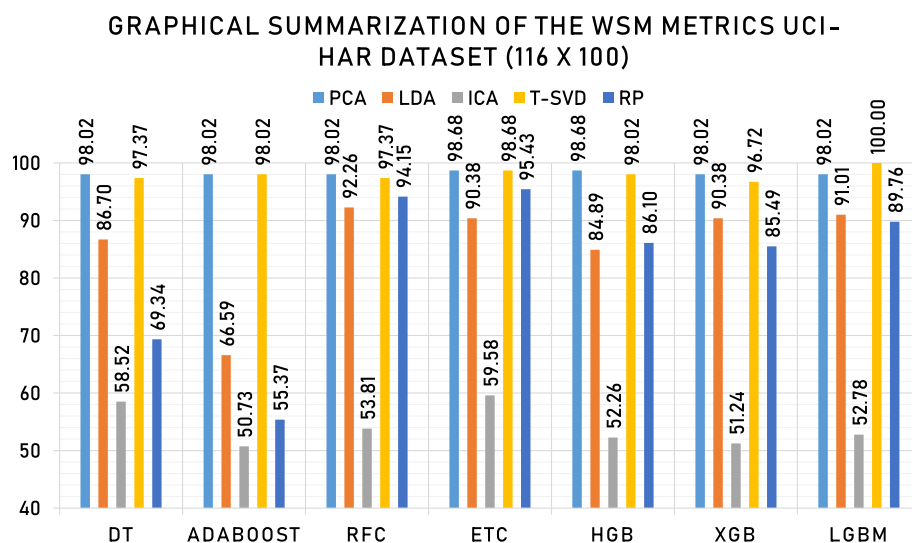
classifier is DT with 0.8 seconds. For the UCI-HAR dataset with 50% overlap, the best reported accuracy, balanced accuracy, precision, specificity, recall, F1-score, IoU, ROC, and NPV are 96.70%, 94.05%, 90.09%, 98.02%, 90.09%, 90.09%, 81.96%, 94.14%, and 98.02%, respectively, by the LGBM classifier. In terms of the elapsed time, the best reported classifier is DT with 0.9 seconds. For the WISDM dataset with 0% overlap, the best reported accuracy, balanced accuracy, precision, specificity, recall,

Table 20 Tabular summarization of the WSM metrics using the “UCI-HAR” dataset (116 × 100)

	AdaBoost (%)	DT (%)	ETC (%)	HGB (%)	LGBM (%)	RFC (%)	XGB (%)	Max value (%)
PCA	98.02	98.02	98.02	98.68	98.68	98.02	98.02	98.68
LDA	86.70	66.59	92.26	90.38	84.89	90.38	91.01	92.26
ICA	58.52	50.73	53.81	59.58	52.26	51.24	52.78	59.58
T-SVD	97.37	98.02	97.37	98.68	98.02	96.72	100	100
RP	69.34	55.37	94.15	95.43	86.10	85.49	89.76	95.43
Max Value	98.02	98.02	98.02	98.68	98.68	98.02	100	100

All the bold refers to the highest WSM score acquired

Fig. 9 Graphical summarization of the WSM metrics using the “UCI-HAR” dataset (116 × 100)



F1-score, IoU, ROC, and NPV are 95.07%, 91.12%, 85.20%, 97.04%, 85.20%, 85.20%, 74.22%, 91.31%, and 97.04%, respectively, by the LGBM classifier. In terms of the elapsed time, the best reported classifier is DT with 54.1 seconds. For the WISDM dataset with 50% overlap, the best reported accuracy, balanced accuracy, precision, specificity, recall, F1-score, IoU, ROC, and NPV are 95.34%, 91.61%, 86.01%, 97.20%, 86.01%, 86.01%, 75.45%, 91.78%, and 97.20%, respectively, by the LGBM classifier. In terms of the elapsed time, the best reported classifier is DT with 114.8 seconds. Table 23 and Fig. 10 summarize the WSM metrics. It shows that the highest WSM values are 90.38%, 92.57%, 89.05%, and 89.62% produced by the UCI-HAR + 0% Overlap, UCI-HAR + 50% Overlap, WISDM + 0% Overlap, and WISDM + 50% Overlap, respectively, by the LGBM classifiers.

5.3.1 Second category experiments remarks

Why was not the TDA feature extraction used over the flatten features instead of sampled ones directly? The

TDA feature extraction technique accepts data with any dimension (i.e., $n \geq 1$) as an input. Hence, performing two processes is not preferred to optimize the time. On the contrary, the traditional feature reduction techniques require 2-dimensional data as an input and hence the flatten step is crucial. **Does the overlapping appliance during the dataset sampling process affect the performance?** According to Table 23, applying 50% overlapping with UCI-HAR and WISDM datasets increases the best-reported WSM by 2.19% and 0.57%, respectively.

5.4 Third category experiments

The current Section presents the experiments applied on both datasets after the sampling step as presented in Fig. 5. The deep learning classifiers are used in this category. Table 24 shows the reported performance metrics for the four generated datasets using the 1D-CNN model. It shows that the highest accuracy, balanced accuracy, precision, specificity, recall, F1-score, IoU, ROC, NPV, and WSM values for the (1) “UCI-HAR + 50% Overlap” are 100%

Table 21 The best combinations applied on the four sampled datasets after TDA

Dataset Keyword	Classifier	Scaling technique	Criterion	Splitter	Max depth	Estimators #	Class weight	Learning rate
UCI-HAR 0% Overlap	DT	MaxAbsScaler	Entropy	Random	None	–	–	–
UCI-HAR 0% Overlap	AdaBoost	RobustScaler	–	–	–	300	–	1.00
UCI-HAR 0% Overlap	RFC	StandardScaler	Gini	–	None	300	None	–
UCI-HAR 0% Overlap	ETC	MinMaxScaler	Entropy	–	None	300	–	–
UCI-HAR 0% Overlap	HGB	StandardScaler	–	–	–	–	–	–
UCI-HAR 0% Overlap	XGB	StandardScaler	–	–	None	300	–	0.10
UCI-HAR 0% Overlap	LGBM	StandardScaler	–	–	None	300	–	0.01
UCI-HAR 50% Overlap	DT	StandardScaler	Gini	Best	None	–	–	–
UCI-HAR 50% Overlap	AdaBoost	StandardScaler	–	–	–	300	–	0.01
UCI-HAR 50% Overlap	RFC	MinMaxScaler	Entropy	–	None	300	Balanced	–
UCI-HAR 50% Overlap	ETC	StandardScaler	Gini	–	None	300	–	–
UCI-HAR 50% Overlap	HGB	StandardScaler	–	–	–	–	–	–
UCI-HAR 50% Overlap	XGB	StandardScaler	–	–	None	300	–	0.01
UCI-HAR 50% Overlap	LGBM	MinMaxScaler	–	–	None	300	–	1.00
WISDM + 0% Overlap	DT	MaxAbsScaler	Entropy	Best	None	–	–	–
WISDM + 0% Overlap	AdaBoost	Normalizer	–	–	–	300	–	0.10
WISDM + 0% Overlap	RFC	StandardScaler	Entropy	–	None	300	Balanced	–
WISDM + 0% Overlap	ETC	RobustScaler	Gini	–	None	300	–	–
WISDM + 0% Overlap	HGB	RobustScaler	–	–	–	–	–	–
WISDM + 0% Overlap	XGB	StandardScaler	–	–	None	300	–	0.10
WISDM + 0% Overlap	LGBM	MinMaxScaler	–	–	None	300	–	0.10
WISDM + 50% Overlap	DT	StandardScaler	Entropy	Best	None	–	–	–
WISDM + 50% Overlap	AdaBoost	Normalizer	–	–	–	300	–	0.10
WISDM + 50% Overlap	RFC	RobustScaler	Entropy	–	None	300	Balanced	–
WISDM + 50% Overlap	ETC	MinMaxScaler	Gini	–	None	300	–	–
WISDM + 50% Overlap	HGB	RobustScaler	–	–	–	–	–	–
WISDM + 50% Overlap	XGB	RobustScaler	–	–	None	300	–	0.10
WISDM + 50% Overlap	LGBM	MaxAbsScaler	–	–	None	300	–	0.10

that is reported by a batch size of 16, (2) “UCI-HAR + 0% Overlap” are 100% that is reported by batch sizes of 64 and 4, (3) “WISDM + 50% Overlap” are 99.90%, 99.81%, 99.70%, 99.94%, 99.68%, 99.69%, 99.38%, 99.81%, 99.94%, and 99.76% that is reported by a batch size of 512, and (4) “WISDM + 0% Overlap” are 99.71%, 99.47%, 99.17%, 99.83%, 99.10%, 99.13%, 98.28%, 99.47%, 99.82%, and 99.33% that is reported by a batch size of 256.

