Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 20 No. 4 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v20i04.43623

PAPER

Human Activity Recognition Using Convolutional Autoencoder and Advanced Preprocessing

Chaimae Zaoui(✉), Faouzia Benabbou, Abdelaziz Ettaoufik, Khadija Sabiri

Laboratory of Modeling and Information Technology Faculty of Sciences Ben M'SIK, University Hassan II, Casablanca, Morocco

chaimae.zaoui1-etu@etu.univh2c.ma

ABSTRACT

E-health systems rely on information and communication technology to support and improve various aspects of health services, delivery, and management. The success of artificial intelligence techniques has led to the emergence of a variety of systems designed to address a wide range of healthcare issues. In particular, gathering data on patient activity and behavior has enabled the development of reliable predictive systems for detecting chronic diseases and forecasting their progression. Human activity detection is a vast and emerging field, and various datasets have been collected for training different machine learning and deep learning (DL) models. The University of Milano Bicocca smartphone-based human activity recognition (UniMiB-SHAR) dataset is widely used for analyzing and recognizing human actions, including walking, running, and other daily activities. However, the autoencoder (AE) technique trained on this dataset yields poor performance. This paper aims to enhance the performance of AEs on the challenging UniMiB-SHAR dataset by introducing a convolutional AE model and employing novel preprocessing techniques, including normalization, magnitude, principal component analysis (PCA), and balancing methods such as SMOTEEN and ADASYNE. The experimental results demonstrate that the proposed AE model achieved successful performance, surpassing the state-of-the-art methods, with accuracies of 96.56% for activities of daily living (ADL), 98.86% for Fall, and 88.47% for the full dataset.

KEYWORDS

human activity detection, deep learning (DL), autoencoder (AE), UniMiB-SHAR

1 INTRODUCTION

Artificial intelligence and ubiquitous computing have recently placed more emphasis on recognizing human activity [1]. This shift can be attributed, in part, to the proliferation of Internet of Things (IoT) devices such as smart home systems and wearable devices, which provide a multitude of data sources offering real-time and continuous information. The field of human activity detection is a research area focused on identifying and comprehending the actions and behaviors of individuals

Zaoui, C., Benabbou, F., Ettaoufik, A., Sabiri, K. (2024). Human Activity Recognition Using Convolutional Autoencoder and Advanced Preprocessing. *International Journal of Online and Biomedical Engineering (iJOE)*, 20(4), pp. 144–159. https://doi.org/10.3991/ijoe.v20i04.43623

Article submitted 2023-08-01. Revision uploaded 2023-10-11. Final acceptance 2023-10-11.

in various scenarios. This field is relevant in numerous domains, including healthcare, video surveillance, human-computer interaction, security, and sports. The field of eHealth encompasses the use of information and communication technologies in the healthcare sector. This includes the use of electronic medical records to store medical information, teleconsultations for remote health monitoring, mobile health for smartphone-based healthcare, and wearables equipped with sensors for various healthcare purposes [2]. These sensors can include heart rate sensors, glucose sensors, motion sensors, and temperature sensors. Additionally, computer vision, machine learning (ML), and deep learning (DL) techniques have significantly improved recognition systems in various domains, such as facial recognition, object recognition, speech recognition, and human activity recognition (HAR) [3]. Recently, there has been a significant increase in the use of machine learning methods for solving activity recognition (AR) problems. ML techniques have proven to be highly effective [4] in detecting human activity. By harnessing the power of ML algorithms, it becomes possible to automatically recognize and classify various human behaviors. The data utilized in the field of human activity detection comes in various formats, including videos, images, sensors, wearable devices, and audio data. The data from sensors includes information from accelerometers, gyroscopes, pressure sensors, heart rate sensors, motion sensors, and more. In our previous work [6], we utilized the UniMiB-SHAR dataset [5] to analyze and classify various activities. We employed a range of ML techniques, such as K-nearest neighbors (KNN), random forest (RF), convolutional neural network (CNN), autoencoder (AE), support vector machine (SVM), and artificial neural network (ANN) [7]. In many studies, AE models did not yield significant results in the University of Milano Bicocca smartphone-based HAR (UniMiB-SHAR) [8] [9] [10] [11]. For this reason, this paper proposes a new architecture for AEs with the implementation of various preprocessing techniques, including magnitude calculation, normalization, and dimensional reduction using principal component analysis (PCA), to enhance performance on the UniMiB-SHAR dataset.

The paper is organized as follows: Section 2 explores the related literature in the field, while Section 3 describes the methods used in the approach. Section 4 presents the experimental results, and Section 5 concludes the findings and discusses future research directions.

2 RELATED WORKS

Different ML and DL techniques have been applied to this dataset. However, the performance of the AE application is generally low on the UniMiB-SHAR dataset.

A study conducted in [8] aims to compare the performance of various algorithms, including hand-crafted features (HC), codebook approaches (CB), AE, multi-layer perceptron (MLP), recurrent neural networks (RNNs), convolutional neural networks (CNNs), long short-term memory networks (LSTM), hybrid convolutional and recurrent networks, and deep residual learning. In this study, the authors used the sliding time window size as an important hyperparameter in models, with T corresponding to 32 (approximately 1 second), 64 (approximately 2 seconds), and 96 (approximately 3 seconds) on different datasets. Using the UniMiB-SHAR dataset with 17 activities (Full-17), the AE model achieved F1-scores of 68.37% (T = 32), 68.24% (T = 64), and 68.39% (T = 96). An LSTM-AE network for fall detection, as proposed in [9], utilized data collected from smartphone accelerometers to capture body acceleration. They conducted experiments on two scenarios: activities of daily living and falls. Using the ADL-9 sub-dataset, their model achieved an accuracy of 94.70%, an AUC of 87.21%, and an F1-score of 87%. For fall detection (Fall-8), the model achieved an accuracy of 55.20%, an AUC of 67.82%, and an F1-score of 55%. A variational AE is proposed for HAR using the UniMiB-SHAR (Full-17) dataset in [10]. The latter achieves an average performance of 24.01% on the Full-17 dataset. Various ML techniques were investigated by [11], including the AE, using two datasets (OPPORTUNITY and UniMiB-SHAR) for sensor-based HAR. Using Leave-One-Subject-Out cross-validation with 30-fold, the AE achieved an accuracy of 65.67%, a weighted F1-score of 64.84%, and an average F1-score of 55.04% on the Full-17 sub-dataset of the UniMiB-SHAR dataset. To tackle the class imbalance in the human activity dataset [12], the Balancing Sensor Data Generative Adversarial Networks (BSDGAN) technique was employed. An AE was utilized in the learning process of BSDGAN on the WISDM and UniMiB-SHAR datasets, as well as on the ADL-9 sub-dataset. An approach called multiclass AE-based active learning (MAAL) is introduced, which is based on multiclass AE and deep support vector data description (Deep SVDD) models as presented in [13]. Two datasets, USC-HAD and UniMiB-SHAR, consist of 10 classes. The fall-8 class is considered one class, and the remaining nine classes are from ADL-9. This approach achieved 90.79% accuracy and an 88.40% F1-score on the USC-HAD dataset, while it achieved 98.66% accuracy and a 94.90% F1 score on the UniMiB-SHAR dataset for the Full-17 sub-dataset.

