A New Time-Series Classification Approach for Human Activity Recognition with Data Augmentation

Youssef Errafik, Younes Dhassi, Adil Kenzi
Laboratoire LISA (Laboratoire d'Ingénierie, Systèmes et Applications), ENSAF,
Sidi Mohamed Ben Abdellah University, Fez, Morocco[1, 2]
Laboratoire LSIA (Laboratoire des Systèmes intelligents & Applications),
FST, Sidi Mohamed Ben Abdellah University , Fez, Morocco[3]

***Abstract*—Accurate classification of multivariate time series data represents a major challenge for scientists and practitioners exploring time series data in different domains. LSTM-Auto-encoders are Deep Learning models that aim to represent input data efficiently while minimizing information loss during the reconstruction phase. Although they are commonly used for Dimensionality Reduction and Data Augmentation, their potential in extracting dynamic features and temporal patterns for temporal data classification is not fully exploited in contrast to the tasks of time-series prediction and anomaly detection. In this article, we present a multi-level hybrid TSC-LSTM-Auto-Encoder architecture that takes full advantage of the incorporation of temporal labels to capture comprehensively temporal features and patterns. This approach aims to improve the performance of temporal data classification using this additional information. We evaluated the proposed architecture for Human activity Recognition (HAR) using the UCI-HAR and WISDM public benchmark datasets. The achieved performance outperforms the current state-of-the-art methods.**

***Keywords*—Deep Learning (DL); multivariate time series; Time Series Classification (TSC); Human Activity Recognition (HAR)**

I. Introduction

Time series classification (TSC) [1] holds crucial significance in machine learning, providing various advantages and significant applications. The main objective of TSC is to assigning a category or class to a time series based on its temporal characteristics and dynamic. In terms of processed data, each time series constitutes a sequence of temporal data, such as univariate or multivariate measurements and values, captured and recorded at regular intervals. TSC currently occupies a central place in many applications spread across various fields: It is used in healthcare to classify medical signals such as electrocardiograms (ECG) [2], electroencephalograms (EEG) [3], and other medical monitoring data [4]. For the financial sector, it makes it possible to understand the dynamic developments of financial markets, in particular the classification of shares [5]. In industry, it is used to analyze sensor data for machine monitoring [6], process planning [7], and production quality control. In security, it helps protect computer systems against advanced cyber-attacks [8] through malware detection [9]. In the environmental field, it facilitates climate change detection and causal inference analysis [10] in climate science, using advanced TSC techniques. Finally, the field of recognition of human activity from sensors presents a classification challenge based on the analysis of time series, in particular those coming from accelerometers and gyroscopes. This field encompasses a variety of broad applications in industries such as healthcare, personal monitoring, security, physical performance tracking, life data recording, elderly care, and home care. Due to its critical importance, paving the way for significant progress in understanding and improving the human way of life at multiple levels, we deliberately chose this area to experiment and analyze our DL model.

Despite the critical importance of implementing temporal data classification, this area faces several significant challenges. Class imbalance is one such concern, with the potential to introduce bias into classification models. Extracting relevant dynamic features from temporal data is crucial for accurate classification, but it represents an extremely complex task. In addition, dealing with missing values in time series requires the adoption of suitable strategies. Additionally, detecting relevant temporal patterns can be difficult, especially when these patterns are subtle with noisy data. To overcome these challenges, the scientific community is exploring various avenues of research, covering both traditional machine learning methods and DL approaches, whether supervised or unsupervised. Currently, the use of generative model-based approaches such as GANs and auto encoders remains limited when it comes to temporal data classification. These methods are mainly reserved for data augmentation and dimensionality reduction. To use directly generative models in the classification of temporal data, we have developed a new multi-step approach. This method aims to take advantage of the power of generative models, in particular those of the auto encoder (AE) type, to extract dynamic and temporal characteristics. The goal is to improve the classification of time series data. Our work mainly focuses, in the first phase, on the integration of a digital time label for each class of human activity once determined.

We integrate this label into the raw data as input for our generative model. Subsequently, we train the model to enhance its ability to reproduce this information accurately. This approach is designed to acquire the skill of extracting dynamic, and temporal characteristics, allowing it to generate this label even when it is absent during testing.

In the second phase, we perform classification using the labels generated in the preceding phase, following a supervised approach guided by a previously validated LSTM model. The following points can mention the main contribution of this paper:

- We conduct a general review of the existing literature on DL-based TSC, providing readers with valuable insights to understand and contrast the trend axes in this area.
- The creation of representative labels and identifiers per class in the form of a coherent temporal variable can effectively capture dynamic and temporal relational characteristics and dependencies.
- Conduct a series of comparative experiments in order to highlight the promising performances of our model compared to other well-established models in the field of temporal data classification, particularly in the context of HAR, such as UCI-HAR, and WISDM.

The remainder of the paper is organized as follows: Section II discusses related work, focusing on methods, and architectures machine learning in temporal data classification and sensor-based HAR. Section III details the proposed method including the process followed for all stages, the architecture adapted in three stages carried out. Section IV presents the experimental metrics and results, as well as details regarding the datasets used. By analyzing the experimental results obtained in Section V of discussion. Finally, Section VI concludes our article.

II. Related Work

Studies on TSC have a rich history, marked by numerous proposals for classification approaches over time. We can distinguish two main axes of these TSC methods, namely Traditional Machine Learning Based Methods, and DL-Based models.

A. *Traditional Machine Learning-based Methods*

Traditional ML-based methods deployed for TSC typically involve extracting meaningful features from temporal data; these methods rely on the application of standard classification techniques, including feature engineering, statistical methods and metrics measuring the similarity between time series data. Traditional ML-based methods are generally classified into two distinct groups [11]: On one hand, distance-based approaches [12] involve the utilization of classifiers based on distance measurements between different time series. These methods concentrate on quantifying the similarity between two given time series, employing specific metrics for classification, such as k-nearest neighbors (KNN) [13] or support vector machines (SVM) [14] with similarity-based kernels. Some studies even hybridize them with hidden Markov models (HMM) [15], as demonstrated by the research [16], which integrates HMM and SVM models for the early classification of multivariate time series. Furthermore, Notable similarity measures include dynamic time warping (DTW) [17], which aligns two time series with dynamic deformation to obtain the best fit a process easily implemented using dynamic programming. Furthermore, in this [18], the Time Series Forest (TSF) method is employed as a tree ensemble approach to increase the accuracy of TSC; it achieves enhanced precision by combining entropy gain with a distance measure for evaluating divisions. TSF stands out for its distinctive feature of randomly sampling features at each tree node, resulting in linear computational complexity relative to time series length and facilitating parallel computing. The proposal of a temporal importance curve aims to capture relevant temporal features for classification. Experimental results demonstrate that TSF, utilizing basic features such as mean, standard deviation, and slope, not only excels in computational efficiency but also outperforms KNN classifiers employing DTW.

