IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 12 July 2022, accepted 24 July 2022, date of publication 8 August 2022, date of current version 18 August 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3197618*

RESEARCH ARTICLE

# Future Activities Prediction Framework in Smart Homes Environment

**MAI MOHAMED, AYMAN EL-KILANY, AND NEAMAT EL-TAZI**
Department of Information Systems, Faculty of Computers and Artificial Intelligence, Cairo University, Giza 12613, Egypt
Corresponding author: Mai Mohamed (mai.ismail.researcher@gmail.com)

**ABSTRACT** Smart homes have been recently important sources for providing Activity of Daily Living (ADL) data about their residents. ADL data can be a great asset while analyzing residents' behavior to provide residents with better and optimized services. A popular example is to analyze residents' behavior to predict their future activities and optimize smart homes performance accordingly. This paper proposes a forecasting framework that utilizes ADL data to predict residents' next activities in a smart home environment. Forecasting is performed via the conjunction of embedding algorithm to encode the data and Bidirectional Long Short-Term Memory (BiLSTM) deep neural networks to process the data. The proposed framework is evaluated over five real ADL datasets where the experiments show the outperformance of the proposed framework with accuracy scores ranging from 98.7% to 93.8%.

**INDEX TERMS** Smart home, human activity recognition, BiLSTM neural networks, sequence prediction.

## I. INTRODUCTION

Smart homes utilize Internet of Things (IoT) concept to provide residents a more comfortable experience. Smart home systems are built around the concept of monitoring and controlling devices using sensors [1] where those sensors are connected through wireless networks to collect Activity of Daily Living (ADL) data. Activity of Daily Living data is the data that describes the resident activity such as watching TV or eating lunch. ADL data can be used to recognize and predict the behavior of residents and their activities, which can be utilized in various applications such as healthcare [2], energy consumption [3], recommendation systems, and anomaly detection [4].

To comprehend and detect residents' ADLs, Human Activity Recognition (HAR) approaches are developed, which involve monitoring and analyzing residents' behavior. Human Activity Recognition (HAR) is the process of identifying the correct activity that has been performed by residents in smart homes. There are three main types of HAR, sensor-based, vision-based, and radio-based [5]. Sensor-based type depends on data collected from sensors to detect human activities. For example, the activation of a light sensor is indicating movement in certain area or related to certain activity. Vision-based type depends on data formats as images and videos to detect human activities. For example, movements in smart home videos that could indicate doing certain activities such as walking or cooking. Radio-based type depends on signals' information and characteristics to detect human activities. For example, wearable motion sensors that detect body parts movement could indicate doing certain activities such as walking or sitting.

Another important field of study related to ADLs is called Human Activity Prediction (HAP), which mainly relies on machine learning techniques to forecast human activities from historical data. HAP is used in various applications such as video surveillance, crime prevention, and health care systems. HAP also identifies and predicts future activities in order to collect knowledge about the resident's experience and behavior [6].

Human activity recognition and predication can be used to help elderly people living alone during their daily lives. HAR and HAP act as early warning system when abnormal activities are detected or predicted [7]. In addition, HAR and HAP can be beneficial in energy conservation models [8].

Recognizing and predicting complex human activities in connected environments can be vital for users inside these environments. Users can benefit from various applications

The associate editor coordinating the review of this manuscript and approving it for publication was Akin Tascikaraoglu.

related to security, performance and efficiency while managing smart homes resources such as electricity and water. For Example, elder, ill or disabled people who live alone can benefit greatly from predicting their next activities and providing some type of home automation. In addition, home automation can be very beneficial in energy consumption model. This research is motivated by such needs to predict users' next activities in smart home environment with high quality.

This paper leverages deep learning to detect and predict smart home future activities. It proposes sensor-based HAR model as the first phase and HAP model as the second phase of future activities prediction framework. The proposed framework utilizes Bidirectional Long Short Memory (BiLSTM) [9] deep neural networks in conjunction with word2vec [10], [11] encoding models to enhance HAR and HAP in sensor-based environment. BiLSTM deep neural network was used given its benefits in learning sequence data while taking into consideration data dependency. The contributions of our work are summarized as follows. We propose a novel BiLSTM with Word2Vec NLP Embedding Technique for sensor-based HAP. The effect of the contribution is measured through a comprehensive evaluation. An extensive evaluation for our proposed framework was performed on several benchmark datasets.

The rest of this paper is organized as follows. Section 2 reviews related work while Section 3 presents the proposed framework for activity recognition and prediction in smart homes. Section 4 reports the experimental results followed by a discussion. Finally, Section 5 presents conclusion and future work.

## II. BACKGROUND AND RELATED WORK

This section presents the background and related work required to formulate the research problem and the proposed framework. First, background is presented to give an overview about word embedding, neural networks, recurrent neural networks. Second, Human Activity Recognition and prediction models in the literature are presented.

### A. BACKGROUND

Machine learning (ML) techniques are playing a critical role in discovering knowledge from data and providing a reliable recognition and prediction of human behavior. A neural network is one of ML techniques which provides a good performance in comparison to other techniques when it comes to prediction [12]. A neural network consists of one input layer, one output layer, and one or multiple hidden layers. Each layer consists of a number of units called neurons. Each neuron has a function over the weighted sum of its inputs called activation function such as sigmoid, relu, tanh, and other functions. Using training data, the weights of this weighted sum are learned through optimizer such as RMSprop, Stochastic Gradient Descent (SGD), AdamaX, and other optimizers [13].

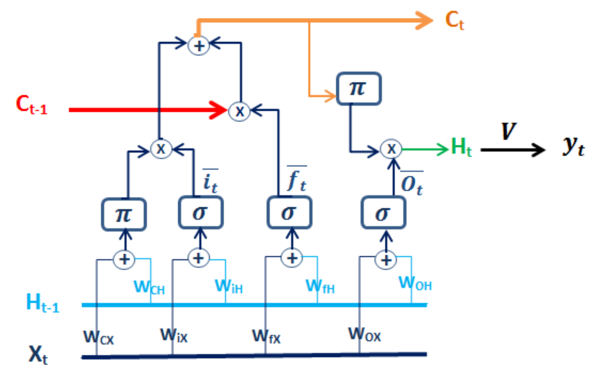


**FIGURE 1. LSTM Architecture.**

A Recurrent Neural Network (RNN) [14] is a type of neural network that includes cyclic connections between different layers, which adds state (i.e., memory) to the model and gives it the ability to "remember" past information. In other words, the training in hidden units depends on the values of the previous units to process the inputs in the given timestamp and return the output.

Given $X_t$ is the input at timestamp t, $H_t$ is the hidden state at timestamp t and contains all historical information up to t. Authors in [15] formulated the function that is performed in each unit in the hidden layer as presented in eq. (1) where F is the activation Function, U and W are vectors of weights over the new inputs and the hidden state respectively.

$$H_t = F(UX_t + WH_{t-1}) \tag{1}$$

A special type of RNN is Long Short-Term Memory (LSTM) network but more complex than RNN because it uses complex unit which called a memory cell [16]. LSTM is considered an enhanced network of RNN that includes a memory cell state $C_t$ which stores information and values from previous states but for longer period than RNN.

As illustrated in fig. 1, LSTM has three types of gates which are input, forget, and output. Input gate $i_t$ as presented in eq. (4) has sigmoid function presented in eq. (2) that takes the previous hidden state and the current input, then decides which information of input vector $X_t$ and hidden unit $H_{t-1}$ should pass to update the cell state $C_t$ presented in eq. (6) which uses tanh function presented in eq. (3). The forget gate $f_t$ as presented in eq. (5) determines whether to forget or keep the information of previous cell state $C_{t-1}$. The output gate $O_t$ as presented in eq. (7) controls the flow of information from the current cell state to the hidden state. The mathematical equations eqs. (2) to (8) [12] show the details of the LSTM gates, where the symbol $\odot$ represents element-wise vectors multiplication and W is the weights vector.

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

$$tanh(x) = \frac{e^x + e^{-x}}{e^x - e^{-x}} \tag{3}$$

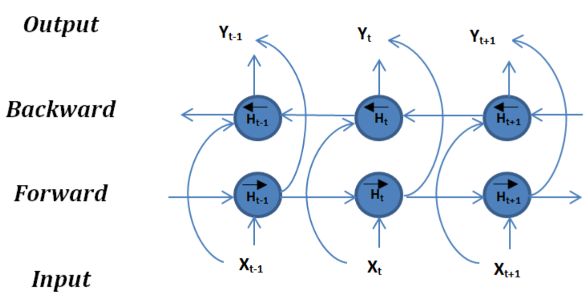


**FIGURE 2.** Bidirectional LSTM (BiLSTM) architecture [17].

$$i_t = sigmoid(W_{ix}X_t + W_{iH}H_{t-1} + b_i) \tag{4}$$
$$f_t = sigmoid(W_{fx}X_t + W_{fH}H_{t-1} + b_f) \tag{5}$$
$$C_t = C_{t-1} \odot f_t + i_t \odot tanh(W_{Cx}X_t + W_{CH}H_{t-1} + b_C) \tag{6}$$
$$O_t = sigmoid(W_{Ox}X_t + W_{OH}H_{t-1} + b_O) \tag{7}$$
$$H_t = O_t \odot tanh(C_t) \tag{8}$$

The BiLSTM [17], [18], [19] is a type of LSTM network that includes two parallel LSTM: forward and backward. It is a neural network that can extract knowledge from past and future sequences to produce an output. Fig. 2 shows the BiLSTM architecture that was introduced in [17]. BiLSTM provides better accuracy than LSTM because the output layer receives information from past nodes and future nodes [20].

The forward LSTM processes information from left to right and its hidden state can be shown as $\overrightarrow{H_t} = LSTM(X_t, H_{t-1})$, while the backward LSTM processes information from right to left and its hidden state can be shown as $\overleftarrow{H_t} = LSTM(X_t, H_{t+1})$. The output of BiLSTM for the hidden state can be summarized by concatenating the forward and backward states as $H_t = [\overrightarrow{H_t}, \overleftarrow{H_t}]$. Mathematical equations eqs. (9) to (19) provide details of the three gates: input, forget, and output gates which are similar to regular LSTM as eqs. (2) to (8) but hidden state $H_t$ calculated based on forward and backward LSTM.

