An Empirical Evaluation of Deep Learning Techniques for Human Activity Recognition

Weijie Lu

A thesis submitted to Auckland University of Technology in partial fulfillment of the requirements for the degree of Master of Computer and Information Sciences (MCIS)

Primary Supervisor: Dr Sira Yongchareon
Secondary Supervisor: Dr Jian Yu
18 February 2020
School of Engineering, Computer and Mathematical Sciences

Abstract

The recent advancement and development of human-activity recognition technology has led to the gradual entrance of smart home induction systems into residents' lives, stimulating the demand for associated products and services. With these developments, human activity recognition based on deep learning models has earned an increasing share of attention.

This research evaluates the ability of nine baseline deep-learning models to classify five CASAS datasets. The study aims to find the baseline deep learning model that best recognises resident activity, and to establish methods that improve the performance of baseline deep-learning models. Specifically, we hypothesise that the bidirectional and hybrid architectures will improve the performance of classifying residential activity. To test this hypothesis, we incorporate the hybrid architecture into the convolutional neural network (CNN), and the bidirectional architecture into the long short-term memory and gated recurrent unit (GRU) classifiers. We then verify whether these extensions improve the performances of the baseline models. Finally, we alter the groupings and compare the performances of the baseline deep learning models by different evaluation metrics and the Friedman test.

Among the nine deep-learning models tested, the BI-GRU model best recognised various human activities. Our hypothetical improvement method, the bidirectional architecture, significantly improved the model's performance.

Keywords: Resident activity recognition, deep learning models, performance improvement

Contents

List of Tables

List of Figures

List of Abbreviations

AAL: Active and Assisted Living

ADL: Activities of Daily Living

ADNI: Alzheimer ‘s Disease Neuroimaging Initiative

ANN: Artificial Neural Network

ANOVA: Analysis of Variance

AI: Artificial Intelligence

AUC: Area Under Curve

BI-GRU: Bi-directional Gated Recurrent Unit

BI-LSTM: Bi-directional Long Short-Term Memory

CASAS: Center for Advanced Studies in Adaptive Systems

CCRBM: Centered Convolutional Restricted Boltzmann Machines

CCDBN: Centered Convolutional Deep Belief Networks

CNN: Convolution Neural Network

DAE: Denoising autoencoders

DBN: Deep Belief Network

DNN: Deep Neural Network

E-BinGRU: Encoded Binarized Gated Recurrent Unit

ECDF: Empirical Cumulative Distribution Function

EEG: Electroencephalogram

E-GRU: Encoded Gated Recurrent Unit

ET-KNN: Evidence Theoretic K-Nearest Neighbors

FN: False Negatives

FP: False Positives

HAR: Human Activity Recognition

HMM: Hidden Markov Model

IoT: Internet-of-Thing

ISOMAP: Isometric Feature Mapping

ITSC: Improved Topology-Based Sparse Coding

GIZ: Group Interaction Zone

GPDM: Gaussian Process Dynamical Model

GRU: Gated Recurrent Units

LSTM: Long Short-Term Memory

LSTM+CNN: Long Short-Term Memory + Convolutional Neural Network

MRI: Magnetic Resonance Imaging

NN: Neural Network

PCA: Principle Component Analysis

pLSC: p-Laplacian Regularized Sparse Coding

PR: Pattern Recognition

RBM: Restricted Boltzmann machine

RNN: Recurrent Neural Networks

ROC: Receiver Operating Characteristics

SAE: Stacked Autoencoders

SRC: Sparse-Reconstruction based Classification

SRU: Simple Recurrent Units

SVM: Support Vector Machines

TN: True Negatives

TP: True Positives

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgments), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature:  Date:18/2/2020

Acknowledgment

This thesis was completed as part of the Master of Computer and Information Sciences course at the School of Engineering, Computer and Mathematical Sciences in the Faculty of Design and Creative Technologies at the Auckland University of Technology, New Zealand. I would like to thank my parents for their support, and my friends for their encouragement during my study in New Zealand. I especially thank my supervisor Dr. Sira Yongchareon for providing me with a lot of help during the learning process. At the beginning of my research project, my supervisor laid out a plan for my thesis. Throughout the following year, we held weekly meetings to discuss and arrange the tasks for the upcoming week. Under his tutelage, I have successfully completed my thesis within the specified time. He always patiently guided my solutions to every problem.

WEIJIE.LU

18/02/2020

Copyright

Chapter 1
Introduction

The first chapter consists of five parts. The first part provides the background of the research. The second part clarifies the motivation of this research project. The third and fourth parts introduce the research questions and contributions of the thesis, respectively. Finally, the thesis structure is presented.

1.1 Background and Motivation

Nowadays, the rapid increase in the world's ageing population. Requires a lot of attention and dedication in the care of elderly dependent people, as the lifestyle and health of older people are affected by these things. Dementia-related problems in the elderly are generating (Umphred, Lazaro, Roller, & Burton, 2013), which has become the major challenge worldwide. Due to this fact, some problems affect physical and mobility. Such as writing, walking and more complex activities (cooking, cleaning, taking medication, etc.)

The continued rapid increase in the world's ageing population has increased the demand for HAR in smart environments and assisted living in smart homes. Increasing numbers of people are interested in smart-home HAR, which can improve the life quality of residents through the information collected by sensors in the smart home (Chen, Hoey, Nugent, Cook, & Yu, 2012) . A smart home learns the daily habits of the residents, providing an independent and comfortable living environment. HAR can intelligently build the data in the intelligent environment into sophisticated modelling, reasoning and decision-making procedures (Aztiria, Augusto, Basagoiti, Izaguirre, & Cook, 2013).

Human activity recognition (HAR) is an important research component of human activity analysis and human–computer interactions. Human activities such as walking, drinking, driving, and more complex activities can be recognised by different machine learning algorithms. HAR is also vital for maintaining the health of elderly persons performing daily activities. Particularly, it can help in detecting and diagnosing serious illness in the elderly. Various types of datasets collected from different sensors are available for detecting human activities and human health states (Cao, Wang, Zhang, Jin, & Vasilakos, 2018). HAR obtains temporal and spatial information through visual or non-visual sensory data, and identifies simple or complex human behaviours and activities in real life (Ranasinghe, Machot, & Mayr, 2016). The adopted sensors can be fused into the living environment by direct connection to the detected objects, or can be worn by the residents. Unlike wearable sensors, object or environmental sensors indirectly detect the

activities of residents and distinguish their similar actions, and does not affect the normal activities of residents (Zolfaghari, Zall, & Keyvanpour, 2016). HAR is applied in three main areas: health care surveillance (Chang, Krahnstoever, Lim, & Yu, 2010), indoor and outdoor activity monitoring (Paola, Naso, Milella, Cicirelli, & Distante, 2008), and active and assisted living (AAL) systems for smart homes (Okeyo, Chen, & Wang, 2014). In recent years, HAR has been increasingly implemented by deep learning technologies (Liciotti, Bernardini, Romeo, & Frontoni, 2019), which learn the sensor data through multiple hidden layers. The advantage of deep learning models is that feature extraction and transformation are performed directly without prior knowledge.

Research on HAR can be divided into these broad categories based on the devices, sensors, and data used for detecting the activity details. Sensors include video-based sensors, wearable sensors, and sensors for mobile phones. Video-based sensors capture the daily activities from images, sounds, or video/surveillance camera functions (Onofri, Soda, Pechenizkiy, & Iannello, 2016).Wearable and embedded sensors placed on different sites of the body can analyse the details and movement patterns of human activities. These sensors are becoming more common with modern advances in mobile phones and wearable sensor technologies.

HAR is a typical pattern recognition (PR) problem. The traditional methods for PR problems are based on machine learning algorithms such as decision trees, support vector machines (SVMs), naive Bayes and hidden Markov models (Lara & Labrador, 2013). Machine learning algorithms deliver excellent performance in HAR problems such as disease detection. However, as human knowledge is limited, machine learning algorithms rely excessively on manual feature extraction. These restrictions prevent machine learning models from learning the deep features and performing unsupervised learning. Traditional PR methods have limited classification accuracy and model performance, and are of limited applicability in HAR.

In recent years, deep learning algorithms have rapidly progressed as alternatives to traditional PR methods. Deep learning algorithms achieve higher performance in HAR

applications than traditional PR. In particular, deep learning reduces the design workload and learns more functions and more advanced functions through deep learning models (Wang, Chen, Hao, Peng, & Hu, 2018). The present research evaluates the ability of different deep learning models to classify human activities; that is, to model the datasets obtained by smart home sensors. The goal is not merely to find the baseline deep-learning model that best classifies human activities, but also to improve the baseline deep-learning model by a simple method. Ultimately, the most appropriate model for classifying human activities is obtained.

To the best of my knowledge, the performance of the deep learning models on HAR sensor datasets remains is a good research topic. Researchers so far studied single-type deep learning model and improvements (Liciotti, Bernardini, Romeo, & Frontoni, 2019). We propose to evaluate the six baseline deep learning models and three improved deep learning models on HAR sensor datasets.

1.2 Research Questions

With the continuous progress of deep learning technology, HAR technology has improved the lives of residents in smart houses. The major goal in this thesis is evaluating the performance of several popular deep learning models for HAR. The questions to be discussed can be summarized as follows:

Which of the baseline deep learning models achieves the best performance for HAR?

In this thesis, the resident activities were extracted from CASAS datasets. Five CASAS datasets were analysed by the six most popular baseline deep-learning models including Convolution Neural Network (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), Deep Neural Network (DNN), Autoencoder, and Sparse Coding. The results were evaluated by the accuracy, precision, recall, F-score and area-under-curve (AUC) measures. The optimal general deep learning model was determined in a statistical analysis.

Which of the baseline models can be improved to achieve the best performance?

Although deep learning techniques are applicable to resident activity recognition, their performance can be improved. We therefore pose the following two hypotheses.

1) Among the six baseline models, we expect that CNN, LSTM and GRU will optimise the resident activity recognition. Hence, the improved model is based on these three outstanding models.
2) We expect that the bidirectional and hybrid architectures will improve the performances of the selected deep learning methods. We therefore improve the CNN, LSTM and GRU by developing bidirectional and hybrid models. In the end, we verify whether these two methods can improve the performance of the models.

1.3 Contributions

This research aims to evaluate different deep learning models for classifying human activities; that is, their ability to process the datasets obtained by smart home sensors. Besides finding the baseline deep learning model that is most suitable for classifying human activities, it attempts to improve the baseline deep-learning model by a simple method, and thereby obtain the most appropriate model for classifying human activities. This thesis performs 1) data pre-processing, 2) HAR by baseline deep-learning models, 3) HAR by improved baseline deep- learning models, and 4) an evaluation analysis. Chapter 2 summarises existing deep learning algorithms and evaluation methods. Chapter 3 provides the deep learning methods and the theoretical basis of assessment methods. Chapter 4 shows and analyses the results.

The overall contributions of this thesis are threefold: 1) we propose HAR based on baseline deep learning, 2) we improve the existing baseline deep-learning models, and 3) we study the applicability of deep learning models to our datasets. Our results meet the current developmental needs of HAR based on deep learning.

1.4 Thesis structure

This thesis consists of five chapters:

Chapter 2 introduces related work in a literature review. We first introduce HAR and its related fields. We then introduce each of the popular deep learning algorithms and HAR models, and systematically discuss the advantages, disadvantages and related work of deep learning methods. Finally, we introduce the evaluation methods and datasets of HAR. In Chapter 2, we learn the results and experience of previous researchers that will guide our following experimental work.

Chapter 3 is dedicated to the methodology of our work, including the collected datasets, data pre-processing, classification models, evaluations, and statistical tests. Chapter 3 introduces our research design, research methods and experimental procedure.

Chapter 4 presents the experimental results and discussion, including the training and test results of different datasets in each deep learning model, and the analysis of the experimental results. We intuitively explain the obtained figures and tables. Finally, we analyse and discuss the experimental results and compare them among the models, thereby identifying which baseline models deliver the best performance and which baseline models can be improved.

Chapter 5 concludes the study, discusses its limitations, and proposes ideas for future work.

Chapter 2
Related Work

This chapter evaluates deep learning models for classifying human activity. By reviewing the past literature and related theories of previous researchers, we can improve our research design and experimental methods. We can also obtain a more comprehensive understanding of the HAR field. In this chapter, we introduce various models and evaluation methods of HAR.

2.1 Introduction

With the continuous development of artificial intelligence (AI) technology, deep learning has become the essence of analysis and recognition (Bharkad, 2013). To understand deep learning techniques for HAR and the best approach to the research, we must study related works. In addition to understanding the technical support of HAR, we need to study the advantages and disadvantages of the different baseline deep-learning algorithms and models. Finally, we need to consider which evaluation method and datasets are suitable for our purpose.

This Chapter, we first introduce overview of HAR. Next, we discuss the most popular deep learning models of structure and application. Next, we present the evaluation method and statistical test for performance of the deep learning models. Finally, we describe the datasets of HAR.

2.2 Human Activity recognition (HAR)

Increasingly, technology is the medium through which healthcare is integrated with society. HAR provides various technical supports that improve residents' quality of life. High-demand areas such as home automation and convenience services are continually growing as the population ages (Röcker, Ziefle, & Holzinger, 2011). HAR processes the data collected from wearable sensors, video frames, or images (Jobanputra, Bavishi, & Doshi, 2019). Accordingly, HAR can monitor and analyse human life information, and thereby introduce more features that provide independence and comfort to the residents (van Kasteren, Englebienne, & Kröse, 2011).

HAR is performed on three main data types: sensor-based, vision-based and radio-based (Mohamad, Sayed-Mouchaweh, & Bouchachia, 2019). The first feasibility studies on activity recognition using body-worn sensors were conducted at the end of the 1990s (Bulling, Blanke, & Schile, 2014). Sensor-based data collection is a traditional method reliant on large volumes of raw input collected from several types of sensors. The features of raw data are manually

extracted based on human knowledge. Finally, these features are employed to train or develop techniques in real HAR tasks (Wang, Chen, Hao, Peng, & Hu, 2018). Vision-based data collection uses image sequences labelled with action tags for action recognition (Poppe, 2010). Radio-based activity recognition is a new approach that utilizes body attenuation and/or channel fading of wireless radio. This method aims for high recognition accuracy while preserving the user's privacy (Wang & Zhou, 2015).