On the other hand, the feature-based approach [19] involves the extraction and selection of deterministic features in the data, which optimize the classification algorithms [20].

Hence, they include a diversity of TSC strategies. The feature extraction aspect relies on the use of a restricted set of features with a solid and easily interpretable theoretical basis. In addition to applying traditional learning algorithms, this approach offers the possibility of analyzing the extracted parameters to obtain additional information. Feature-based approach are divided into three categories:

Firstly, statistical methods focus on using a set of certain statistical characteristics [21], such as the mean, the standard deviation, the skewness to assess the asymmetry of the values compared to the mean, and the kurtosis to measure the relative flatness of the distribution values relative to a normal distribution. These parameters are mobilized in order to resolve the challenges related to statistical process control models intended for classification. Secondly, transformation-based methods [22] aim to improve classification performance through the transformation of data to another alternative data space where discriminatory features are more easily determined, This recent survey in [23] give many of several examples. By adding, the work in [24] proposes a shapelet transformation; this method makes it possible to extract the best shapelets (a subsequence of time series identifying membership in a class) from a dataset to improve the overall accuracy of the classification. Finally, TSC approaches based on the fusion of various characteristics [25] to significantly improve the accuracy of TSC. In this perspective, we mention multi-dimensional [26], multi-channel [27] fusion methods of characteristics and data, as well as network fusion techniques [28], and adaptive feature fusion [29] to improve the accuracy of TSC.

B. *Deep Learning-Based Models*

In the domain of DL and the application of deep neural networks [1, 30] for TSC, the processes of feature extraction and classification are automated, eliminating the necessity for expert intervention. Four key axes emerge from this perspective for modeling and solving intricate classification tasks. First is the utilization of Convolutional Neural Networks (CNN), tailored to the sequential nature of temporal data. While CNNs are renowned for their proficiency in recognizing spatial patterns in two-dimensional data, such as images, their application to time series necessitates adaptation to exploit sequential structures and temporal relationships [31, 32]. In this context, models like the Multi-Scale Convolutional Neural Network (MCNN) [33] illustrate the concept of applying diverse transformations, such as varying scales and frequencies, to temporal data. This approach enables the model to capture features at multiple levels and scales, enhancing its ability to represent complex temporal patterns. Furthermore, another MCNN for TSC in this work [34] dynamically extracts multi-scale feature representations from time series to classify them.

Secondly, recurrent models, including Recurrent Neural Networks (RNN) and architectures based on Long-Short-Term Memory (LSTM), play a pivotal role. These models are designed to process sequences of data, rendering them particularly suitable for time series analysis. RNNs [35] can memorize previous contextual information, while LSTMs and GRU [36] overcome the limitations of RNNs in classification by avoiding gradient vanishing problems. Our expertise lies in effectively combining these two components [37], demonstrating significant performance improvements in TSC.

Thirdly, the convergence of the aforementioned axes has led to the development of hybrid models such as Recurrent Convolutional Neural Networks (CRNN) [38], merging convolutional layers to capture spatial patterns with recurrent layers to handle temporal dependencies. This amalgamation leverages the strengths of both approaches, enhancing the model's capacity to extract spatial and temporal characteristics. Hybrid models, combining the advantages of CNNs for spatial pattern recognition and LSTMs for modeling long-term sequential dependencies, are increasingly gaining popularity and proving effective in TSC applications such as musical classification [39] and electrocardiogram classification [40]. Several comparative studies [41, 42] demonstrate that this combination constitutes a robust solution to overcome the complexity of time series in terms of classification, notwithstanding their intricate and heterogeneous nature.

Finally and recently, several studies have looked into the application of generative models such as auto encoders for TSC: Research [43] proposed a representation 2D of time series by fusing temporal and frequency features using the AE model, in order to construct a classification network. Another research [44] optimized an LSTM based network layer design to create an AE improving ECG signal classification, eliminating the need for complex preprocessing. A method based on a Conditional Variational AE proposed [45] to solve the no distribution problem related to identifying feature importance for time series classifiers. Auto encoders based on RNN have also been used for TSC [46], where different variants of RNN auto encoders were compared for their performance in feature extraction. An automated label generation method for TSC using representation learning with AECS and VAE [47] demonstrated promising results in reducing labeling costs for training. This method synthetically boosts representative time series using VAE, showing performance close to supervised classification and even exceeding baseline performance in some cases. Finally, the authors of the article [48] introduce a VAE model based on an RNN network, incorporating a constrained loss function. This model is designed to generate more meaningful EEG features from raw data, with the goal of enhancing the performance of classification in speech recognition. The approach aims to leverage the inherent independence features within the latent representation of VAE to improve TSC performance.

Having carefully reviewed the state of the art of various research directions focused on developing approaches for TSC in this section, our approach in this work will be to leverage multiple methods. We plan to hybridize these concepts to create a robust multi-stage method that will take advantage of the many advantages offered by this integrative approach. The specific details of our method will be explained in the following section, highlighting our innovative contribution to the advancement of TSC techniques.

III. Proposed Method

In this section, we will describe a multi-stage DL framework for multivariate TSC (see Fig. 1). The general organization of our proposed model, TSC-LSTM-AE is divided into three distinct components: The first component concerns the pre-training phase, which encompasses data augmentation and creation of identification variables (labels) per class. The second component is characterized by the use of the "Auto-Encoder" model, designed to extract dynamic temporal and relational features. Finally, the third component exploits the "Classifiers" model, dedicated to the classification task using the raw data and the features extracted during the previous phase.

Our method incorporates an additional variable designed to capture temporal and dynamic aspects of sequential data, thereby improving feature extraction. Once this identification variable (label) is determined for each data class, it is merged with the initial sequential data according to their respective classes. Then, our Auto-Encoder model is trained to learn to reconstruct the initial data combined with the injected data, thus generating this label from the raw data, even in the absence of this label as input. In this way, we are able to generate an identification label rich in meaningful characteristics, which we use in the classification phase through a classifier to guarantee the improvement of this task. We combine this classifier with other classifiers in the classifier model, such as LSTM, GRU and CNN. Ultimately, aggregating the ratings helps determine the final output class more accurately. Next, we will examine each phase in detail, presenting its importance, its objectives, and its associated steps and methods.