**Forward LSTM**

$$i_t = sigmoid(W_{ix}X_t + W_{iH}H_{t-1} + b_i) \tag{9}$$
$$f_t = sigmoid(W_{fx}X_t + W_{fH}H_{t-1} + b_f) \tag{10}$$
$$C_t = C_{t-1} \odot f_t + i_t \odot tanh(W_{Cx}X_t + W_{CH}H_{t-1} + b_C) \tag{11}$$
$$O_t = sigmoid(W_{Ox}X_t + W_{OH}H_{t-1} + b_O) \tag{12}$$
$$\overleftarrow{H_t} = O_t \odot tanh(C_t) \tag{13}$$

**Backward LSTM**

$$i_t = sigmoid(W_{ix}X_t + W_{iH}H_{t+1} + b_i) \tag{14}$$
$$f_t = sigmoid(W_{fx}X_t + W_{fH}H_{t+1} + b_f) \tag{15}$$
$$C_t = C_{t+1} \odot f_t + i_t \odot tanh(W_{Cx}X_t + W_{CH}H_{t+1} + b_C) \tag{16}$$
$$O_t = sigmoid(W_{Ox}X_t + W_{OH}H_{t+1} + b_O) \tag{17}$$
$$\overrightarrow{H_t} = O_t \odot tanh(C_t) \tag{18}$$

**Hidden State Update**

$$H_t = [\overrightarrow{H_t}, \overleftarrow{H_t}] \tag{19}$$

In recent years, there have been significant improvements of deep learning techniques, as they have been successfully applied to natural language processing (NLP). NLP was used in smart home applications in order to provide remote monitoring and controlling of smart appliances [21]. Natural Language Processing (NLP) is used in various domains, including text classification [22]. Word embedding is a text classification technique that can be obtained using neural networks, where words are represented as real-valued vectors based on their context in natural language before using them through different models. Different word embedding algorithms are used to build vectors such as word2vec, GloVe [23] and term frequency [24].

Word2Vec technique [10] utilizes a neural network with a single hidden layer in order to learn the word representations by involving both target word and context words. Word2vec offers two approaches: Continuous Bag of Words (CBOW) and Skip-gram [10]. CBOW learns the representations by predicting the current word based on its context while Skip-gram learns representations by predicting surrounding words given the current word.

### B. RELATED WORK

Recent research aims to improve humans' quality of living by addressing their needs within a smart home environment. Consequently, recognition and prediction of human activities has become an important research field with a wide range of applications. One of these applications presented in [25] used Markov Logic Network to recognize the residents' profiles based on their activities and preference. A resident's profile might contain: resident role (father, mother, son…), gender (male or female), age range (young, middle-age, etc.), and job (worker, student, doctor, etc.). The residents' profiles can be used to improve the performance of recognition and prediction tasks.

#### 1) HUMAN ACTIVITY RECOGNITION

Machine Learning (ML) techniques have been used to effectively solve human activity recognition problem. For instance, the widespread of smartphones and wearable accelerometer sensors could be used to build Bayesian stream-based active learning and Conditional Restricted Boltzmann Machine (CRBM) classifier that could label sensory sequential data as proposed in [5].

Various algorithms such as Decision Tree (DT) [26], Naive Bayes (NB) [27], Support Vector Machine (SVM) [28], K-Nearest Neighbor (KNN) [29], and Logistic Regression (LR) [30] are widely used in human activity recognition. Some of these algorithms were used beside Linear Discriminant Analysis (LDA) [31] and Ensemble Learning (EL) [32] to recognize activities of elderly people who are living alone as proposed in [7]. The proposed model was evaluated using CASAS datasets [33] where it showed 90% accuracy.

In [34], the authors used DT, KNN, NB, LR as a classifier chain that was accumulated through Majority Voting Ensemble classifier to solve multi-resident activity recognition problem. While evaluating the model using ARAS datasets [34], the experimental results showed that the proposed model achieved 90% average accuracy.

A Probabilistic neural network (PNN) was integrated with H2O autoencoder approach for recognizing the activities performed in a smart home, then separating the normal from the anomalous activities [4]. Although the model provided an overall accuracy of 90%, PNN also needs a lot of memory size to be implemented. Convolutional Neural Network (CNN), Conditional Random Fields (CRF), and Hidden Markov Model (HMM) were used in [35] and [36] over the Kasteren datasets [36] to recognize activity label. The main limitation of their work is that the model encapsulated the events in one input matrix for each activity without considering the order of these events, which could differ according to human nature.

Several studies have been utilized Convolutional Neural Network (CNN) as in [37], a layer-wise training CNN with local loss for HAR using wearable sensors, which was evaluated using different datasets UCI HAR dataset, OPPORTUNITY dataset, UniMib-SHAR dataset, PAMAP dataset, and WISDM dataset. Similar models were evaluated using same datasets, which considered cross-channel communication in HAR scenario [38]. Learned offsets and feature amplitudes were added into standard convolution in [39]. In addition, in [40], multi-CNNs were integrated with attention mechanism to enhance HAR task. Otherwise, in [41], a multi-branch CNN is proposed, which utilized a selective kernel mechanism for HAR.

Different deep learning (DL) models such as LSTM, Uni-LSTM, BiLSTM, Casc-LSTM, Ens2-LSTM, and CascEns-LSTM were used in [42] to learn how to recognize human activities. The proposed algorithms were compared to Conditional Random Fields (CRF) and Hidden Markov Model (HMM) results over CASAS datasets. The experiments showed that LSTM provides the best accuracy.

#### 2) HUMAN ACTIVITY PREDICTION

Human Activity Prediction (HAP) models have suffered from less attention than activity recognition in literature. HAP models are important to many real applications such as energy consumption [43], [44] prediction which could save a lot of smart homes energy.

In [45], the authors proposed a new hybrid HAP model using HMM (Hidden Markov Model) and SVM (Support Vector Machines). The experiment's results were compared with Hybrid model of HMM (Hidden Markov Model) and MLP (Multi-Layer Perceptron). These two approaches were invented to recognize ADLs from home environments using binary sensors' datasets. They used Kasteren datasets [36] and OrdoneZ datasets to evaluate the two approaches. The HMM-SVM approach showed better results with almost 67% average F-Score.

Furthermore, LSTM proved that it is the most efficient algorithm in activity prediction either used alone or with other algorithms. This was clear in multiple cases whether it was to predict the next activity [6], predict the sensors designated to the next activity [8], or predict the time that will elapse until the next event [12].

Another direction is to detect activity via object usage as presented in [46]. Passive RFID Tags and LSTM are used over OrdoneZ datasets to recognize and predict the activity label. Similar to previous research [45], a sensor-based model was presented in [47] where authors have managed to predict the next activity and the timestamp of next sensor event using LSTM. They used different datasets such as: CASAS [33], MITB [48], and Kasteren datasets [36] to evaluate their work. The experiments showed that the best accuracy was 52.9%. Their framework was separated into two phases. In the first phase, data was preprocessed and cleaned. Fast Fourier transform was used in order to capture the frequency characteristics. Data was then divided into sequences of the same size using sliding window. In the second phase, LSTM was trained to predict the next activity.

In addition, the authors in [49] implemented different algorithms such as ALZ, LZ78, and LSTM to predict the next activity performed in a smart home. Their proposed model achieved 54.6% accuracy while evaluating using Aruba dataset and 45.4% while using Cairo dataset. Another similar study [50] proposed complex approaches but its best approach achieved 82.35% accuracy while evaluating using Milan and Aruba datasets. Authors in [51] have utilized feature engineering to generate a strong set of features before training their model on them. Their approach managed to achieve an average accuracy of 89%. Table 1 summarizes the most related HAP models to our work showing their features, algorithms and their accuracy's results.

All previous models have considered HAR and HAP as separate tasks where each model has its own perspective while processing the ADL data. Motivated by the successful results of the LSTM-based network demonstration for several recognition and prediction applications, this paper utilizes LSTM to combine HAR and HAP in a unified framework in order to predict future activities with high accuracy.

## III. PROPOSED FRAMEWORK

In this paper, we aim to enhance the performance of methods that recognize and predict human activities in smart home environment. Consequently, the proposed framework tackles the two problems of activity recognition and prediction in sequence where the output of recognition algorithm is used

**TABLE 1. Summary of results, features and methods used in the literature papers for activity prediction.**

| Ref | Features | Methods / Algorithms | Datasets | Performance |
|---|---|---|---|---|
| [6] | • Graphical features (videos of the activities) | 3D CNN + LSTM | • MSR Daily Action Dataset<br>• UCF101 Dataset | Accuracy = 93.7% |
| [46] | • Object usage<br>• Activity class | LSTM + Object-usage | • Ordonez Dataset | Accuracy = 78% |
| [47] | • Sensor signals | LSTM | • CASAS Datasets<br>• Kasteren Dataset<br>• MITB Dataset | Accuracy = 52% |
| [49] | • Duration of activity<br>• Count of the sensor events within an activity<br>• Count of activity times performed per day<br>• Sensor states | DNN +OCD-AE +LSTM | • Aruba Dataset<br>• Cairo Dataset | Accuracy = 57% |
| [50] | • Activities ordered by activity time | CRF<br>HMM | • Aruba Dataset<br>• Milan Dataset | Accuracy = 82% |
| [51] | • Activity class<br>• Place where the activity occurs<br>• Hour of the day when the activity occurs<br>• Day of the week when the activity occurs<br>• Number of Openings of (doors, cupboards, and so on) during the activity<br>• Number of Light events recorded during the activity<br>• Number of Motion events recorded during the activity | PSINES DBN | • HH102 Dataset<br>• HH103 Dataset<br>• HH104 Dataset<br>• HH105 Dataset<br>• HH106 Dataset<br>• Orange4Home Dataset | Accuracy = 89.5% |

as an input for prediction algorithm. Detection problem is formulated as follows. Given several sensors' values that create an event, we need to recognize the human activity that is related to that event and the state of such activity as being in the beginning, middle, or an end of an activity. The prediction algorithm is formulated as follows. Given a number of activities and their states, the proposed framework should predict the next activity/activities that the smart home resident is expected to perform. The proposed framework that is presented in fig. 3 is detailed in the following subsections.