Various state-of-the-art methods have been proposed for activity recognition tasks. Figure 1 summarises the techniques applied in HAR. The traditional HAR models employ machine learning algorithms such as decision trees, SVMs (Erfani, Rajasegarar, Karunasekera, & Leckie, 2016), naive Bayes, and hidden Markov models (HMMs). These methods are widely used in HAR and similar researches.

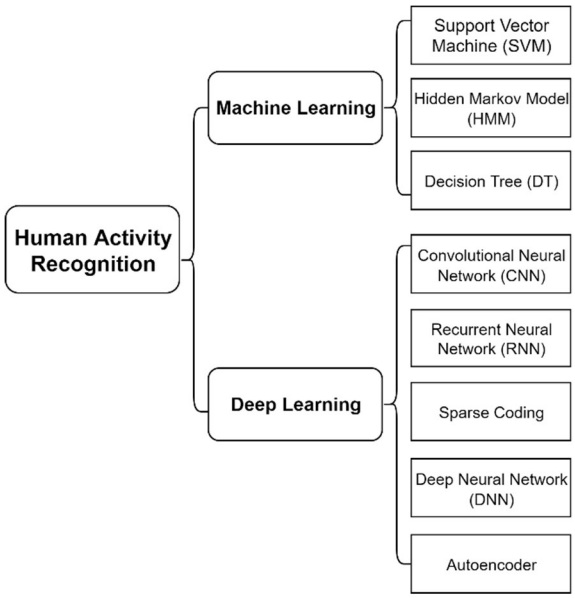


Figure 1 Several state-of-the-art methods for HAR

Data for monitoring purposes are collected by advanced sensor technologies with excellent performance, low weight, and low power consumption. However, the current machine learning technology of HAR depends on human skill prior to the training phase, so is costly and unreliable in non-stationary environments (Mohamad, Sayed-Mouchaweh, & Bouchachia, 2019). The replacement of traditional sensors with

smartphones for HAR has greatly improved the functionality of HAR devices. The smartphone era has ushered in mobile-phone motion sensors, which have become a popular choice for collecting the physiological signals of daily human activities (Duffner, Berlemont, Lefebvre, & Garcia, 2014). Smartphones can collect and monitor the daily life information of users automatically and without interference. However, their detection ability is limited; that is, mobile phone sensors cannot reliably recognise activities such as drinking, eating and typing, as they are normally stored in pockets or equivalent positions. To recognise relatively complex activities, the signals from two or more motion sensors at different positions must be evaluated in combination (Shoaib, Bosch, Incel, Scholten, & Havinga, 2016).

The learned results of traditional machine learning results are not entirely satisfactory, because feature extraction heavily relies on human knowledge or experience. In more general environments and tasks, human knowledge limits the success of a recognition system. Moreover, training the model requires a large amount of well-labelled data, whereas most of the data in real-world applications are unlabelled, causing unexpected performance of the model in unsupervised learning tasks (Wang, Chen, Hao, Peng, & Hu, 2018). Motivated by the desire to improve human lifestyles, researchers have recently developed HAR with higher recognition and classification accuracy under more realistic settings than previous efforts (Wang, Cang, & Yu, 2019). Modern deep-learning approaches can extract HAR features in an unsurprised manner (Wiretunga & Cooper, 2017), by techniques such as natural language processing and image pattern recognition.

2.3 Deep learning Models

Feature extraction by current HAR relies on handcrafted and human knowledge, so cannot identify complex activities among the current influx of data from multimodal, high dimensional sensors (Nweke, Teh, Al-garadi, & Alo, 2018). Recently, deep learning and AI methods have overcome these challenges by automatically extracting the diverse representation of HAR features. CNNs, recurrent neural networks (RNNs) and deep

belief networks (DBNs) have delivered especially promising results (Hammerla, Halloran, & Ploetz, 2016).

Deep learning methods are based on neural networks with multiple processing layers, which automatically discover the required representations through multiple levels of abstraction. A deep learning machine is fed by raw input and automatically extracts the features for detection or classification. Obviously, deep learning enables advanced problem solving, which has resisted the best attempts of AI for many years (LeCun, Bengio, & Hinton, 2015).

In applications that produce big data, such as Internet-of-Things (IoT), deep learning brings important improvements over traditional machine learning approaches. Deep learning models improve the recognition accuracy by extracting features other than handcrafted and human-derived features (Mohammadi, Al-Fuqaha, Sorour, & Guizani, 2018). Deep learning has also benefitted from advanced tools developed for analysing large volumes of raw business data. Deep learning algorithms extract the high-level, complex abstractions of data by a hierarchical learning process with a system for collecting massive amounts of information for Big Data Analytics (Najafabadi, et al., 2015).

Deep learning operates through a series of consecutive artificial neural networks (ANNs). Deep learning architectures typically contain dozens or even hundreds of consecutive processing layers. Each layer processes data with different functions, resulting in increasingly rich information results (Mohammadi, Al-Fuqaha, Sorour, & Guizani, 2018). A deep learning architecture composed of many different systems has now been proposed. These models can have the same kind of stack or a stack with different architectures to improve the architecture functionality. Deep learning provides powerful system performance, flexibility and functionality. CNNs and recursive upgrade networks are very popular in HAR (Nweke, Teh, Al-garadi, & Alo, 2018).

Many deep learning methods have been proposed (LeCun, Bengio, & Hinton, 2015). The popular deep leaning methods adopted in HAR are classified into three broad

categories: generative models, discriminative models and hybrid models (see Fig. 2). Generative models include the three most common methods: restricted Boltzmann machines, autoencoders, and sparse coding. Generative models are graphical models that generate data distributions with random variance, either independently or dependently. The discriminative models are CNN (the most popular discriminative model), RNN, and deep neural models. The third category, hybrid models, includes convolutional sparse coding and recurrent CNN (Nweke, Teh, Al-garadi, & Alo, 2018). These models combine generative and discriminative models, and also involve the pre-training data. The various deep learning methods in each category are outlined in Fig. 2.

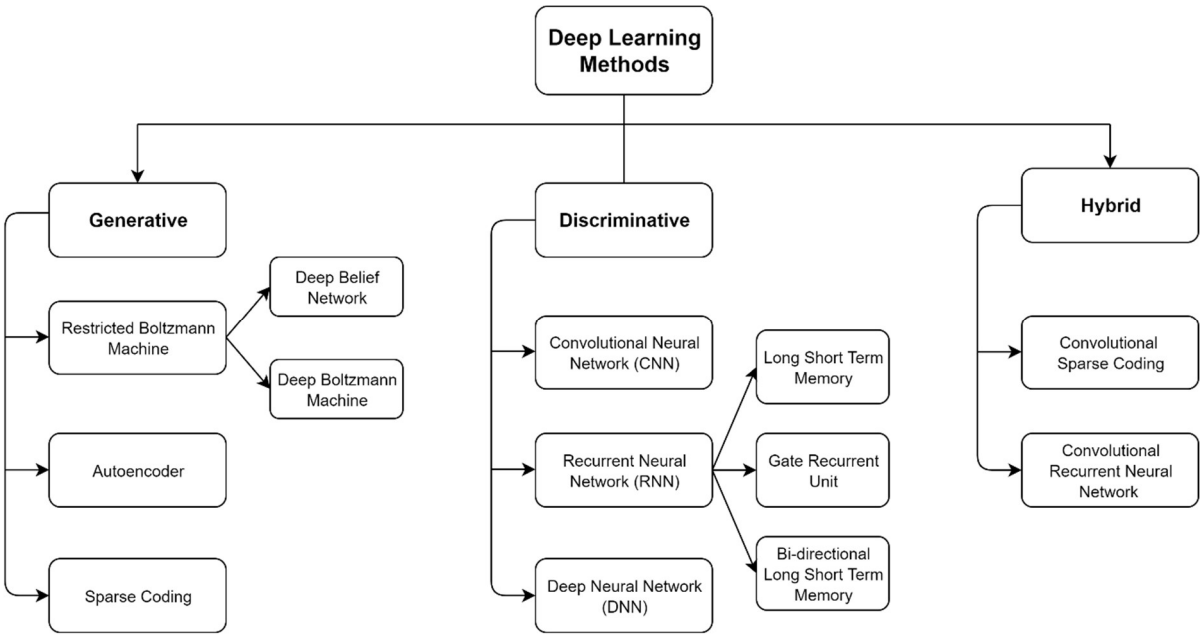


Figure 2. Deep learning method classification based on (Nweke, Teh, Al-garadi, & Alo, 2018).

2.3.1 Discriminative Deep Learning Models

Discriminative feature learning models are modelled with posterior distribution classes to boost their classification and recognition powers (Nweke, Teh, Al-garadi, & Alo, 2018). In recent years, discriminative deep learning methods based on CNNs, RNNs and similar approaches have been applied in activity recognition. In this section, we discuss these applications for HAR.

2.3.1.1 Convolutional Neural Network （CNNs）

CNN performs convolution operations to extract features from large-scaled input data (M.Sarigui, B.M.Ozyildirim, & M.Avci, 2019). Like ordinary neural networks, CNNs are composed of neurons with learnable weights and biases (see Fig. 3). A CNN contains an input layer, one or more convolution and pooling layers, and one or more fully connected layers (Yang, Nguyen, San, Li, & Krishnaswamy, 2015). Each convolutional layer in a CNN consists of several convolutional units. The convolution budget must extract the different features from the dataset. Low-level features are generally extracted by the first convolutional layer, and more complex features are extracted by increasingly higher convolution layers. The pooled layer following the convolutional layer increases the size of the extracted features. CNNs are popularly used to convolve data sets. They have been widely applied in image recognition, speech recognition, and HAR (Liu, Liang, Lan, Hao, & Chen, 2016).

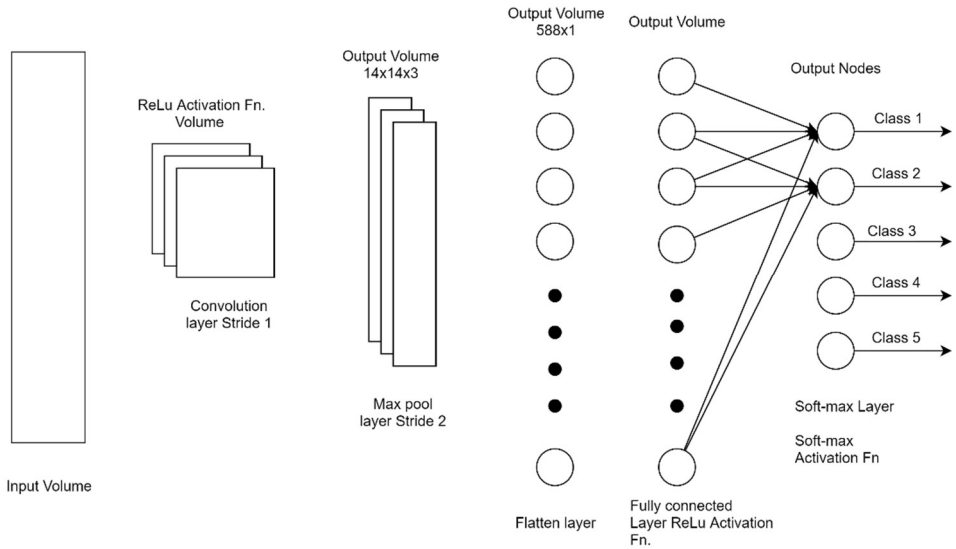


Figure 3. Structure of a convolutional neural network

Ha etc.team (2015) proposed an activity recognition method using a CNN model with a 2D kernel (Ha, Yun, & Choi, 2015). The activity information in their dataset was collected from multiple sensors. Data preprocessing enabled the 2D kernel model to identify the dataset. The data of the various sensors were separated by padding zeros. The

human activity identification performances were compared among CNN with a 2D kernel, CNN with a 1D kernel, CNN−pf (In the first convolutional layer, partial weight sharing, and in the second convolutional layer, full weight sharing.), CNN−pff (In the first convolutional layer, partial and full weight sharing, and in second convolutional layer, full weight sharing), and mechanical learning. The CNN achieved higher performance with fewer parameters and more effective multi-mode data than mechanical learning (Ha & Choi, 2016).

The Stefan team proposed a CNN-based method that classifies 3D gestures using mobile devices. The sensed dataset is compiled into a matrix-data input model of fixed sized by accelerating the instrument and the gyroscope. The results of different settings and different datasets were evaluated in a comparative study. The obtained results were at least equal to the state-of-the-art method at that time (Duffner, Berlemont, Lefebvre, & Garcia, 2014). Zeng (2014) proposed a CNN-based method that automatically extracts the discriminative features in three classes of human activities learned from three public datasets: Skoda (for assembly line activities), Opportunity (for dishwashing, cooking, and other householder activities), and Actitracker (jogging, walking, and other outdoors activities). They exploited the local dependency and scale invariance of CNN, and improved the accelerometer signals by partial weight-sharing technology (Zeng, et al., 2014). Liu, Ying, Han, and Ruan (2018) proposed a HAR for analysing video data. They detected four behavioural activities (walking, running, punching, and tripping) by the Bayes classifier and CNN. The input data were extracted from the KTH dataset, and the extracted features (length-width ratio, entropy, and Hu invariant moment) were detected by a Kalman filter. The accuracy of CNN exceeded that of the Bayes Classifier (Liu, Ying, Han, & Ruan, 2018).

The Kyoung–Soub team proposed a new method that classifies real-time motions by CNN-based HAR. The HAR dataset they selected is about one healthy subject and five rehabilitation motions. They built an intelligent system that collects data from smartphones and smart watches, and segments the data by a time window. The

performances of each CNN were compared by 5-fold cross-validation technology. Finally, the motion information was classified by the CNN, and the optimal time-window size yielding the highest HAR accuracy was determined. The classification ability of the HAR was found to be improved by only minimizing the sample size (Lee, Chae, & Park, 2019).

2.3.1.2 Recurrent Neural Networks（RNNs）

An RNN combines recurrent paths, meaning that the recurring process is a path of information flow. RNN maps the previous input history information to the target vector that can be processed by an internal sequence in memory. Accordingly, the RNN is suitable for unsegmented, continuous handwriting recognition and speech recognition. Whereas the traditional neural network only establishes the connections between adjacent layers, the RNN also establishes connections between the units in directed loops. The advantage of this model is the identification of continuous sequence data (Ye, Yang, Chen, & Wang, 2019). The loops in an RNN store the information in the network structure, enabling connection of the previous information to the execution and analysis of the current task. (Schrauwen, 2013). These connections cannot be made by traditional neural networks. As shown in Fig. 4, the RNN is expanded over time. The neurons between the input and output layers of the RNN contain a non-linear (Mohammadi, Al-Fuqaha, Sorour, & Guizani, 2018).