A. *Pre-Processing Phase*

The initial phase of our method consists of preprocessing the time series data, comprising four distinct steps:

1) Synchronized windowing method: To ensure the alignment of variables and prevent any temporal overlap between the samples, we implemented a unified windowing method specifically designed for this purpose, applicable to all classes of activities. Our inspiration for this method comes from an approach used to estimate an optimal time series mean [49], employing a multi-task auto-encoder architecture for the estimation. Due to the intricacy of this approach, we have limited the variables in this work to three accelerometers. The methodology involves calculating the average of these three variables for each sample. Subsequently, we extract 100, 80, and 60 successive variables, respectively, from the maximum value of this average. The objective of this method temporally synchronizes the data, considerably reducing the gap between the variables of the two samples, two by two.

In the next section, we will use the t-SNE method to approve our choice of the appropriate window size.

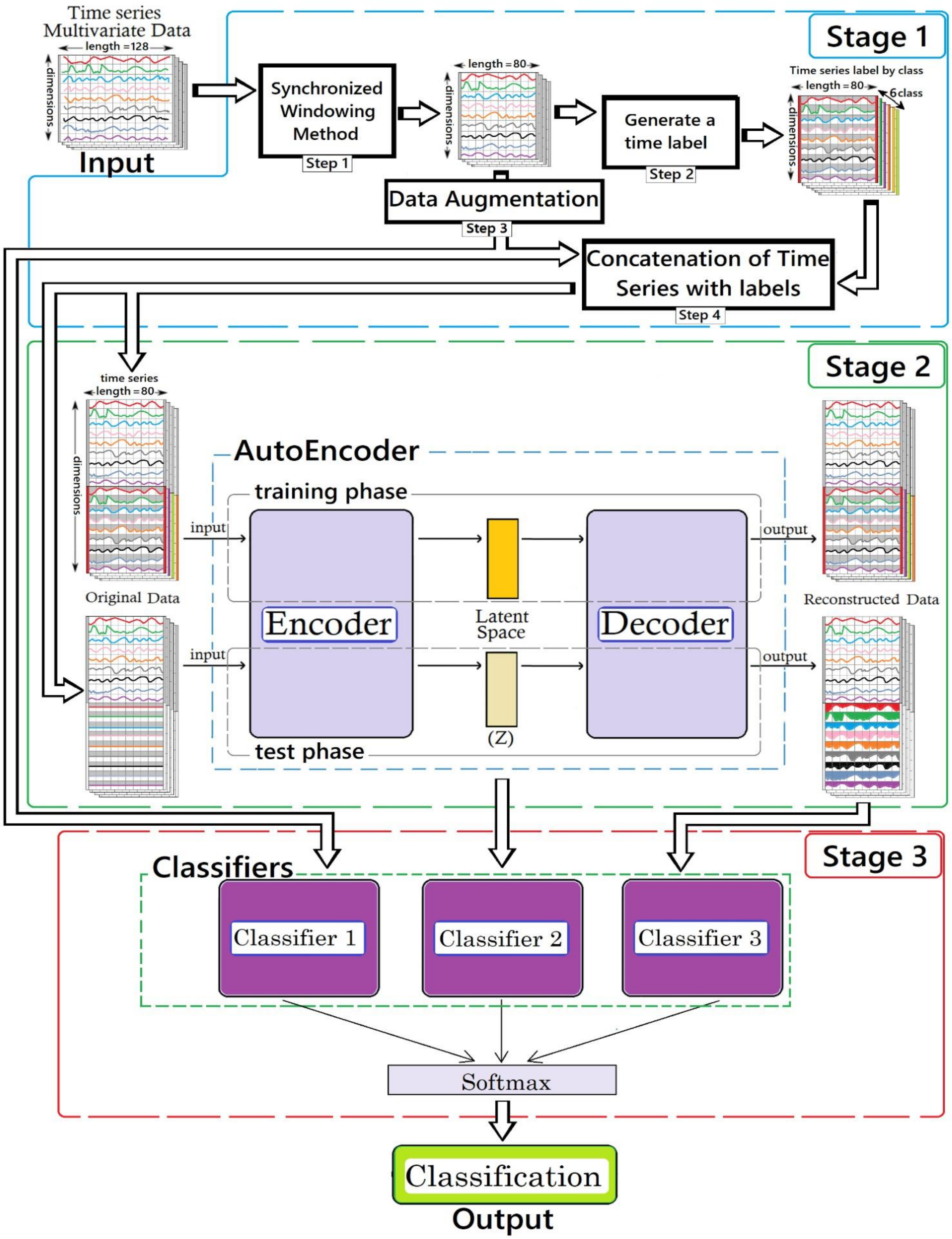

Fig. 1. Schema general of TSC-LSTM-AE.

2) Generate a time label by class: After the temporal alignment of all the time series data and the cutting of its data to length 80, the second step aims to create a variable λt | $Ck = (\lambda t1,\ \lambda t2,\ \ .\ .\ \ ,\ \lambda tn)$ for each class k examined for classification. In order to have a variable capable of simultaneously capturing the dynamic temporal aspects of sequential data and correctly identifying each class studied, it is essential that it is consistent with temporal data belonging to the same class k. Thus, it can function as a unified time label for data of the same class, moreover, in order to minimize as much as possible the negative impact on the relational dependencies between the different characteristics, we chose average arithmetic as the means of calculation for this temporal label per class. In Fig. 2 below, we display the calculated time for the "Walking" class.

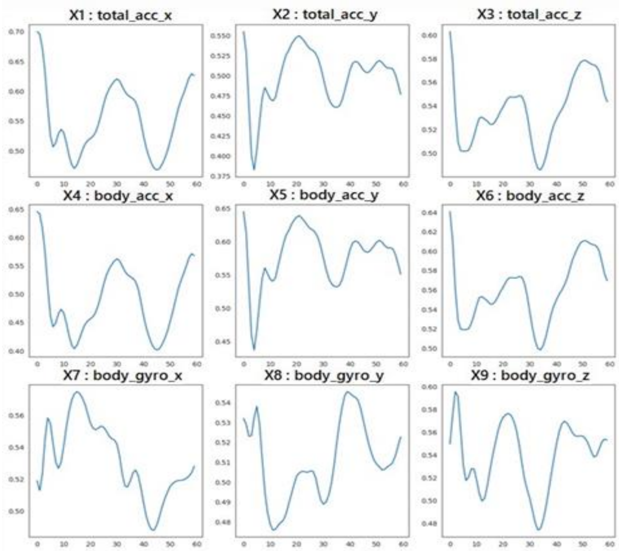

Fig. 2. The time label for the "WALKING" activity generated.