### A. SMART HOME CASAS DATASET AND PREPROCESSING

Researchers employed many datasets in a smart home environment for HAR and HAP solutions. Because of variety, cost and time-consuming challenges that may be faced while collecting real-world data, public benchmark datasets became critical for academic researchers. CASAS smart home project [33], [52], [53] provides several benchmark datasets that contain sequential sensor data collected from a residents' homes during a period of time. Five annotated datasets, named Cairo [54], Kyoto7 [54], Aruba [54], HH104 [54], and Milan [54] were selected among all available CASAS datasets. These datasets are represented as tuples of date and time, the SensorID, the state/value, and the activity label (take medicine, watch TV, etc.).

The Cairo dataset contains sensor data collected from home of a volunteer two residents R1 and R2, during a period of two months. The dataset contains sequential sensor data collected from a residents' home that was equipped with only two kinds of sensors; motion (M) and temperature (T) sensors. The dataset has 13 unique activities, and total number of events 726,500 event.

The Kyoto7 dataset contains sensor data collected from home of a volunteer two residents R1 and R2, during a period of two months with different kinds of sensors such as motion (M), kitchen item (I), door (D), burner (AD1-A), hot water (AD1-B), cold water (AD1-C), and temperature (T) sensors. The dataset has 16 unique activities, and total number of events 138,000 event.

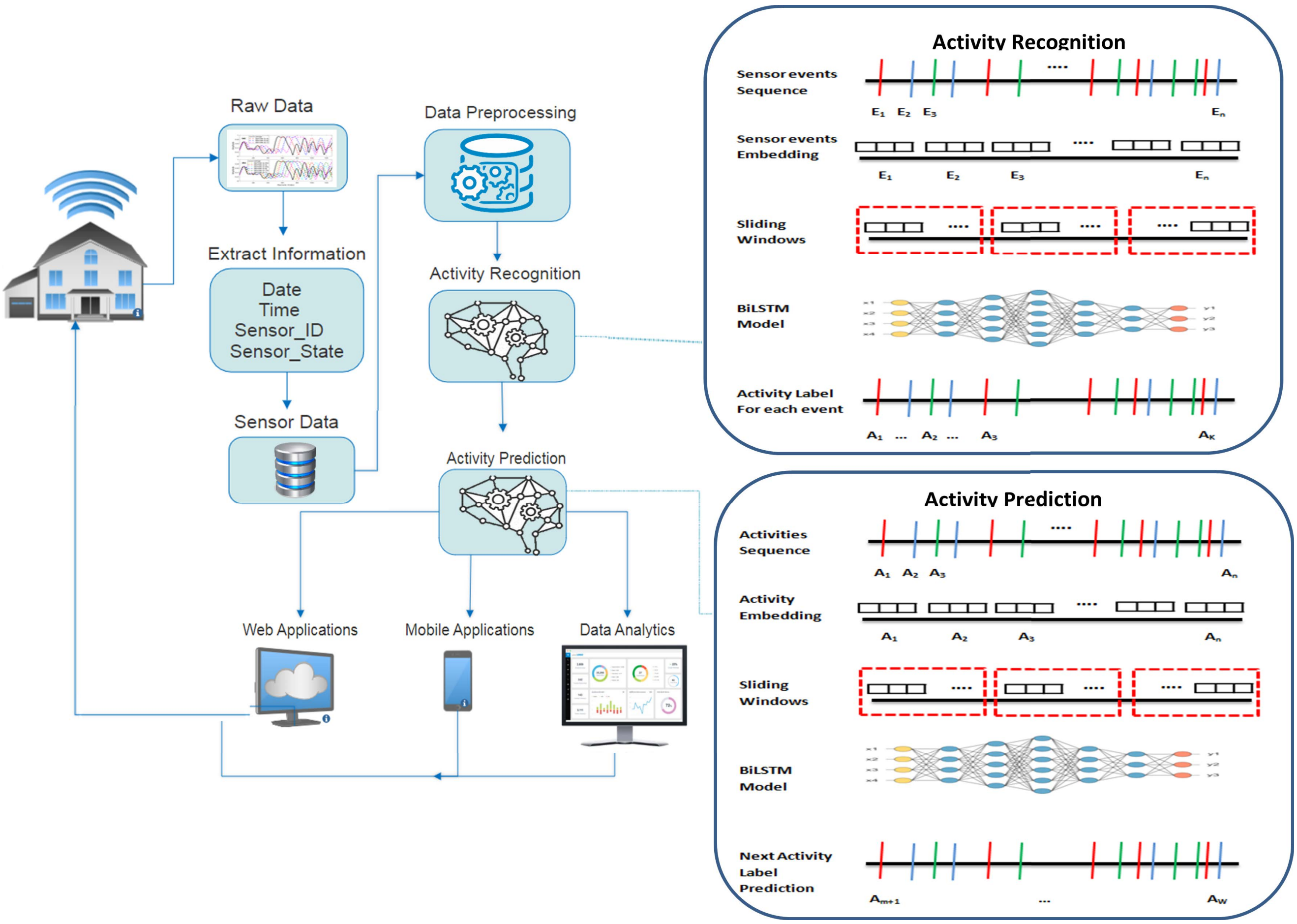


**FIGURE 3.** Proposed framework.

The Aruba dataset contains sensor data collected from home of a volunteer woman, during a period of seven months with different kinds of sensors such as motion (M), door (D), and temperature (T) sensors. The dataset has 11 unique activities, and total number of events 1,719,500 event.

The HH104 dataset contains sensor data collected from home of a volunteer, during a period of two months with different kinds of sensors such as such as door (D), motion (M), temperature (T), and light (LS) sensors. The dataset has 30 unique activities, and total number of events 478,000 event.

The Milan dataset contains sensor data collected from home of a volunteer woman, during a period of three months with different kinds of sensors such as such as motion (M), door (D), and temperature (T) sensors. The dataset has 15 unique activities, and total number of events 433,500 event.

A sample of these datasets is presented in table 2 and the set the unique activities of these datasets are presented in table 3.

The data collected in the datasets can be described as follows. Let Sensors $\in \{BATV001, LS012, \ldots\}$ where Sensors set contains the sensors identifiers/names, and generally formulated as S $= \{S_1, S_2, \ldots, S_Q\}$. Let Sensor State $\in \{ON, OFF, \ldots\}$, and generally formulated as C $= \{C_1, C_2, \ldots, C_B\}$. Let Events $\in \{M021\_OFF, LS013\_7, \ldots\}$, and generally formulated as E $= \{E_1, E_2, \ldots, E_M\}$. Let Activity $\in \{Work, Sleep, Cook, \ldots\}$, and generally formulated as A $= \{A_1, A_2, \ldots, A_N\}$, and Activity State $\in \{Begin, Intermediate, End\}$. Each event (E) consists of

**TABLE 2.** Example of sensors data.

| Date | *Time* | *SensorID* | *State* | *Activity* |
|---|---|---|---|---|
| 2011-06-15 | *00:06:32.834414* | *M021* | *ON* | *Sleep="begin"* |
| 2011-06-15 | *00:06:33.988964* | *M021* | *OFF* | |
| 2011-06-15 | 00:12:32.670631 | BATV012 | 9540 | |
| 2011-06-15 | 00:15:01.957718 | LS013 | 6 | |
| . | . | . | . | |
| . | . | . | . | |
| . | . | . | . | |
| 2011-06-15 | 03:38:28.21206 | M021 | ON | |
| 2011-06-15 | 03:38:29.213955 | MA020 | OFF | |
| 2011-06-15 | 03:38:29.32819 | M021 | OFF | Sleep="end" |

**TABLE 3.** Daily living activities in multiple datasets.

| *HH104* | *Aruba* | *Milan* | *Kyoto7* | *Cairo* |
|---|---|---|---|---|
| Bathe | - | Master_Bathroom | - | - |
| Toilet | - | | | |
| | - | Guest_Bathroom | | |
| Bed_Toilet_Transition | Bed_to_Toilet | Bed_to_Toilet | R1_bed_to_toilet | Bed_to_Toilet |
| | | | R2_bed_to_toilet | |
| Cook | Meal_Preparation | Kitchen_Activity | - | - |
| Cook_Breakfast | | | - | |
| Cook_Dinner | | | R2_prepare_dinner | |
| Cook_Lunch | | | R2_prepare_lunch | |
| Eat | Eating | Dining_Rm_Activity | - | - |
| Eat_Breakfast | | | R1_breakfast | Breakfast |
| | | | R2_breakfast | |
| Eat_Dinner | | | - | Dinner |
| Eat_Lunch | | | - | Lunch |
| Enter_Home | Enter_Home | - | - | - |
| Groom | - | - | R1_groom | - |
| Dress | | | R2_groom | |
| Leave_Home | Leave_Home | Leave_Home | - | Leave_Home |
| Morning_Meds | - | Morning_Meds | - | - |
| Personal_Hygiene | - | - | - | - |
| Watch_TV | Relax | Watch_TV | R2_watch_TV | - |
| Read | | Read | - | |
| Relax | | - | - | |
| Sleep | Sleeping | Sleep | R1_sleep | R1_sleep |
| Sleep_Out_Of_Bed | | | R2_sleep | R2_sleep |
| Take_Medicine | - | Eve_Meds | - | R2_take_medicine |
| Wash_Breakfast_Dishes | Wash_Dishes | - | Cleaning | - |
| Wash_Dinner_Dishes | | | | |
| Wash_Dishes | | | | |
| Wash_Lunch_Dishes | | | | |
| - | Housekeeping | | | Laundry |
| Work | Work | Desk_Activity | R1_work_at_computer | R1_work_in_office |
| | | Chores | R2_work_at_computer | - |
| | | - | R1_work_at_dining_room_table | |
| - | - | Meditate | - | - |
| - | - | - | - | R1_wake |
| | - | - | - | R2_wake |
| Others (O) | Others (O) | Others (O) | Others (O) | Others (O) |

Sensor(S) and the sensor state. Each activity (A) consists of a number of events (E).