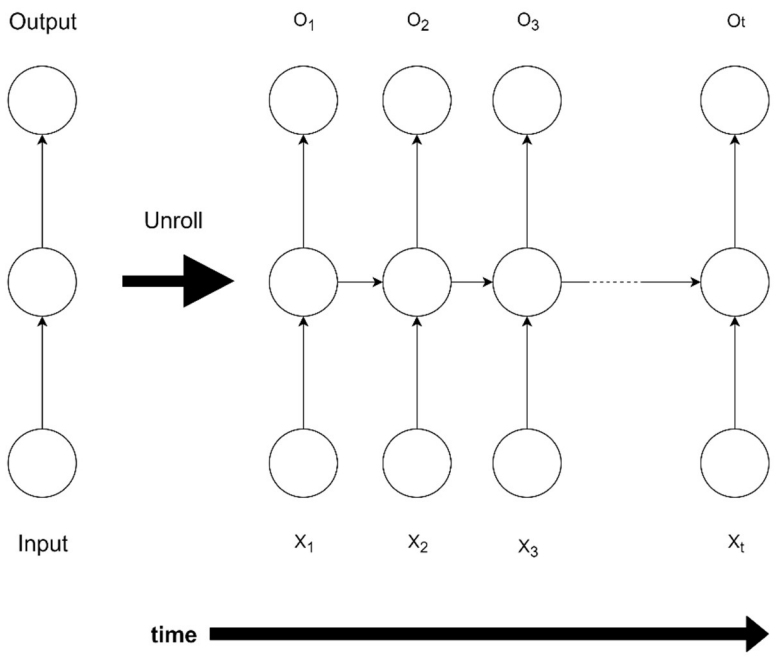


Figure 4. Structure of the RNN, adapted from Mohammadi, Al Fuqaha, Sorour, and Guizani (2018).

2.3.1.2.1 Long Short-Term Memory (LSTM)

RNN has recently achieved great success in various fields. These success factors are mainly attributed to LSTM, an extension of RNN that solves the long-term dependency problem in RNN. The LSTM processes and predicts data with very long delays in time-series intervals (Chen, Baca, & Tou, 2017). In addition to the feedback path that stores information, LSTM embeds an input gate, an output gate, and a forgotten gate in each neuron. Figure 5 shows the cell structure of LSTM (Mohammadi, Al-Fuqaha, Sorour, & Guizani, 2018).

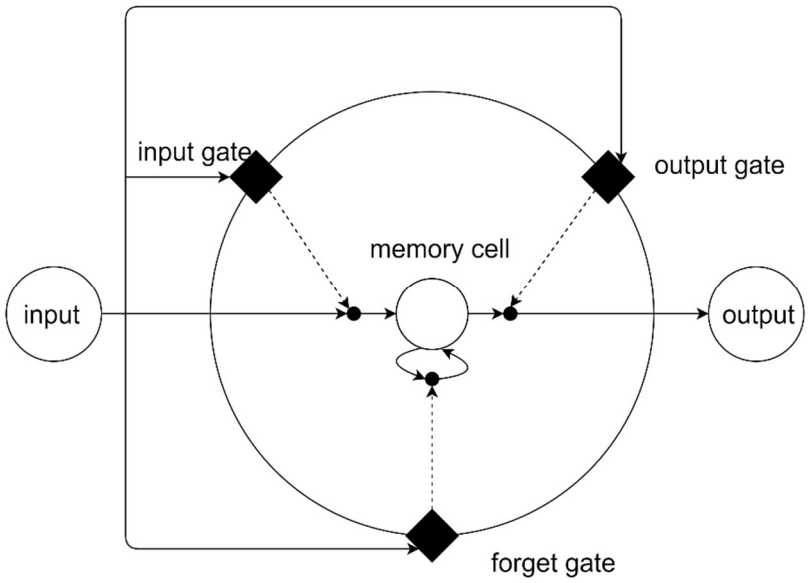


Figure 5. Cell structure of LSTM (Mohammadi, Al-Fuqaha, Sorour, & Guizani, 2018)

LSTM outperforms the RNN model in the analysis of input sensor data. For example, LSTM maintains the time gradient flow by a forgetting gate that does not decay over time. However, the binding efficiency of related data in the LSTM structure is very low (Park, Jang, & Yang, 2018).The output of any neuron is determined by the input of the previous neuron. An RNN stores the previously learned information. During training, a traditional RNN may develop a vanishing or exploding gradient phenomenon, causing diminishment or rapid growth of the error function, respectively. This phenomenon destroys the identification ability of the RNN. LSTM avoids this problem by the memory in each LSTM unit, which can be used by the gate to store, delete or update prior information as needed. Each gate has different weights, biases and activation functions. Bi-directional LSTM achieves superior recognition performance because its neuronal connections can proceed in the backward and forward directions, instead of only to previous cells (Sarma, Chakraborty, & Banerjee, 2019).

Chen etc. team (2017) analysed the HAR data collected by accelerometers and gyroscopes by LSTM. The training model was learned by a deep CNN. The data were images collected from wearable cameras. To mitigate the degradation of the final recognition accuracy by the dynamics and uncertainty of human activities, they replaced the error information by the location information (Chen, Baca, & Tou, 2017). Carfi etc.

team (2018) proposed a recognition method that explains the information delivered by gestures, a type of natural human communication. Gesture interpretation is important for communications between humans and robots. They proposed the SLOTH classifier, a probabilistic classifier that detects 15 gesture sequences based on LSTM analysis of data from a wearable triaxial accelerometer. This system is advantaged by 1) high recognition accuracy, 2) immediate system reaction, and 3) continuous gesture detection (Carfi, Motolese, Bruno, & Mastrogiova, 2018). Li etc. team (2019) researched a bi-directional LSTM using the MOCAP dataset from Carnegie Mellon University which contains high-resolution radar range profiles. They selected six human activities for identification and classification. Although bi-directional LSTM outperforms unidirectional LSTM, its results are sensitive to the length of the data. The best duration of data is around 0.6 S (Li, He, Yang, Hong, & Jing, 2019). Pham and colleagues (2017) developed deep-care LSTM for predicting diabetic disease, and compared its performance with that of Markovian and support vector machine approaches (Pham, Tran, Phung, & Venkatesh, 2017).

A HAR method based on smart phones was proposed by Yu and Qin (2018). Collecting data from a user's cellphone reduces the inconvenience of wearing multiple sensors. Activity recognition was performed by a bi-directional LSTM structure that analyses data from an accelerometer sensor and a gyro sensor in the smartphone. Their network proved superior to other models (machine learning models, DBN and baseline LSTM), reaching accuracies of up to 93.79% (Yu & Qin, 2018).Although HAR has been widely accepted, the spatial complexity and long-time span of human activities have defied many traditional machine learning methods. Accordingly, deep learning algorithms have largely replaced machine learning algorithms. LSTM is ideally suited to processing time series data. Increasing the depths of the LSTM model, the gradient vanishing arises. LSTM recognition minimizes the pre-processing of the raw data, and reduces the versatility of the model in reducing the experimental error (Yu, 2018).

2.3.1.2.2 Gated Recurrent Units (GRU)

Identifying human activities from multimodal human sensor data is a very effective care technique for the elderly or disabled in smart medical environments. Traditional machine learning techniques with a single induction method are unsuitable for medical care. A method that builds simple recurrent units (SRUs) with the GRU of neural networks has been proposed. The SRU processes a multi-mode data series through internal storage states and stores the past states of information in the deep GRU for analysis of future states. GRU resolves the instability and gradient issues, and outperforms currently available state-of-the-art methods (Gumaei, Hassan, Alelaiwi, & Alsalman, 2019).

GRUs do not recognise human activities, but group each human into a local group that represents his or her relationship in the entire scene. The important movement information is maximized by modelling both the individual human motions and the local group relationships. Using a GRU model, Lee, Kim and Lee (2017) captured multiple-relationship time dynamics of multiple lengths. Their method outperformed the Group Interaction Zone (GIZ) and Gaussian Process Dynamical Model (GPDM) methods developed without local grouping (Lee, Kim, & Lee, 2017). He etc. team (2018) proposed a unified architecture based on CNN and GRU for classifying medical relationships in Chinese and English clinical records. Their model uses bi-directional GRU to capture the long-term dependencies of phrase-level features (He, Guan, & Dai, 2018).

Hao etc. team studied a variant GRU with an encoder that preprocesses the data of payload-aware intrusion detection. The payload is extracted from raw traffic data, which is split into individual fixed-length packets. They compared the abilities of two variant GRUs which are encoded gated recurrent unit (E-GRU) and encoded binarized gated recurrent unit (E-BinGRU) and other algorithms in learning network packets. Both variant algorithms provided accurate network-packet rules without requiring human experience, and automatically generated those rules in the correct format (requiring no

format conversion). The detection accuracy of E-GRU and E-BinGRU exceeded 99% (Hao, Sheng, & Wang, 2019).

Compagnon et al. (2019) proposed an activity classification pipeline based on GRU and inertial sequences. They exploited the feature-extraction capabilities of the neural network instead of manually defining the rules or extracting artificial features. Their method resamples the dataset and designs a new GRU model with improved performance. The dataset consisted of two major groups: 1) five common behavioural postures (sitting, lying, standing, walking, and transfer), and 2) activities of daily living (ADL) and falling mishaps provided by the MobiAct V2 dataset. In experimental assessments, the new GRU model improved the performance and enhanced the system deployment potential (Compagnon, Lefebvre, Duffner, & Garcia, 2019).

2.3.1.3 Deep Neural Networks (DNNs)

DNNs are developed through ANNs, but (unlike ANNs) possess many hidden layers. Increasing the number of hidden layers increases the recognition ability of a DNN. DNNs are typically applied as dense layers in other network models (Wang, Chen, Hao, Peng, & Hu, 2018). A model with a 5-layer hidden layer improves the classification performance by automatic feature extraction and classification. The authors demonstrated that a larger number of hidden layers improves the recognition performance of the model when identifying complex HA data (Hammerla, Halloran, & Ploetz, 2016).

Amroun et al. (2017) developed an activity that distinguishes between calling and management using a smart phone and a remote control. The performance of the DNN algorithm was compared with that of decision tree and SVM. By removing the need for data preprocessing, DNN improves the recognition accuracy (Amroun, Temkit, & Ammi, 2017). Later, Amroun et al. (2017) presented a HAR method based on DNN, decision tree and SVM, which analyses the data of smart watches, smartphones and remote controls while the user performs four types of movement (walking, standing, sitting, and lying). The DNN algorithm improved the recognition rate of the system (Amroun,

Temkit, & Ammi, 2017). Hayashi etc. team (2015) proposed a DNN-based method for daily HAR which identifies multi-mode signals (ambient sound and object acceleration). Over a 72-h period, they recorded real data for a comparative experiment of the DNN algorithm, SVM, and other algorithms. In the evaluation, the DNN algorithm outperformed the other algorithms (SVM, K-nearest neighbour, decision tree) (Hayashi, Nishida, Kitaoka, & Takeda, 2015).Cheng etc. team (2017) proposed a DNN-based HAR model that monitors the activity status of patients with early Parkinson's disease. They found significant differences in the percentage of walking time and the frequency of posture changes between the Parkinson's patients and healthy controls (Cheng, Scotland, Lipsmeier, & Kilchenman, 2017).

The configurability and scalability of the DNN classifier in HAR has also been investigated. Bhat's model (2019) inputs the number of inputs, weights, and characteristics of the neurons, and parameterises the multiply accumulated block, weight store, rectified linear unit and MAX functions. As the parameterised neurons are derived from the hidden and output layers, the parametrisation affects the architecture of the hidden DNN. Eventually, the authors designed an activity-aware 2-level HAR engine that recognises two types of recognition activities. This method improves the recognition accuracy while reducing the power consumption of the HAR engine (Bhat, Tuncel, An, Lee, & Ogras, 2019).

2.3.2 Generative Deep Learning Models

Generative deep learning models extract the features from the data by identifying their associated statistical distribution. In the past few years, many HAR studies have applied generative deep learning models. In this section, we introduce related experiments that have been previously studied using this model.

2.3.2.1 Autoencoder

Autoencoder is a type of self-monitoring neural network model. Autoencoders are often used in feature extraction, rebuilding data from corrupted data, and other tasks. Conventional autoencoders have fully connected input layers and output layers of the same size and a fully connected hidden layer of smaller size. Autoencoder algorithms have been used in recommendation systems (Liu, Qiu, Ma, & Wu, 2019). Autoencoders generate feature learning, and are superior to the feature extraction methods of principle component analysis (PCA) and empirical cumulative distribution function (ECDF) (Plötz, Hammerla, & Olivier, 2011). Feng etc. team (2018) proposed a method that improves the efficiency of Autoencoder and avoids falling into a local optimum (Feng, Chen, & Fu, 2018). Their method can be implemented in conjunction with quantum particle swarm optimisation and autoencoder algorithms. Autoencoder includes two main functions: an encoder and a decoder. The encoder accepts the input data and converts it into a new representation. This step is often referred to as code or latent variable. The decoder receives the new code generated by the encoder and converts it into the original input form. Figure 6 shows the main structure of Autoencoder. The basic autoencoder has many extensions such as denoising Autoencoder, sparse Autoencoder and variant Autoencoder (Mohammadi, Al-Fuqaha, Sorour, & Guizani, 2018) .