Table 1 presents the studies that have focused on enhancing the performance of AEs using the Full UniMiB-SHAR dataset or its sub-datasets, according to the specified criteria.

- Study ref: it presents the article reference.
- Description: it presents the idea for each article.
- Preprocessing: it represents the techniques used to prepare the data set.
- Dataset: it presents the dataset or sub-dataset used in each work.
- Performance: is validation score employed like Accuracy, F1-score and AUC.

Table 1. AE's performance in the state of the art

Ref.	Description	Preprocessing	Dataset	Performance
[8]	Proposition of ARN model for HAR and its comparison with other models such as: HC, CB, AE, Multi-Layer Perceptron (MLP), Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Long-Short Term Memory Networks (LSTM), Hybrid Convolutional and Recurrent Networks, Deep Residual Learning.	Sliding Time Window Size	Full-17	F1-scores : 68.37% (T = 32), 68.24% (T = 64) and 68.39% (T = 96).
[9]	Proposition of recurrent autoencoders model for HAR, and comparison with other models such as SVM and OC-SVM.	Normalization	ADL-9	Accuracy: 94.70% AUC: 87.21% F1-score: 87%
			Fall-8	Accuracy: 55.20% AUC: 67.82% F1-score: 55%
[10]	Proposition of Generalizable Independent Latent Excitation (GILE) model and variational autoencoder (VAE) model for HAR and comparison with DIVA, DDNN, DeepConvLSTM, the CODATS time series model.	Cross validation with leave-one-domain-out strategy	Full-17	Accuracy: 24.01%
[11]	Comparison of deep learning methods integrating autoencoding for HAR.	Leave-One-Subject-Out cross-validation	Full-17	Accuracy: 65.67% F1-score: 64.84%
[13]	Proposition of Multiclass Autoencoder-Based Active Learning (MAAL and deep Support Vector Data Description (Deep SVDD) for HAR.	Combine all fall type in single type.	ADL+ Fall in single type	Accuracy: 98.66% F1-score: 94.90%

Table 1 demonstrates that a range of ML or DL techniques have been utilized with great success on the UniMiB-SHAR dataset. Some studies focus on the entire dataset (Full-17), while others concentrate on the ADL-9 or Fall-8 sub-datasets, often approaching the problem as a binary classification task. It is worth noting that the

Fall-8 subset exhibits lower performance, and the AE performances remain subpar and constrained on this dataset [9], [10].

The UniMiB-SHAR dataset has been widely used in the literature, as evidenced in our previous research [6]. Numerous models have been developed and trained using this dataset to support a wider variety of classification tasks.

A detailed description of the strategy proposed in [14], which seems to be based on deep learning networks and a margin-based approach. This section describes how margins are utilized to enhance the accuracy of activity recognition in three datasets: OPPORTUNITY, PAMAP2, and UniMiB-SHAR. The proposed method yields higher accuracy than the methods already mentioned in the state of the art. An approach called "Channel-Selectivity" [15] focuses on selecting the most informative sensor channels using two deep learning models, CNN and ResNet, to improve recognition accuracy in five datasets: UCI-HAR, OPPORTUNITY, UniMib-SHAR, WISDM, and PAMAP2. CNN with SelectConv achieves an accuracy of 96.77% on UCI-HAR, 79.67% on OPPORTUNITY, 77.26% on UniMib-SHAR, 97.44% on WISDM and 94.33% on PAMAP2. For ResNet with SelectConv, the accuracy rates are 97.28%, 82.36%, 78.25%, 98.52%, and 94.33% on UCI-HAR, OPPORTUNITY, UniMib-SHAR, WISDM, and PAMAP2 respectively. A unified deep learning approach [16], based on ResNet, was applied to the UniMiB-SHAR dataset for two classification scenarios: binary and multi-class. For binary classification, the proposed method achieves an accuracy rate of 99.87% using 5-fold cross-validation (CV) and 98.48% using the leave-one-subject-out (LOO) method. For multi-class classification, the performance was 97.39% with 5-fold CV, 98.07% with 10-fold CV, and 80.09% with 10-fold leave-one-out (LOO) cross-validation. An approach using neural architecture search (NAS) to search for suitable models for HAR tasks, called HARNAS, is being tested on the Opportunity dataset and UniMiB-SHAR as proposed in [17].

3 MATERIEL AND METHOD

This study aims to improve the performance of the AEs model using the UniMiB-SHAR dataset. The key steps and procedures involved in this approach include portioning the dataset into three sub-datasets (ADL-9, Fall-8, AF-2, and Full-17), preprocessing the data using normalization, dimension reduction, magnitude, and balancing the data on AF-2 and Full-17, training the proposed AE model, and classification. The performance is evaluated using various metrics, including accuracy, precision, recall, AUC, and loss. Figure 1 depicts the flowchart of the proposed method.

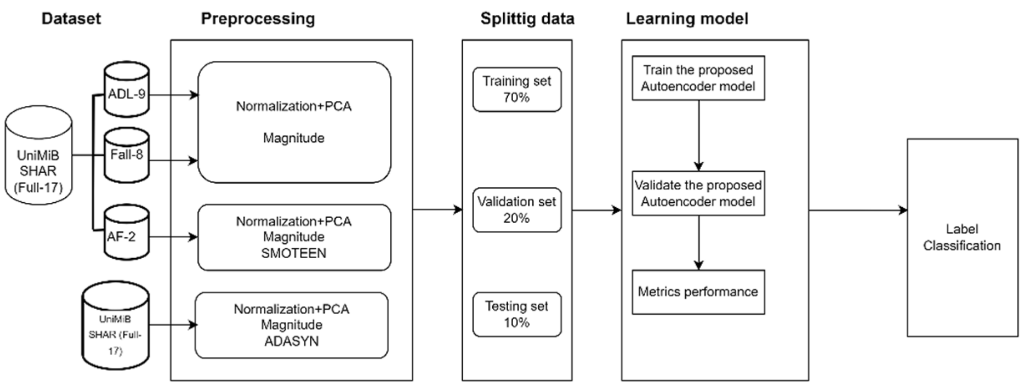


Fig. 1. The proposed approach

More details are provided in the sub-sections.