Data Preprocessing is a fundamental stage of the proposed framework as it is performed before the recognition phase and the prediction phase. In our framework, we ignore the Date and Time of an event, as we focus on the sequence of the events that happened in a certain period. Consequently, the input features for our framework will be the SensorID, State and the Activity features. In addition, the Activity related to each sensor event is modified to include an activity state; either start of the activity, middle, or the end of the activity (e.g., start of the sleep activity will be "B-Sleep", intermediate of the sleep activity will be "I-Sleep", and the end of sleep activity will be "E-Sleep"). Sensor events that are not in the beginning, the middle, or the end of an activity will be annotated as ("O") activity. Another step is performed to improve the model's performance where "SensorID" and "State" are combined to a new feature "SensorID/State". For instance, if SensorID = M021 and State = ON then the new feature value will be M021_ON. Table 4 shows the features that are used after all preprocessing steps.

### B. PROPOSED FRAMEWORK

The proposed framework contains four main stages: Sensor Events Embedding, BiLSTM Recognition model, Activities

**TABLE 4. Example of Sensor data after ignoring, merging and enhancing features values.**

| *SensorID/State* | *Activity* |
|---|---|
| M021_ON | B_Sleep |
| M021_OFF | I_Sleep |
| BATV012_9540 | I_Sleep |
| MA013_OFF | I_Toilet |
| T014_24 | I_Toilet |
| LAS013_54 | I_Toilet |
| MA013_ON | I_Toilet |

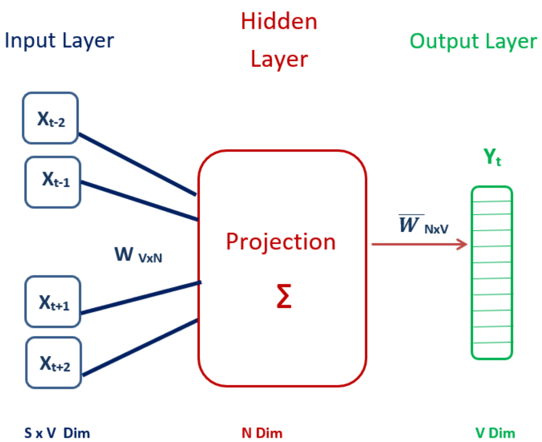


**FIGURE 4. Continuous Bag of Words (CBOW) Architecture [10].**

Embedding, and BiLSTM Prediction model. These stages are described in the following subsections.

### 1) SENSOR EVENTS EMBEDDING

Both words sequences in any text and sensors events are generated from a closed set of options where the context of sequence item defines the item meaning. To calculate the embeddings in our model, we employed the Word2Vec technique, a widely used embedding model. Word2vec algorithm is used to embed sensor events into vector of $D$ dimension for each sensor event. We chose the Word2Vec technique because of its effectiveness, memory efficient, and it is easy-implemented unsupervised learning method.

Word2vec is utilized to map events to numeric vectors given the context where the event appeared. The Continuous Bag of Words (CBOW) neural network architecture is utilized to train the word2vec model where the network predicts the current event using its surrounding events in a specific window size. The weights of hidden layer neurons in the neural network is used as the event vector. The fig. 4 Illustrate the CBOW neural network architecture.

After converting each sensor event into a vector, the sequence of sensor events is divided into a subsequence of size equals to $N$ sensor events. The subsequence is the main block in the learning process as all the models will learn the behavior of sensor events and related activities from each subsequence. Each subsequence is represented as a matrix of sensor events vectors which is called *embedding Matrix*, as shown in fig. 5.

$$\textit{Sequence embedding matrix} = \begin{bmatrix} V_{0,0} & V_{0,1} & \cdots & V_{0,D} \\ V_{1,0} & V_{1,1} & \cdots & V_{1,D} \\ \cdots & \cdots & \cdots & \cdots \\ V_{N,1} & V_{N,1} & \cdots & V_{N,D} \end{bmatrix}$$

**FIGURE 5. Embedding Matrix.**

The embedding matrix has $N$ rows, where $N$ is the number of sensor events in each subsequence and $D$ dimension vector for each sensor event. This *embedding Matrix* will be fed to the first BiLSTM model that recognizes the activities.

### 2) BILSTM RECOGNITION MODEL

The BiLSTM Recognition model is formulated as follows. Given a sequence of events, our model will classify each event in the sequence based on the order of the sequence into its activity. Also, this model will show the activity state (begin, middle, and end) when this event happens. The order of the sequence is necessary because the event classification will differ based on past and future events. The BiLSTM Recognition model takes sensor events embedded vectors as an input and produces the activity and the state of this activity for each sensor event in the sequence given to the model. An LSTM layer was built above the BiLSTM model to enhance the final recognition results.

The Human Activity Recognition Algorithm is depicted in Algorithm 1, where eventsList is a list of previous and current sequential events generated by environmental sensors. The embedding vector dimension (D) and the subsequence size (N) are provided as input assuming $D = 100$ and $N = 100$. This algorithm attempts to generate the corresponding activity for each event.

### 3) ACTIVITIES EMBEDDING

After getting results from recognition BiLSTM model, word2vec embedding technique is used to embed and map each activity in the result into a 100-dimension vectors using word2vec model. The sequence of activities is divided into a subsequence of size equals to N activities which will be fed to the second BiLSTM model to predict the future activities.

### 4) BILSTM PREDICTION MODEL

The BiLSTM prediction model is formulated as follows. Given a sequence of previous activities which represents the history, the model will predict future sequence of activities and their states while considering the order of the previous sequence. The BiLSTM prediction model takes recognized activities states embedded vectors as an input and produces the next activity and its state as an output. An LSTM layer was built above our BiLSTM model to enhance the final prediction results.

**Algorithm 1** Human Activity Recognition Algorithm

**Input:** $eventsList$ ← List of sequential events $D$ ← Embedded vector dimension $N$ ← SubSequences size
**Output:** $activitiesList$ ← List of related activity and its status for each event
**for each** $uniqueEvent \in eventsList$ **do**
  $eventvector$ ←Word2Vec($uniqueEvent, D$)
  $normalizedvec$ ←Normalization($eventvector$)
  $embeddingMap[uniqueEvent] = normalizedvec$
**end for**
**for each** $event \in eventsList$ **do**
  $eventvector \leftarrow embeddingMap[event]$
  $eSequence \leftarrow append(eventvector)$
**end for**
**for** $i = 0$ to *eSequence Length* **do**
  $Subsequences \leftarrow append(eSequence[\,i : i + N)$
  $i \leftarrow i + N$
**end for**
*model ← Hybrid model of BiLSTM and LSTM Configuration*
**for each** $embeddingMatrix \in Subsequences$ **do**
  $y \leftarrow model.predict(embeddingMatrix)$
  $activity \leftarrow OneHotDecoding(y)$
  $activitiesList \leftarrow append(activity)$
**end for**
**return** activitiesList

The Human Activity Prediction Algorithm is depicted in Algorithm 2, where activitiesList is a list of previous and current sequential activities. The embedding vector dimension (D) and the subsequence size (N) are provided as input. This algorithm attempts to generate the future activities for the next k activities. This algorithm could be adapted as needed to predict activities that might happen in the next M minutes.

## IV. PERFORMANCE EVALUATION

The objective of performance evaluation is to prove the capability of the proposed framework to predict future activities with high accuracy. Toward this objective, the proposed framework performance is evaluated in terms of accuracy and F-score for predicting the smart home resident activities in two settings of experiments.

### A. EXPERIMENTAL SETUP

All the experiments of this study were implemented on a one core-i7 and 16GB RAM Computer. The algorithms were implemented using Python 3.9.7, Gensim 4.1.2, Tensorflow v2.8 [55], and Keras 3.1 [56].

### B. GENERAL HYPERPARAMETERS

To generate the embedding values for events and activities, Word2vec model of Gensim package was utilized. Each event/activity was represented by a vector of 100 continuous value. We used mincount equals to one, which means that Word2Vec model didn't ignore any event/activity in any sequence. In addition, 10 worker threads were used to train word2vec models.

**Algorithm 2** Human Activity Prediction Algorithm

**Input:** $activitiesList$ ← List of previous and current activities recognized by Algorithm 1 $K$ Number of activities need to be predicted $M$ Time Length of activities need to be predicted $D$ ← Embedded vector dimension $N$ ← SubSequences size
**Output:** $predictedactivitiesList$ ← List of sequential future activities and their status
**for each** $uniqueAct \in activitiesList$ **do**
  $activityvector$ ←Word2Vec($uniqueAct, D$)
  $normalizedvec$ ←Normalization($activityvector$)
  $embeddingMap[uniqueAct] = normalizedvec$
**end for**
**for each** $activity \in activitiesList$ **do**
  $activityvector \leftarrow embeddingMap[activity]$
  $activitySequence \leftarrow append(activityvector)$
**end for**
*model ← Hybridd model of BiLSTM and LSTM Configuration*
**if** $M \geq 0$ **then**
  **for** $i = 0$ to *activitiesList Length* **do**
    $duration \leftarrow 0$
    $count \leftarrow 0$
    **while** $duration \leq t$ **do**
      $count \leftarrow count + 1$
      $count \leftarrow activitiesList[i+1] - activitiesList[i]$
      $i \leftarrow i + 1$
    **end while**
    $SubsequencesCount \leftarrow append(count)$
  **end for**
  *K ← most frequent count in SubsequencesCount*
**end if**
**if** $K \geq 0$ **then**
  **for** $i = 0$ to $K$ **do**
    $y \leftarrow model.predict(activitySequence)$
    $activitySequence \leftarrow append(y)$
    $activity \leftarrow embeddingMap(y)$
    $predictedactivitiesList \leftarrow append(activity)$
  **end for**
**end if**
**return** predictedactivitiesList

In Recognition BiLSTM model, the categorical cross-entropy was used as a loss function since there are more than two class labels. In prediction BiLSTM model, Root Mean Square Error (RMSE) was used as a loss function which is calculated using eq. (20), where $Y_n$ is the expected output and $\bar{Y}_n$ is the predicted output. The training loss and validation loss curves for predicting the activities' embedding vectors are shown in fig. 6.