Table 25 shows the reported performance metrics for the four generated datasets using the GRU model. It shows that the highest accuracy, balanced accuracy, precision, specificity, recall, F1-score, IoU, ROC, NPV, and WSM values for the (1) “UCI-HAR + 50% Overlap” are 99.86%, 99.74%, 99.57%, 99.91%, 99.57%, 99.57%, 99.14%, 99.74%, 99.91%, and 99.67% that is reported by a batch size of 32, (2) “UCI-HAR + 0% Overlap” are 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, and 100% that is reported by a batch size of 16, (3)

“WISDM + 50% Overlap” are 98.83%, 97.75%, 96.81%, 99.37%, 96.13%, 96.47%, 93.18%, 97.76%, 99.23%, and 97.28% that is reported by a batch size of 256, and (4) “WISDM + 0% Overlap” are 98.11%, 96.34%, 94.88%, 98.99%, 93.69%, 94.28%, 89.18%, 96.38%, 98.74% and 95.62% that is reported by a batch size of 256.

Table 26 shows the reported performance metrics for the four generated datasets using the BiLSTM model. It shows that the highest accuracy, balanced accuracy, precision, specificity, recall, F1-score, IoU, ROC, NPV, and WSM value for the (1) “UCI-HAR + 50% Overlap” are 99.71%, 99.48%, 99.14%, 99.83%, 99.14%, 99.14%, 98.29%, 99.48%, 99.83%, and 99.34% that is reported by a batch size of 8, (2) “UCI-HAR + 0% Overlap” are 99.71%, 99.48%, 99.14%, 99.83%, 99.14%, 99.14%, 98.29%, 99.48%, 99.83%, and 99.34% that is reported by a batch size of 8, (3) “WISDM + 50% Overlap” are 94.98%, 89.59%, 87.53%, 97.68%, 81.50%, 84.41%, 73.02%,

Table 22 The corresponding performance metrics using each classifier applied on the four sampled datasets after TDA

Dataset Keyword	Classifier	Elapsed time (s) (%)	Accuracy (%)	Balanced accuracy (%)	Precision (%)	Specificity (%)	Recall (%)	F1 (%)	IoU (%)	ROC (%)	NPV (%)
UCI-HAR 0% Overlap	DT	0.8	92.53	86.55	77.59	95.52	77.59	77.59	63.38	87.01	95.52
UCI-HAR 0% Overlap	AdaBoost	36.8	82.47	68.45	47.41	89.48	47.41	47.41	31.07	71.61	89.48
UCI-HAR 0% Overlap	RFC	71.3	94.83	90.69	84.48	96.90	84.48	84.48	73.13	90.90	96.90
UCI-HAR 0% Overlap	ETC	15.3	94.54	90.17	83.62	96.72	83.62	83.62	71.85	90.41	96.72
UCI-HAR 0% Overlap	HGB	18.3	91.38	84.48	74.14	94.83	74.14	74.14	58.90	85.11	94.83
UCI-HAR 0% Overlap	XGB	49.2	93.10	87.59	79.31	95.86	79.31	79.31	65.71	87.98	95.86
UCI-HAR 0% Overlap	LGBM	16.5	95.69	92.24	87.07	97.41	87.07	87.07	77.10	92.39	97.41
UCI-HAR 50% Overlap	DT	0.9	94.40	89.91	83.19	96.64	83.19	83.19	71.22	90.16	96.64
UCI-HAR 50% Overlap	AdaBoost	42.7	77.16	58.88	31.47	86.29	31.47	31.47	18.67	64.95	86.29
UCI-HAR 50% Overlap	RFC	81.5	95.55	91.98	86.64	97.33	86.64	86.64	76.43	92.14	97.33
UCI-HAR 50% Overlap	ETC	16.5	96.12	93.02	88.36	97.67	88.36	88.36	79.15	93.13	97.67
UCI-HAR 50% Overlap	HGB	37.5	95.98	92.76	87.93	97.59	87.93	87.93	78.46	92.88	97.59
UCI-HAR 50% Overlap	XGB	70.9	95.55	91.98	86.64	97.33	86.64	86.64	76.43	92.14	97.33
UCI-HAR 50% Overlap	LGBM	23.3	96.70	94.05	90.09	98.02	90.09	90.09	81.96	94.14	98.02
WISDM + 0% Overlap	DT	54.1	92.72	86.90	78.16	95.63	78.16	78.16	64.15	87.33	95.63
WISDM + 0% Overlap	AdaBoost	1491.9	89.60	81.27	68.79	93.76	68.79	68.79	52.42	82.23	93.76
WISDM + 0% Overlap	RFC	5458.8	94.72	90.50	84.17	96.83	84.17	84.17	72.67	90.72	96.83
WISDM + 0% Overlap	ETC	338.3	94.74	90.54	84.23	96.85	84.23	84.23	72.75	90.76	96.85
WISDM + 0% Overlap	HGB	206.2	94.86	90.74	84.57	96.91	84.57	84.57	73.26	90.95	96.91
WISDM + 0% Overlap	XGB	6106.3	94.88	90.78	84.63	96.93	84.63	84.63	73.35	90.99	96.93
WISDM + 0% Overlap	LGBM	338.3	95.07	91.12	85.20	97.04	85.20	85.20	74.22	91.31	97.04
WISDM + 50% Overlap	DT	114.8	92.90	87.22	78.70	95.74	78.70	78.70	64.89	87.64	95.74
WISDM + 50% Overlap	AdaBoost	3024.4	89.75	81.55	69.25	93.85	69.25	69.25	52.96	82.47	93.85
WISDM + 50% Overlap	RFC	11858.8	94.99	90.99	84.98	97.00	84.98	84.98	73.89	91.19	97.00
WISDM + 50% Overlap	ETC	678.5	95.00	90.99	84.99	97.00	84.99	84.99	73.89	91.19	97.00
WISDM + 50% Overlap	HGB	305.0	95.10	91.18	85.30	97.06	85.30	85.30	74.37	91.37	97.06
WISDM + 50% Overlap	XGB	13159.5	95.23	91.42	85.70	97.14	85.70	85.70	74.98	91.60	97.14

Table 22 (continued)

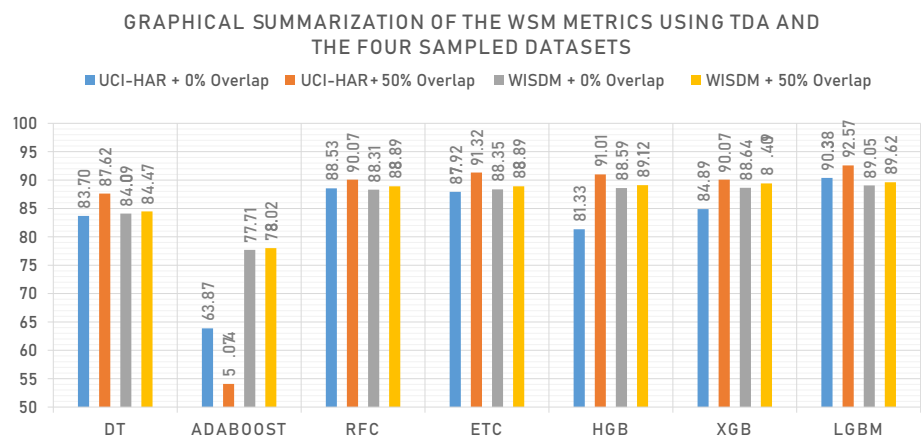
Dataset Keyword	Classifier	Elapsed time (s) (%)	Accuracy (%)	Balanced accuracy (%)	Precision (%)	Specificity (%)	Recall (%)	F1 (%)	IoU (%)	ROC (%)	NPV (%)
WISDM + 50% Overlap	LGBM	554.3	95.34	91.61	86.01	97.20	86.01	86.01	75.45	91.78	97.20

Table 23 Tabular summarization of the WSM metrics using TDA and the four sampled datasets

Dataset Keyword	DT (%)	AdaBoost (%)	RFC (%)	ETC (%)	HGB (%)	XGB (%)	LGBM (%)	Max value (%)
UCI-HAR 0% Overlap	83.70	63.87	88.53	87.92	81.33	84.89	90.38	90.38
UCI-HAR 50% Overlap	87.62	54.07	90.07	91.32	91.01	90.07	92.57	92.57
WISDM + 0% Overlap	84.09	77.71	88.31	88.35	88.59	88.64	89.05	89.05
WISDM + 50% Overlap	84.47	78.02	88.89	88.89	89.12	89.40	89.62	89.62

All the bold refers to the highest WSM score acquired

Fig. 10 Graphical summarization of the WSM metrics using TDA and the four sampled datasets



89.95%, 96.35%, and 88.33% that is reported by a batch size of 256, and (4) “WISDM + 0% Overlap” are 94.78%, 88.98%, 87.36%, 97.68%, 80.29%, 83.68%, 71.94%, 89.41%, 96.12%, and 87.80% that is reported by a batch size of 256.