3.1 Datasets

The University of Milano Bicocca smartphone-based HAR (UniMiB-SHAR) dataset is a recently created collection containing 11,771 samples specifically designed for HAR and fall detection purposes [18]. The dataset consists of data collected from 30 participants, including 6 males and 24 females, aged between 18 and 60 years. The data was captured with a Bosch BMA220 3D accelerometer integrated within a Samsung Galaxy Nexus I9250 smartphone. Sampling of the data was conducted at a rate of 50 Hz, a common rate utilized in relevant literature concerning HAR. The dataset includes a total of 17 distinct classes, which are further divided into 9 types of ADLs (activities of daily living) and 8 types of falls. Additionally, the dataset stores associated information that facilitates the selection of samples based on various criteria, including the ADL type performed, gender, age, and more [19]. In UniMiB-SHAR, both the data and their corresponding labels are provided in the form of time windows with a fixed length of T = 151, which is approximately equivalent to 3 seconds. An energy-based segmentation method is used to establish these time periods, which involves locating peaks in the acceleration data. When the amplitude of the signal reaches 1.5g (where g is the gravitational acceleration constant) and is less than 0 at time t-1, an acceleration peak is identified at a specific time point, t. The dataset comprises 11,771 time periods, each with three dimensions. In this experiment, we utilized the UniMiB-SHAR dataset as a distinct sub-dataset.

To study the AE model, the original dataset was divided into three sub-datasets: ADL-9, Fall-8, and AF-2. Table 2 provides details of the four datasets.

Table 2. Dataset description

Dataset	Description	Number of Data	Activities	Class
ADL-9	9 classes activities of Daily Living	7579	9	ADL
Fall-8	8 classes of fall activities	4192	8	Fall
AF-2	2 classes fall and ADL (Not fall).	11771	2	Fall and Not-Fall
Full-17	17 classes	11771	17	ADL+ Fall

3.2 Preprocessing

Preprocessing techniques play a crucial role in improving the performance of ML algorithms. The study investigated four methods: the standard scaler normalization method, magnitude transformation, dimension reduction using PCA, and balancing datasets with the SMOTEEN and ADASYN methods. However, depending on the nature of the dataset, specific techniques have been applied. The Full-17 dataset and AF-2 are both highly imbalanced. The most effective techniques for improving performance were ADASYN for Full-17 and SMOTEEN for AF-2, as well as Magnitude. The ADL-9 and Fall-8 datasets are balanced. Normalization and PCA were applied to the Full-17, AF-2, ADL-9, and Fall-8 datasets.

Magnitude. The original dataset consists of 453 columns, each representing the X, Y, and Z coordinates of the sensor across 151 different instances within a three-second time frame. Each data point was replaced by its magnitude value in order to simplify the data and capture the essential information, each data point was replaced by its magnitude value [5]. The magnitude represents the distance from the origin of each data point [20], as shown in Equation 1. The magnitude, or size, of a vector is represented by its norm. Regardless of the direction of a vector, its magnitude provides a measurement of its length or intensity.

$$\left\| V \right\| = \sqrt{a^2 + b^2 + c^2} \tag{1}$$

Normalization. Normalization refers to the adjustment of data features to a single common scale (with a mean of 0 and a standard deviation of 1) to improve convergence, reduce scalar asymmetry, and enhance model stability [21]. This can be achieved using methods suah as MinMaxScaler, StandardScaler, and Robust Scaling [22]. The data used in our experiment is normalized using StandardScaler, which scales each feature to a specified range between 0 and 1 to minimize loss and expedite model convergence.

Dimensionality reduction: principal component analysis. In our study, we utilized PCA on our dataset. We retained components that accounted for variance exceeding a threshold of 1%, leading to a reduction in the dataset's dimensionality to only six components. The reason for this choice is to disregard fewer representative axes, thereby eliminating noise from the data. These new variables are uncorrelated with each other and retain a significant amount of information from the original data [23] [24]. Table 3 displays the quantity of data items following the implementation of principal component analysis.

Table 3. Quantity of data

Dataset	Number of Data	Number of Features before PCA	Number of Features after PCA
ADL-9	7579	453	6
Fall-8	4192	453	390
AF-2	11771	453	434
Full-17	11771	453	434

Balancing data. The problem of data imbalance is one of the challenges encountered in prediction and classification tasks, and it is a common and foreseeable issue. Balancing a dataset facilitates model creation by preventing biases in favor of a specific class. However, we have explored various approaches to address this challenge, including resampling, undersampling, oversampling, and hybrid approaches. **SMOTEEN** is a valuable and effective technique that combines the advantageous features of SMOTE (synthetic minority over-sampling technique) and edited nearest neighbors (ENN) [25], while **ADASYN** represents an improved version of SMOTE [26].

We explored other techniques, such as SMOTE and resampling. However, the best results have been achieved by applying SMOTEEN to the AF-2 dataset and ADASYN to the Full-17 dataset.

3.3 Splitting the dataset

Data splitting is a standard procedure in data analysis, ML, and data science. It divides a dataset into two or more subsets. In our study, we divided the dataset into three parts. The first part, comprising 70% of the data, was used for training machine learning models. The second part, consisting of 20% of the data, was allocated for refining the model's hyperparameters and conducting performance evaluations during training. Finally, the remaining 10% was designated for testing the model's generalization and performance in real-world scenarios.