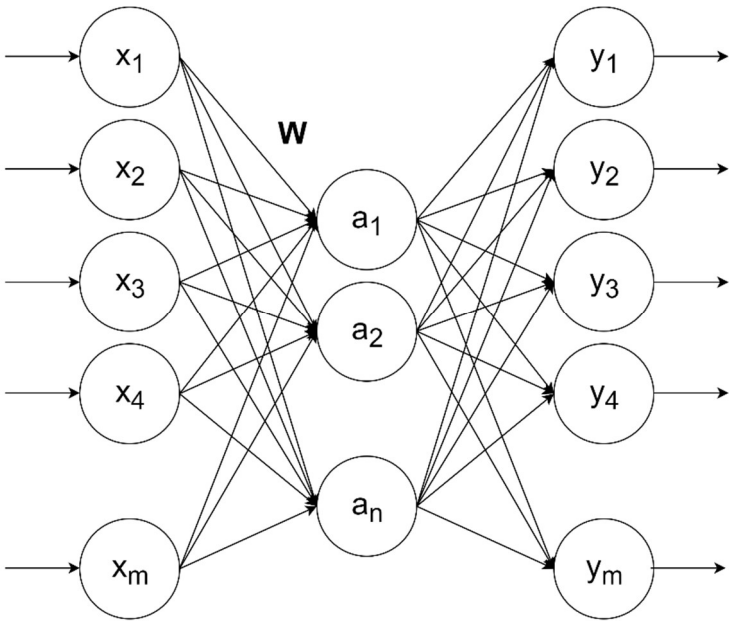


Figure 6. Structure of an Autoencoder network (Xu, Wu, Yang, & Zhang, 2017)

Zhou etc. team (2018) proposed a method for embedding a local autoencoder into an architecture. Local embedding on each cluster-based subgraph tightly integrates the LAN embedding and graph clustering through mutual enhancement, thus saving memory (Zhou, et al., 2018). Zhang etc. team (2018) proposed an Autoencoder-based speech recognition system in noisy environments. This system identifies speech in noisy environments and also the noise features by a multitasking autoencoder combining a denoising autoencoder and a de-speeching autoencoder (Zhang, Liu, Inoue, & Shinoda, 2018).

Fraiwan and Lweesy (2017) identified the sleep states of newborn infants using an Autoencoder. The identification system performs feature extraction and classification. As the statistical parameters to be extracted in the time and spectral domains, they selected 29 EEG records (14 preterm and 15 full-term). To improve the results, they used two autoencoder layers and one softnet output layer, and applied the10-fold cross validation technique. The recognition accuracy was 0.804 (Fraiwan & Lweesy, 2017). Autoencoder is an important branch of deep learning in tasks such as denoising hybrid noises in images. Ye etc. team (2015) removed hybrid noises in images using a sparse denoising autoencoder. The model achieved good performance in single noise recognition and demonstrated outstanding performance in mixed noise. Autoencoder

algorithms are widely used in feature extraction programs because they properly represent the input data (Ye, Wang, Xing, & Huang, 2015).

Denoising autoencoders (DAE) and stacked autoencoders (SAE) have been proposed for low- accuracy acquisition of voices in noisy environments. Autoencoders are powerful tools for extracting and recognising sound features. To improve the accuracy of classifying speech conversations, Janod etc. Team (2017) proposed a method of obtaining combined feature extraction with manual and automatic transcription functions. Exploiting the bottleneck features of Autoencoder, this function combines the advantages of both noisy and clean transcription to improve the robustness of error-prone representations (Janod, Morchid, Dufour, Linarès, & Mori, 2017).

Fournier and Aloise (2019) proposed an autoencoder algorithm that reduces the dimensions of high-dimensional data (images, sentences or recordings). Autoencoder is widely used because it is more flexible than PCA and isometric feature mapping (LSOMAP). They applied four dimension-reduction algorithms to three commonly used image datasets (MINST, Fashion-MNIST and CIFAR-10), and projected the data into a low-dimensional space for identification. For the three datasets classified by k-nearest neighbour, the recognition accuracies of PCA and Autoencoder were almost identical, but PCA greatly reduced the computational time and resource use (Fournier & Aloise, 2019). Badem, Caliskan, Basturk, and Yuksel (2016) proposed a HAR method using two autoencoder layers and one softmax layer. When tested on common datasets of HAR using smartphones, the proposed method achieved better classification results than other techniques (k-nearest neighbour, naive Bayes, and decision tree). (Badem, Caliskan, Basturk, & Yuksel, 2016).

Gu etc. team (2018) identified indoor positioning and navigation-related activities by a method based on stacked denoising autoencoders. They used four sensors (accelerometers, gyroscopes, magnetometers and barometers) to collect the data. The advantages of this method are high recognition accuracy and no manual feature

extraction. The algorithm demonstrated high performance in identifying acceleration data (Gu, Khoshelham, Valaee, Shang, & Zhang, 2018).

2.3.2.2 Sparse Coding

Sparse coding is a deep learning algorithm proposed by Olshausen team (Olshausen & Field, 1997). This method well extracts the characteristics of a dataset. Diego, Reichinnek, Both, and Hamprecht (2013) proposed a method based on sparse coding, which analyses and recognises neuronal activity in calcium imaging data for neuronal activity studies. They used wavelet changes and watersheds to identify the image segmentation of a single unit, and sparse coding to analyse the transient signals. The sensitivity of this method is approximately 94%, significantly higher than those of other published algorithms (Diego, Reichinnek, Both, & Hamprecht, 2013). Manifold regularized sparse coding, which include sparse representations and manifold structures, delivers high performance in motion recognition. Liu etc. team (2016) proposed p-Laplacian regularized sparse coding (pLSC) for HAR, which preserves the local shape by p-Laplacian regularisation. They also proposed a fast-iterative contraction enthalpy algorithm to optimise the pLSC. This method was evaluated in HAR experiments using the non-structural social activity attribute dataset HMDB51 (a human motion database). The pLSC algorithm achieved higher recognition performance than traditional Laplacian regularity, sparse coding and the Hessian regularisation sparse coding algorithm (Liu, Zha, Wang, Lu, & Tao, 2016).

Umakanthan etc. team (2015) proposed sparse coding for identifying actions. In a comparison study of sparse coding and three other algorithms (SRC, Shared Dictionary + SVM, Class Dictionary + SVM), sparse coding delivered the highest recognition accuracy (96.8%) (Umakanthan, Denman, Fookes, & Sridharan, 2015). Wang and Wang (2018) proposed an improved topology-based sparse coding (ITSC) that recognises Alzheimer's disease signals in magnetic resonance imaging (MRI) data sourced from the ADNI dataset. The method is divided into four steps: 1) data preprocessing (such as

correction, registration, segmentation), 2) training of the ITSC datasets, 3) optimisation by an iterative algorithm, and 4) identification of the MRI data using a SoftMax classifier and the auxiliary fine-tuning method. This method is superior to PCA and a self-learning neural network (Wang & Wang, 2018). Zhang and Ma (2012) proposed a new image classification framework that leverages low-rank sparse matrix decomposition and group sparse coding. The local features of the image are decomposed into a low-rank matrix and a sparse matrix by local feature extraction (related terms and specific terms) of adjacent faces of the image. When trained on the low-rank and sparse parts of the datasets, the recognition accuracy of this method was 75.83% ± 0.71% (Zhang & Ma, 2012).

2.3.2.3 Restricted Boltzmann Machine (RBM)

An RBM contains two layers: a stochastic hidden layer and a stochastic visible layer. Each neuron in an RBM is assigned an energy. As shown in Figure 7, the nodes in each layer are connected only to nodes which have no connection and are conditionally independent between the layers (Fang & Hu, 2014).

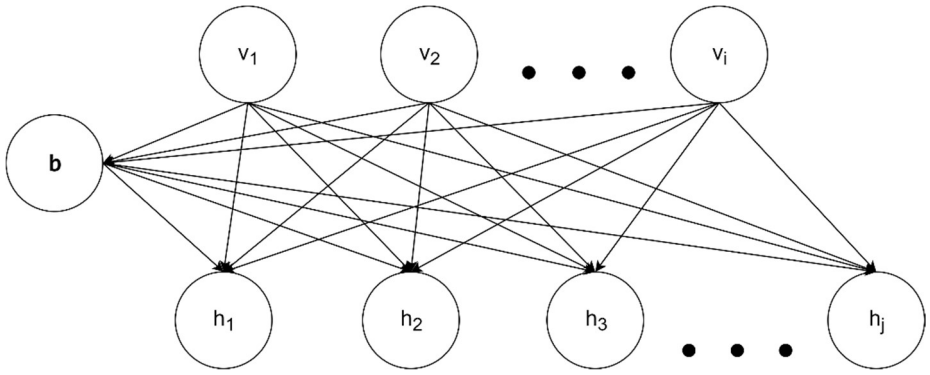


Figure 7. Structure of a Restricted Boltzmann Machine, adapted from Fang and Hu (2014)

Katsageorgiou, Zanotto, Tucci, Murino, and Sona (2017) analysed electrophysiological data using RBMs. Using a latent variable model and a mean-covariance RBM, they found a meaningful configuration corresponding to regularities in the input data. (Katsageorgiou, Zanotto, Tucci, Murino, & Sona, 2017).

RBMs have demonstrated outstanding performance in video detection, behaviour recognition, and other fields. Ajmal etc.team(2019) proposed to identify identified and

evaluated the interactions between people or between people and objects on a surveillance video dataset based on an RBM. In terms of recognition accuracy, this method outperformed the latest technology on an actual monitoring dataset (Ajmal, Ahmad, Naseer, & Jamjoom, 2019). Phan, Dou, Piniewski, and Kil (2015) developed a social RBM that identifies human modeling and predictions in healthy social networks. The model consists of three layers: historical layers, visible layers, and hidden layers. Experiments confirmed the superiority of RBM over divergence and backpropagation algorithms (Phan, Dou, Piniewski, & Kil, 2015).

Lei and Todorovic (2016) proposed a conditional deep Boltzmann machine that detects the remote motions of an active 3D human skeleton. Their model extends the conditional RBM and the factored conditional RBM by introducing additional hidden layers and removing the style layer, while retaining high computational efficiency. The new hidden variables can properly capture the spatiotemporal interactions between human joints, enabling simulations of 3D motion sequences without active and motion patterns (Lei & Todorovic, 2016).

2.3.3 Hybrid Deep Learning Models

Hybrid deep learning methods consist of multiple subsystems, usually controlled by two or three algorithms (Yinghao Chu, 2018). In the HAR literature, the recognition accuracy of hybrid algorithms is usually better than those of single algorithms of the same type.

Akkad and He (2019) estimated the remaining useful life (RLU) of industrial machinery by a hybrid deep learning algorithm. Their hybrid algorithm combines long- and short-term memory and CNNs. They entered the data into the first layer (a one-dimensional convolution layer with 3 filters and a kernel size of 1), and obtained a 3-dimensional tensor feature map by a convolution operation. The feature map generated from the convolutional layer was then input to the LSTM layer. The second and third

layers were LSTM layers with 100 and 50 hidden units, respectively. Finally, the RUL was estimated through the dense layer (Akkad & He, 2019).

Zhang etc. team (2019) proposed a facial expression recognition method based on a hybrid deep learning model. They processed static facial images and optical flow images by two separate CNNs. The processed segment-level spatial and temporal features were input to the DBN model and a deep fusion network was constructed. Finally, the facial expressions were classified by an SVM) (Zhang, Pan, Cui, Zhao, & Liu, 2019).

Li etc. team (2017) proposed a multi-sensor identification method based on a hybrid deep learning algorithm that combines CNN and LSTM. They input the captured spatial features into the CNN layer and the time-extraction features into the LSTM layer. The extracted features were then merged, and a decision was made. This model is scalable and is easily trained and deployed (Li, et al., 2017). Ordóñez and Roggen (2016) also proposed an activity recognition architecture based on CNN and LSTM, which is applicable to multi-mode wearable sensors, naturally fuses the different sensor data, requires no artificial feature extraction, and clarifies the time dynamics of modeling feature activation (Ordóñez & Roggen, 2016).

RBMs can effectively represent complex data in feature extraction from scenes. However, RBMs are unsuitable for processing large images because the calculation becomes very complex. Gao etc.team (2016) proposed a hybrid method called centered convolutional restricted Boltzmann machines (CCRBM) for scene recognition. This method redefines the visible units of the network using a central factor. The hidden function is learned by a distribution function and a modified energy function. The learned hidden unit is used to rebuild the visible unit, and the CCDBN is used for greedy stratification training. Finally, scene recognition is performed by SoftMax regression. Exploiting the convolution characteristics, the CCDBN achieves excellent stability and generalization, and is applicable to discriminative methods for natural scene image recognition (Gao, Yang, Wang, & Li, 2016).

2.4 Evaluation Method and Statistical Test

In this section, we introduce the related work of evaluation method and statistical test for evaluate performance of deep learning models. First, we describe the evaluation metric for deep learning models in recent years. Second, we discuss the statistical tests of performance of deep learning models.

2.4.1 Evaluation Method

The functional performance of HAR is evaluated by preset evaluation techniques such as accuracy, computational time and complexity, robustness, diversity, data size, scalability, and types of sensors. By evaluating deep learning methods, we can understand how parameter changes affect the performance of the model during training. (Nweke, Teh, Al-garadi, & Alo, 2018)

Hold-out cross-validation technology is also one of the Evaluation techniques. Hold-out cross-validation technology can be used to test the performance of models on different datasets. Hold-out cross-validation techniques include: leave one-out, leave one person out when testing the performance of single-user, 10-fold cross validation and so on (Hammerla, et al., 2015).

The most common performance metrics are accuracy, accuracy, recall, confusion matrix, Area Under the Curve (AUC) and Receiver Operating Characteristics (ROC)ROC curve is a performance graph that demonstrates whether the classification model falls below the classification threshold. The ROC curve is also called the precision-recall rate. The ROC curve can compare the true positive rate with the true negative rate give as (FPR). However, the ROC curve is only applicable to the detection model; It cannot be used in the imbalanced datasets commonly used in human activity recognition based on deep learning. (Nweke, Teh, Al-garadi, & Alo, 2018)

2.4.2 Statistical Test

- Analysis of Variance (ANOVA)

The one-way ANOVA determines whether data from different groups have the same mean. Whether the averages of the population differ is assessed by checking the internal variation in each sample (relative to the amount of change between samples). Therefore, the one-way ANOVA calculates two variances: the variance between samples and the variance inside the sample. The two estimated population variances are then compared using the significance value (*p* value) of the F-test. Kolekar's team proposed human activity recognition based on hidden Markov models. They analyzed the features through the ANOVA test and result of the p-value is 0.04, indicating that the features are valid (Kolekar & Dash, 2016).

The Friedman test is a non-parametric statistical test of multiple sets of measures. It can be used to approve the null hypothesis that multiple groups of measures have the same variance at a certain level of significance. On the other hand, failure to approve null hypotheses indicates that they have different values of variance. Oda etc. team introduced the application of neural network (NN) for user identification in the Tor network. They use the Friedman test to analyze the data result. From the results, they adopt null hypothesis H_0 for all activation functions since $p < 0.05$. Since the activation function of softsign/x has the smallest p-value. Then, the softsign/x is moistest suitable for bad user identification in Tor networks (Oda, et al., 2016). The Kaur team proposes to use six machine learning models to predict the software quality of five open-source software. They compare machine learning models results by Friedman's test. The Friedman test can indicate whether there is a statistical difference between the classifiers used (Kaur & Kaur, 2018).