Figure 11 shows a graphical comparison of the best reported results between the three approaches (i.e., 1D-CNN, GRU, and BiLSTM). It shows that the 1D-CNN approach outperforms the other two approaches. Also, the overlapping mechanism shows better results compared with the non-overlapping approach.

5.4.1 Third category experiments remarks

Does the overlapping appliance during the dataset sampling process affect the performance? According to Tables 24, 25, and 26, applying 50% overlapping with WISDM dataset increases the best-reported WSM values by (1) 0.43% for the 1D-CNN model, (2) 1.66% for the GRU model, and (3) 0.53% for the BiLSTM model. However, it does not increase the best-reported WSM values for the UCI-HAR dataset.

Table 24 The performance metrics using the 1D-CNN model applied on the four generated datasets

Dataset Keyword	BS	Accuracy (%)	Balanced accuracy (%)	Precision (%)	Specificity (%)	Recall (%)	F1 (%)	IoU (%)	ROC (%)	NPV (%)	WSM (%)
UCI-HAR + 50% Overlap	128	99.43	98.97	98.28	99.66	98.28	98.28	96.61	98.97	99.66	98.68
UCI-HAR + 50% Overlap	64	99.86	99.74	99.57	99.91	99.57	99.57	99.14	99.74	99.91	99.67
UCI-HAR + 50% Overlap	32	99.14	98.45	97.41	99.48	97.41	97.41	94.96	98.45	99.48	98.02
UCI-HAR + 50% Overlap	16	100	100	100	100	100	100	100	100	100	100
UCI-HAR + 50% Overlap	8	95.26	86.29	98.26	99.74	72.84	83.66	71.91	87.33	94.84	87.79
UCI-HAR + 50% Overlap	4	97.27	94.91	92.17	98.45	91.38	91.77	84.80	94.98	98.28	93.78
UCI-HAR + 0% Overlap	128	99.43	98.97	98.28	99.66	98.28	98.28	96.61	98.97	99.66	98.68
UCI-HAR + 0% Overlap	64	100	100	100	100	100	100	100	100	100	100
UCI-HAR + 0% Overlap	32	98.85	97.93	96.55	99.31	96.55	96.55	93.33	97.94	99.31	97.37
UCI-HAR + 0% Overlap	16	98.42	96.98	95.65	99.14	94.83	95.24	90.91	97.01	98.97	96.35
UCI-HAR + 0% Overlap	8	98.28	96.90	94.83	98.97	94.83	94.83	90.16	96.92	98.97	96.07
UCI-HAR + 0% Overlap	4	100	100	100	100	100	100	100	100	100	100
WISDM + 50% Overlap	2048	99.73	99.49	99.20	99.84	99.15	99.18	98.36	99.50	99.83	99.36
WISDM + 50% Overlap	1024	99.83	99.68	99.49	99.90	99.47	99.48	98.96	99.68	99.89	99.60
WISDM + 50% Overlap	512	99.90	99.81	99.70	99.94	99.68	99.69	99.38	99.81	99.94	99.76
WISDM + 50% Overlap	256	99.83	99.69	99.51	99.90	99.48	99.50	99.00	99.69	99.90	99.61
WISDM + 0% Overlap	2048	99.33	98.75	98.13	99.63	97.86	98.00	96.07	98.75	99.57	98.45
WISDM + 0% Overlap	1024	99.37	98.84	98.18	99.64	98.04	98.11	96.29	98.84	99.61	98.55
WISDM + 0% Overlap	512	99.27	98.61	97.96	99.59	97.64	97.80	95.69	98.62	99.53	98.30
WISDM + 0% Overlap	256	99.71	99.47	99.17	99.83	99.10	99.13	98.28	99.47	99.82	99.33

5.5 Overall remarks

From the experiments performed in the current study and partitioned into three categories, the best approach is 1D-CNN concerning the “WISDM” and “UCI-HAR” datasets, respectively. Additionally, applying 50% overlapping with “UCI-HAR” and “WISDM” datasets increases the best-reported metrics. For the “WISDM” dataset,

concerning the first category of experiments, the accuracy, balanced accuracy, precision, specificity, recall, F1-score, IoU, ROC, and NPV had been increased by 0.39%, 0.67%, 1.11%, 0.22%, 1.11%, 1.11%, 1.58%, 0.63%, and 0.22%, respectively. Concerning the second category of experiments, the accuracy, balanced accuracy, precision, specificity, recall, F1-score, IoU, ROC, and NPV had increased by 0.27%, 0.49%, 0.81%, 0.16%, 0.81%, 0.81%, 1.23%,

Table 25 The performance metrics using the GRU model applied on the four generated datasets

Dataset Keyword	BS	Accuracy (%)	Balanced accuracy (%)	Precision (%)	Specificity (%)	Recall (%)	F1 (%)	IoU (%)	ROC (%)	NPV (%)	WSM (%)
UCI-HAR + 50% Overlap	128	99.71	99.48	99.14	99.83	99.14	99.14	98.29	99.48	99.83	99.34
UCI-HAR + 50% Overlap	64	98.99	97.67	98.23	99.66	95.69	96.94	94.07	97.69	99.14	97.57
UCI-HAR + 50% Overlap	32	99.86	99.74	99.57	99.91	99.57	99.57	99.14	99.74	99.91	99.67
UCI-HAR + 50% Overlap	16	99.14	98.45	97.41	99.48	97.41	97.41	94.96	98.45	99.48	98.02
UCI-HAR + 50% Overlap	8	94.25	89.66	82.76	96.55	82.76	82.76	70.59	89.92	96.55	87.31
UCI-HAR + 50% Overlap	4	99.28	98.71	97.84	99.57	97.84	97.84	95.78	98.71	99.57	98.35
UCI-HAR + 0% Overlap	128	99.71	99.48	99.14	99.83	99.14	99.14	98.29	99.48	99.83	99.34
UCI-HAR + 0% Overlap	64	99.71	99.48	99.14	99.83	99.14	99.14	98.29	99.48	99.83	99.34
UCI-HAR + 0% Overlap	32	99.43	98.97	98.28	99.66	98.28	98.28	96.61	98.97	99.66	98.68
UCI-HAR + 0% Overlap	16	100	100	100	100	100	100	100	100	100	100
WISDM + 50% Overlap	2048	96.72	93.36	91.73	98.41	88.30	89.99	81.79	93.49	97.68	92.39
WISDM + 50% Overlap	1024	98.02	96.14	94.71	98.96	93.32	94.01	88.70	96.18	98.67	95.41
WISDM + 50% Overlap	512	98.27	96.62	95.38	99.09	94.15	94.76	90.05	96.65	98.83	95.98
WISDM + 50% Overlap	256	98.83	97.75	96.81	99.37	96.13	96.47	93.18	97.76	99.23	97.28
WISDM + 0% Overlap	2048	94.98	89.25	88.23	97.85	80.66	84.27	72.82	89.67	96.20	88.21
WISDM + 0% Overlap	1024	95.38	90.63	88.14	97.75	83.50	85.76	75.06	90.91	96.73	89.32
WISDM + 0% Overlap	512	97.40	94.97	92.95	98.62	91.32	92.13	85.40	95.04	98.27	94.01
WISDM + 0% Overlap	256	98.11	96.34	94.88	98.99	93.69	94.28	89.18	96.38	98.74	95.62

0.47%, and 0.16%, respectively. Concerning the third category of experiments, the accuracy, balanced accuracy, precision, specificity, recall, F1-score, IoU, ROC, and NPV had increased by 0.19%, 0.34%, 0.53%, 0.11%, 0.58%, 0.56%, 0.1%, 0.34%, 0.12%, and 0.43%, respectively. For the “UCI-HAR” dataset, concerning the second category of experiments, the accuracy, balanced accuracy, precision, specificity, recall, F1-score, IoU, ROC, and NPV had increased by 1.01%, 1.81%, 3.02%, 0.61%, 3.02%, 3.02%, 4.86%, 1.75%, and 0.61%, respectively. However, for the first and third categories of experiments, the reported metrics had not been affected when applying the overlapping.