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(Y_n - \bar{Y}_n)^2} \tag{20}$$

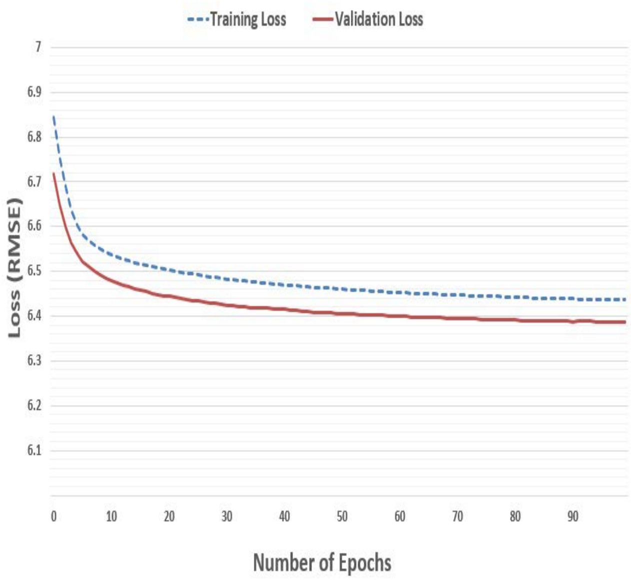


**FIGURE 6.** Training loss curves.

Adamax [57] is an extension of the Adam optimizer [58] and it is a variant of Adam based on the infinity norm. In Recognition BiLSTM model and Prediction BiLSTM model, Adamax was used which is more preferable than Adam especially in models with embedding.

In order to train the recognition and prediction models, all the training input subsequences were assumed to be with equal length to ensure that there is enough information to learn during the training process. The subsequence length was set to 100, which was found to improve the performance of the models and consequently improve the final performance of the framework.

### C. EVALUATION MEASURES

Accuracy is used as the main evaluation metric in order to find the correctness of the future activities' prediction. On the other hand, F-score is used to find out which activity has the best prediction quality. Both Accuracy and F-score are calculated as shown in eqs. (21) to (24), where $T_p$ denotes the True Positives, and $T_n$ the True Negatives.

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{21}$$

$$Precision = \frac{T_P}{T_P + F_P} \tag{22}$$

$$Recall = \frac{T_P}{T_P + F_N} \tag{23}$$

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{24}$$

The proposed framework was evaluated using 10-folds cross validation procedure where the accuracy and F-score for the 10 folds were averaged to obtain the final results. The mean accuracy which is the average of accuracy of all folds is calculated as eq. (25), where k is 10. A Ten-Fold Cross Validation splits data into 10 equally-sized folds. The model is trained on K-2 folds, one fold for validation and one fold for testing.

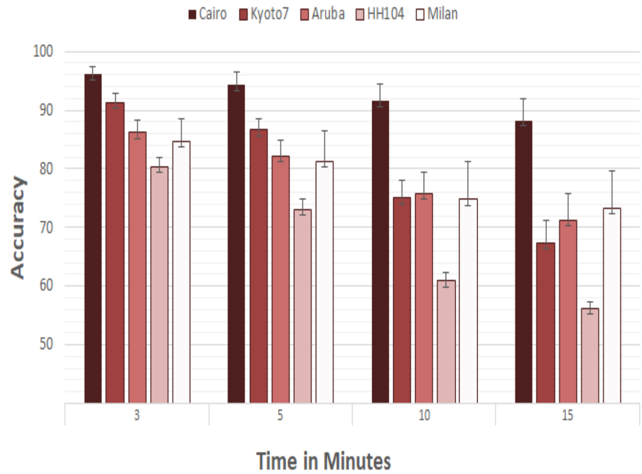


**FIGURE 7.** Mean Accuracy and Standard Deviation for our framework with different duration of time (Minutes).

$$MeanAccuracy = \frac{1}{K}\sum_{n=1}^{K} Accuracy_{(n)} \tag{25}$$

### D. EXPERIMENTAL EVALUATION

The proposed framework performance is evaluated in two settings over the five datasets presented earlier. The first setting is to predict the resident activities for the next M minutes in the future. The objective of performing several experiments in this setting is to find the maximum duration of minutes which we can predict activities within while having the highest accuracy. To get the best duration, fixed batch size 500 and fixed number of epochs 100 were used in different length of time duration as presented in fig. 7. The best duration was found to be three minutes.

The second setting is to predict a fixed number K of activities that the resident will do in the future without restricting it to a given time. The objective of performing several experiments in this setting is to find the maximum fixed number of activities which can be predicted with the highest accuracy. To get the best number, fixed batch size 500 and fixed number of epochs 100 were used in different number of activities as presented in fig. 8. The best fixed number of activities was found to be ten activities.

In this paper, after performing several experiments, 0.001 was found to be the best learning rate value for both recognition and prediction models. Drop out approach was performed to randomly drop out nodes during training to reduce overfitting and improve generalization error. Several experiments presented in table 5 were performed to select the most suitable dropout. A dropout value of 0.25 was used to reduce overfitting.

**TABLE 5.** Mean accuracy for our framework with different dropout before enhancement.

| Datasets | Dropout = 0.1 | | Dropout = 0.2 | | Dropout = 0.25 | | Dropout = 0.3 | | Dropout = 0.4 | | Dropout = 0.5 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Next 10 | 3 min | Next 10 | 3 min | Next 10 | 3 min | Next 10 | 3 min | Next 10 | 3 min | Next 10 | 3 min |
| **Cairo** | 98.727 | 95.688 | 98.621 | **96.075** | **98.727** | 96.065 | 98.545 | 95.924 | 98.631 | 95.716 | 98.106 | 95.531 |
| **Kyoto7** | 96.254 | 93.818 | 96.479 | 93.977 | **96.356** | **94.334** | 94.066 | 88.262 | 96.167 | 91.933 | 95.558 | 94.022 |
| **Aruba** | 91.921 | 82.824 | 93.811 | 85.979 | **93.912** | 86.213 | 93.605 | **86.226** | 93.574 | 86.189 | 92.341 | 85.198 |
| **HH104** | 94.401 | 78.982 | 93.635 | 77.945 | **95.171** | **80.365** | 94.25 | 79.896 | 93.563 | 79.25 | 93.422 | 78.405 |
| **Milan** | 93.04 | 84.537 | 92.788 | 84.695 | **93.844** | **84.708** | 92.259 | 84.295 | 91.716 | 84.22 | 90.732 | 83.237 |

**TABLE 6.** Precision Recall and F1 Score for next 3 minutes and next 10 activities for Cairo dataset.

| Activities | Next 3 Minutes | | | Next 10 Activities | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1-Score | Precision | Recall | F1-Score |
| **Laundry** | 0.962598 | 0.946886 | 0.954677 | 0.1 | 0.1 | 0.1 |
| **Take medicine** | 0.075457 | 0.078809 | 0.077097 | 0.6 | 0.52406 | 0.559465 |
| **Work in office** | 0.086127 | 0.041232 | 0.055766 | 0.2 | 0.2 | 0.2 |
| **Night wandering** | 0.465722 | 0.455218 | 0.46041 | 0.57 | 0.566667 | 0.568328 |
| **Leave home** | 0.944181 | 0.988903 | 0.966025 | 0.08 | 0.2 | 0.114286 |
| **Sleep** | 0.626473 | 0.520528 | 0.568607 | 0.776389 | 0.775287 | 0.775838 |
| **Lunch** | 0.635246 | 0.663256 | 0.648949 | 0.788235 | 0.8 | 0.794074 |
| **Bed to toilet** | **0.986127** | **0.984982** | **0.985554** | 0.1 | 0.1 | 0.1 |
| **other Activity** | 0.858973 | 0.857667 | 0.85832 | **0.994501** | **0.995315** | **0.994908** |
| **Dinner** | 0.767268 | 0.752288 | 0.759704 | 0.9 | 0.896667 | 0.89833 |
| **Breakfast** | 0.813042 | 0.781754 | 0.797091 | 1 | 0.98 | 0.989899 |
| **Wake** | 0.615192 | 0.665905 | 0.639545 | 0.886127 | 0.884982 | 0.885554 |

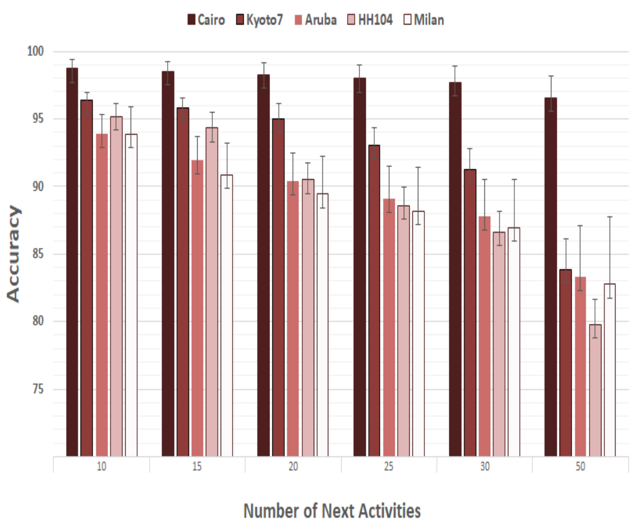


**FIGURE 8.** Mean Accuracy and Standard Deviation for our framework with different number of activities.

## *E. RESULTS AND DISCUSSION*

The chosen configuration would yield the best results while predicting activities in the next three minutes and almost the best results while predicting the next ten activities. The best accuracy for predicting the next ten activities was found to be 98.727% on Cairo dataset while the accuracy for predicting activities that would happen in next 3 minutes was found to be 96.065% on Cairo dataset.

Precision, Recall and F1 Score were used to evaluate the effectiveness of the proposed framework in the chosen configuration. Precision, Recall, and F1 Score of each activity for the two experiments on each dataset are presented in tables 6 to 10, and Confusion matrices for each dataset are shown in figs. 9 to 13. For example, activities such as 'Cook_Breakfast', 'Cook' and 'Personal_Hygiene' have the best prediction performance on the results where their F1 score is near to 1. On the other hand, activities such as 'Take_Medicine' or 'Morning_med' have F1 score near to Zero which may be related to their scarceness on the training datasets.