3) Data augmentation: Generative GAN models integrate temporal dependencies and spatial relationships to generate long, high-fidelity time series. Their demonstrated capability to surpass standards in terms of fidelity, diversity, and predictive performance is noteworthy. Time series GANs have achieved considerable success in diverse fields, including finance, healthcare, data imputation, and anomaly detection. In this step, with a focus on augmenting the quantity of data utilized for training DL models, we have opted to adopt and customize the TimeGAN [50] generative method.

4) Concatenation of time series with labels: In order to prepare the temporal data for the second phase. We proceeded to concatenate the raw data with its previously generated temporal labels. The essence of this approach lies in scrupulous respect for the belonging of this data to each respective class. In other words, each time series was enriched by adding its own temporal labels, thus creating a precise link between the raw data and their specific temporal context. This concatenation operation aims to enrich the dependency relationships between the raw data and the joint temporal label, thus facilitating the extraction of temporal features and adequately preparing the data for the next phase of the process.

B. Feature Extraction Phase

Having successfully completed the initial data pre-processing phase, we move on to the second stage of the process. This step aims to leverage the AE approach to extract the hidden temporal and dynamic characteristics within the sequential data. Additionally, a novelty is introduced this time: the inclusion of an additional specialized variable for classification. This phase is subdivided into two fundamental steps:

The first step is dedicated to the training of an LSTM type AE model. The goal here is to establish a model to excel in extracting features from time series data, also this model is trained to learn the complex dependency relationships between raw data and associated labels. This helps enrich the internal representation of the model by more precisely capturing the nuances and temporal structures present in the data, thus facilitating the next step. Our recurrent AE consists of three layers of encoding and decoding; we used LSTM layers to capture temporal sequences. The objective is to minimize the MSE loss between the input and output data to learn a compact representation of the input temporal sequences. The first coding layer is an LSTM layer composed of 64 units, designed to process Input data as a sequence of 80 steps. Then, a second LSTM coding layer, consisting of 32 units, is added, followed by a third LSTM coding layer with 16 units. This last layer generates a latent space (Z) of dimension 16 as output. The decoding phase begins with a 'Repeat-Vector' layer, which replicates the previous output to obtain a sequence of the same length as the input. This creates a link between the encoding and decoding stages of the model. The three layers of the decoder are also LSTM type, comprising 16, 32 and 64 units respectively. Finally, the model ends with a Time Distributed layer wrapping a dense layer of four units. This configuration allows the application of a dense operation at each time step of the output sequence.

After training our auto encoder, the second step of this phase focuses on testing the AE model. Our main contribution lies in using the AE model to extract features to reconstruct the data. However, this time we leverage these capabilities to generate our injected variable in the absence of it at the input. In order to capture the extracted features in the temporal labels, we eliminate the label values at the input by replacing them with zero values. Then, we recover at the output the reconstructed label with dynamic and temporal characteristics. These characteristics used to identify and recognize the class of membership in the next phase.

C. Classification Phase

In the final phase of our approach, we will develop our classifier using three different classification models:

Classifier 1: The first classifier adopted is a hybrid CNN-LSTM model. This model initially consists of a convolutional layer with filters of size 128. Continuing with this, we have five successive layers of LSTM, each having a size equal to 64 units. To regularize the model, a dropout layer with a rate of 30% is introduced after the LSTM layers. Then, three fully connected (FC) layers follow, with respective sizes of 100, 32, and 10. This hybrid model allows capturing complex patterns both spatial and temporal in the raw input data. The convolutional layer acts as an initial feature extractor, while the sequential LSTM layers process the temporal sequence, thus preserving important temporal dependencies. The introduction of a dropout layer contributes to the regularization of the model by reducing over fitting; the two fully connected layers at the end of the model allow classification to be carried out by consolidating the information extracted in the previous layers.

The first FC layer reduces the dimensionality, while the second layer produces the outputs corresponding to the target classes. By combining these different layers, our CNN-LSTM classifier is designed to provide a robust representation of the input data, taking advantage of the model's spatial and temporal processing capabilities, while minimizing the risks of over fitting thanks to built-in regularization.

Classifier 2: The second selected classifier adopts a more simplified approach than the first, mainly due to the reduced dimension of the input data, structured in the form of a vector of 16 values. The objective of this classifier is to optimize the overall performance of the model while considering the particular nature of the input data, which represents the latent space of the AE model with the essential features extracted. With an input of dimension 16, the second classifier is more efficient in terms of computational resources, while offering adequate classification capacity for the characteristics contained in this restricted vector. The simplicity of the model also helps reduce the risk of over fitting, which is particularly important when input data is limited. Despite its simplicity, the second classifier is designed to extract discriminant information from the input vector and produce accurate class predictions. The architecture of the model includes a Conv1d convolution layer with 16 Kernels, activation is 'Relu' ,and a single layer of FC neurons of size 32 units adapted to the size of the input, then a Dense layer of 10 neurons.

Classifier 3: The third classifier stands out for its use to classify temporal data through the temporal labels generated during the previous phase. We opted for adopting the same structure as the first classifier in order to maintain consistency in the classification approach. This structural consistency between the first and third classifiers arises for several reasons; they have the same structure and dimensionality of the data at the input and the results at the output. Furthermore, the reuse of the structure of the first classifier demonstrates its robustness and efficiency, which motivated the decision to apply it for the classification of temporal data. By adapting the same architecture, including a convolutional layer followed by five LSTM layers, a dropout layer, and finally two fully connected layers, the third classifier is able to capture the complex temporal features captured by the temporal labels. This structure makes it possible to take advantage of CNN and LSTM mechanisms for better representation of temporal dependencies, thus contributing to accurate classification of time series data.

IV. PERFORMANCE EVALUATION

A. Dataset

1) Here is a brief overview of the standard datasets we used: UCI-HAR dataset [51]: Comes from the public repository "Machine Learning at the University of California, Irvine (UCI)".This dataset was constituted from the activity of 30 participants aged between 19 and 48 years, who carried out several activities of daily life of the following six activities: 'Sitting', 'Standing', 'Walking' , 'Lying' , 'Walking Upstairs', and 'Walking Downstairs'. Data collection was carried out using a waist-mounted Samsung Galaxy smartphone equipped with a built-in accelerometer and gyroscope. The dataset was collected in a laboratory environment under appropriate supervision. In total, this dataset has 10.299 examples. The 3D linear acceleration and angular velocity measurements were recorded at a constant sampling rate of 50 instances per second.