Why were the original datasets not used directly?

The original datasets, specially the WISDM dataset, are large and time-demanding (i.e., the training process is time-consuming) and suffer from imbalanced data issue. Additionally, the time-series behaviour is not employed.

Why were the resulting datasets from the balancing phase not used directly?

As with the original datasets, the balancing-resulting datasets are large and time-demanding. Moreover, time-series behaviour data will not be employed.

Is the TDA better than the traditional ML features reduction techniques? From Tables 11, 14, 17, 20, and 23, TDA feature extraction is better than traditional feature reduction techniques. Concerning the WSM value,

Table 26 The performance metrics using the BiLSTM model applied on the four generated datasets

Dataset Keyword	BS	Accuracy (%)	Balanced accuracy (%)	Precision (%)	Specificity (%)	Recall (%)	F1 (%)	IoU (%)	ROC (%)	NPV (%)	WSM (%)
UCI-HAR + 50% Overlap	128	95.04	88.41	90.55	98.36	78.45	84.06	72.51	88.96	95.80	88.02
UCI-HAR + 50% Overlap	64	93.68	87.24	83.33	96.90	77.59	80.36	67.16	87.77	95.58	85.51
UCI-HAR + 50% Overlap	32	98.42	96.81	96.05	99.22	94.40	95.22	90.87	96.84	98.88	96.30
UCI-HAR + 50% Overlap	16	98.35	96.94	95.24	99.05	94.83	95.03	90.53	96.96	98.97	96.21
UCI-HAR + 50% Overlap	8	99.71	99.48	99.14	99.83	99.14	99.14	98.29	99.48	99.83	99.34
UCI-HAR + 50% Overlap	4	98.78	97.72	96.54	99.31	96.12	96.33	92.92	97.73	99.22	97.18
UCI-HAR + 0% Overlap	128	99.57	99.05	99.13	99.83	98.28	98.70	97.44	99.05	99.66	98.97
UCI-HAR + 0% Overlap	64	75.43	51.47	19.78	87.41	15.52	17.39	9.52	62.78	83.80	47.01
UCI-HAR + 0% Overlap	32	97.99	96.38	93.97	98.79	93.97	93.97	88.62	96.41	98.79	95.43
UCI-HAR + 0% Overlap	16	96.70	93.53	91.15	98.28	88.79	89.96	81.75	93.65	97.77	92.40
UCI-HAR + 0% Overlap	8	99.71	99.48	99.14	99.83	99.14	99.14	98.29	99.48	99.83	99.34
UCI-HAR + 0% Overlap	4	95.40	91.03	87.50	97.59	84.48	85.96	75.38	91.27	96.92	89.50
WISDM + 50% Overlap	2048	93.21	83.35	88.02	98.13	68.58	77.09	62.72	84.65	93.98	83.30
WISDM + 50% Overlap	1024	93.65	85.76	86.01	97.60	73.92	79.51	65.99	86.57	94.93	84.88
WISDM + 50% Overlap	512	92.47	82.00	85.24	97.70	66.29	74.58	59.46	83.49	93.54	81.64
WISDM + 50% Overlap	256	94.98	89.59	87.53	97.68	81.50	84.41	73.02	89.95	96.35	88.33
WISDM + 0% Overlap	2048	88.76	70.15	81.41	98.07	42.23	55.61	38.52	75.50	89.46	71.08
WISDM + 0% Overlap	1024	91.85	77.03	93.66	99.26	54.81	69.15	52.85	80.18	91.65	78.94
WISDM + 0% Overlap	512	93.23	83.83	87.04	97.92	69.74	77.43	63.18	85.01	94.18	83.51
WISDM + 0% Overlap	256	94.78	88.98	87.36	97.68	80.29	83.68	71.94	89.41	96.12	87.80

this conclusion was reached because of the following: (1) for the WISDM dataset with 50% overlapping, TDA has outperformed the best reported traditional technique by 3.77% and (2) for the WISDM dataset with 0% overlapping, TDA has outperformed the best reported traditional technique by 6.74%.

According to Tables 10, 13, 16, 19, and 22, why did the precision, recall, and F1-score have the same values and also for the specificity and NPV? As mentioned

before, the micro-average is used as an averaging method. In this method, if there is a false positive, there will always also be a false negative and vice versa, as always one class is predicted. Hence, increasing only FP or FN but not both is not possible (i.e., the resulting FP and FN have the same values). According to the equations mentioned in Table 7, precision, recall, and F1-score will always have the same values. The same for specificity and NPV values. **Is the DL approach better than the traditional ML approach with**

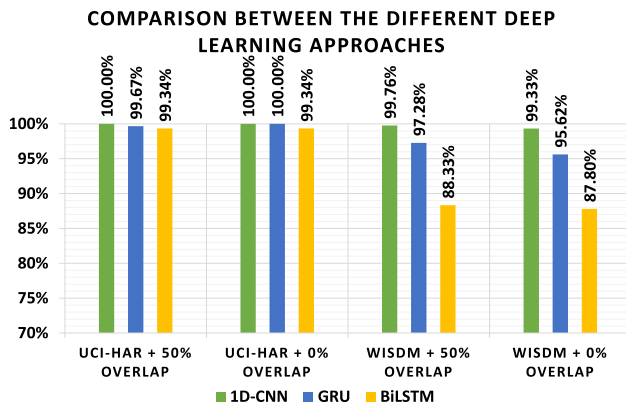


Fig. 11 Graphical comparison of the best reported results between the three approaches (i.e., 1D-CNN, GRU, and BiLSTM)

the traditional ML features reduction techniques? From Tables 11, 14, 17, 20, 24, 25, and 26, DL approach achieved better WSM values than the traditional ML approach with the traditional ML features reduction techniques. **Is the DL approach better than the traditional ML approach with the TDA features?** From Tables 23, 24, 25, and 26, DL approach achieved better WSM values than the traditional ML approach with the TDA features. It is worth mentioning that when the results were conducted from the three categories of experiments, the DL algorithms used with both datasets after the sampling step reported the best results so far.

5.6 Related studies comparison

The performance of the algorithms from the most recent studies and those proposed was compared. In both datasets,

the suggested approach performed well in terms of classification. Table 27 shows a comparison between the results of the related studies and the presented approach concerning the two utilized datasets. In the WISDM dataset, the best-reported accuracy and F1-score of the proposed algorithm were 1.08% and 0.88% better than those of Teng’s layer-wise CNN approach. Concerning the UCI-HAR dataset, the accuracy score of the proposed approach was 3.02% and 0.85% better than those of Teng’s layer-wise CNN and Wang’s hierarchical deep LSTM network, respectively.

6 Limitations

Although the current study presented the potential of using machine and deep learning models to perform the Human Activity Recognition (HAR) task, some limitations are presented. The main limitation is the instantaneity, the high-dimensional features results are consuming a considerable amount of time in the classifier training stage. Additionally, only features extraction using TDA and 5 dimensionality reduction techniques were used. Also, GS is not utilized with the suggested deep learning models. To overcome the imbalanced data problem, only the over-sampling techniques were used.