3.4 Learning model

Background. An AE is a specific type of neural network [27] designed to encode and decode an input into a meaningful and compressed representation. It ensures that the reconstructed input is as close as possible to the original. Similar to other DL architectures, the AE operates through neural layers and is trained using backpropagation. The layers are divided into multiple encoding and decoding layers. The input is connected to the initial encoding layer, and in each subsequent encoding layer, the number of neurons is reduced until reaching the final encoding layer. In each coding layer, the number of neurons progressively decreases until the final coded layer, which has the highest neuron count and represents the features of the bottleneck. Beyond this layer, the decoding layers are initialized. In each decoder layer, the number of neurons in each subsequent layer is gradually increased until the output layer matches the number of neurons in the input layer [28]. There are various types of autoencoders, classified according to structures and specific objectives. These include variational autoencoders (VAE) [29] [30], generative adversarial networks (GAN) designed for unsupervised learning tasks [31], and convolutional autoencoders (CAE) [32]. In this study, we employed a CAE model, which generally consists of two layers: the encoder and decoder. This model is intended to produce a program code for each input sample, minimizing the mean square error (MSE) between inputs and outputs. Convolutional layers enable the capture of patterns and spatial relationships in the input data. CAEs learn sparse representations by encouraging the activation of only a limited number of neurons in the hidden layer at any given time. The proposed approach involves incorporating a regularization term into the loss function, which encourages the learning of concise and informative representations. CAEs are well-suited for tasks such as image reduction, denoising, and feature extraction. They achieve this by minimizing the loss function through the adjustment of the network's weights and biases.

Convolutional autoencoder model. In our experiment, we utilized a CAE consisting of the following layers:

- **Input layer:** This layer contains the input data from the dataset.
- **Encoder:** The encoder is constructed using a CNN with three convolution sub-layers, each followed by a corresponding pooling layer. These layers extract and compress the pertinent features from the input data.
- **Decoder:** The decoder part utilizes deconvolution sub-layers to decode the output of the encoder. These layers reconstruct the original input using the compressed features obtained from the encoder.
- **Classifier:** The classifier is responsible for categorizing the data. It is based on two layers: GRU and dense. We evaluate its performance using various metrics such as accuracy, precision, recall, area under the curve (AUC), and loss.

Figure 2 illustrates the different layers of our proposed AE model. The first phase involves the input layer, followed by the encoder component, which includes convolutional and pooling layers to extract pertinent features from the input data.

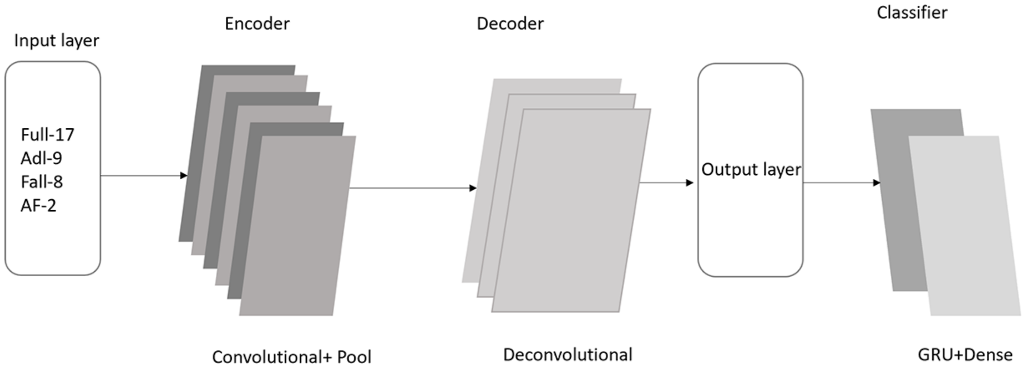


Fig. 2. The proposed AE model

The convolutional operations filter the input data, capturing local patterns and structures. The pooling layer downsamples the output of the convolutional layer, reducing spatial dimensions while preserving the most important information. The decoder section includes deconvolutional layers that decode the data, reconstructing it in its original form. Lastly, the classifier section includes a GRU model that processes the output data as a sequence. The GRU model is designed to capture sequential dependencies within the data. The classification is carried out using the dense layer, which is responsible for the final classification task. Table 4 displays the hyperparameters linked to our model.

Table 4. CAE model parameters

	ADL-9	Fall-8	AF-2	Full-17
Neuron	77618	80443	76561	76514
Optimizer	Adam	Adam	Adam	Adam
Activation Function	softmax	softmax	softmax	softmax
Loss Function	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy
Epochs	100	100	100	20
Batch size	24	24	24	24

4 EXPERIMENT AND ANALYSIS

In this study, we divided our dataset processing into three phases: model training, model validation, and model testing to assess its performance on unseen data. Tables 5–8 represent the performance results of the approach when training the CAE model with the Full-17 dataset and subset datasets: ADL-9, Fall-8, and AF-2, respectively.

Table 5. CAE performance using Full-17 dataset

Metrics	Magnitude			PCA + Normalization		
	Train	Validation	Test	Train	Validation	Test
Accuracy	91.48%	88.21%	**88.47%**	65.42%	84.48%	66.18%
Precision	94.57%	91.68%	**92.08%**	79.77%	79.66%	80.53%
Recall	89.40%	85.76%	**86.44%**	52.21%	51.40%	53.43%
F1-score	91.47%	88.22%	**88.47%**	65.45%	84.50%	66.18%
AUC	98.90%	98.62%	**98.60%**	96.05%	96.07%	96.26%
Loss	0.007	0.01	**0.009**	0.02	0.02	0.02

Table 6. CAE performance using ADL-9 sub-dataset

Metrics	Magnitude			PCA + Normalization		
	Train	Validation	Test	Train	Validation	Test
Accuracy	99.72%	95.02%	96.17%	99.99%	97.36%	**96.56%**
Precision	99.76%	95.59%	95.76%	99.98%	97.36%	**96.30%**
Recall	99.70%	94.86%	95.64%	99.98%	97.36%	**96.17%**
F1-score	99.72%	95.02%	96.16%	99.98%	97.36%	**96.56%**
AUC	100.00%	99.13%	99.43%	99.99%	99.51%	**99.28%**
Loss	7.62	6.92	7.28	0.12	0.15	**0.17**

Table 7. CAE performance using AF-2 sub-dataset

Metrics	Magnitude			PCA + Normalization		
	Train	Validation	Test	Train	Validation	Test
Accuracy	99.92%	99.31%	**98.86%**	99.97%	99.49%	98.81%
Precision	99.92%	99.31%	**98.86%**	99.98%	99.94%	99.81%
Recall	99.92%	99.31%	**98.86%**	99.49%	99.98%	98.81%
F1-score	99.94%	99.30%	**98.86%**	99.97%	99.50%	98.81%
AUC	99.99%	99.80%	**99.65%**	99.98%	99.88%	99.30%
Loss	0.03	0.03	**0.04**	0.09	0.02	0.07