2) Wireless Sensor Data Mining dataset (WISDM) [52]: characterized by imbalance, comprising almost a million samples. The most common activities represent almost 39% of the total, while the least frequent account for almost 4%. Additionally, 36 subjects were selected to perform specific daily tasks, while carrying an Android phone in their front pocket, which made them the participants of the WISDM experiment. A 3D accelerometer operating at a sampling rate of 20 instances per second was used as a sensor, also being a motion sensor integrated into smartphones. The activities recorded in this dataset include six different activities: 'Standing', 'Sitting', 'Walking', 'Upstairs' , 'Downstairs', and 'Jogging'. To ensure complete capture of each activity, the length of each sample was set to 128 consecutive instances, which equates to approximately "seconds. This duration is considered sufficient to adequately represent the activity performed. The data partitioning following activities for the two datasets used illustrate in Fig. 3.

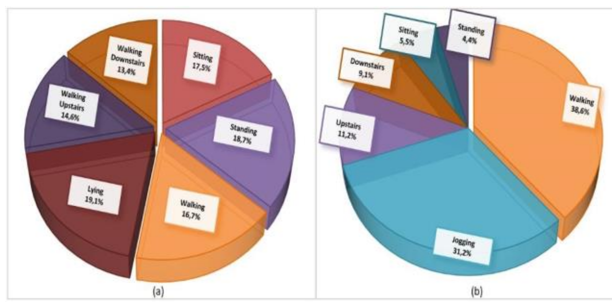

Fig. 3. Percentage of different activities (a) in the UCI-HAR and (b) in the WISDM dataset.

B. Metrics Used

The multiclass classification problem, commonly encountered in the field of TSC in general, and more specifically in HAR using sensors, is generally solved using approaches supervised learning. Each processed data is assigned to one of the classes of human activities and is labeled accordingly. To evaluate the performance of these approaches in our current work, we used the following metrics: Precision (P), F1 score (F1), Recall (R), accuracy, and Confusion Matrix (CM).These metrics are the most commonly used in this research area. The standard equations corresponding to these performance measures are as follows:

$$Precesion = \frac{\sum_{i=1}^{n} Precesion_i}{n} \quad (1)$$

$$Recall = \frac{\sum_{i=1}^{n} Precesion_i}{n} \quad (2)$$

$$F1_{score} = \frac{2 \times Precesion \times Recall}{Precesion + Recall} \quad (3)$$

The multi-class confusion matrix is an extension of the confusion matrix used in the context of multiclass classification problems. Unlike the binary confusion matrix, which is used to evaluate a model's performance in two-class scenarios, the multiclass confusion matrix helps visualize a model's performance when confronted with multiple classes.

For a multi-class classification problem, the confusion matrix is a rectangular table with rows and columns, where each row corresponds to the actual class and each column corresponds to the class predicted by the model. The matrix entries represent the number of observations belonging to a particular class. The main elements of the multiclass confusion matrix are:

True positives (TP): The number of observations for which the model correctly predicted the positive class.

True negatives (TN): The number of observations for which the model correctly predicted a negative class.

False Positives (FP): The number of observations for which the model incorrectly predicted a positive class.

False Negatives (FN): The number of observations for which the model wrongly omitted a positive class.

Each cell of the matrix represents an actual and predicted class pair. The objective is to maximize diagonal elements (true positives) while minimizing classification errors (false positives and false negatives). This matrix is useful for evaluating the performance of the model on each class individually and for identifying classes for which the model has difficulty.

C. Experimental Results

In this section, we will present the results obtained on the UCI-HAR data set during the experimental phase, detailing each phase and its steps carried out to evaluate and validate our method. During the first "Synchronized Windowing" step of the pre-training phase, we used the t-SNE (t-distributed stochastic neighbor embedding) technique to determine the optimal size of the data to cut. This was done by setting three different configurations, namely 60, 80 and 100 length units. Fig. 4 visually presents the corresponding t-SNE representations for each of these configurations, illustrating the dispersion of the sliced data. Visualization of t-SNE provides a graphical understanding of relationships between data in a reduced-dimensional space, making it easier to select the optimal size for cutting. Looking at Fig. 4, we assess that the division into 80 sizes is preferred due to the accumulated clarity of the t-SNE representation at this scale, thus allowing clearer distinction of data by class compared to other sizes. This technique made it possible to optimize the division of the data, thus contributing to a better representation of the essential characteristics during the following phases of the experiment.

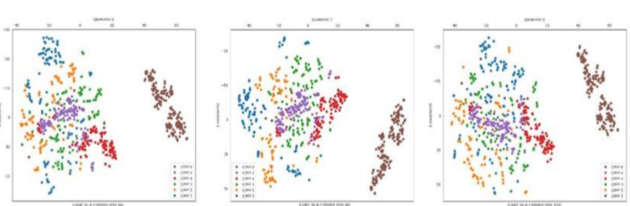

Fig. 4. t-SNE extracted data of size 60, 80 and 100.

In the first preprocessing stage of our method, we sought to improve the classification performance of temporal data by integrating the TimeGAN data augmentation method, a generative approach. The objective was to enrich our dataset and strengthen the robustness of our classification model. To assess the impact of this increase, we used the t-SNE plot and principal component analysis (PCA) to visualize the spatial distribution of the data generated (synthetic) by TimeGAN compared to the real data (see Fig. 5). In summary, the incorporation of TimeGAN in the preprocessing phase aimed to strengthen the ability of our model to effectively deal with temporal variability in the data. PCA and t-SNE visualizations provide visual tools demonstrating the superior quality of the data generated.

In the experiments we conducted to establish, train and test the two DL models in the last two stages, we use the Google Colaboratory platform with the GPU T4 execution type. We leverage Tensorflow 2.9.2 and Keras API to perform everything from data preprocessing to final evaluation. We build the DL models by Keras sequential model based on Tensorflow python architecture as backend.

In the second stage, to calculate the errors between the input data and the reconstructed data, we used the "cross entropy error", with a learning rate equal to 0.0001, with a rate of training set to 0.0025, a learning loss set to 0.0015 and the batch size is set to 128. Additionally, we apply a mean square loss function “MSE” with the “Adam” optimizer to back propagation errors across network layers in order to improve the hyper-parameters of the objective function of two composite models of our proposed model.

The final results of our method is displayed in Fig. 6 and 7, we present the two confusion matrices obtained when testing the UCI-HAR and WISDM datasets.