7 Conclusions

Recently, HAR has earned a lot of interest and emerged as a promising approach. It has a wide range of possible applications (e.g., intelligent assistance for people suffering

Table 27 Comparison between the results of the current study and the related studies utilizing the same dataset(s)

Reference	WISDM				UCI-HAR			
	Accuracy (%)	F1-score (%)	Recall (%)	Precision (%)	Accuracy (%)	F1-score (%)	Recall (%)	Precision (%)
Ignatov et al. [69]	93.32	–	–	–	97.63	97.62	–	–
Zhang et al. [34]	96.4	–	–	–	–	–	–	–
Teng et al. [58]	98.82	98.81	–	–	96.98	96.97	–	–
Wang and Liu [52]	–	–	–	–	99.15	–	–	–
Xia et al. [68]	95.85	–	95.75	–	95.7	–	95.8	–
Khan and Ahmad [71]	98.18	97.20	97.29	97.12	95.38	95.37	95.42	95.48
Current Study (1st Category Experiments)	93.92	81.76	81.76	81.76	100	100	100	100
Current Study (2nd Category Experiments)	95.34	86.01	86.01	86.01	96.70	90.09	90.09	90.09
Current Study (3rd Category Experiments)	99.90	99.69	99.68	99.70	100	100	100	100

from cognitive disorders and elderly people). In this research, a comprehensive analysis for recognizing activities of humans with the help of traditional feature reduction techniques, feature extraction, ML, and DL algorithms. The sensor-based data retrieved from two public datasets (i.e., WISDM and UCI-HAR) are used to train and evaluate different machine and deep learning models to recognize several human activities. Nine different oversampling techniques were utilized to deal with the problem of imbalanced data. Additionally, a sampling mechanism with two overlapping percentages (i.e., 50% and 0%) is applied to each balanced dataset to use the advantage of time-series data. For feature extraction and dimensionality reduction, five traditional techniques were applied (i.e., PCA, LDA, ICA, RP, and T-SVD). Additionally, feature extraction using Topological Data Analysis (TDA) was used. Seven types of machine learning algorithms are used where six of them are ensemble classifiers (i.e., LGBM, XGB, Ada-Boost, HGB, RF, ETs, and DT). For the DL experiments, three types of algorithms are used (i.e., 1D-CNN, GRU, and BiLSTM). Three categories of experiments were created. The first category is constructed using traditional feature reduction techniques and ML algorithms, while the second category is conducted using TDA feature extraction and ML algorithms. For the first two categories of experiments, grid search is used to perform the hyperparameter optimization process. For the third category experiments, automatic feature extraction is performed using the three chosen DL algorithms. For the first category experiments, the best-reported scores concerning the WISDM dataset are accuracy, F1-score, recall, and precision of 93.92%, 81.76%, 81.76%, and 81.76%, respectively, achieved by XGB classifier with RP as feature reduction technique. When concerning the UCI-HAR dataset, the best-reported scores are accuracy, F1-score, recall, and precision of 100% achieved by LGBM classifier with T-SVD as feature reduction technique. For the second category experiments, the best-reported scores concerning the WISDM dataset are accuracy, F1-score, recall, and precision of 95.34%, 86.01%, 86.01%, and 86.01%, respectively, achieved by LGBM classifier. When concerning the UCI-HAR dataset, the best-reported scores are accuracy, F1-score, recall, and precision of 96.70%, 90.09%, 90.09%, and 90.09%, respectively, achieved by LGBM. For the third category, the best-reported scores concerning the WISDM dataset are accuracy, F1-score, recall, and precision of 99.90%, 99.69%, 99.68%, and 99.70%, respectively, achieved by 1D-CNN classifier. When concerning the UCI-HAR dataset, the best-reported scores are accuracy, F1-score, recall, and precision of 100% achieved by 1D-CNN classifier. The utilized data was gathered through accelerometers that were worn on various body parts of people. The gathered information is a time series that depicts the acceleration

along all three dimensions. There are two aspects to the data (i.e., time steps and the values of the acceleration in three axes). In the current study, the used time series data had a strong time locality that can be recovered by convolutions, hence 1D-CNN performed the best. This is understandable given that a 1D convolution on a time series roughly computes its moving average (using terms from digital signal processing) and applies a filter to the time series, giving some hints about the trend of the data. The reported results were then compared with 6 of prior related works utilizing the same dataset(s). This showed that the current study had outperformed all the mentioned prior works.

7.1 Future work

In future studies, a meta-heuristic optimizer, e.g., Aquila Optimizer and Sparrow Search Algorithm, can be used to optimize the hyperparameters of the deep learning models. The experiments can be applied with different datasets with more features such as the heart rate. The undersampling techniques can be tested and compared with the oversampling ones on the datasets. To enhance used datasets, additional augmentation methods, such as deep learning-based generative models (e.g., conditional generative adversarial networks and variationally auto-encoders), will be incorporated.

Author Contributions All the authors have participated in writing the manuscript and have revised the final version. All authors read and approved the final manuscript.

Funding No funding was received for this work (i.e., study).

Data Availability The datasets, if existing, that are used, generated, or analyzed during the current study (A) if the datasets are owned by the authors, they are available from the corresponding author on reasonable request, (B) if the datasets are not owned by the authors, the supplementary information including the links and sizes are included in this published article.

Declarations

Conflict of interest No conflict of interest exists. We wish to confirm that, there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Intellectual property We confirm that, we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, concerning intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

Research ethics We further confirm, if existing, that any aspect of the work covered in this manuscript that has involved human patients has been conducted with the ethical approval of all relevant bodies and that such approvals are acknowledged within the manuscript. Written consent to publish potentially identifying information, such as details of the case and photographs, was obtained from the patient(s) or their legal guardian(s).

Authorship We confirm that the manuscript has been read and approved by all named authors. We confirm that the order of authors listed in the manuscript has been approved by all named authors.

Compliance with ethical standards The current study does not contain any studies with human participants and/or animals performed by any of the authors.

Consent to participate There is no informed consent for the current study.

Consent for publication Not Applicable.