2.5 Datasets

Datasets are important for human activity recognition for evaluate the performance of deep learning. In fact, many data sets can be used in activity recognition. HAR benchmark datasets based on deep learning include data collection schemes which can be divided into two parts (Wang, Chen, Hao, Peng, & Hu, 2018): self-data collection (Bhattacharya & Lane, 2016) and public datasets (Hammerla, Halloran, & Ploetz, 2016). Self-data collection is performed their own data collection, but it is rather tedious to process the collected data. Public datasets are adopted by most researchers. In human activity recognition. most the researchers choose Opportunity Dataset, ARAS datasets and CASAS datasets.

OPPORTUNITY Dataset (Rogge, et al., 2010) is a complex and hierarchical dataset which is the activity of daily living that collects from multiple sensors of different modalities, collect relevant data on the human body and in the environment by using different kinds of sensors. The OPPORTUNITY dataset includes composed of sessions, daily living activities and drills. In the activities of daily life, the subjects were required to perform kitchen-related activities (drinking coffee, eating, opening and closing the refrigerator, etc.). While in the drill session, each action requires 20 repetitive activities for 20 hours. The OPPORTUNITY dataset includes a total of 17 activities and 12 subjects. ARAS human activity dataset (Benmansour, Bouchachia, & Feham, 2015) collected from two real-life with two pairs of residents. The first pair is two males and the second pair is a couple. ARAS data can better study and compare activity recognition algorithms. The critical feature of ARAS data is that it contains a variety of human activities, and a large number of activities occur. A total of 27 activities were carried out.

The center for advanced studies in adaptive systems (CASAS) dataset collects information on the behavior and environmental status of residences surrounding Washington State University. Sensors in the houses collect the relevant data as the residents perform their daily work and life activities. Through these datasets, the research team can realize the automation of smart homes that satisfy human needs (Cook & Das,

Smart Environments: Technology, Protocols, and Applications, 2014). The apartment of CASAS project has one bathroom, one living room, three bedrooms and one kitchen. The sensors are evenly distributed throughout the apartment. The sensor can be divided into several categories:1) motion sensor.2) item sensor for selected items in the Specific area (living room, kitchen).4) experimenter switch (manual trigger such as light switch, fan) and so on. Each different sensor can collect different information (Power consumption, times, status, etc.)

Kyoto (Singla, Cook, & Schmitter-Edgecombe, 2010) (Tested on 2009-2012) and Aruba (Cook, 2010) (Tested on 2010-2012) are most referenced of the CASAS datasets. Kyoto include variety of daily activities (such as filling medication dispenser, setting out ingredients for dinner, reading a magazine, simulating the payment of an electric bill, gathering food for a picnic, retrieving dishes from a kitchen cabinet and packing supplies in the picnic basket and etc.) from two resident. Kyoto dataset collected from environment sensors (motion, item, cabinet, water, burner, phone and temperature sensors), this one contains information include the date and time of each event, the sensor ID and value (binary or numeric) of each sensor activated during the event. Aruba contains activities collected from an older volunteer woman, and her children and grandchildren which includes movement from bed to bathroom, eating, getting home, housework, leaving home, preparing food, relaxing, sleeping, washing dishes and working. The Aruba datasets contains information which are the date and time of each event, the sensor ID and value (binary or numeric) of each sensor activated during the event. In the works of activity recognition, the most referenced CASAS datasets are ARAS, Cairo, Aruba, Kyoto, DOMUS and Tokyo. The most work about these datasets is used different classification models were used to compare the one or more datasets (De-La-Hoz-Franco, Ariza-Colpas, Quero, & Espinilla, 2018).

Fahad team (Fahad, Tahir, & Rajarajan, 2015) propose an activity recognition which can improved the performance of daily activities performed in a smart home, the datasets they selected are Kyoto1, Kyoto7, Kasteren7 and Kasteren10. The accuracy of

Evidence Theoretic K-Nearest Neighbors (ET-KNN) achieve 97%. Fang team (Fang, Srinivasan, & Cook, 2012) propose three probabilistic models (Naive Bayes, Forward procedure of a Hidden Markov Model and Viterbi algorithm based on HMM) recognize human activities in smart home environment. The selected smart apartment dataset is CASAS dataset.

Twomey team (Twomey, Diethe, Craddock, & Flach, 2017)propose two methods that is attempt automatically learn about the sensor network topology, and shown how adjacency matrices between sensors can be constructed. They selected CASAS twor.2009 dataset, because which is multi-class and multi-resident problem with a high proportion of the data annotated.

2.6 Summary

This chapter mainly review the past literature. It introduces several of the most popular deep learning models available today, analyses the framework of several deep learning models, and discusses the applications of related fields. Our experience of model architectures, evaluation methods and datasets guide our subsequent experimental design and project research.

Chapter 3
Methodology

This chapter introduces the specific methodology of our project. We first introduce the necessity and process of the research design, dataset selection and data preprocessing. Next, we introduce the six baseline deep-learning models. Three of these baseline models are selected for extension and improvement. Finally, we describe the evaluation and statistical test for analysing the model performance.

3.1 Introduction

Through studying the published literature and related works, we gain a general understanding of HAR based on deep learning. This chapter introduces the research design, dataset selection, data preprocessing, classification models, evaluation methods and statistical test.

3.2 Research Design

The main aim of this thesis is to evaluate the performance of deep learning models in HAR. Project design is a necessary prerequisite of project development. Figure 8 shows the project flow of the present study. Our specific research structure consists of three main steps which are datasets processing, deep learning models implemented and deep learning models evaluation.

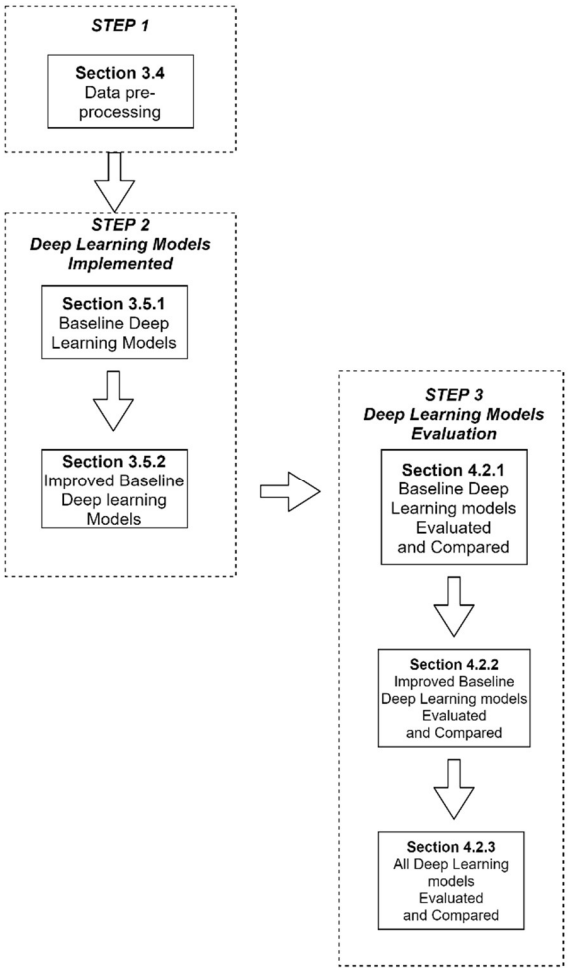


Figure 8. Stepwise evaluation of deep learning models

The first step selects the datasets (Cairo, Kyoto7, Kyoto8, Kyoto11 and Milan) of human activities and determines the data pre-processing. The second step trains the six baseline deep-learning models (CNN, LSTM, GRU, DNN, Autoencoder and sparse coding) to classify the five CASAS datasets, and presumes which baseline deep learning models will deliver excellent HAR performance. These deep learning models are selected for improvement, obtains the three improved baseline deep learning models (BI-LSTM, BI-GRU and LSTM+CNN). Trains the three improved baseline deep learning models to classify these datasets. The third step divides the deep learning models into three groups for evaluation: baseline deep-learning models, improved baseline deep-learning models, and all deep learning models. The training results are analysed by their accuracy, precision, recall, F-measure, AUC, and statistical analysis (Friedman).

3.3 Datasets

The CASAS datasets (Cook, n.d.) were collected by Washington State University's smart apartment research project. The CASAS Smart Home Project collects the daily life information of residents from different types of sensors (e.g., motion sensors, temperature sensors, kitchen-selected item sensors, and electricity sensors) installed in smart apartments. All CASAS datasets contain the date and time of the sensor's data collection, the type of sensor, and the status or value of the sensor. Among the many CASAS datasets, we selected the Cairo, Kyoto7, Kyoto8, Kyoto11 and Milan datasets. Table 1 shows the contents of a portion of the Kyoto8 raw dataset.

Table 1. Part of the Kyoto8 Datasets

DATE AND TIME	***SENSOR***	***STATE/VALUE***
2009-05-29 00:11:27.054181	*T004*	*26*
2009-05-29 00:11:28.014679	*T002*	*23*
2009-05-29 00:11:28.737119	*T003*	*25*
2009-05-29 00:11:29.249039	*T001*	*22.5*
2009-05-29 00:11:29.083165	*T005*	*27.5*

2009-05-29 00:15:36.024549	*P001*	*547*
2009-05-29 00:16:10.343369	*M047*	*ON*
2009-05-29 00:16:11.938839	*M047*	*OFF*
2009-05-29 00:16:12.766999	*M047*	*ON*
2009-05-29 00:16:13.027624	*M048*	*ON*

The Cairo dataset was collected from the sensors of two adult volunteers (residents R1 and R2). The "residents" of the house were men, women, dogs and children entering the house at least once. The sensor types are motion sensors (indicated by "M") and temperature sensors (indicated by "T"). Eleven different activities (wake, night wandering, work in office, laundry, take medicine, sleep, leave home, breakfast, dinner, lunch, bed to toilet) were recorded. Figure 9 depicts the specific deployment of sensors in this room.

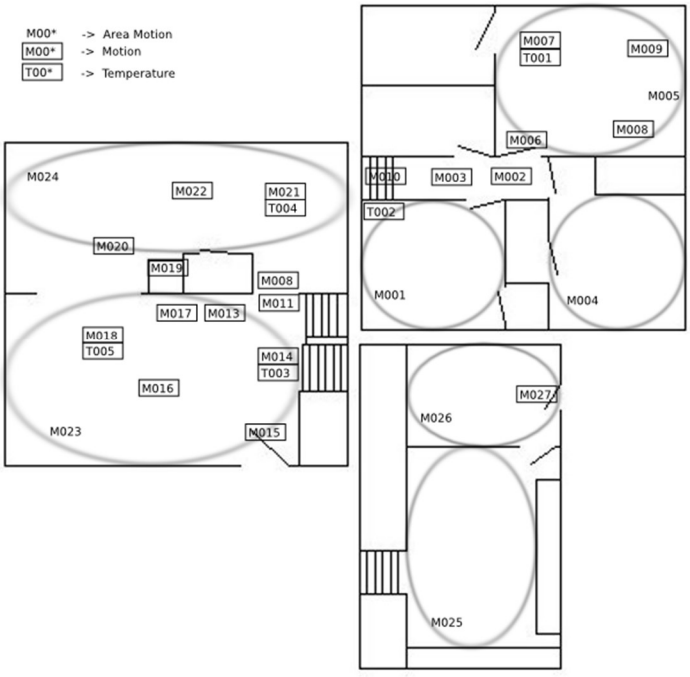


Figure 9. Sensor deployment in Cairo dataset (WSU CASAS Datasets, 2009)

The Kyoto 7, 8, and 11 datasets recorded the daily lives of residents R1 and R2 in their apartment. These three datasets detect signals from the same sensors (motion sensor M, sensor I for the kitchen, door sensor D, temperature sensor T, burner sensor AD1-A, hot water sensor AD1-B, cold water sensor AD1-C and electricity sensors P001),

but recognise different types of activity (see Table 2). Figure 10 is sensor deployment in Kyoto dataset.

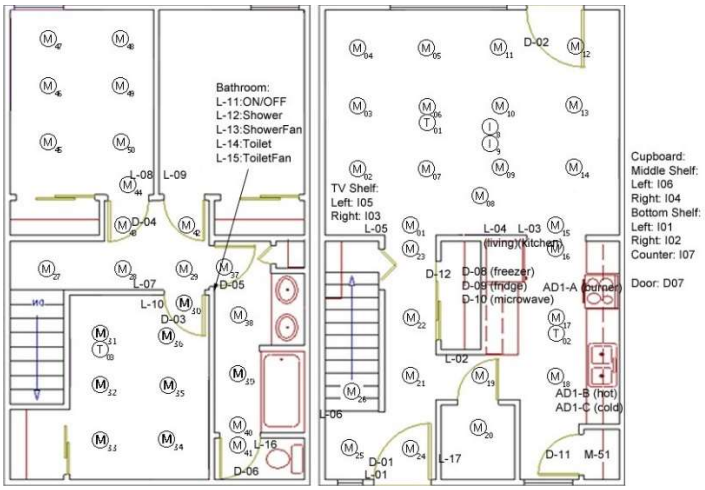


Figure 10 Sensor deployment in Kyoto dataset (WSU CASAS Datasets, 2009)

The Milan dataset is also collected from sensors in smart apartments. Refer to Figure 11. The Milan dataset records the data of two humans (an adult woman and her child) and a dog. The woman's child visited the woman several times. The data are recoded from motion sensors (M), door sensors (D), and temperature sensors (T).