In Table I, we display the Classification report for our model TSC-LSTM-AE model with two used in this work: UCI-HAR dataset, and WISDM Dataset.

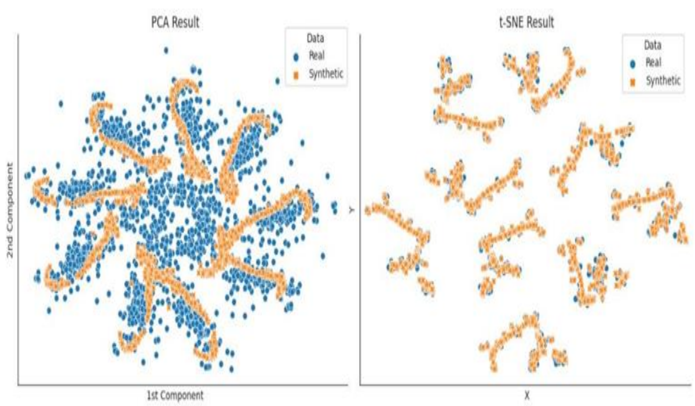

Fig. 5. Qualitative Assessment of Diversity of data generated.

TABLE II. CLASSIFICATION REPORT FOR TSC-LSTM-AE MODEL WITH UCI-HAR DATASET

Class	***P***	***R***	***F1***	***Class***	***P***	***R***	***F1***
WALK	0.97	0.98	0.97	***DOWNS***	0.94	0.98	0.94
WALK_U	0.98	0.95	0.97	***JOGG***	0.99	0.99	0.99
WALK_D	0.99	0.98	0.99	***SITT***	0.91	0.96	0.91
SITTING	0.97	0.93	0.95	***STAND***	0.98	0.99	0.98
STAND	0.89	0.91	0.9	***UPSTAIR***	0.96	0.95	0.96
LAYING	0.98	0.99	0.99	***WALK***	0.97	0.95	0.97
Accuracy	96,1%				96,5%		
Dataset	**UCI-HAR**				**WISDM**		

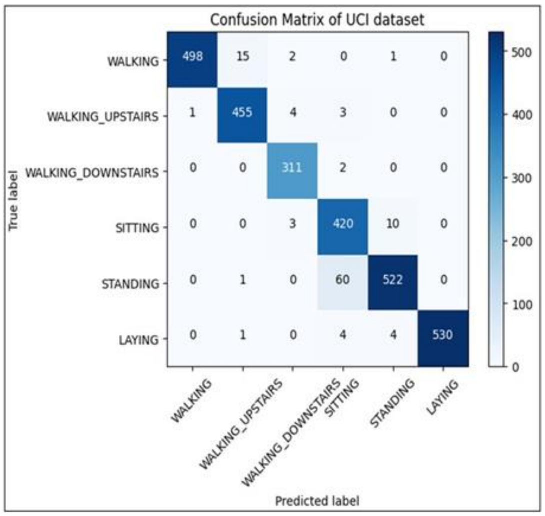

Fig. 6. Confusion Matrix of testing for UCI-HAR Dataset.

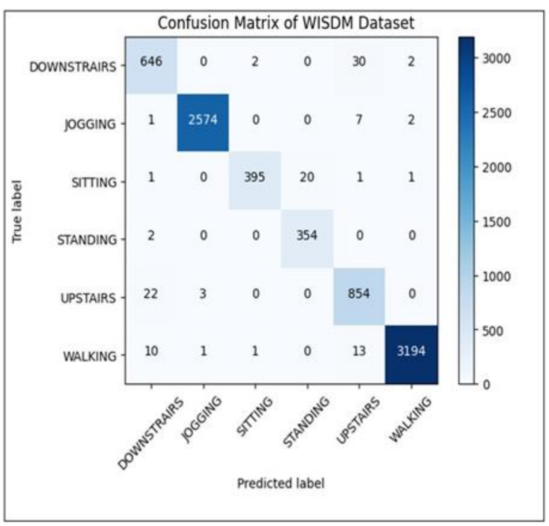

Fig. 7. Confusion Matrix of testing for WISDM Dataset.

In Table II, we compare the average accuracy of our TSC-LSTM-AE model with that of the mentioned approaches. TSCLSTM-AE achieved the best human activity recognition accuracy 96.1% among all the tested approaches for the UCI-HAR dataset and similarly 96.5% for the WISDM dataset.

In Table III, we show the results illustrating the impact of data augmentation performed by TimeGAN on the classification accuracy of temporal data. Data augmentation, in this context, refers to the creation of additional synthetic data through the TimeGAN algorithm, designed specifically to generate realistic time series.

TABLE III. AVERAGE ACCURACY COMPARISON OUR MODEL WITH THREE OTHER MODELS

	Dataset Use	**Our model**	**SVM**	**CNN-LSTM**	**KNN**
Testing accuracy	***UCI -HAR***	**96,1%**	93,6%	95 %	94,3%
	WISDM	**96,5 %**	95,1%	96,2%	94,9%

TABLE IV. COMPARISON OF AVERAGE ACCURACY BETWEEN OUR MODEL WITH, AND WITHOUT DATA AUGMENTATION

	Dataset Use	**With data augmentation**	**Without data augmentation**
Testing accuracy	***UCI -HAR***	96,1%	91,3%
	WISDM	96,5 %	89,5 %

V. DISCUSSION

In this research, we present a Deep Learning method with several steps and components. This method merges unsupervised learning to extract dynamic features using an LSTM AE model [44], with the supervised approach to classify time series, focusing on recognizing human activity at Using smartphone sensors, in this hybrid way, we take advantage of the advantages of supervised learning and those of unsupervised learning and generative models to improve the performance of TSC.

On the one hand, the AE approach enables the complete extraction of various features and the reconstruction of synthetic data similar to the original multivariate data. This is why we combine the power of LSTM blocks with the architecture of auto encoders in an innovative structure. This structure teaches the model to reconstruct appropriate and identifiable temporal labels for the classes. We then intelligently remove this variable from the model input, which prompts the model to complete this variable in the output based solely on the other variables and the relational dependencies captured. In this way, we adopt a method that uses an auto-encoder to extract dynamic features and functional interdependencies between various variables. This strategy aims to considerably strengthen the classification task. The results presented in Table I demonstrate the performance and effectiveness of our proposed model in recognizing both static and dynamic human activities. Our model achieves classification rates surpassing 96% on data from two datasets. All these layers, including the fully connected (FC) layers, have undergone training and fine-tuning to classify the input data proficiently. Our proposed model surpasses other machine learning models including SVM [14], KNN [13] and CNN-LSTM (Classifier 1) [41] models. The results presented in Table II clearly illustrate that our model exhibits significantly improved performance quality and accuracy. Furthermore, our approach is highly recommended for the recognition of human activities, especially in emergencies, such as the automated monitoring of Parkinson's disease and the elderly.