Several studies [6], [8], [46], [47], [49], [50], [51] were proposed to predict the next activity only not an entire future sequence of activities as proposed in this paper. Previous approaches have utilized different algorithms such as LSTM, Conditional Random Fields (CRF), Hidden Markov Model (HMM), Naive Bayes, and other approaches [49], [51].

**TABLE 7.** Precision Recall and F1 Score for next 3 minutes and next 10 activities for HH104 dataset.

| | 3 Minutes | | | Next 10 activities | | |
|---|---|---|---|---|---|---|
| **Activity** | **Precision** | **Recall** | **F1-Score** | **Precision** | **Recall** | **F1-Score** |
| **Bathe** | 0.693007974 | 0.715571785 | 0.701480808 | 0.786666667 | 0.754761905 | 0.767099567 |
| **Bed_Toilet_Transition** | 0.587294516 | 0.458332632 | 0.503589775 | 0.687856072 | 0.663218391 | 0.67200627 |
| **Cook** | **0.813338357** | **0.787561376** | **0.797434783** | **0.98875** | **0.99375** | **0.990813929** |
| **Cook_Breakfast** | 0.81564784 | 0.598681158 | 0.687719238 | 0.874242424 | 0.897826087 | 0.885048285 |
| **Cook_Dinner** | 0.836526309 | 0.587841422 | 0.68352018 | 0.885324675 | 0.877142857 | 0.879851157 |
| **Cook_Lunch** | 0.355235471 | 0.305284736 | 0.327956739 | 0.397142857 | 0.4 | 0.398550725 |
| **Dress** | 0.631999939 | 0.433606375 | 0.507714669 | 0.7 | 0.666666667 | 0.68 |
| **Eat** | 0.523700954 | 0.321920177 | 0.386065214 | 0.58 | 0.6 | 0.588888889 |
| **Eat_Breakfast** | 0.249482402 | 0.199584882 | 0.204143778 | 0.275 | 0.4 | 0.318283677 |
| **Eat_Dinner** | 0.776136283 | 0.360433799 | 0.457692954 | 0.86 | 0.858823529 | 0.849074074 |
| **Eat_Lunch** | 0.211764706 | 0.093166667 | 0.125726654 | 0.24 | 0.290909091 | 0.252380952 |
| **Enter_Home** | 0.000008 | 0.000025 | 0.000018 | 0.00001 | 0.000078 | 0.000975 |
| **Groom** | 0.655480558 | 0.460420458 | 0.535906677 | 0.778333333 | 0.790909091 | 0.783780247 |
| **Leave_Home** | 0.329583333 | 0.089746809 | 0.130200192 | 0.355 | 0.413235294 | 0.378303303 |
| **Morning_Meds** | 0.066177908 | 0.118181818 | 0.07987013 | 0.18 | 0.2 | 0.188888889 |
| **Personal_Hygiene** | 0.789815226 | 0.872185345 | 0.828098968 | 0.981373358 | 0.984667137 | 0.982944468 |
| **Read** | 0.440364151 | 0.323415766 | 0.361090334 | 0.48 | 0.5 | 0.488888889 |
| **Relax** | 0.809025672 | 0.51979037 | 0.625348307 | 0.954813242 | 0.949352297 | 0.950974826 |
| **Sleep** | 0.336657455 | 0.503777296 | 0.397723524 | 0.978926927 | 0.986067462 | 0.982321882 |
| **Sleep_Out_Of_Bed** | 0.717997532 | 0.551194034 | 0.591757715 | 0.7725 | 0.783333333 | 0.776056892 |
| **Take_Medicine** | 0.1 | 0.052272727 | 0.068656716 | 0.1 | 0.1 | 0.1 |
| **Toilet** | 0.671644558 | 0.275763889 | 0.376473889 | 0.941034188 | 0.897319989 | 0.91674765 |
| **Wash_Breakfast_Dishes** | 0.2 | 0.084244265 | 0.118509856 | 0.2 | 0.2 | 0.2 |
| **Wash_Dinner_Dishes** | 0.822074009 | 0.495464654 | 0.586347755 | 0.88 | 0.879411765 | 0.877413479 |
| **Wash_Dishes** | 0.385545228 | 0.201248794 | 0.234896419 | 0.45 | 0.5 | 0.469736842 |
| **Wash_Lunch_Dishes** | 0.22155243 | 0.264319249 | 0.232227768 | 0.3 | 0.295238095 | 0.297560976 |
| **Watch_TV** | 0.353139595 | 0.267651809 | 0.290344433 | 0.4 | 0.4 | 0.4 |
| **Work** | 0.628779778 | 0.461645363 | 0.484681443 | 0.775 | 0.8 | 0.785714286 |
| **Unknown Activity (o)** | 0.720081862 | 0.82811627 | 0.769495616 | 0.950817127 | 0.95082656 | 0.950587155 |

**TABLE 8.** Precision Recall and F1 score for next 3 minutes and next 10 activities for Kyoto7 dataset.

| **Activities** | **Next 3 Minutes** | | | **Next 10 Activities** | | |
|---|---|---|---|---|---|---|
| | **Precision** | **Recall** | **F1-Score** | **Precision** | **Recall** | **F1-Score** |
| **Prepare_lunch** | 0.461244412 | 0.200994 | 0.279981 | 0.917593 | 0.87668 | 0.894103 |
| **Breakfast** | 0.298504017 | 0.463173 | 0.363039 | 0.972705 | 0.981606 | 0.976327 |
| **Watch_TV** | 0.215513722 | 0.445265 | 0.290447 | 0.982332 | 0.974184 | 0.977759 |
| **Bed_to_toilet** | 0.321112362 | 0.467556 | 0.380738 | 0.969667 | 0.925757 | 0.945586 |
| **Cleaning** | 0.647168234 | 0.50714 | 0.568661 | 0.155195 | 0.19 | 0.170238 |
| **Wash_bathtub** | 0.596363636 | 0.522456 | 0.556969 | 0.1 | 0.1 | 0.1 |
| **Sleep** | 0.395572686 | 0.480268 | 0.433825 | **0.98908** | **0.993236** | **0.99107** |
| **Work_at_computer** | 0.355249792 | 0.471118 | 0.405061 | 0.987029 | 0.986148 | 0.986479 |
| **Groom** | 0.252924241 | 0.1302 | 0.171906 | 0.973443 | 0.97005 | 0.971409 |
| **Prepare_dinner** | 0 | 0.321604 | 0 | 0.950653 | 0.968673 | 0.958534 |
| **Work_at_dining_room_table** | **0.996363636** | **0.967777** | **0.981862** | 0.496364 | 0.480268 | 0.487923 |

**TABLE 9.** Precision Recall and F1 score for next 3 minutes and next 10 activities for Aruba dataset.

| Activities | Next 3 Minutes | | | Next 10 Activities | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1-Score | Precision | Recall | F1-Score |
| **Relax** | 0.652563 | 0.672349 | 0.662308 | 0.918537 | 0.909013 | 0.912964 |
| **Enter_Home** | 0.260725 | 0.064601 | 0.103546 | 0.291667 | 0.360833 | 0.266933 |
| **Meal_Preparation** | 0.726105 | 0.772226 | 0.748455 | 0.938407 | 0.963434 | 0.950167 |
| **Eating** | 0.577663 | 0.3485 | 0.434731 | 0.74 | 0.728454 | 0.732315 |
| **Work** | 0.526344 | 0.419305 | 0.466767 | 0.656 | 0.632748 | 0.642197 |
| **Wash_Dishes** | 0.679906 | 0.413039 | 0.513892 | 0.7855 | 0.75939 | 0.770613 |
| **Leave_Home** | 0.139058 | 0.03519 | 0.056167 | 0.153571 | 0.2 | 0.173714 |
| **other Activity** | **0.892492** | **0.860306** | **0.876103** | 0.955985 | 0.95708 | 0.956494 |
| **Bed_to_Toilet** | 0.094584 | 0.0266 | 0.041522 | 0.14 | 0.102198 | 0.118116 |
| **Housekeeping** | 0.682933 | 0.688985 | 0.685945 | 0.771111 | 0.794118 | 0.780905 |
| **Sleeping** | 0.770791 | 0.832754 | 0.800575 | **0.984058** | **0.958981** | **0.970642** |

**TABLE 10.** Precision Recall and F1 score for next 3 minutes and next 10 activities for Milan dataset.

| Activities | Next 3 Minutes | | | Next 10 Activities | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1-Score | Precision | Recall | F1-Score |
| **Guest_Bathroom** | 0.313079166 | 0.277805 | 0.25373 | 0.559881 | 0.649636 | 0.572218 |
| **Meditate** | 0.265809586 | 0.14752 | 0.184045 | 0.295 | 0.267857 | 0.279167 |
| **Dining_Rm_Activity** | 0.381480243 | 0.479883 | 0.393942 | 0.43 | 0.5 | 0.446154 |
| **Read** | 0.829475558 | 0.748652 | 0.780868 | 0.955309 | 0.902547 | 0.923936 |
| **Master_Bathroom** | 0.744391649 | 0.597065 | 0.633136 | 0.895632 | 0.859341 | 0.864846 |
| **Desk_Activity** | 0.31445891 | 0.349772 | 0.320515 | 0.385462 | 0.380476 | 0.382842 |
| **Watch_TV** | 0.657127022 | 0.569928 | 0.589337 | 0.774583 | 0.759551 | 0.76629 |
| **Kitchen_Activity** | **0.83155905** | **0.851214** | **0.839701** | **0.962377** | **0.963803** | **0.96257** |
| **Sleep** | 0.784332937 | 0.845592 | 0.806839 | 0.917541 | 0.953425 | 0.933309 |
| **Eve_Meds** | 0.1 | 0.039286 | 0.05641 | 0.1 | 0.1 | 0.1 |
| **other Activity** | 0.853726761 | 0.847097 | 0.849595 | 0.93295 | 0.938259 | 0.935286 |
| **Master_Bedroom_Activity** | 0.295352247 | 0.299248 | 0.285942 | 0.441154 | 0.408422 | 0.418693 |
| **Leave_Home** | 0.291422159 | 0.200632 | 0.194356 | 0.7775 | 0.638138 | 0.664684 |
| **Bed_to_Toilet** | 0.206111418 | 0.317424 | 0.221342 | 0.261379 | 0.358088 | 0.284042 |
| **Morning_Meds** | 0.252380952 | 0.133303 | 0.171882 | 0.3 | 0.3 | 0.3 |
| **Chores** | 0.38743724 | 0.367677 | 0.376394 | 0.386395 | 0.387273 | 0.386607 |

**TABLE 11.** Activity prediction accuracy comparison.

| | Aruba | Cairo | HH104 | Kyoto7 | Milan |
|---|---|---|---|---|---|
| **DNN+OCD-AE+LSTM [49]** | 57.2391 | 51.0639 | 59.6492 | 59.5239 | 69.7116 |
| **Proposed Framework** | 93.912 | 98.727 | 95.171 | 96.355 | 93.843 |

Although these studies used the same benchmark datasets that was utilized in this paper during evaluation, several considerations should be considered. For example, different training and testing splits in addition to the nature of the framework itself where the proposed activity prediction model did the prediction using the results of activity recognition model. While considering all the difference between other approaches and the proposed framework, the proposed framework has achieved the highest accuracy comparing with the evaluation results reported for other approach. On more specific level, the proposed framework has achieved an average accuracy score of 95.6% on five different datasets while other models [8], [49], [50] have achieved accuracy scores ranging from 45.5% to 89.7% on the same datasets. Although this is not a fair comparison between models, but it would give an indication about the quality of the proposed framework.