Table 8. CAE performance using Fall-8 sub-dataset

Metrics	Magnitude			PCA + Normalization		
	Train	Validation	Test	Train	Validation	Test
Accuracy	58.14%	47.32%	**48.86%**	48.13%	46.99%	47.15%
Precision	60.15%	57.30%	**65.67%**	60.15%	57.30%	65.70%
Recall	32.49%	28.23%	**27.41%**	30.12%	31.67%	31.88%
F1-score	58.14%	47.33%	**48.87%**	48.12%	49.00%	47.17%
AUC	91.27%	86.83%	**86.88%**	88.50%	83.16%	84.45%
Loss	1.11	1.34	**1.3**	2.12	2.10	1.99

According to Table 5, the Full-17 sub-dataset achieved the best results using the magnitude technique, with 88.47% accuracy, 92.08% precision, 86.44% recall, 88.47% F1-score, 98.60% AUC, and 0.009 loss. For the ADL-9 data subset, Table 6 presents compelling evidence that superior performance was achieved by applying PCA and normalization techniques. The results indicate that during training, an impressive accuracy of 99.99% was attained, while validation accuracy, precision, F1 score, and recall attained significant scores of 97.36%. Test precision and recall were also impressive, achieving scores of 96.30% and 96.17%, respectively. In the training, precision and recall were both respectable at 99.98%. In terms of the AUC metric, the data subset performed exceptionally well, achieving scores of 99.99%, 99.51%, and 99.28% for training, validation, and test sets, respectively. In relation to AF-2, Table 7 illustrates the performance for various metrics and indicates that the disparities between the two pre-processing methods are minimal. However, the magnitude method performed slightly better because of its lower values. Specifically, it achieved an accuracy of 99.92% for training, 99.31% for validation, and 98.86% for testing. Precision scores were also high, with training, validation and testing achieving 99.92%, 99.31%, and 98.86%, respectively. Similarly, the recall scores for training, validation, and testing were 99.92%, 99.31%, and 98.86%, respectively. In the case of the Fall-8 sub-dataset, Table 8 shows low results despite the application of two pre-processing methods and the balancing of the data. The difficulty may arise from the close similarity between the eight classes, making it challenging to distinguish between them. Further efforts should be made to investigate ways to enhance the performance of the CAE model. One potential way to improve is to explore data preprocessing techniques. By carefully selecting and engineering features, we can emphasize the key characteristics that distinguish each class.

To prevent overfitting our models, we evaluated the performance of the CAE by analyzing accuracy curves. Figure 3 provides a visual representation of accuracy curves. Figures 3a–d show the accuracy of the ADL-9, Fall-8, AF-2, and Full-17 categories, respectively, demonstrating the convergence of training and validation data towards 1. This convergence indicates a steady improvement in accuracy with each epoch. While Figure 4 represents the loss curves, Figures 4a–d represent the loss graphs for the following three sub-datasets: ADL-9, Fall-8, AF-2, and Full-17, respectively.

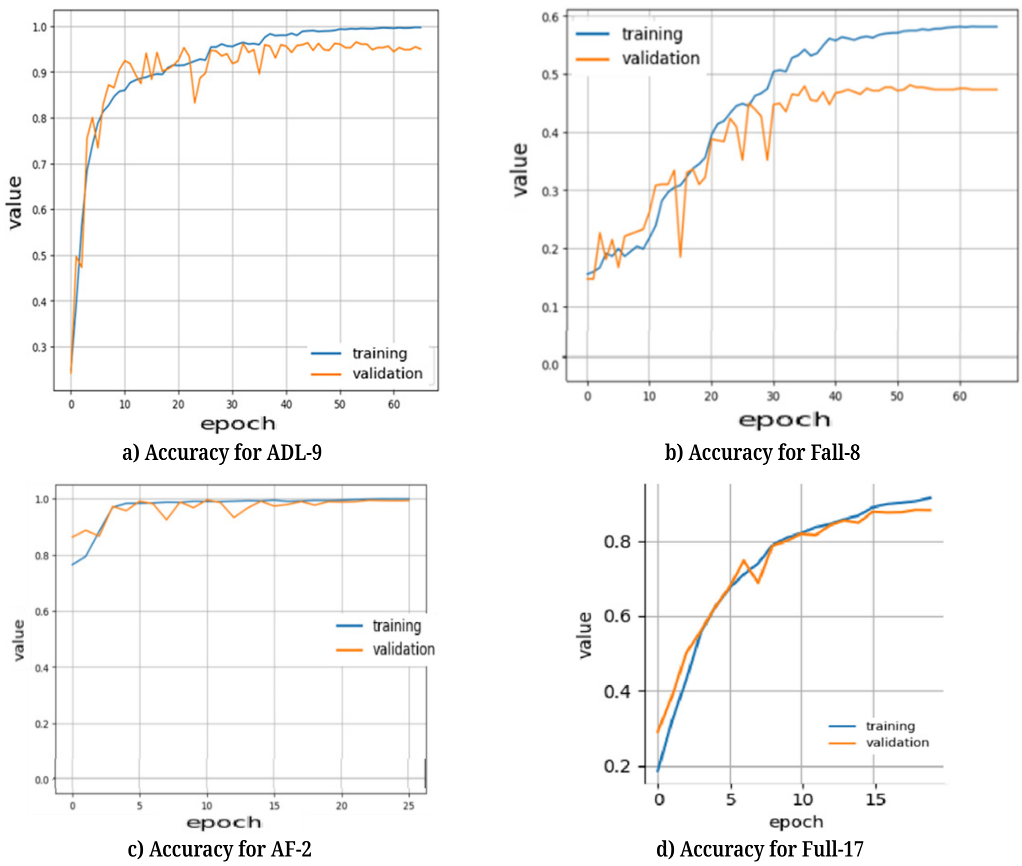
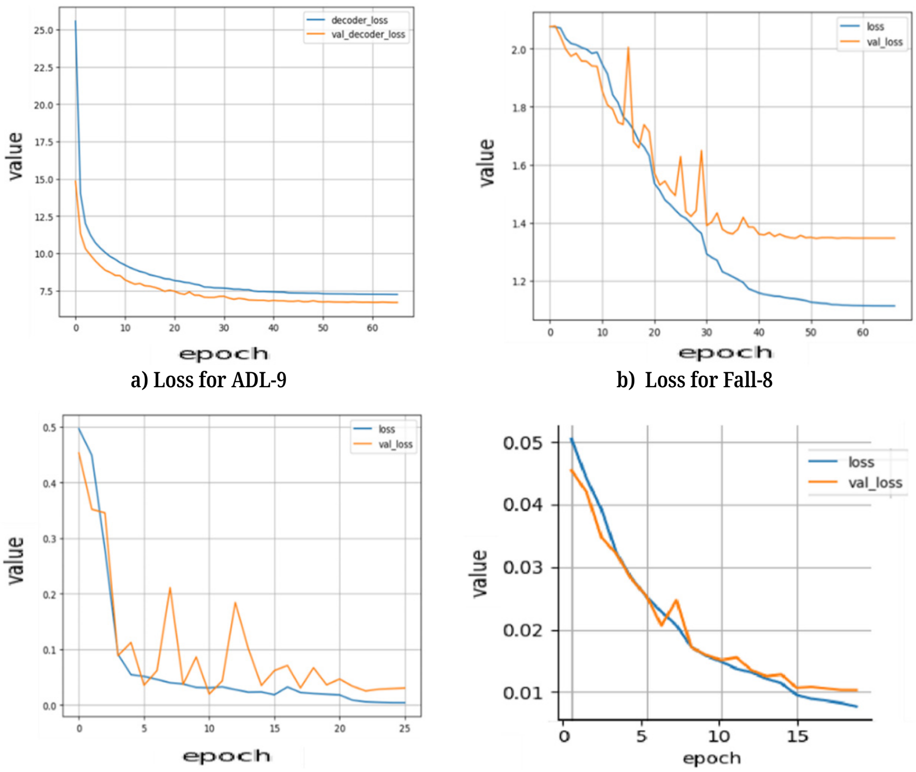