On the other hand, to considerably improve the efficiency and performance of our model, we have incorporated data augmentation through the TimeGAN model, improving. The application of the TimeGAN algorithm [50] aims to increase the original dataset, introducing more diversity and representativeness in the classification models. The results in Table III demonstrate the significant impact of integrating realistic synthetic data on the accuracy of the classification process. This comparative analysis sheds light on the effectiveness of the data augmentation approach proposed by TimeGAN, resulting in an improvement of TSC accuracy by approximately 7% for both balanced and unbalanced datasets.

In this research, we introduce a comprehensive DL method that combines unsupervised learning with an LSTM AE model for dynamic feature extraction and a supervised approach for TSC, specifically targeting human activity recognition using smartphone sensors. The innovative structure incorporates LSTM blocks and auto encoders to reconstruct temporal labels, strategically removing a variable to prompt the model to predict it based on relational dependencies. Classification further validates the model's ability to capture dynamic features. Based on the experimental results, we can conclude that our proposed approach enhances the prediction accuracy of temporal data classification on datasets well recognized in the literature.

VI. Conclusion

In conclusion, this study introduces TSC-LSTM-AE, a novel method designed for the classification of multivariate time series in the realm of human activity recognition utilizing sensor data. The efficacy of our proposed approach is evident through its superior performance in comparison to alternative methods, showcasing its proficiency in handling temporal data across various evaluation criteria. The results obtained underscore the superiority of our methodology over other experimental benchmarks. As our approach remains dynamic, continual refinement can be achieved through additional experimental studies to comprehensively address diverse aspects. In light of these findings, we advocate for the further development and integration of real-time capabilities, aiming to enhance the responsiveness of our approach for both professional and personal applications that demand swift and efficient processing.

In future work, we plan to explore the applicability of this approach to other application domains. In addition, we plan to improve our method to better handle real-time aspects. Finally, we will seek to refine our model so that it is more sensitive to subtle variations in the data, which could lead to significant improvements in classification robustness.

References

[1] A. Gupta, H. P. Gupta, B. Biswas and T. Dutta, “Approaches and Applications of Early Classification of Time Series: A Review,” in IEEE Transactions on Artificial Intelligence, vol. 1, no. 1, pp. 47-61, Aug. 2020.

[2] P. Kanani and M. Padole, “ECG heartbeat arrhythmia classification using time-series augmented signals and deep learning approach,” Procedia Computer Science, vol. 171, pp. 524–531, 2020.

[3] S. Pahuja and K. Veer, “Recent approaches on classification and feature extraction of EEG signal: A review,” Robotica, vol. 40, no. 1, pp. 77-101, Jan. 2022.

[4] T. Liang and Y. J. Yuan, “Wearable medical monitoring systems based on wireless networks: A review,” IEEE Sensors Journal, vol. 16, no. 23, pp. 8186–8199, 2016.

[5] E. Fons, P. Dawson, X.-j. Zeng, J. Keane, and A. Iosifidis, “Evaluating data augmentation for financial time series classification,” arXiv preprint arXiv:2010.15111, 2020.

[6] Md. M. Rahman, M. A. Farahani, and T. Wuest, “Multivariate Time-Series Classification of Critical Events from Industrial Drying Hopper Operations: A Deep Learning Approach,” Journal of manufacturing and materials processing, Sep. 08, 2023.

[7] N. Mehdiyev, J. Lahann, A. Emrich, D. Enke, P. Fettke, and P. Loos, “Time Series Classification using Deep Learning for Process Planning: A Case from the Process Industry,” Procedia Computer Science, Jan. 01, 2017.

[8] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. “Adversarial attack on deep neural networks for time series classification,” in Proc. IJCNN 2019, pages 1–8, 2019.

[9] S. Tobiyama, Y. Yamaguchi, H. Shimada, T. Ikuse and T. Yagi, “Malware detection with deep neural network using process behavior,” Proc. IEEE 40th Annu. Comput. Softw. Appl. Conf. (COMPSAC), vol. 2, pp. 577-582, Jun. 2016.

[10] C. Papagiannopoulou, S. Decubber, D. G. Miralles, M. Demuzere, N. Verhoest, and W. Waegeman, “Analyzing Granger Causality in Climate Data with Time Series Classification Methods,” Lecture Notes in Computer Science, Jan. 01, 2017.

[11] B. Zhao, H. Lu, S. Chen, J. Liu and D. Wu, “Convolutional neural networks for time series classification,” in Journal of Systems Engineering and Electronics, vol. 28, no. 1, pp. 162-169, Feb. 2017.

[12] A. Abanda, U. Mori, and J. Lozano, “A review on distance based time series classification,” Data Mining and Knowledge Discovery, 33(2):378–412, 2019.

[13] Y.-H. Lee, C.-P. Wei, T.-H. Cheng, and C.-T. Yang, “Nearest-neighborbased approach to time-series classification,” Decision Support Systems, vol. 53, no. 1, pp. 207 – 217, 2012.

[14] D. Zhang, W. Zuo, D. Zhang and H. Zhang, “Time Series Classification Using Support Vector Machine with Gaussian Elastic Metric Kernel,” 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 2010, pp. 29-32.

[15] C. Neukirchen and G. Rigoll. “Time Series Classification using Hidden Markov Models and Neural Networks.” In Proc. of the IAR Annual Meeting. Duisburg, Germany, November 1997.

[16] M. F. Ghalwash, D. Ramljak, and Z. Obradovi´c, “Early classification of multivariate time series using a hybrid hmm/svm model,” in Proc. BIBM, 2012, pp. 1–6.

[17] P. Senin, “Dynamic time warping algorithm review,” Information and Computer Science Department University of Hawaii at Ma-noa Honolulu, USA, 2008, vol. 855, no. 1-23, p. 40.

[18] H. Deng, G. Runger, E. Tuv and M. Vladimir, “A time series forest for classification and feature extraction”, Inf. Sci., vol. 239, pp. 142-153, 2013.

[19] F. J. Baldan and J. M. Benítez, “Complexity measures and features for times series classification,” Expert Systems with Applications, Mar. 01, 2023.

[20] L. Khrissi, E. A. Nabil, H. Satori, and K. Satori, “A Feature Selection Approach Based on Archimedes’ Optimization Algorithm for Optimal Data Classification,” Jan. 01, 2023.