On the other hand, we compared the proposed framework against the model presented in [49] to provide evidence about

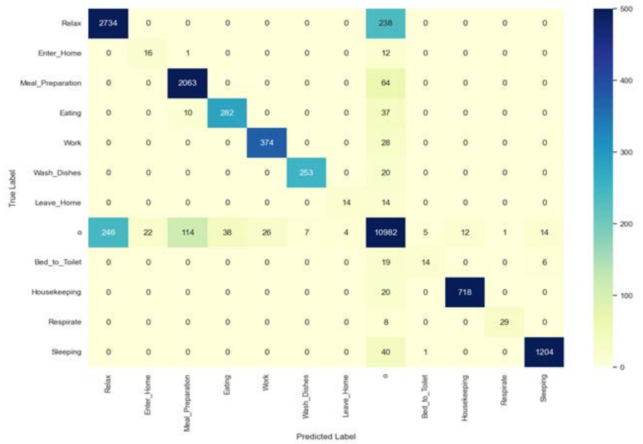


**FIGURE 9.** Confusion matrix for Aruba dataset.

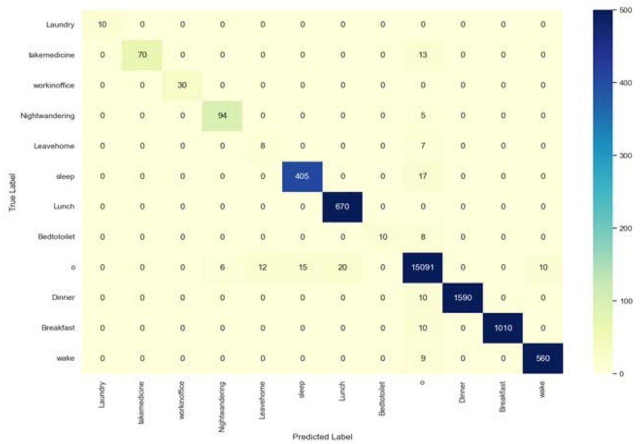


**FIGURE 10.** Confusion matrix for Cairo dataset.


**FIGURE 11.** Confusion matrix for Kyoto dataset.

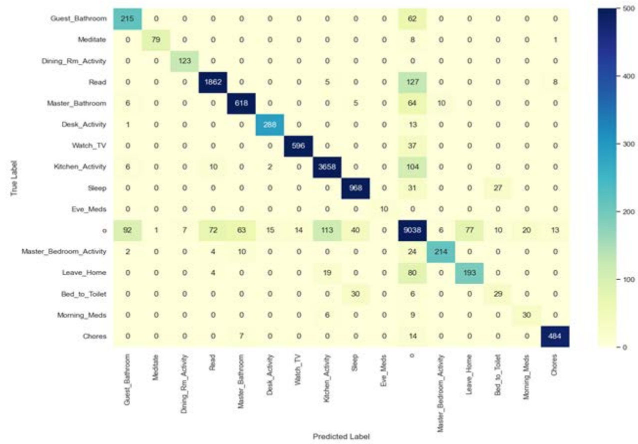


**FIGURE 12.** Confusion matrix for Milan dataset.

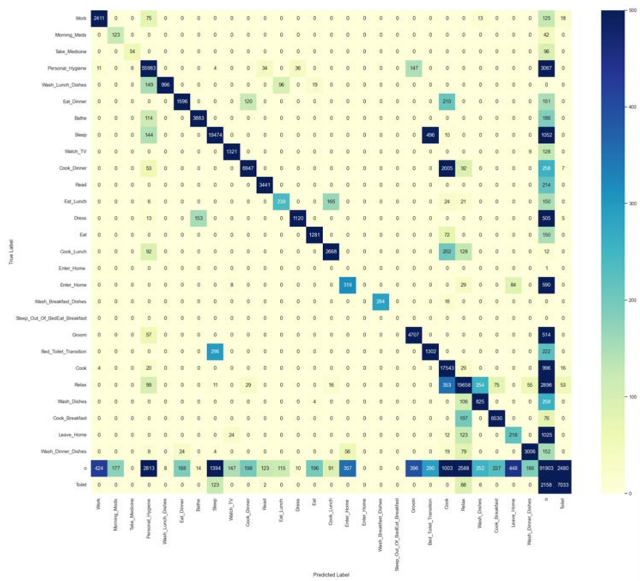

**FIGURE 13.** Confusion matrix for HH104 dataset.

the superiority of proposed framework against one of the very recent models. In [49], they utilize an overcomplete-deep autoencoder (OCD-AE) to distinguish between normal and abnormal activities. In addition, (LSTM) neural network was used to forecast the next activity. In order to provide a fair experiment, we re-implemented the model presented in [49] before evaluating its next activity forecasting performance on the same five dataset we used to evaluate the proposed framework. Table 11 present the results of accuracy comparison between the proposed framework and the model presented in [49]. The results show the outperformance of our proposed framework. In addition, the results acquired for the model presented in [49] for Aruba and Cairo datasets are synchronized with the results reported in their paper for the same datasets. The proposed framework outperformance is thought to be related to the utilization of Word2Vec embeddings especially that LSTM in [49] used one-hot encoder to encode the data at input layer.

Embeddings have proved a huge success through NLP models and managed to improve the performance of different NLP tasks with a huge margin. The similarity between words and sensor readings where each of them has a closed set of vocabulary was the key to utilize Word2Vec embeddings in order to generate sensor readings representations. The embeddings allowed sensor readings that happen within similar contexts to have similar representations. Consequently, the recognition and prediction Bi-LSTM networks were fed a better input data representations of sensor readings at their input layer which enhanced the recognition and prediction performance of future activities.

## V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a framework for recognizing and predicting future activities for smart home inhabitants. The findings given in this study demonstrated that the Word2Vec embedding technique and BiLSTM may be utilized together to generate a hybrid solution with high activity recognition and prediction ability. The suggested framework demonstrated that basic techniques, such as Word2Vec, which is utilized in a variety of domains, may be leveraged to improve the performance of HAR and HAP tasks. By comparing our model to previous models, we were able to demonstrate its efficiency and acceptability. The proposed approach reached average accuracy for predicting the next ten activities 95.602% while the accuracy for predicting activities that would happen in next 3 minutes 88.337%.

In the future, several tests on various sorts of datasets with extremely high volatility would be carried out. To improve accuracy, several parameter tweaking approaches may be considered.

## REFERENCES

[1] C. Stolojescu-Crisan, C. Crisan, and B.-P. Butunoi, "An IoT-based smart home automation system," *Sensors*, vol. 21, no. 11, p. 3784, May 2021.

[2] S. Krishnamoorthy, A. Dua, and S. Gupta, "Role of emerging technologies in future IoT-driven healthcare 4.0 technologies: A survey, current challenges and future directions," *J. Ambient Intell. Humanized Comput.*, vol. 12, pp. 1–47, May 2021.

[3] M. R. Haque, S. Jaman, M. G. Saklayen, M. M. Khondoker, A. B. Siddik, U. Sara, and M. S. Uddin, "Towards the development of an energy-efficient smart home through IoT," *Int. J. Adv. Technol. Eng. Explor.*, vol. 6, no. 58, pp. 208–216, Dec. 2019.

[4] L. G. Fahad and S. F. Tahir, "Activity recognition and anomaly detection in smart Homes," *Neurocomputing*, vol. 423, pp. 362–372, Jan. 2021.

[5] S. Mohamad, M. Sayed-Mouchaweh, and A. Bouchachia, "Online active learning for human activity recognition from sensory data streams," *Neurocomputing*, vol. 390, pp. 341–358, May 2020.

[6] R. Alfaifi and A. M. Artoli, "Human action prediction with 3D-CNN," *Social Netw. Comput. Sci.*, vol. 1, no. 5, pp. 1–15, Sep. 2020.

[7] Z. Xu, G. Wang, and X. Guo, "Comparative studies on activity recognition of elderly people living alone," in *Proc. Chin. Intell. Syst. Conf.* (Lecture Notes in Electrical Engineering). Singapore: Springer, 2020, pp. 276–291.

[8] M. Khan, M. M. Saad, M. A. Tariq, J. Seo, and D. Kim, "Human activity prediction-aware sensor cycling in smart home networks," in *Proc. IEEE Globecom Workshops (GC Wkshps*, Dec. 2020, pp. 1–6.

[9] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2013, PP. 1–9.

[12] N. Tax, "Human activity prediction in smart home environments with LSTM neural networks," in *Proc. 14th Int. Conf. Intell. Environ. (IE)*, Jun. 2018, pp. 40–47.

[13] J. Pomerat, A. Segev, and R. Datta, "On neural network activation functions and optimizers in relation to polynomial regression," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2019, pp. 6183–6185.