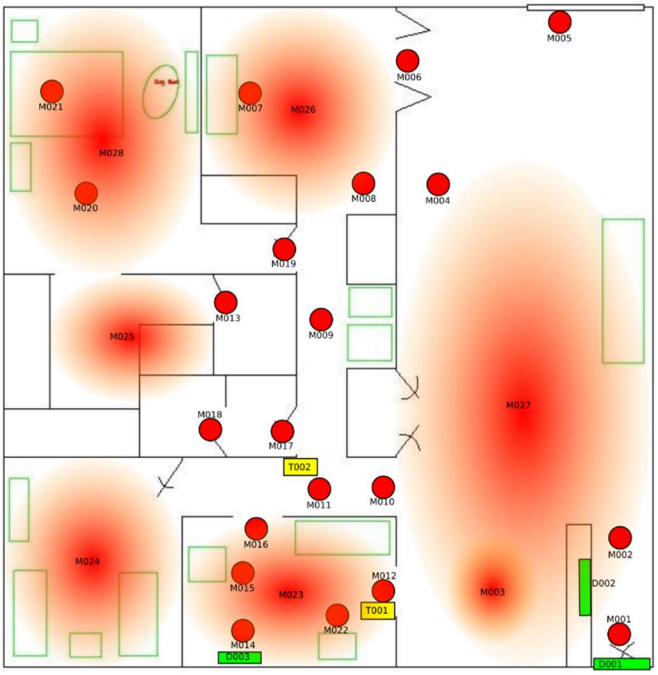


Figure 11 Sensor deployment in Milan dataset. (WSU CASAS Datasets, 2009)

Table 2 summarises the residents, sensors types, activities and activity types of the five CASAS datasets.

Table 2. Contents of the CASAS Dataset

Casas Dataset	Cairo	Kyoyo7	Kyoto8	Kyoto11	Milan
Residents	R1, R2, pet	R1, R2	R1, R2	R1, R2	R1, R2, pet
Sensors Types	M, T	M, D, T, I, AD1A/B/C, P001	M, D, T, I, AD1A/B/C, P001	M, D, T, I, AD1A/B/C, P001	M, D, T
Activities	13	13	12	25	15
Types of Activity	R1 wake, R2 wake, Night Wandering, R1 work in office, Laundry, R2 take medicine, R1 sleep, R2 sleep, Leave home, Breakfast, Dinner, Lunch, Bed to toilet	R1-Bed to Toilet, R2-Bed to Toilet, Meal Preparation, R1-Personal Hygiene, R2-Personal Hygiene, Watch TV, R1-Sleep, R2-Sleep, Clean, R1-Work, R2-Work, Study, Wash-Bathtub	R1_shower, R2_shower, Bed toilet transition, Cooking, R1-sleep, R2-sleep, Cleaning, R1-work, R2-work, Grooming, R1-wakeup, R2-wakeup,	R1-Wandering in room, R2-Wandering in room, R1-Work, R2-Work, R1-Housekeeping, R1-Sleeping Not in Bed, R2-Sleeping Not in Bed, R1-Sleep, R2-Sleep, R1-Watch TV, R2-Watch TV, R1-Personal Hygiene, R2-Personal Hygiene, R1-Leave Home, R2-Leave Home, R1-Enter Home, R2-Enter Home, R1-Eating, R2-Eating, R1-Meal Preparation, R2-Meal Preparation, R1-Bed Toilet Transition, R2-Bed Toilet Transition, R1-Bathing, R2-Bathing	Master Bedroom Activity, Meditate, Chores, Desk Activity, Morning Meds, Eve Meds, Sleep, Read, Watch TV, Leave Home, Dining Rm Activity, Kitchen Activity, Bed to Toilet, Master Bathroom, Guest Bathroom

3.4 Data pre-processing

As high-quality data sets can effectively improve the training results of deep learning, data preprocessing is a necessary prerequisite to data training and analysis. Data preprocessing renders a more suitable dataset format for deep learning models. The process consists of three steps: deleting errors and missing data, reclassifying the dataset,

and transforming the dataset format. The analysis and evaluations of each step are described below:

- Step 1 browses the entire data set and removes the erroneous and missing data to ensure the correctness and integrity of the dataset.
- Step 2 reclassifies the entire dataset into 11 activity categories. As shown in Table 3, each CASAS dataset (Cairo, Kyoto7, Kyoto8, Kyoto11 and Milan) records various types of activities, which inconveniences the overall comparison and analysis of results. For an effective analysis, we re-classified the original datasets into 11 activities category of daily life (work, take medicine, sleep, leave home, eat, bed-to-toilet, bathing, enter home, personal hygiene and other). These activities are summarised in Table 3. The "other" category accumulates the specific activities that cannot be included in any other category. Activities unrelated to the datasets are labelled as "-".
- Step 3 converts the dataset format from the original data format ". excel" (which is a large dataset) to the smaller-volume ". npy" format.

Table 3. Details of the Cairo, Kyoto and Milan datasets

	Cairo	**Kyoyo7**	**Kyoto8**	**Kyoto11**	**Milan**
Other	R1-wake, R2-wake, Night wandering	Study, Wash Bathtub	Bed toilet transition, Grooming, R1-wakeup, R2-wakeup	R1-Wandering in room, R2-Wandering in room	Master Bedroom Activity, Meditate
Work	R1-work in office, Laundry	Clean, R1-Work, R2-Work	Cleaning, R1-work, R2-work	R1-Work, R2-Work, R1-Housekeeping	Chores, Desk Activity
Take medicine	R2-take medicine,	-	-	-	Morning Meds, Eve Meds
Sleep	R1-sleep R2-sleep	R1-Sleep, R2-Sleep	R1-sleep, R2-sleep	R1-Sleeping Not in Bed, R2-Sleeping Not in Bed, R1-Sleep, R2-Sleep	Sleep
Leave Home	Leave home	-	-	R1-Leave Home, R2-Leave Home	Leave Home

Eat	Breakfast, Dinner, Lunch	-	-	R1-Eating, R2-Eating	Dining Rm Activity
Bed to toilet	Bed to toilet	R1-Bed to Toilet, R2-Bed to Toilet	-	R1-Bed Toilet Transition, R2-Bed Toilet Transition	Bed to Toilet
Bathing	-	-	R1-shower, R2-shower	R1-Bathing, R2-Bathing	Master Bathroom, Guest Bathroom
Enter home	-	-	-	R1-Enter Home, R2-Enter Home	-
Personal hygiene	-	R1-Personal Hygiene, R2-Personal Hygiene	-	R1-Personal Hygiene, R2-Personal Hygiene	-
Relax	-	Watch TV	-	R1-Watch TV, R2-Watch TV	Read, Watch TV
Cook	-	Meal Preparation	Cooking	R1-Meal Preparation, R2-Meal Preparation	Kitchen Activity

3.5 Deep learning Models

This section introduces the structure, parameters and formulas of the six-baseline deep- learning models selected in this thesis: CNN, LSTM, GRU, Autoencoder, Sparse Coding and DNN. We also introduce the structure, parameters and formulas of the three improved baseline deep-learning methods, namely, LSTM+CNN, bi-directional LSTM (BI-LSTM) and bi-directional GRU (BI-GRU). The selected baseline deep-learning models are the most popular deep learning models in HAR research. Three of these baseline deep-learning models were then selected for extension and improvement.

3.5.1 Baseline Deep-learning Models

After reviewing the previous literature, we selected six baseline deep-learning models for evaluation. These baseline deep-learning models are separately described in the following sections.

3.5.1.1 Convolutional Neural Network (CNN)

The CNN extracts hierarchies from sensor data, while maintaining invariant features during the conversion process. Figure 12 shows the architecture and parameters of our five-layer CNN model (Zeng, et al., 2014)`.

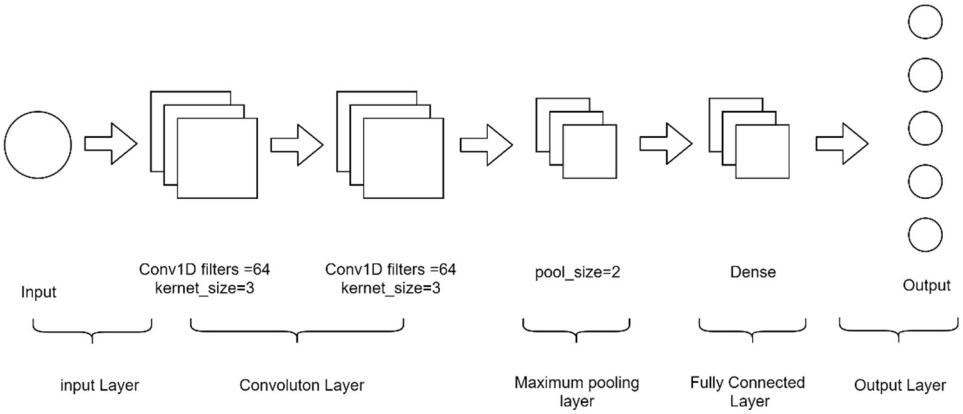


Figure 12 Architecture of the CNN

- Input layer

The first layer of the CNN structure is the input layer, which accepts the data. The format of the input data must be consistent with the requirements of the neural network.

- Convolution layer

The convolutional layer is the feature-extraction layer of the CNN. In this layer, the input data are passed through a feature filter that performs local filters and inner product operations on the input data. The output is converted to its corresponding value in the convolution output matrix for the next operation. We assume an *N*-unit input layer and a filter size of *m*. The output value is sized ($N - m + 1$) units. The convolution layer is formulated as

$$x_i^{l,j} = f(\sum_{a=1}^{m} w_a^j x_{i+a-1}^{l-1,j} + b_j) \quad . \qquad (3.1)$$

where $x_i^{l,j}$ is the output of the jth feature map in the ith unit of convolutional layer l, w_a^j defines the convolutional kernel matrix, and b_j denotes the bias in the convolutional feature maps. The weight is obtained by summing the bias and the result of the

convolution operation on the output feature map of the previous layer. The nonlinear mapping is then performed by an activation function *f*.

- Maximum pooling layer

The maximum pooling layer performs sparse processing on the feature map. When a convolutional layer detects a particular feature, it needs only to retain its approximate position relative to other features (no exact location is required). The maximum pooling layer reduces the sensitivity of the output and the amount of data calculation. $x_i^{l,j}$ Indicates the output after the pooling process. *r* is the size of the pooled kernel. The activation function of the largest pooling layer in the CNN is given by

$$x_i^{l,j} = max_{i,j=1}^{r}(x_{i,j}) . \qquad (3.2)$$

- Fully connected layer

The fully connected layer re-fits the extracted features to prevent loss of feature information.

- Output layer

The target result of the output layer for preparing the output

As shown in Figure 12, the CNN model consists of two identical convolutional layers. In each convolutional layer, the filter size is 64 and the kernel size is 3. The size of the maximum pooling layer is 2. The classifier is SoftMax.

3.5.1.2 Long Short-Term Memory (LSTM)

The LSTM model performs well in complex activities and related datasets with dynamic time (Nweke, Teh, Al-garadi, & Alo, 2018). LSTM is one of our selected baseline deep-learning models. The most important structural units of LSTM are the gates that store the long-term states. The main components of the LSTM model are the input gate, output gate and forget gate. Each gate allows the selective transmission of information through a neural layer with sigmoidal functions and point-by-point

multiplication operations. Figure 13 illustrates the structure of an LSTM. The functions and conversion equations of each gate are described below.

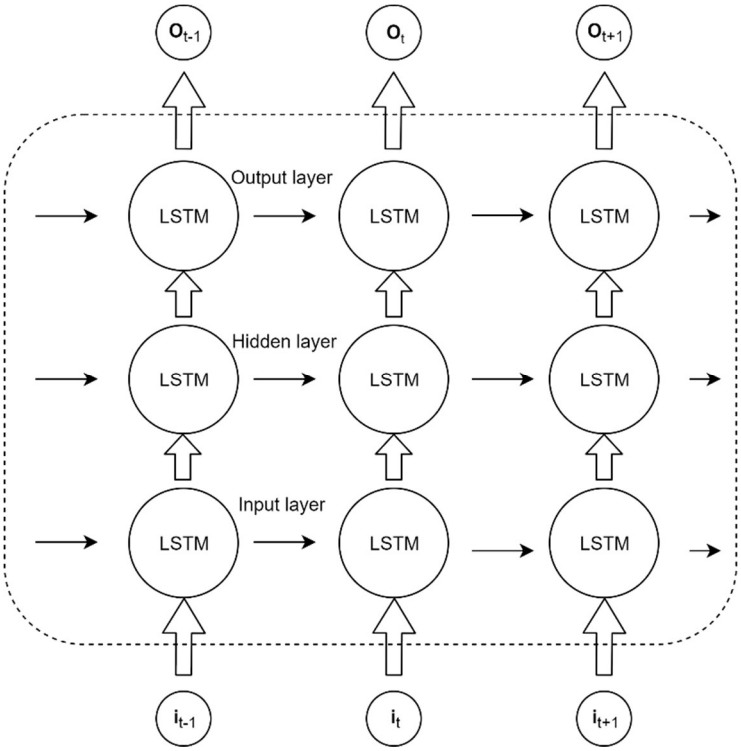


Figure 13 Structure of an LSTM, based on Liciotti (2019)

- Input Gate

The input gate decides how much of the current network input is saved to the cell state. Let x_t be the current input, and h_{t-1} be the previous state. When the current input and the previous state enter the input gate at the same time, the calculation result is multiplied by the weight matrix and passes through a sigmoid or a tanh layer to determine which information needs to be updated. We define the input gate by i_t, the internal memory cell state by C_t, and the weight matrix by W. σ is the sigmoid function, and tanh is the hyperbolic tangent activation function. The related equations (Li, Wang, Liu, & Chen, 2018) are given as

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i)\,, \qquad (3.3)$$

$$\widetilde{C_t} = tan\,h(W_c * [h_{t-1}, x_t] + b_C). \qquad (3.4)$$

- Forget Gate

The forget gate decides which cells of the upper layer should be forgotten in the current layer, and saves the remaining cells to the current cell state. The forget gate gets the current input x_t and the previous state h_{t-1}, then outputs a probability in the range 0–1. An output of 0 or 1 means complete abandonment and complete reservation, respectively. f_t is the output gate. The related equation (Li, Wang, Liu, & Chen, 2018) is as follows:

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \ , \qquad (3.5)$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t}. \qquad (3.6)$$

- Output Gate

The output gate is the output of the cell with the new state. The sigmoid layer determines the parts of the cell to be exported, and the cell state is input to the tanh layer which outputs a probability from -1 to 1. Finally, the probability values are multiplied by the output of the sigmoid layer. o_t is the output gate. The related equations are given by (Li, Wang, Liu, & Chen, 2018):

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \ , \qquad (3.7)$$

$$h_t = o_t * tan\, h(C_t). \qquad (3.8)$$

3.5.1.3 Gated Recurrent Unit (GRU)

The GRU is one of the deep learning models that we selected for data training. GRU is an improved version of standard RNN (itself a simplified version of LSTM), and appears to perform comparably to LSTM but with a simpler structure and lower computational complexity. The internal modellings of GRU and LSTM are similar, being composed of different gates, but unlike LSTM, GRU has no separate storage unit. GRU has two gates, a reset gate and an update gate.