References

- Balaha HM, Hassan AE-S (2022) Skin cancer diagnosis based on deep transfer learning and sparrow search algorithm. *Neural Comput Appl* 35:1–39
- Balaha HM, Hassan AE-S (2022) A variate brain tumor segmentation, optimization, and recognition framework. *Artif Intell Rev* <https://doi.org/10.1007/s10462-022-10337-8>
- Baghdadi NA, Alsayed SK, Malki GA, Balaha HM, Farghaly Abdelaliem SM (2023) An analysis of burnout among female nurse educators in Saudi Arabia using k-means clustering. *EEur J Investig Health Psychol Educ* 13(1):33–53
- Balaha HM, Saafan MM (2021) Automatic exam correction framework (aecf) for the mcqs, essays, and equations matching. *IEEE Access* 9:32368–32389
- Balaha MH, El-Ibiary MT, El-Dorf AA, El-Shewaikh SL, Balaha HM (2022) Construction and writing flaws of the multiple-choice questions in the published test banks of obstetrics and gynecology: Adoption, caution, or mitigation? *Avicenna J Med* 12(03):138–147
- Balaha HM, El-Gendy EM, Saafan MM (2022) A complete framework for accurate recognition and prognosis of COVID-19 patients based on deep transfer learning and feature classification approach. *Artif Intell Rev* 55(6):5063–5108
- Baghdadi NA, Malki A, Balaha HM, AbdulAzeem Y, Badawy M, Elhosseini M (2022) Classification of breast cancer using a manta-ray foraging optimized transfer learning framework. *PeerJ Comput Sci* 8:1054
- Balaha HM, Shaban AO, El-Gendy EM, Saafan MM (2022) A multi-variate heart disease optimization and recognition framework. *Neural Comput Appl* 34(18):15907–15944
- Balaha MM, El-Kady S, Balaha HM, Salama M, Emad E, Hassan M, Saafan MM (2022) A vision-based deep learning approach for independent-users Arabic sign language interpretation. *Multimed Tools Appl* <https://doi.org/10.1007/s11042-022-13423-9>
- Baghdadi NA, Malki A, Balaha HM, AbdulAzeem Y, Badawy M, Elhosseini M (2022) An optimized deep learning approach for suicide detection through Arabic tweets. *PeerJ Comput Sci* 8:1070
- Yousif NR, Balaha HM, Haikal AY, El-Gendy EM (2022) A generic optimization and learning framework for Parkinson disease via speech and handwritten records. *J Ambient Intell Human Comput* <https://doi.org/10.1007/s12652-022-04342-6>
- Yassine A, Singh S, Alamri A (2017) Mining human activity patterns from smart home big data for health care applications. *IEEE Access* 5:13131–13141
- Gupta HP, Chudgar HS, Mukherjee S, Dutta T, Sharma K (2016) A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors. *IEEE Sensors J* 16(16):6425–6432
- Subasi A, Radhwan M, Kurdi R, Khateeb K (2018) IoT based mobile healthcare system for human activity recognition. In: 2018 15th Learning and Technology Conference (L &T), pp. 29–34. IEEE
- Sridevi G, Kumar SS (2019) Image inpainting based on fractional-order nonlinear diffusion for image reconstruction. *Circuits Syst Signal Process* 38(8):3802–3817
- Babiker M, Khalifa OO, Htike KK, Hassan A, Zaharadeen M (2017) Automated daily human activity recognition for video surveillance using neural network. In: 2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA), pp. 1–5. IEEE
- Park S.Y, Ju H, Park C.G (2016) Stance phase detection of multiple actions for military drill using foot-mounted imu sensors 14:16
- Sathyannarayana A, Ofli F, Fernandez-Luque L, Srivastava J, Elmagarmid A, Arora T, Taheri S (2016) Robust automated human activity recognition and its application to sleep research. In: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), pp. 495–502. IEEE
- Rezaie H, Ghassemian M (2017) An adaptive algorithm to improve energy efficiency in wearable activity recognition systems. *IEEE Sens J* 17(16):5315–5323
- Karantonis DM, Narayanan MR, Mathie M, Lovell NH, Celler BG (2006) Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Trans Inform Technol Biomed* 10(1):156–167
- Wang J, Chen Y, Hao S, Peng X, Hu L (2019) Deep learning for sensor-based activity recognition: a survey. *Pattern Recognit Lett* 119:3–11
- Chen L, Hoey J, Nugent C.D, Cook D.J, Yu Z (2012) Sensor-based activity recognition. *IEEE Trans Syst Man Cybernet Part C Appl Rev* 42(6):790–808
- Azorin-Lopez J, Saval-Calvo M, Fuster-Guillo A, Garcia-Rodriguez J (2016) A novel prediction method for early recognition of global human behaviour in image sequences. *Neural Process Lett* 43(2):363–387
- Abdulazeem Y, Balaha HM, Bahgat WM, Badawy M (2021) Human action recognition based on transfer learning approach. *IEEE Access* 9:82058–82069
- Vavoulas G, Padiaditis M, Spanakis EG, Tsiknakis M (2013) The mobifall dataset: An initial evaluation of fall detection algorithms using smartphones. In: 13th IEEE International Conference on BioInformatics and BioEngineering, pp. 1–4. IEEE
- Ali S, Khan NA, Haneef M, Luo X (2017) Blind source separation schemes for mono-sensor and multi-sensor systems with application to signal detection. *Circuits Syst Signal Process* 36(11):4615–4636
- Lu Y, Wei Y, Liu L, Zhong J, Sun L, Liu Y (2017) Towards unsupervised physical activity recognition using smartphone accelerometers. *Multimed Tools Appl* 76(8):10701–10719
- Jain A, Kanhangad V (2016) Investigating gender recognition in smartphones using accelerometer and gyroscope sensor readings. In: 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), pp. 597–602. IEEE

29. Lim WK, Davila S, Teo JX, Yang C, Pua CJ, Blöcker C, Lim JQ, Ching J, Yap JJJ, Tan SY et al (2018) Beyond fitness tracking: the use of consumer-grade wearable data from normal volunteers in cardiovascular and lipidomics research. *PLoS Biol* 16(2):2004285
30. Tison GH, Sanchez JM, Ballinger B, Singh A, Olgin JE, Pletcher MJ, Vittinghoff E, Lee ES, Fan SM, Gladstone RA et al (2018) Passive detection of atrial fibrillation using a commercially available smartwatch. *JAMA Cardiol* 3(5):409–416
31. Lu T-C, Fu C-M, Ma MH-M, Fang C-C, Turner AM (2016) Healthcare applications of smart watches. *Appl Clin Inform* 7(03):850–869
32. Lunney A, Cunningham NR, Eastin MS (2016) Wearable fitness technology: a structural investigation into acceptance and perceived fitness outcomes. *Comput Human Behav* 65:114–120
33. Alani AA, Cosma G, Taherkhani A (2020) Classifying imbalanced multi-modal sensor data for human activity recognition in a smart home using deep learning. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 . IEEE
34. Zhang Y, Zhang Z, Zhang Y, Bao J, Zhang Y, Deng H (2019) Human activity recognition based on motion sensor using u-net. *IEEE Access* 7:75213–75226
35. Buda M, Maki A, Mazurowski MA (2018) A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw* 106:249–259
36. Ravi D, Wong C, Lo B, Yang G-Z (2016) Deep learning for human activity recognition: A resource efficient implementation on low-power devices. In: 2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN), pp. 71–76 . IEEE
37. Wang J, Chen Y, Hao S, Peng X, Hu L (2018) Deep learning for sensor-based activity recognition: a survey, pattern recognition letters. Elsevier, Oxford
38. Panwar M, Dyuthi SR, Prakash KC, Biswas D, Acharyya A, Maharatna K, Gautam A, Naik GR (2017) Cnn based approach for activity recognition using a wrist-worn accelerometer. In: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 2438–2441 . IEEE
39. Balaha HM, Ali HA, Saraya M, Badawy M (2021) A new arabic handwritten character recognition deep learning system (ahcr-dls). *Neural Comput Appl* 33:6325–6367
40. Sainath TN, Vinyals O, Senior A, Sak H (2015) Convolutional, long short-term memory, fully connected deep neural networks. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4580–4584 . IEEE
41. Reining C, Niemann F, Moya Rueda F, Fink GA, ten Hompel M (2019) Human activity recognition for production and logistics—a systematic literature review. *Information* 10(8):245
42. Anguita D, Ghio A, Oneto L, Parra X, Reyes-Ortiz JL (2012) Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In: International Workshop on Ambient Assisted Living, pp. 216–223 . Springer
43. Jiang W, Yin Z (2015) Human activity recognition using wearable sensors by deep convolutional neural networks. In: Proceedings of the 23rd ACM International Conference on Multimedia, pp. 1307–1310
44. Shi J, Zuo D, Zhang Z (2020) Transition activity recognition system based on standard deviation trend analysis. *Sensors* 20(11):3117
45. Garcia-Gonzalez D, Rivero D, Fernandez-Blanco E, Luaces MR (2020) A public domain dataset for real-life human activity recognition using smartphone sensors. *Sensors* 20(8):2200
46. Ahmed N, Rafiq JI, Islam MR (2020) Enhanced human activity recognition based on smartphone sensor data using hybrid feature selection model. *Sensors* 20(1):317
47. Reyes-Ortiz J-L, Oneto L, Sama A, Parra X, Anguita D (2016) Transition-aware human activity recognition using smartphones. *Neurocomputing* 171:754–767
48. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. *IEEE Trans Signal Process* 45(11):2673–2681
49. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444
50. Chen K, Zhang D, Yao L, Guo B, Yu Z, Liu Y (2021) Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Comput Surv (CSUR)* 54(4):1–40
51. Barut O, Zhou L, Luo Y (2020) Multitask lstm model for human activity recognition and intensity estimation using wearable sensor data. *IEEE Internet Things J* 7(9):8760–8768
52. Wang L, Liu R (2020) Human activity recognition based on wearable sensor using hierarchical deep lstm networks. *Circuits Syst Signal Process* 39(2):837–856
53. Pham C, Nguyen-Thai S, Tran-Quang H, Tran S, Vu H, Tran T-H, Le T-L (2020) Senscapsnet: Deep neural network for non-obtrusive sensing based human activity recognition. *IEEE Access* 8:86934–86946
54. Gil-Martín M, San-Segundo R, Fernandez-Martinez F, Ferreiros-López J (2020) Improving physical activity recognition using a new deep learning architecture and post-processing techniques. *Eng Appl Artif Intell* 92:103679
55. Xu C, Chai D, He J, Zhang X, Duan S (2019) Innohar: a deep neural network for complex human activity recognition. *Ieee Access* 7:9893–9902
56. Kwapisz JR, Weiss GM, Moore SA (2011) Activity recognition using cell phone accelerometers. *ACM SigKDD Explor Newsletter* 12(2):74–82
57. Chavarriaga R, Sagha H, Calatroni A, Digumarti S.T, Tröster G, Millán J.d.R, Roggen D (2013) The opportunity challenge: a benchmark database for on-body sensor-based activity recognition. *Pattern Recognit Lett* 34(15):2033–2042
58. Teng Q, Wang K, Zhang L, He J (2020) The layer-wise training convolutional neural networks using local loss for sensor-based human activity recognition. *IEEE Sens J* 20(13):7265–7274
59. Roggen D, Calatroni A, Rossi M, Hollecsek T, Förster K, Tröster G, Lukowicz P, Bannach D, Pirkl G, Ferscha A, et al (2010) Collecting complex activity datasets in highly rich networked sensor environments. In: 2010 Seventh International Conference on Networked Sensing Systems (INSS), pp. 233–240. IEEE
60. Micucci D, Mobilio M, Napolitano P (2017) Unimib shar: a dataset for human activity recognition using acceleration data from smartphones. *Appl Sci* 7(10):1101
61. Reiss A, Stricker D (2012) Introducing a new benchmarked dataset for activity monitoring. In: 2012 16th International Symposium on Wearable Computers, pp. 108–109 . IEEE
62. Ronao CA, Cho S-B (2016) Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Syst Appl* 59:235–244
63. Lee Y-S, Cho S-B (2011) Activity recognition using hierarchical hidden markov models on a smartphone with 3d accelerometer. In: International Conference on Hybrid Artificial Intelligence Systems, pp. 460–467 . Springer
64. Bianchi V, Bassoli M, Lombardo G, Fornacciari P, Mordonini M, De Munari I (2019) Iot wearable sensor and deep learning: An integrated approach for personalized human activity recognition in a smart home environment. *IEEE Internet Things J* 6(5):8553–8562
65. Lawal IA, Bano S (2020) Deep human activity recognition with localisation of wearable sensors. *IEEE Access* 8:155060–155070