Fig. 3. Accuracy curve



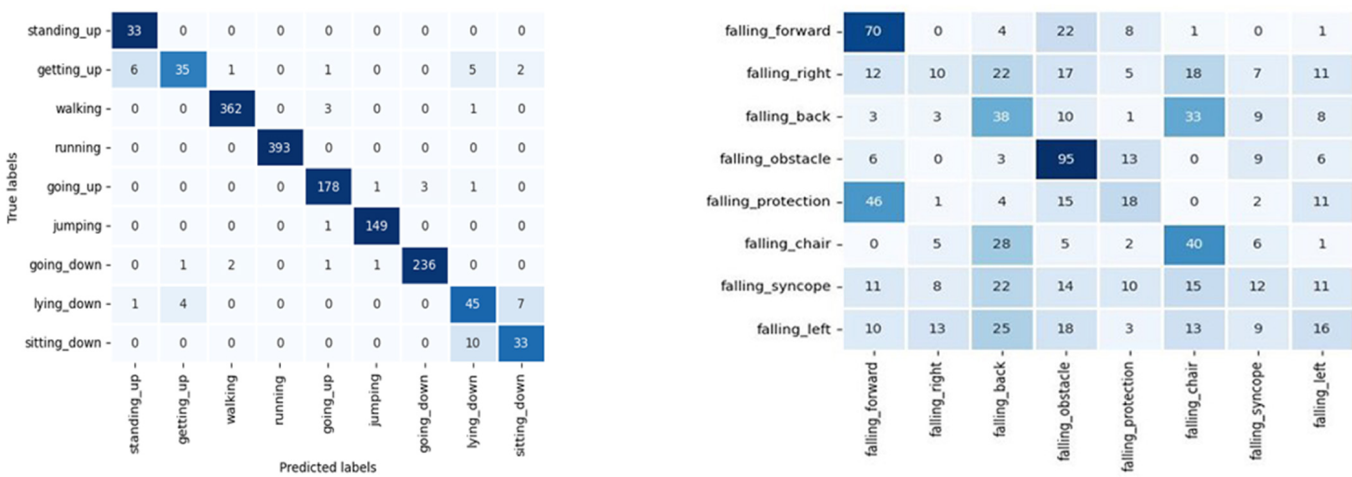
a) Loss for ADL-9 b) Loss for Fall-8

c) Loss for AF-2 d) Loss for Full-17

Fig. 4. Loss curve

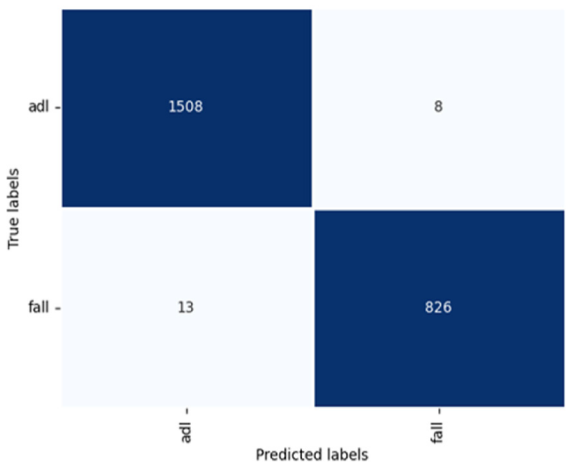
Furthermore, the training loss and validation loss curves exhibit a consistent decrease with each epoch while maintaining a relatively small difference between them. The convergence of accuracy and the decrease in loss indicate that our models learn from the data efficiently without overfitting.

Figure 5 displays the confusion matrix used to evaluate performance by comparing the algorithm's predictions with the actual data labels. Figure 5a depicts the confusion matrix for ADL-9, while Figure 5b illustrates the confusion matrix for Fall-8. In the same manner, Figure 5c displays the confusion matrix for AF-2, and finally, Figure 5d presents a confusion matrix for Full-17.

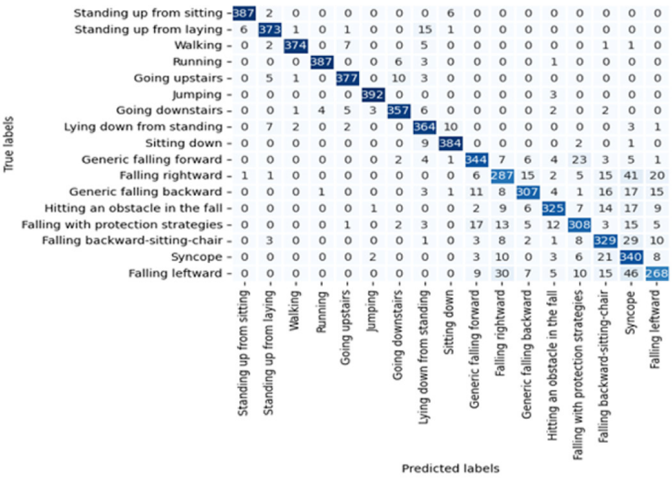


a) Confusion matrix for ADL-9 b) Confusion matrix for Fall-8

Fig. 5. *(Continued)*



c) Confusion matrix for AF-2



d) Confusion matrix for Full-17

Fig. 5. Confusion matrix for AE model

For the AF-2 dataset, Figure 5c shows that 1508 activities are classified correctly. This indicates that the model has correctly identified the ADLs, but there are 13 ADL activities that are incorrectly classified as falls. There are 826 fall activities correctly classified as such, and there are 8 actual images of fall that are incorrectly classified as ADLs. Figure 5a depicts the confusion matrix for the ADL-9. The diagonal indicates that the classes are well classified. Outside the diagonal, there are no significant values for false positives and false negatives, except for the sitting-down activity, which has 10 false positives. In the case of the Full-17 dataset, Figure 5d indicates that the diagonal contains logical values. This suggests that the model has accurately identified the various activities within their respective classes, even though some activities may have been misclassified into other classes.