The reset gate defines the combination of the previous memory and the new input, which is applied directly to the previous state. The reset gate mainly determines whether the current state needs to be combined with previous information. The update

gate combines the functions of the input and forget gates to determine the information to be discarded and the information to add into memory.

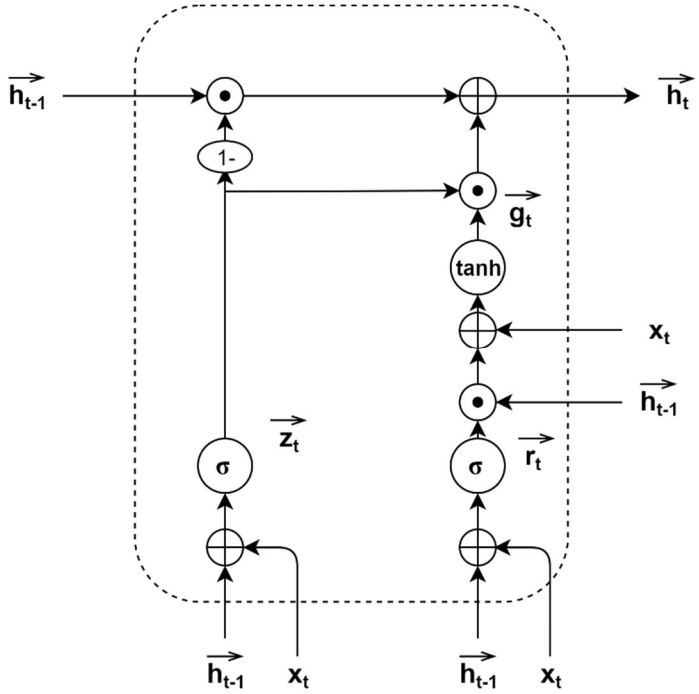


Figure 14 . Structure of a GRU cell (Deng, Wang, Jia, Tong, & Li, 2019)

Figure 14 is a schematic of a GRU cell. At time interval *t*, the reset gate $\overrightarrow{r_t}$ adjusts the incorporation of the new input with the previous memory. The reset gate corresponds with the weight matrices $\overrightarrow{W_{xr}}$ and $\overrightarrow{W_{hr}}$, and with the bias $\overrightarrow{b_r}$. Once the reset gate is closed, all information that is irrelevant to the current hidden state is discarded. Meanwhile, the update gate $\overrightarrow{z_t}$ determines how much of the information from the previous state flows to the current hidden state. The update gate corresponds with weight matrices $\overrightarrow{W_{xz}}$ and $\overrightarrow{W_{hz}}$, and with the bias $\overrightarrow{z_t}$. The candidate cell state $\overrightarrow{g_t}$ is dictated by the previous cell state $\overrightarrow{h_{t-1}}$ and the input vectors x_t . It corresponds with the weight matrices $\overrightarrow{W_{xg}}$ and $\overrightarrow{W_{hg}}$, and with the bias $\overrightarrow{b_g}$. The final cell state $\overrightarrow{h_t}$ is determined in two parts: one part calculated by the elementwise product of $(1 - \overrightarrow{z_t})$ and $\overrightarrow{h_{t-1}}$, the other part calculated by the elementwise product of $\overrightarrow{z_t}$ and $\overrightarrow{g_t}$. During time step *t*, the cell states of a GRU are calculated as (Deng, Wang, Jia, Tong, & Li, 2019):

$$\overrightarrow{r_t} = \sigma\left(\overrightarrow{W_{xr}}x_t + \overrightarrow{W_{hr}}h_{t-1} + \overrightarrow{b_r}\right), \quad (3.9)$$

$$\overrightarrow{Z_t} = \sigma\left(\overrightarrow{W_{xz}}x_t + \overrightarrow{W_{hz}}h_{t-1} + \overrightarrow{b_z}\right), \quad (3.10)$$

$$\overrightarrow{g_t} = tanh\left(\overrightarrow{W_{xg}}x_t + \overrightarrow{W_{hg}}(\overrightarrow{r_t} \odot h_{t-1}) + \overrightarrow{b_g}\right), \quad (3.11)$$

$$\overrightarrow{h_t} = (1 - \overrightarrow{z_t}) \odot h_{t-1} + \overrightarrow{z_t} \odot \overrightarrow{g_t}. \quad (3.12)$$

In the above expressions, $\overrightarrow{W_{xr}}$, $\overrightarrow{W_{xz}}$, and $\overrightarrow{W_{xg}}$ are the weight matrices for connection to the input vector x_t. $\overrightarrow{W_{hr}}$, $\overrightarrow{W_{hz}}$ and $\overrightarrow{W_{hg}}$ are the weight matrices for connection to the state vector $\overrightarrow{h_{t-1}}$ of the previous cell, and $\overrightarrow{b_r}$, $\overrightarrow{b_z}$ and $\overrightarrow{b_g}$ are the bias vectors. All of the above weight matrices and biases are shared by all time steps and are learned during the model training. The σ and tanh are nonlinear activation functions. Specifically, σ is a logistic sigmoid ($\sigma(x) = (1/1 + e^{-x})$) and tanh is the hyperbolic tangent function ($\tanh(x) = (e^x - \frac{e^{-x}}{e^x} + e^{-x})$). The elementwise product of two vectors is denoted by $\odot$.

3.5.1.4 Autoencoder

Another of our selected deep learning methods is Autoencoder, an unsupervised learning algorithm often used in dimensionality reduction or feature learning. Autoencoder is a multi-layered neural network containing an input layer, a hidden layer, and an output layer. Like other deep learning methods, the input and output layers accept the input data and dispense the output data, respectively. The input and output layers of an autoencoder have the same number of dimensions, but the dimensionality of the middle layer (hidden layer) is set low to achieve the desired dimensional reduction. The hidden layer has two partial encoders and decoders which pass the input dataset to the output layer. Figure 15 is the structure of an Autoencoder network. The mapping functions of the encoder and decoder are described below.

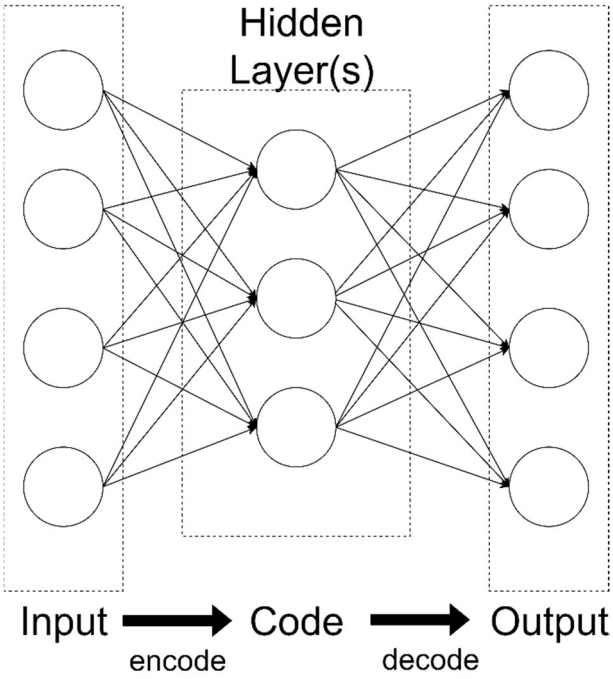


Figure 15 Structure of an Autoencoder network based on (Mohammadi, Al-Fuqaha, Sorour, & Guizani, 2018)

- Encoder

The hidden layer compresses the input data vectors $\left(x_i \in R^d\right)$ into $m(m < d)$ neurons. The activation equation for the neurons is given by (Chen, Yeo, Lee, & Lau, 2018)

$$h_i = f_\theta(x) = s(\sum_{j=1}^{n} W_{ij}^{input} x_j + b_i^{input}) . \qquad (3.13)$$

In Eq. (3.13), x is the input data, and θ represents the parameter $\{W_{ij}^{input}, b_i^{input}\}$. W is the weight matrix of the encoder, and b is an m-dimensional deviation vector. This equation transforms the input data into low-dimensionality data through the encoder.

- Decoder

The following equation decodes the data through the hidden layer into the original input space. The decoder parameters are W^{hidden} and b^{hidden}. The decoding equation is given by (Chen, Yeo, Lee, & Lau, 2018)

$$x_i' = g_{\theta'}(h) = s\left(\sum_{j=1}^{n} W_{ij}^{hidden} x_j + b_i^{hidden}\right). \qquad (3.14)$$

3.5.1.5 Sparse Coding

Sparse coding is an unsupervised learning method that seeks a set of base vectors by learning the dataset. The base vectors provide an efficient representation of the sample data. One assessment criterion is the extent to which the code describes the input. This can be measured by the squared error between the input and its reconstruction by the network (Li, Shi, Li, & Shi, 2009):

$$Error(A,S) = \sum_{x,y}[I(x,y) - \sum_i a_i \Phi_i(x,y)]^2. \qquad (3.15)$$

The cost of seeking sparse codes (called the *sparseness*) is an additional standard of sparse coding. The sparse cost is calculated as

$$Sparseness(A,S) = \sum_i S(\frac{a_i}{\sigma_i}), \qquad (3.16)$$

where $S(x)$ is a nonlinear function. The Sparseness tries to minimise the number of non-zero coefficients. The found coefficients are statistically independent of each other. Any higher-order statistical structure in the input data can be captured by the following sparsity cost function:

$$E(a,\Phi) = \sum_{x,y}[I(x,y) - \sum_i a_i \Phi_i(x,y)]^2 + \lambda \sum_i S(\frac{a_i}{\sigma_i}). \qquad (3.17)$$

3.5.1.6 Deep Neural Network (DNN)

A DNN contains an input layer, two or more hidden layers, and an output layer. A DNN has more parameters than traditional three-layer ANN because it contains more hidden layers and more units per hidden layer. Owing to the large number of parameters, a DNN can automatically learn the suitable classification functions from raw sensor data. The parameters of a DNN include weights and biases. The data vector of a frame is fed into the input units of the DNN. The activation probability y_j of hidden unit *j* is calculated using the inputs from the previous layer. The equation is as follows (Zhang, Wu, & Luo, 2015):

$$y_j = \frac{1}{1+e^{-m_j}}, m_j = b_j + \sum_i y_i w_{ij}, \quad (3.18)$$

where b_j is the bias of unit j, i indexes the unit of the previous layer, and y_i is the input of unit i in the previous layer. w_{ij} is the weight between unit i and unit j. In multi-category classification, the SoftMax function in output unit j converts the input of the previous layer to a class probability p_j, calculated as follows (Zhang, Wu, & Luo, 2015):

$$p_j = \frac{e^{m_j}}{\sum_k e^{m_k}}, m_j = b_j + \sum_i y_i w_{ij}, \quad (3.19)$$

where k indexes the classes.

3.5.2 Improved baseline deep-learning models

After investigating previous related papers, we noted that CNN, LSTM and GRU performed well in HAR applications. In a CNN architecture, Zebin, Scully, and Ozanyan (2017) identified multi-channel lines of time series (human behaviour recognition) acquired from a set of wearable sensors. The classification accuracy reached 97.01%. Yu, Chen, Yan, and Liu (2018) classified human activities in an LSTM network, achieving a classification accuracy of 94.34%. Compagnon, Lefebvre, Duffner, and Garcia (2019) classified the common postures of five people by GRU. The identification accuracy of their method was 91.1%.

We assume that CNN, LSTM and GRU will deliver excellent performance in HAR of residents in the CASAS dataset. These three methods are expected to outperform DNN, Autoencoder and sparse coding. We hope to optimise CNN, LSTM and GRU by changing the architecture of the model. We therefore improve the CNN and LSTM by developing hybrid models, and obtain CNN+LSTM. We also improve the LSTM and GRU by developing bidirectional, obtain BI-LSTM and BI-GRU. This section introduces the three improved deep model architecture of CNN, LSTM and GRU.

3.5.2.1 Bi-directional Long Short-Term Memory (BI-LSTM)

LSTM can solve gradient disappearances and explosion problems to some extent. Time-dependent learning on long-term axes can be simplified when the LSTM reaches a sufficient depth. However, the CASAS datasets record the continuous activity tracks of the residents, and LSTM may be unable to accurately predict the current state based on the previous information. To resolve this problem, we apply the BI-LSTM model, which predicts the current state not only from the previous information, but also from the subsequent information (Yu & Qin, 2018).

The BI-LSTM can obtain information from different directions, meaning that past and future information can be obtained horizontally. Moreover, the information of the lower layer can be obtained from the vertical direction. Figure 16 shows the architecture of our bidirectional LSTM.

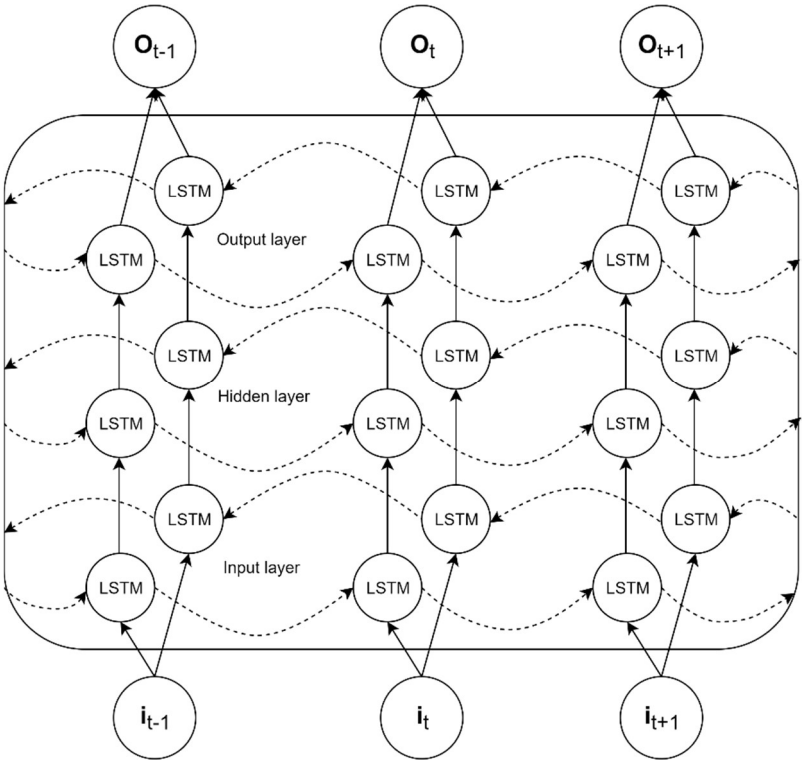


Figure 16 Bidirectional LSTM architecture based on Liciotti (2019).