66. Ordóñez FJ, Roggen D (2016) Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* 16(1):115
67. Zappi P, Lombriser C, Stiefmeier T, Farella E, Roggen D, Benini L, Tröster G (2008) Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection. In: *European Conference on Wireless Sensor Networks*, pp. 17–33. Springer
68. Xia K, Huang J, Wang H (2020) Lstm-cnn architecture for human activity recognition. *IEEE Access* 8:56855–56866
69. Ignatov A (2018) Real-time human activity recognition from accelerometer data using convolutional neural networks. *Appl Soft Comput* 62:915–922
70. Anguita D, Ghio A, Oneto L, Parra X, Reyes-Ortiz JL et al (2013) A public domain dataset for human activity recognition using smartphones. *Esann* 3:3
71. Khan ZN, Ahmad J (2021) Attention induced multi-head convolutional neural network for human activity recognition. *Appl Soft Comput* 110:107671
72. Twomey N, Diethel T, Kull M, Song H, Camplani M, Hannuna S, Fafoutis X, Zhu N, Woznowski P, Flach P, et al (2016) The sphere challenge: Activity recognition with multimodal sensor data. *arXiv preprint arXiv:1603.00797*
73. Grzeszick R, Lenk JM, Rueda FM, Fink GA, Feldhorst S, ten Hompel M (2017) Deep neural network based human activity recognition for the order picking process. In: *Proceedings of the 4th International Workshop on Sensor-based Activity Recognition and Interaction*, pp. 1–6
74. Feldhorst S, Masoudenijad M, ten Hompel M, Fink GA (2016) Motion classification for analyzing the order picking process using mobile sensors. In: *Proc. Int. Conf. Pattern Recognition Applications and Methods*, pp. 706–713
75. Sun Y, Wong AK, Kamel MS (2009) Classification of imbalanced data: a review. *Int J Pattern Recognit Artif Intell* 23(04):687–719
76. He H, Garcia EA (2009) Learning from imbalanced data. *IEEE Trans Knowl Data Eng* 21(9):1263–1284
77. Branco P, Torgo L, Ribeiro R (2015) A survey of predictive modelling under imbalanced distributions. *arXiv preprint arXiv:1505.01658*
78. He H, Ma Y (2013) Imbalanced learning: foundations, algorithms, and applications
79. Tomek I (1976) A generalization of the k-nn rule. *IEEE Trans Syst Man Cybern* 2:121–126
80. Lemaître G, Nogueira F, Aridas CK (2017) Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *J Mach Learn Res* 18(1):559–563
81. Yap BW, Abd Rani K, Abd Rahman HA, Fong S, Khairudin Z, Abdullah NN (2014) An application of oversampling, under-sampling, bagging and boosting in handling imbalanced datasets. In: *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, pp. 13–22. Springer
82. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. *J Artif Intell Res* 16:321–357
83. Stanfill C, Waltz D (1986) Toward memory-based reasoning. *Commun ACM* 29(12):1213–1228
84. Cost S, Salzberg S (1993) A weighted nearest neighbor algorithm for learning with symbolic features. *Mach Learn* 10(1):57–78
85. Han H, Wang W-Y, Mao B-H (2005) Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: *International Conference on Intelligent Computing*, pp. 878–887. Springer
86. He H, Bai Y, Garcia EA, Li S (2008) Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1322–1328. IEEE
87. Last F, Douzas G, Bacao F (2017) Oversampling for imbalanced learning based on k-means and smote. *arXiv preprint arXiv:1711.00837*
88. Nguyen HM, Cooper EW, Kamei K (2011) Borderline over-sampling for imbalanced data classification. *Int J Knowl Eng Soft Data Paradig* 3(1):4–21
89. Yan W, Yu L (2019) On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach. *arXiv preprint arXiv:1908.09238*
90. Domingos P (2012) A few useful things to know about machine learning. *Commun ACM* 55(10):78–87
91. Brownlee J (2014) Discover feature engineering, how to engineer features and how to get good at it. *Machine Learning Process*
92. Balaha HM, Ali HA, Youssef EK, Elsayed AE, Samak RA, Abdelhaleem MS, Tolba MM, Shehata MR, Mahmoud MR, Abdelhameed MM et al (2021) Recognizing arabic handwritten characters using deep learning and genetic algorithms. *Multimed Tools Appl* 80:32473–32509
93. Chatzimparmpas A, Martins RM, Kucher K, Kerren A (2022) Featureenvi: Visual analytics for feature engineering using stepwise selection and semi-automatic extraction approaches. *IEEE Trans Visual Comput Graphic* 28(4):1773–1791
94. Heaton J (2016) An empirical analysis of feature engineering for predictive modeling. In: *SoutheastCon 2016*, pp. 1–6. IEEE
95. Liu H, Motoda H (1998) Feature transformation and subset selection. *IEEE Intell Syst Their Appl* 13(2):26–28
96. Patel K, Fogarty J, Landay JA, Harrison B (2008) Investigating statistical machine learning as a tool for software development. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 667–676
97. Markovitch S, Rosenstein D (2002) Feature generation using general constructor functions. *Machine Learn* 49(1):59–98
98. Schuller B, Reiter S, Rigoll G (2006) Evolutionary feature generation in speech emotion recognition. In: *2006 IEEE International Conference on Multimedia and Expo*, pp. 5–8. IEEE
99. Kankanige Y, Bailey J (2014) Improved feature transformations for classification using density estimation. In: *Pacific Rim International Conference on Artificial Intelligence*, pp. 117–129. Springer
100. Storcheus D, Rostamizadeh A, Kumar S (2015) A survey of modern questions and challenges in feature extraction. In: *Feature Extraction: Modern Questions and Challenges*, pp. 1–18. PMLR
101. Molina LC, Belanche L, Nebot À (2002) Feature selection algorithms: a survey and experimental evaluation. In: *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pp. 306–313. IEEE
102. Chandrashekar G, Sahin F (2014) A survey on feature selection methods. *Comput Electr Eng* 40(1):16–28
103. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. *J Mach Learn Res* 3(Mar):1157–1182
104. Khalid S, Khalil T, Nasreen S (2014) A survey of feature selection and feature extraction techniques in machine learning. In: *2014 Science and Information Conference*, pp. 372–378. IEEE
105. Tharwat A (2016) Principal component analysis-a tutorial. *Int J Appl Pattern Recognit* 3(3):197–240
106. Fisher RA (1938) The statistical utilization of multiple measurements. *Ann Eugen* 8(4):376–386