For the Fall-8 subset, the performance is very low despite applying a range of pre-processing techniques. This can be explained by the fact that all fall activities are symmetrical, making it difficult to differentiate between them. Additionally, when working with the magnitude calculator, the axes do not provide the possibility of determining the direction of the fall.

Table 9 presents a comprehensive analysis of the performance of the autoencoders used in current research and the autoencoder we have proposed in this study. The results clearly demonstrate that our suggested model outperformed existing approaches. The results obtained also demonstrate that our proposed model outperformed existing approaches and achieved superior performance, attributed to the application of various preprocessing techniques and the proposed architecture layers.

Table 9. AE performance comparison

Ref.	Model	Dataset	Performance
[8]	AE	Full-17	F1-scores : 68.37% (T = 32), 68.24% (T = 64) and 68.39% (T = 96).
[9]	AE-RE	ADL-9	Accuracy: 94.70%
		Fall-8	Accuracy: 55.20%
[10]	VAE	Full-17	Accuracy: 24.01%

(Continued)

Table 9. AE performance comparison *(Continued)*

Ref.	Model	Dataset	Performance
[11]	AE	Full-17	Accuracy: 65.67%
[13]	Multiclass Auto-encoder-Based Active Learning (MAAL)	ADL+ Fall in single class (10 classes)	Accuracy: 98.66%
The proposed approach	CAE	Full-17	Accuracy:88.47%
			F1-score: 88.47%
		ADL-9	Accuracy: 96.56%
		Fall-8	Accuracy: 48.86%
		AF-2	Accuracy: 98.86%

5 CONCLUSION AND FUTURE WORKS

This study aims to enhance the performance of autoencoders using the FULL-17 dataset. The proposed model architecture comprises three main elements: an encoder with convolutional layers and pooling, a decoder with deconvolutional layers, and a classifier consisting of a GRU model that processes the output as a sequence, followed by a dense layer for classification. By implementing preprocessing techniques such as calculating magnitude, reducing dimensionality using PCA, normalizing, and balancing data using SMOTEEN and ADASYNE, the performance of the CAE model has significantly improved. By systematically integrating these strategies, the approach achieved a higher classification accuracy of 88.47% on the FULL-17 dataset. The model achieved an accuracy of 96.56% on the ADL-9 sub-dataset and 98.86% on the AF-2 sub-dataset. The proposed CAE achieved only 48.86% accuracy in the Fall-8 sub-dataset, which is still very low. Further investigation should be conducted. In our future work, we plan to concentrate on DL techniques and their application to eHealth classification problems.

6 REFERENCES

[1] S. Pinitkan and N. Wisitpongphan, "Abnormal activity detection and notification platform for real-time ad hoc network," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 16, no. 15, pp. 45–63, 2020. https://doi.org/10.3991/ijoe.v16i15.16065

[2] C. Free *et al.*, "The effectiveness of mobile-health technology-based health behaviour change or disease management interventions for health care consumers: A systematic review," *PLoS Medicine*, vol. 10, no. 1, p. e1001362, 2013. https://doi.org/10.1371/journal.pmed.1001362

[3] B. Zaabar, O. Cheikhrouhou, F. Jamil, M. Ammi, and M. Abid, "HealthBlock: A secure blockchain-based healthcare data management system," *Computer Networks*, vol. 200, p. 108500, 2021. https://doi.org/10.1016/j.comnet.2021.108500

[4] S. Ramasamy Ramamurthy and N. Roy, "Recent trends in machine learning for human activity recognition—A survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, 2018. https://doi.org/10.1002/widm.1254

[5] D. Micucci, M. Mobilio, and P. Napoletano, "UniMiB SHAR: A dataset for human activity recognition using acceleration data from smartphones," *Applied Sciences*, vol. 7, no. 10, p. 1101, 2017. https://doi.org/10.3390/app7101101