[21] Y. Lei and Z. Wu, ‘‘Time series classification based on statistical features,’’ EURASIP J. Wireless Commun. Netw., vol. 2020, no. 1, pp. 1–13, Feb. 2020.

[22] A. Bagnall, L. Davis, J. Hills, and J. Lines, “Transformation based ensembles for time series classification,” in Proc. 12th SDM, 2012, vol. 12, pp. 307–318.

[23] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, “Transformers in time series: A survey,” 2022, arXiv:2202.07125.

[24] J. Lines, L. Davis, J. Hills, and A. Bagnall, "A shapelet transform for time series classification," in Proc. 18th ACM Int. Conf. Knowl. Discovery Data Mining, 2012, pp. 289–297.

[25] D. I. K. A. BUZA, "Fusion methods for time-series classification," Update, 2011, vol. 2, p. 27.

[26] S. Quan, M. Sun, X. Zeng, X. Wang, and Z. Zhu, "Time Series Classification Based on Multi-Dimensional Feature Fusion," IEEE Access, vol. 11, pp. 11 066–11 077, 2023.

[27] X. B. Jin, A. Yang, T. Su, J. L. Kong, and Y. Bai, "Multi-channel fusion classification method based on time-series data," Sensors, vol. 21, no. 13, 2021.

[28] B. K. Iwana and S. Uchida, "Time series classification using local distance-based features in multi-modal fusion networks," Pattern Recognition, vol. 97, pp. 1-12, 2019.

[29] T. Wang, Z. Liu, T. Zhang, S. F. Hussain, M. Waqas and Y. Li, "Adaptive feature fusion for time series classification," Knowledge-Based Systems, vol. 243, pp. 108459, 2022.

[30] M. Abouelnaga, J. Vitay, and A. Farahani, "Multivariate Time Series Classification: A Deep Learning Approach," arXiv.org, Jul. 05, 2023.

[31] B. Zhao, H. Lu, S. Chen, J. Liu and D. Wu, "Convolutional neural networks for time series classification," J. Syst. Eng. Electron., vol. 28, no. 1, pp. 162-169, Feb. 2017.

[32] L. Sadouk, "CNN approaches for time series classification," in Time Series Analysis: Data Methods and Applications, London, U.K.:IntechOpen, pp. 1-23, 2019.

[33] Z. Cui, W. Chen and Y. Chen, "Multi-scale convolutional neural networks for time series classification," arXiv:1603.06995.

[34] B. Qian, Y. Xiao, Z. Zheng, M. Zhou, W. Zhuang, S. Li, and Q. Ma, "Dynamic multi-scale convolutional neural network for time series classification," IEEE Access, vol. 8, pp. 109 732–109 746, 2020.

[35] M. Hüsken and P. Stagge, "Recurrent neural networks for time series classification," Neurocomputing, vol. 50, pp. 223-235, Jan. 2003.

[36] F. Zhu, H. Wang and Y. Zhang, "GRU Deep Residual Network for Time Series Classification," 2023 IEEE 6th Information Technology,Networking,Electronic and Automation Control Conference (ITNEC), Chongqing, China, 2023, pp. 1289-1293.

[37] Y. Errafik, A. Kenzi, and Y. Dhassi, "Proposed Hybrid Model Recurrent Neural Network for Human Activity Recognition," Lecture notes in networks and systems, Jan. 01, 2023.

[38] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, arXiv:1506.00019.

[39] K. Choi, G. Fazekas, M. Sandler and K. Cho, "Convolutional recurrent neural networks for music classification," Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP), pp. 2392-2396, Mar. 2017.

[40] M. Zihlmann, D. Perekrestenko, and M. Tschannen, "Convolutional recurrent neural networks for electrocardiogram classication," in Proc. Comput.Cardiol. (CinC), Sep. 2017.

[41] C. I. Garcia, F. Grasso, A. Luchetta, M. C. Piccirilli, L. Paolucci, and G. Talluri, "A comparison of power quality disturbance detection and classification methods using CNN, LSTM and CNN-LSTM," Applied Sciences, vol. 10, no. 19, pp. 6755, Sep. 2020.

[42] F. Liu, X. Zhou, T. Wang, J. Cao, Z. Wang, H. Wang, et al., "An attention-based hybrid LSTM-CNN model for arrhythmias classification," Proc. Int. Joint Conf. Neural Netw. (IJCNN), pp. 1-8, Jul. 2019.

[43] S. P. Chakka, S. K. Vengalil, and N. Sinha, "Identification of Stochasticity by Matrix-decomposition: Applied on Black Hole Data," arXiv.org, Jul. 15, 2023.

[44] P. Liu, X. Sun, Y. Han, Z. He, W. Zhang and C. Wu, "Arrhythmia classification of LSTM autoencoder based on time series anomaly detection," Biomed. Signal Process. Control, vol. 71, Jan. 2022.

[45] H. Meng, C. Wagner and I. Triguero, "Feature importance identification for time series classifiers," 2022 IEEE International Conference on Systems Man and Cybernetics (SMC), pp. 3293-3298, 2022.

[46] W. Yu, I. Y. Kim, and C. K. Mechefske, "Analysis of different RNN autoencoder variants for time series classification and machine prognostics," Mechanical Systems and Signal Processing, Feb. 01, 2021.

[47] S. Bandyopadhyay, A. Datta, and A. Pal, "Automated Label Generation for Time Series Classification with Representation Learning: Reduction of Label Cost for Training," arXiv.org, Jul. 12, 2021.

[48] G. Krishna, C. Tran, M. Carnahan, and A. Tewfik, ‘‘Constrained variational autoencoder for improving EEG based speech recognition systems,’’ 2020, arXiv:2006.02902.

[49] T. Terefe, M. Devanne, J. Weber, D. Hailemariam, and G. Forestier, "Time series averaging using multi-tasking autoencoder," in IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 2020.

[50] J. Yoon, D. Jarrett, and M. van der Schaar, "Time-series generative adversarial networks," in Proc. of Advances in Neural Information Processing Systems. Curran Associates, Inc., 2019, pp. 5509–5519.

[51] D. Anguita, A. Ghio, L. Oneto, X. Parra and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in Proc. Esann, vol. 3, pp. 3, Apr. 2013.

[52] J. R. Kwapisz, G. M. Weiss and S. A. Moore, "Activity recognition using cell phone accelerometers," ACM SIGKDD Explor. Newslett., vol. 12, no. 2, pp. 74-82, Mar. 2011.