107. Fukunaga K (2013) Introduction to Statistical Pattern Recognition. Elsevier, Newyork
108. Tang B, Shepherd M, Milios E, Heywood MI (2005) Comparing and combining dimension reduction techniques for efficient text clustering. In: Proceeding of SIAM International Workshop on Feature Selection for Data Mining, pp. 17–26
109. Dasgupta S (2000) Experiments with random projection. Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference (UAI-2000)(pp. 143–151). Morgan Kaufmann
110. Halko N, Martinsson P-G, Tropp JA (2011) Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev* 53(2):217–288
111. Euler L (1741) *Solutio problematis ad geometriam situs pertinentis*. *Commentarii academiae scientiarum Petropolitanae*, 128–140
112. Carlsson G (2009) Topology and data. *Bull Am Math Soc* 46(2):255–308
113. Chazal F (2016) High-dimensional topological data analysis. CRC Press
114. Rosenstock S (2021) Learning from the shape of data. *Philos Sci* 88(5):1033–1044
115. Edelsbrunner H, Letscher D, Zomorodian A (2000) Topological persistence and simplification. In: Proceedings 41st Annual Symposium on Foundations of Computer Science, pp. 454–463 . IEEE
116. Zomorodian A, Carlsson G (2005) Computing persistent homology. *Discret Compu Geom* 33(2):249–274
117. Singh G, Mémoli F, Carlsson GE et al (2007) Topological methods for the analysis of high dimensional data sets and 3d object recognition. *PBG @ Eurograp* 2:91–100
118. Li L, Cheng W-Y, Glicksberg BS, Gottesman O, Tamler R, Chen R, Bottinger EP, Dudley JT (2015) Identification of type 2 diabetes subgroups through topological analysis of patient similarity. *Sci Trans Med* 7(311):311–174311174
119. Nicolau M, Levine AJ, Carlsson G (2011) Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proc Natl Acad Sci* 108(17):7265–7270
120. Balaha HM, Balaha MH, Ali HA (2021) Hybrid covid-19 segmentation and recognition framework (hmb-hcf) using deep learning and genetic algorithms. *Artif Intell Med* 119:102156
121. Ahsan MM, Mahmud M, Saha PK, Gupta KD, Siddique Z (2021) Effect of data scaling methods on machine learning algorithms and model performance. *Technologies* 9(3):52
122. Smaczny C (2018) Feature preprocessing in hep at the example of a susy classification problem
123. Shaheen H, Agarwal S, Ranjan P (2020) Minmaxscaler binary pso for feature selection. In: First International Conference on Sustainable Technologies for Computational Intelligence, pp. 705–716 . Springer
124. Abosamra G, Faloudah A (2019) Machine learning based marks prediction to support recommendation of optimum specialization and study track. *Int J Comput Appl* 975:8887
125. Iglewicz B (1983) Robust scale estimators and confidence intervals for location. *Understanding robust and exploratory data analysis*, 405431
126. Hu J, Niu H, Carrasco J, Lennox B, Arvin F (2020) Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. *IEEE Trans Veh Technol* 69(12):14413–14423
127. Ribeiro M, Grolinger K, Capretz MA (2015) Mlaas: Machine learning as a service. In: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp. 896–902 . IEEE
128. Marr B (2018) What is deep learning ai? a simple guide with 8 practical examples. *Forbes*. Retrieved October 9, 2021
129. Baghdadi NA, Malki A, Abdelaliem SF, Balaha HM, Badawy M, Elhosseini M (2022) An automated diagnosis and classification of covid-19 from chest ct images using a transfer learning-based convolutional neural network. *Comput Biol Med* 144:105383
130. Sejnowski TJ (2018) *The Deep Learning Revolution*. MIT press, Cambridge
131. Balaha HM, Saif M, Tamer A, Abdelhay EH (2022) Hybrid deep learning and genetic algorithms approach (hmb-dlgaha) for the early ultrasound diagnoses of breast cancer. *Neural Comput Appl* 34(11):8671–8695
132. Baghdadi NA, Malki A, Balaha HM, Badawy M, Elhosseini M (2022) A3c-tl-gto: Alzheimer automatic accurate classification using transfer learning and artificial gorilla troops optimizer. *Sensors* 22(11):4250
133. Straczekiewicz M, James P, Onnela J-P (2021) A systematic review of smartphone-based human activity recognition methods for health research. *NPJ Dig Med* 4(1):1–15
134. Polikar R (2012) Ensemble learning. In: *Ensemble Machine Learning*, pp. 1–34. Springer
135. Brownlee J (2020) Gradient boosting with scikit-learn, xgboost, lightgbm, and catboost. *Machine Learning Mastery*
136. Kopitar L, Kocbek P, Cilar L, Sheikh A, Stiglic G (2020) Early detection of type 2 diabetes mellitus using machine learning-based prediction models. *Sci Rep* 10(1):1–12
137. Quinlan JR (1987) Simplifying decision trees. *Int J Man-mach Stud* 27(3):221–234
138. Rawat R, Mahor V, Chirgaiya S, Shaw RN, Ghosh A (2021) Analysis of darknet traffic for criminal activities detection using tf-idf and light gradient boosted machine learning algorithm. In: *Innovations in Electrical and Electronic Engineering*, pp. 671–681. Springer
139. Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, pp. 785–794
140. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci* 55(1):119–139
141. Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
142. Ho TK (1995) Random decision forests. In: *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, pp. 278–282 . IEEE
143. Ho TK (1998) The random subspace method for constructing decision forests. *IEEE Trans Pattern Anal Mach Intell* 20(8):832–844
144. Breiman L, Friedman J, Olshen R, Stone C (1984) *Classification and regression trees (wadsworth, belmont, ca)*. ISBN-13, 978-0412048418
145. Hastie T, Tibshirani R, Friedman J (2009) *The elements of statistical learnin*. Cited on, 33
146. Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. *Mach Learn* 63(1):3–42
147. Brownlee J (2018) 1D convolutional neural network models for human activity recognition
148. Kiranyaz S, Avci O, Abdeljaber O, Ince T, Gabbouj M, Inman DJ (2021) 1d convolutional neural networks and applications: a survey. *Mech Syst signal Process* 151:107398
149. Cho K, Van Merriënboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*
150. Gers FA, Schmidhuber J, Cummins F (2000) Learning to forget: continual prediction with lstm. *Neural Comput* 12(10):2451–2471

151. Britz D (2015) Recurrent neural network tutorial, part 4 implementing a gru/lstm rnn with python and theano. URL <http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano>
152. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint [arXiv:1412.3555](https://arxiv.org/abs/1412.3555)
153. Zhao C, Huang X, Li Y, Yousaf Iqbal M (2020) A double-channel hybrid deep neural network based on cnn and bilstm for remaining useful life prediction. *Sensors* 20(24):7109
154. Refaeilzadeh P, Tang L, Liu H (2009) Cross-validation. *EEncycl Database Syst* 5:532–538
155. Sharaby I, Alksas A, Nashat A, Balaha HM, Shehata M, Gayhart M, Mahmoud A, Ghazal M, Khalil A, Abouelkheir RT et al (2023) Prediction of wilms' tumor susceptibility to preoperative chemotherapy using a novel computer-aided prediction system. *Diagnostics* 13(3):486
156. Bahgat WM, Balaha HM, AbdulAzeem Y, Badawy MM (2021) An optimized transfer learning-based approach for automatic diagnosis of covid-19 from chest x-ray images. *PeerJ Comput Sci* 7:555
157. Balaha HM, El-Gendy EM, Saafan MM (2021) Covh2sd: A covid-19 detection approach based on harris hawks optimization and stacked deep learning. *Expert Syst Appl* 186:115805
158. Kapp MN, Sabourin R, Maupin P (2012) A dynamic model selection strategy for support vector machine classifiers. *Appl Soft Comput* 12(8):2550–2565
159. MONTGOMERY D.C (2013) Design and analysis of experiments, eight ed., a john wiley & sons. Inc
160. Elgendy M, Balaha HM, Shehata M, Alksas A, Ghoneim M, Sherif F, Mahmoud A, Elgarayhi A, Taher F, Sallah M et al (2022) Role of imaging and ai in the evaluation of covid-19 infection: a comprehensive survey. *Front biosci (Landmark edition)* 27(9):276

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.