[6] C. Zaoui, F. Benabbou, and A. Ettaoufik, "Edge-fog-cloud data analysis for eHealth-IoT," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 19, no. 7, pp. 184–199, 2023. https://doi.org/10.3991/ijoe.v19i07.38903
[7] D. Pham and T. Le, "Auto-encoding variational bayes for inferring topics and visualization," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020. https://doi.org/10.18653/v1/2020.coling-main.458
[8] J. Long, W. Sun, Z. Yang, and O. I. Raymond, "Asymmetric residual neural network for accurate human activity recognition," *Information*, vol. 10, no. 6, p. 203, 2019. https://doi.org/10.3390/info10060203
[9] E. Boutellaa, "Detecting falls with recurrent autoencoders and body acceleration data," in *6th International Conference on Image and Signal Processing and Their Applications (ISPA)*, 2019. https://doi.org/10.1109/ISPA48434.2019.8966875
[10] H. Qian, S. J. Pan, and C. Miao, "Latent independent excitation for generalizable sensor-based cross-person activity recognition," in *Proc.AAAI Conference on Artificial Intelligence*, 2021, vol. 35, no. 13. https://doi.org/10.1609/aaai.v35i13.17416
[11] F. Li, K. Shirahama, M. Nisar, L. Köping, and M. Grzegorzek, "Comparison of feature learning methods for human activity recognition using wearable sensors," *Sensors*, vol. 18, no. 3, p. 679, 2018. https://doi.org/10.3390/s18020679
[12] Y. Hu, "BSDGAN: Balancing sensor data generative adversarial networks for human activity recognition," in *2023 International Joint Conference on Neural Networks (IJCNN)*, 2023. https://doi.org/10.1109/IJCNN54540.2023.10191928
[13] H. Park, G. H. Lee, J. Han, and J. K. Choi, "Multiclass autoencoder-based active learning for sensor-based human activity recognition in Iot applications," *SSRN Preprint,* 2023. https://doi.org/10.2139/ssrn.4469523
[14] T. Lv, X. Wang, L. Jin, Y. Xiao, and M. Song, "Margin-based deep learning networks for human activity recognition," *Sensors*, vol. 20, no. 7, p. 1871, 2020. https://doi.org/10.3390/s20071871
[15] W. Huang, L. Zhang, Q. Teng, C. Song, and J. He, "The convolutional neural networks training with channel-selectivity for human activity recognition based on sensors," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 10, pp. 3834–3843, 2021. https://doi.org/10.1109/JBHI.2021.3092396
[16] T. Stampfler, M. Elgendi, R. R. Fletcher, and C. Menon, "The use of deep learning for smartphone-based human activity recognition," *Frontiers in Public Health*, vol. 11, p. 1086671, 2023. https://doi.org/10.3389/fpubh.2023.1086671
[17] X. Wang, X. Wang, T. Lv, L. Jin, and M. He, "HARNAS: Human activity recognition based on automatic neural architecture search using evolutionary algorithms," *Sensors*, vol. 21, no. 20, p. 6927, 2021. https://doi.org/10.3390/s21206927
[18] Y. Tang, Q. Teng, L. Zhang, F. Min, and J. He, "Layer-wise training convolutional neural networks with smaller filters for human activity recognition using wearable sensors," *IEEE Sensors Journal*, vol. 21, no. 1, pp. 581–592, 2021. https://doi.org/10.1109/JSEN.2020.3015521
[19] A. Ferrari, D. Micucci, M. Mobilio, and P. Napoletano, "Personalization in human activity recognition," *arXiv*, 2020. https://doi.org/10.48550/arXiv.2009.00268
[20] F. Jeannot, "Norme et fonction norme de Frobenius," 2023.
[21] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv*, 2016. https://doi.org/10.48550/arXiv.1607.06450
[22] S. Aksoy and R. M. Haralick, "Feature normalization and likelihood-based similarity measures for image retrieval," *Pattern Recognition Letters*, vol. 22, no. 5, pp. 563–582, 2001. https://doi.org/10.1016/S0167-8655(00)00112-4
[23] L. C. Paul and A. A. Sumam, "Face recognition using principal component analysis method," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 1, no. 9, pp. 135–139, 2012.

[24] A. Singh, R. K. Ranjan, and A. Tiwari, "Credit card fraud detection under extreme imbalanced data: A comparative study of data-level algorithms," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 34, no. 4, pp. 571–598, 2021. https://doi.org/10.1080/0952813X.2021.1907795

[25] A. Muaz, M. Jayabalan, and V. Thiruchelvam, "A comparison of data sampling techniques for credit card fraud detection," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 6, 2020. https://doi.org/10.14569/IJACSA.2020.0110660

[26] I. Dey and V. Pratap, "A comparative study of SMOTE, borderline-SMOTE, and ADASYN oversampling techniques using different classifiers," in *3rd International Conference on Smart Data Intelligence (ICSMDI)*, 2023. https://doi.org/10.1109/ICSMDI57622.2023.00060

[27] G. Zhang, Y. Liu, and X. Jin, "A survey of autoencoder-based recommender systems," *Frontiers of Computer Science*, vol. 14, no. 2, pp. 430–450, 2019. https://doi.org/10.1007/s11704-018-8052-6

[28] M. Sewak, S. K. Sahay, and H. Rathore, "An overview of deep learning architecture of deep neural networks and autoencoders," *Journal of Computational and Theoretical Nanoscience*, vol. 17, no. 1, pp. 182–188, 2020. https://doi.org/10.1166/jctn.2020.8648

[29] G. San Martin, E. López Droguett, V. Meruane, and M. das Chagas Moura, "Deep variational auto-encoders: A promising tool for dimensionality reduction and ball bearing elements fault diagnosis," *Structural Health Monitoring*, vol. 18, no. 4, pp. 1092–1128, 2018. https://doi.org/10.1177/1475921718788299

[30] S. W. A. Canchumuni, A. A. Emerick, and M. A. C. Pacheco, "Towards a robust parameterization for conditioning facies models using deep variational autoencoders and ensemble smoother," *Computers & Geosciences*, vol. 128, pp. 87–102, 2019. https://doi.org/10.1016/j.cageo.2019.04.006

[31] A. Ng, "Sparse autoencoder," Lecture Notes CS294A, Standford Univ., Stanford, CA, USA, 2011.

[32] X. Guo, X. Liu, E. Zhu, and J. Yin, "Deep clustering with convolutional autoencoders," in *Lecture Notes in Computer Science*, 2017, pp. 373–382. https://doi.org/10.1007/978-3-319-70096-0_39

7 AUTHORS

Chaimae Zaoui has a Master's degree in Data Science and Big Data from Hassan II University in Casablanca, Morocco in 2021. Currently, she is pursuing PhD at the Laboratory of Information Processing and Modeling (LTIM) at the Ben M'sik Faculty of Science. Her research interests include Internet of Things, cloud computing, fog computing, edge computing, machine learning, and deep learning (E-mail: chaimae.zaoui1-etu@etu.univh2c.ma).

Faouzia Benabbou is a Professor of Computer Science in the Department of Mathematics and Computer Science at the Ben M'Sick Faculty of Science, Hassan II University of Casablanca. She received Ph.D. in Computer Science from the Faculty of Sciences, University Mohamed V, Morocco, in 1997. She is a member of the Computer Science and Modeling Laboratory and head of the team of Cloud Computing, Network and Systems Engineering (ICCNSE) team. Her research interests include cloud computing, data mining, machine learning, and natural language processing (E-mail: faouzia.benabbou@univh2c.ma).

Abdelaziz Ettaoufik is a Professor at Department of Mathematics and Computer Science Faculty of sciences S Ben M'SIK Hassan II University Casablanca. His research interest includes big data, cloud computing and security, IoT, blockchain, IA, and smart cities (E-mail: abdelaziz.ettaouik@etu.univh2c.ma).

Khadija Sabiri has completed Ph.D. degree in the cloudification legacy system to a cloud-native application from the Science Faculty of Ben M'sik in Casablanca-Morocco. She moved to the University of Beira Interior in Portugal as Postdoc researcher as part of cloud computing competence center-C4 projects. Her overarching research is to explore rigorous software development methodologies and cloud-based technology solutions that increase and guarantee citizen rights, such as privacy, transparency (E-mail: khadija.sabiry@gmail.com).