[14] H. Fang, H. Si, and L. Chen, "Recurrent neural network for human activity recognition in smart home," in *Proc. Chin. Intell. Automat. Conf.* (Lecture Notes in Electrical Engineering). Berlin, Germany: Springer, 2013, pp. 341–348.

[15] M. Taksir and S. Aktar, "Data-driven time series based prediction in smart home appliance energy consumption," *Int. J. Comput. Appl.*, vol. 178, no. 15, pp. 41–46, May 2019.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Dec. 2013, pp. 273–278.

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

[19] Z. Hameed and B. Garcia-Zapirain, "Sentiment classification using a single-layered BiLSTM model," *IEEE Access*, vol. 8, pp. 73992–74001, 2020.

[20] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of LSTM and BiLSTM in forecasting time series," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2019, pp. 3285–3292.

[21] S. Kumar, S. Benedict, and S. Ajith, "Application of natural language processing and IoTCloud in smart Homes," in *Proc. 2nd Int. Conf. Intell. Commun. Comput. Techn. (ICCT)*, Sep. 2019, pp. 20–25.

[22] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, Apr. 2019.

[23] J. Pennington, R. Socher, and C. D. Manning. *Glove: Global Vectors for Word Representation*. Accessed: Feb. 10, 2022. [Online]. Available: https://nlp.stanford.edu/projects/glove/

[24] V. Vichianchai and S. Kasemvilas, "A new term frequency with Gaussian technique for text classification and sentiment analysis," *J. ICT Res. Appl.*, vol. 15, no. 2, pp. 152–168, Oct. 2021.

[25] Q. Li, W. Huangfu, F. Farha, T. Zhu, S. Yang, L. Chen, and H. Ning, "Multi-resident type recognition based on ambient sensors activity," *Future Gener. Comput. Syst.*, vol. 112, pp. 108–115, Nov. 2020.

[26] S. Mitrofanov and E. Semenkin, "An approach to training decision trees with the relearning of nodes," in *Proc. Int. Conf. Inf. Technol. (InfoTech)*, Sep. 2021, pp. 1–5.

[27] B. Priyoko and A. Yaqin, "Implementation of naive Bayes algorithm for spam comments classification on Instagram," in *Proc. Int. Conf. Inf. Commun. Technol. (ICOIACT)*, Jul. 2019, pp. 508–513.

[28] M. Sanjeev, M. Israyelu, and S. Sashidhar, "Comparative analysis of direct SVM and indirect SVM techniques for direct matrix converter," in *Proc. Int. Symp. Asian Control Assoc. Intell. Robot. Ind. Autom. (IRIA)*, Sep. 2021, pp. 95–100.

[29] A. Kumar, A. Verma, G. Shinde, Y. Sukhdeve, and N. Lal, "Crime prediction using K-nearest neighboring algorithm," in *Proc. Int. Conf. Emerg. Trends Inf. Technol. Eng. (IC-ETITE)*, Feb. 2020, pp. 1–4.

[30] T. Kim and J. Park, "Logistic regression for LDPC decoding failure prediction," in *Proc. IEEE 18th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2021, pp. 1–6.

[31] A. Zaib, T. Ballal, S. Khattak, and T. Y. Al-Naffouri, "A doubly regularized linear discriminant analysis classifier with automatic parameter selection," *IEEE Access*, vol. 9, pp. 51343–51354, 2021.

[32] J. Jung, Y. C. Choi, and S.-I. Choi, "Ensemble learning using pressure sensor for gait recognition," in *Proc. IEEE Region Symp. (TENSYMP)*, Aug. 2021, pp. 1–3.

[33] D. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, "CASAS: A smart home in a box," *Computer*, vol. 46, no. 7, pp. 62–69, Jul. 2013.

[34] M. Jethanandani, A. Sharma, T. Perumal, and J.-R. Chang, "Multi-label classification based ensemble learning for human activity recognition in smart home," *Internet Things*, vol. 12, Dec. 2020, Art. no. 100324.

[35] D. Singh, E. Merdivan, S. Hanke, J. Kropf, M. Geist, and A. Holzinger, "Convolutional and recurrent neural networks for activity recognition in smart environment," in *Towards Integrative Machine Learning and Knowledge Extraction* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2017, pp. 194–205.

[36] T. van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate activity recognition in a home setting," in *Proc. 10th Int. Conf. Ubiquitous Comput.*, New York, NY, USA, 2008, pp. 1–9.
[37] Q. Teng, K. Wang, L. Zhang, and J. He, "The layer-wise training convolutional neural networks using local loss for sensor-based human activity recognition," *IEEE Sensors J.*, vol. 20, no. 13, pp. 7265–7274, Jul. 2020.
[38] W. Huang, L. Zhang, W. Gao, F. Min, and J. He, "Shallow convolutional neural networks for human activity recognition using wearable sensors," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.
[39] S. Xu, L. Zhang, W. Huang, H. Wu, and A. Song, "Deformable convolutional networks for multimodal human activity recognition using wearable sensors," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2022.
[40] H. Zhang, Z. Xiao, J. Wang, F. Li, and E. Szczerbicki, "A novel IoT-perceptive human activity recognition (HAR) approach using multihead convolutional attention," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1072–1080, Feb. 2020.
[41] W. Gao, L. Zhang, W. Huang, F. Min, J. He, and A. Song, "Deep neural networks for sensor-based human activity recognition using selective kernel convolution," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.
[42] D. Liciotti, M. Bernardini, L. Romeo, and E. Frontoni, "A sequential deep learning application for recognising human activities in smart Homes," *Neurocomputing*, vol. 396, pp. 501–513, Jul. 2020.
[43] R. E. Alden, H. Gong, C. Ababei, and D. M. Ionel, "LSTM forecasts for smart home electricity usage," in *Proc. 9th Int. Conf. Renew. Energy Res. Appl. (ICRERA)*, Sep. 2020, pp. 434–438.
[44] F. U. M. Ullah, A. Ullah, I. U. Haq, S. Rho, and S. W. Baik, "Short-term prediction of residential power energy consumption via CNN and multi-layer bi-directional LSTM networks," *IEEE Access*, vol. 8, pp. 123369–123380, 2020.
[45] F. Ordóñez, P. de Toledo, and A. Sanchis, "Activity recognition using hybrid generative/discriminative models on home environments using binary sensors," *Sensors*, vol. 13, no. 5, pp. 5460–5477, 2013.
[46] Du, Lim, and Tan, "A novel human activity recognition and prediction in smart home based on interaction," *Sensors*, vol. 19, no. 20, p. 4474, Oct. 2019.
[47] C. Wang and Z. Peng, "Deep learning model for human activity recognition and prediction in smart Homes," in *Proc. Int. Conf. Intell. Transp., Big Data Smart City (ICITBS)*, Jan. 2020, pp. 741–744.
[48] H. D. Mehr, H. Polat, and A. Cetin, "Resident activity recognition in smart Homes by using artificial neural networks," in *Proc. 4th Int. Istanbul Smart Grid Congr. Fair (ICSG)*, Apr. 2016, pp. 1–5.
[49] K. A. Alaghbari, M. H. M. Saad, A. Hussain, and M. R. Alam, "Activities recognition, anomaly detection and next activity prediction based on neural networks in smart Homes," *IEEE Access*, vol. 10, pp. 28219–28232, 2022, doi: 10.1109/access.2022.3157726.
[50] I. Fatima, M. Fahim, Y.-K. Lee, and S. Lee, "A unified framework for activity recognition-based behavior analysis and action prediction in smart Homes," *Sensors*, vol. 13, no. 2, pp. 2682–2699, 2013, doi: 10.3390/s130202682.
[51] J. Cumin, G. Lefebvre, F. Ramparany, and J. L. Crowley, "PSINES: Activity and availability prediction for adaptive ambient intelligence," *ACM Trans. Auton. Adapt. Syst.*, vol. 15, no. 1, pp. 1–12, Mar. 2020, doi: 10.1145/3424344.
[52] D. Cook and M. Schmitter-Edgecombe, "Assessing the quality of activities in a smart environment," *Methods Inf. Med.*, vol. 48, no. 5, pp. 480–485, 2009.
[53] D. Cook, "Learning setting-generalized activity models for smart spaces," *IEEE Intell. Syst.*, vol. 27, no. 1, pp. 32–38, Jan./Feb. 2012.
[54] *Casas Datasets*. Accessed: Feb. 10, 2022. [Online]. Available: http://casas.wsu.edu/datasets/
[55] *Tensorflow*. Accessed: Feb. 10, 2022. [Online]. Available: https://www.tensorflow.org/
[56] *Keras Documentation: Keras API Reference*. Accessed: Feb. 10, 2022. [Online]. Available: https://keras.io/api/
[57] *Keras Documentation: Adamax Optimizer*. Accessed: Feb. 10, 2022. [Online]. Available: https://keras.io/api/optimizers/adamax/
[58] *Keras Documentation: Adam Optimizer*. Accessed: Feb. 10, 2022. [Online]. Available: https://keras.io/api/optimizers/adam/

**MAI MOHAMED** received the bachelor's degree from the Faculty of Computers and Artificial Intelligence, Cairo University, in 2016. She is a Researcher with the Faculty of Computers and Artificial Intelligence, Cairo University.

**AYMAN EL-KILANY** received the M.Sc. and Ph.D. degrees from the Information Systems Department, Faculty of Computers and Artificial Intelligence, Cairo University, in 2012 and 2018, respectively. He is an Assistant Professor and a Researcher with the Faculty of Computers and Artificial Intelligence, Cairo University.

**NEAMAT EL-TAZI** received the Ph.D. degree in data management and data analytics jointly from Cairo University and the University of Michigan. She is an Associate Professor with the Department of Information Systems, Faculty of Computers and Artificial Intelligence, Cairo University. She is also the Co-Founder of AbuErdan, a powerful SaaS platform that helps poultry companies increase their profitability while reducing their production waste using the latest technologies in artificial intelligence, machine learning, and blockchain. She has more than 18 years of experience in product development and management in multiple industries in different countries. She is working in both academia and industry, she always strives to reduce the gap between the two worlds. Her research interests include data engineering, data management, and data analytics in social networks.