Let i denote the input data. The first layer of BI-LSTM is the input layer, which receives the preprocessed data. The entered data are then input to the hidden layer, which contains a forward sequence $\vec{h}$ and a backward sequence $\overleftarrow{h}$. The forward and backward

dotted arrows show the left-to-right and right-to-left directions of reading the input data i, respectively. The final predicted output is the weighted sum of the two predicted scores (forward and backward tracks). From left to right, the time sequence is t-1, t, $t+1$.The forward sequence, backward sequence, and input layer are respectively calculated as (Yu & Qin, 2018):

$$\vec{h}_t = g(U_{\vec{h}}x_t + W_{\vec{h}}\vec{h}_{t-1} + b_{\vec{h}}), \quad (3.20)$$

$$\overleftarrow{h}_t = g\left(U_{\overleftarrow{h}}x_t + W_{\overleftarrow{h}}\overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}\right), \quad (3.21)$$

$$y_t = g(V_{\vec{h}}\vec{h}_t + V_{\overleftarrow{h}}\overleftarrow{h}_t + b_y). \quad (3.22)$$

3.5.2.2 Bi-directional Gated Recurrent Unit (BI-GRU)

In theory, GRU accurately classifies long-term datasets. However, the performance of GRU may degrade during actual experiments, because the GRU can only access past information, not the future information. To solve this limitation problem of the GRU model, Deng, Wang, Jia, Tong, and Li (2018) proposed Bi-GRU, which simultaneously learns the past and future information in the sequence to understand the meaning of the sequence.

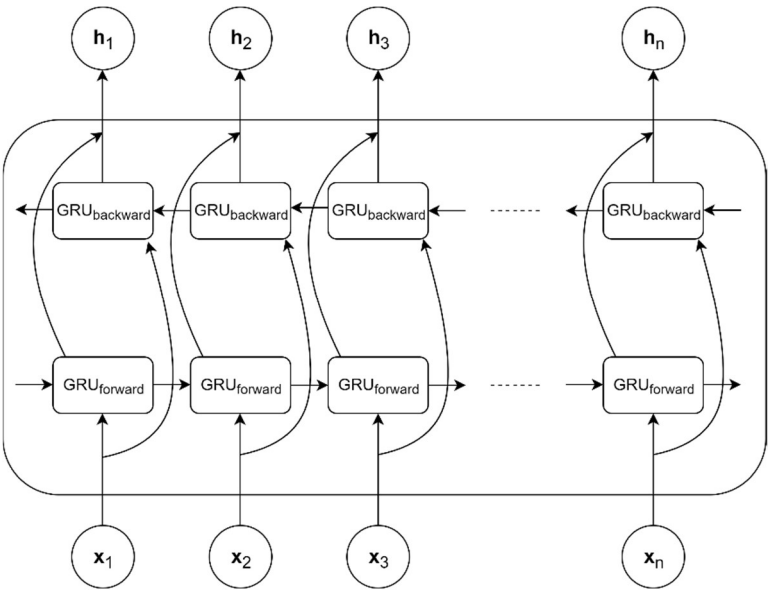


Figure 17 Structure of BI-GRU (Deng, Wang, Jia, Tong, & Li, 2018)

Figure 17 shows the framework of our adopted BI-GRU model. When data are input to the BI-GRU, the output depends on the states of the previous frame *t*-1 and the next frame *t*+1. The BI-GRU then extracts the available information from the bidirectional data. As shown in Figure 17, when one GRU is flowing forward and calculating the forward hidden state ($\underrightarrow{h_1}, \underrightarrow{h_2}, \underrightarrow{h_3}, ..., \underrightarrow{h_n}$), the other GRU flows backwards and calculates the backward hidden layer state ($\underleftarrow{h_1}, \underleftarrow{h_2}, \underleftarrow{h_3}, ..., \underleftarrow{h_n}$). The final output of the BI-GRU is calculated from the hidden states in both directions. The complete hidden element of the BI-GRU, represented by h_t, is the concatenated vector of the outputs in the forward and backward directions as follows:

$$h_t = \overrightarrow{h_{t-1}} \oplus \overleftarrow{h_{t+1}} . \qquad (3.23)$$

3.5.2.3 Long Short-Term Memory + Convolutional Neural Network (LSTM+CNN)

LSTM excels in time-related datasets, and CNN performs excellently in feature extraction. Wu, Zheng, & Zhao (2019) proposed an L-CNN model that combines an LSTM layer and a CNN layer. The model is combined into a new model through the complementarity of CNN and LSTM. Here, we combine CNN and LSTM into a new hybrid model based on the deep models in section 3.5.1. Figure 18 shows the structure of our proposed LSTM+CNN. The time information in the signal is extracted by the LSTM layer. The CNN model performs the feature extraction and behaviour classification on the LSTM output.

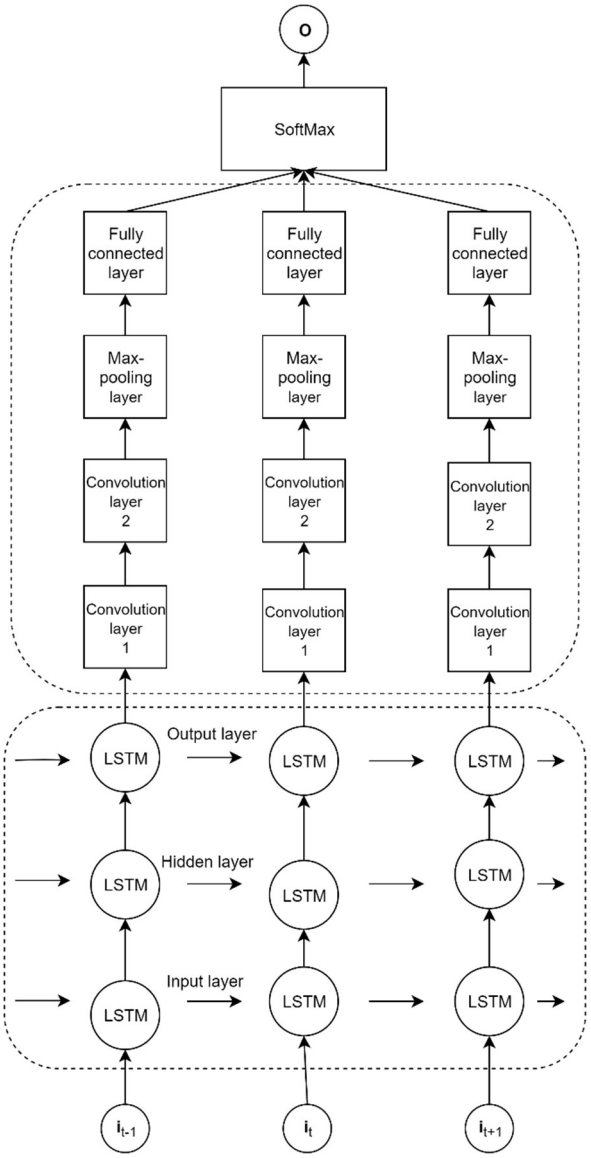


Figure 18 Structure of LSTM+CNN (Wu, Zheng, & Zhao, 2019)

As shown in Figure 18, the LSTM-CNN model consists of eight layers. The data are received by the input layer of the LSTM, and are output after passing through the hidden layer of the LSTM. The data leaving the output layer of the LSTM are input to the first and second convolutional layers for convolution. Next, they are operated by the max-pooling layer and transferred to the fully connected layer. Finally, the classification result is output through SoftMax.

3.6 Evaluation Method

Many evaluation algorithms are available for deep learning algorithms. In this project, the performances of the nine models were evaluated by the accuracy, precision, recall, F-measure and AUC.

The evaluation methods require four values: True positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). When an activity is correctly identified, the number of classes reduces to two: TP and TN. When an activity is misclassified, it can be FP or FN. The meanings and calculation formulae of each evaluation method are outlined below.

- Accuracy

Accuracy provides the number of correctly classified instances. Accuracy equals the sum of the correct classification divided by the total number of classifications.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+F} \quad (3.24)$$

- Precision

Precision defines the number of correct positive predictions among the total number of positive predictions.

$$Precision = \frac{TP}{TP+FP} \quad (3.25)$$

- Recall

Recall defines the number of correct predictions among the forward data

$$Recall = \frac{TP}{TP+FN} \quad (3.26)$$

- F-measure

The F-measure is the weighted harmonic average of precision and recall

$$F - measre = 2 * \frac{Precision*Recall}{Precision+Recal} \quad (3.27)$$

- AUC

The AUC defines the two-dimensional area under the ROC curve. This performance indicator is constant for unequal error costs and unbalanced class sample

sizes. The AUC of the basic classifier is 0.5. An AUC of 1.0 indicates an ideal classifier. The AUC is calculated as follows (Moin & Parviz, 2009):

$$AUC = \frac{\sum_{X_0} \sum_{X_1} I(h(X_1) > h(X_0))}{|X_0||X_1|} \quad , \qquad (3.28)$$

where $|X_0|$ and $|X_1|$ represent the number of instances of each class in binary classification problem, and $I(u)$ is the indicator function.

3.7 Statistical Test

Data analysis comprehensively checks the results of the data. The useful data and information can then be integrated into the evaluation results. Statistical inference is an important tool for evaluating sample datasets collected through research and experiments. Referring to Demšar (2016), I compared the performances of the selected classifiers by a statistical test. Specifically, I benchmarked multiple classifiers on multiple datasets by the Friedman test.

The Friedman test analyses the deep learning models based on their ranks on each dataset. Given n datasets and m deep learning models, each deep learning model is individually ranked on each dataset based on its evaluation metric (accuracy, precision, recall, F-score and AUC). For instance, if the performance accuracy pa of learning model M_j on dataset D_i satisfies $pa_{ij} > pa_{ij'}, \ \forall j', j, j' \in \{1,2,\dots,k\}, j \neq j'$, then model M_j is Rank 1.Under the null hypothesis, all classifiers perform equivalently, The Friedman test is calculated as follows:

$$X_F^2 = \frac{12*n}{m(m+1)} \left(\sum_{j=1}^{m} R_j^2 - \left(\frac{m(m+1)^2}{4}\right)\right) , \qquad (3.29)$$

where X_F^2 has $m-1$ degrees of freedom.

3.8 Summary

This section introduced the source and content of the datasets and our pre-processing of the data. The section on the models introduced the basic framework, mathematical equations and parameters of each deep learning algorithm. Next, we introduced the evaluation methods and statistical test. The next chapter presents and summarises our experimental results.

Chapter 4
Results and Discussion

This chapter focusses on the evaluation results of each deep learning model for HAR. These results discuss in three parts. First, we discuss the six baseline models. Second, we discuss the three improved baseline models. Finally, we comprehensively compare the nine deep learning models.

4.1 Introduction

In the previous chapter, we introduced the research design, dataset selection and data preprocessing, and provided the structure of each deep learning model. After training these deep learning models, we obtained the evaluation results, namely, the accuracy, precision, recall, F-score, and AUC. The evaluation results were statistically analysed by the Friedman test. This chapter presents and analyses the evaluation results.

4.2 Experimental Results

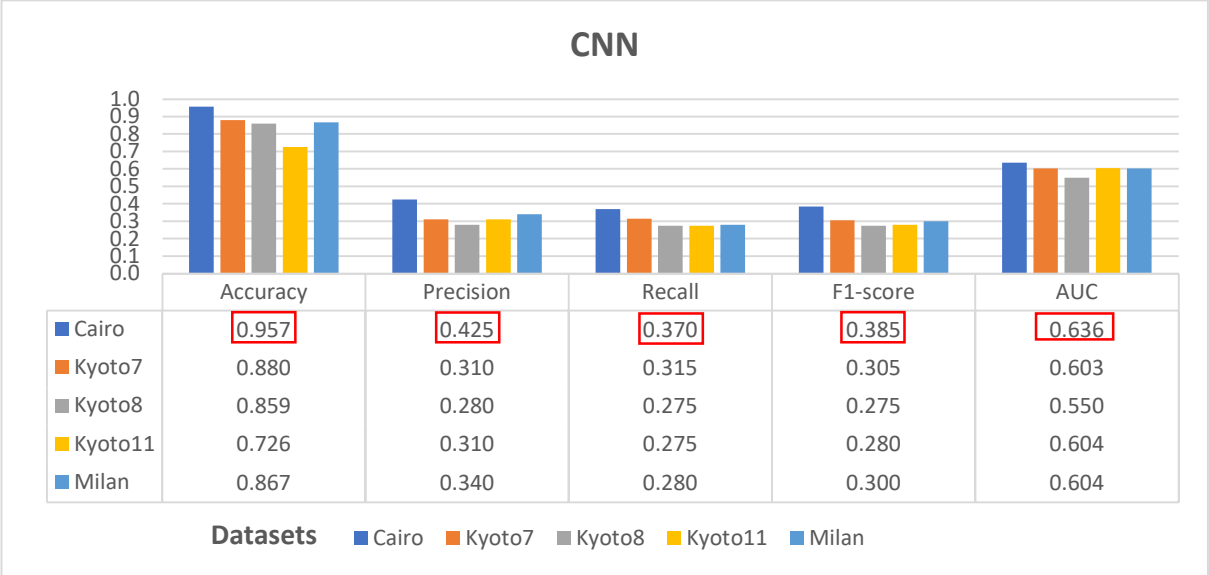
To ensure a fair evaluation, the nine deep learning models (six baseline models and three improved baseline models) were tested under the same experimental settings. First, the experimental dataset was preprocessed by the method introduced in Section 3.4. The stability and correctness of the evaluation was checked by a 5-fold cross-validation program. The evaluation result was the average of all folds. In this thesis, the experimental parameters were decided as follows: seeds = 7, units = 64, epochs = 200. Twenty percent of the training data were reserved as the verification data in the experimental dataset.

4.2.1 Baseline models

In this section, we show and discuss the evaluation results and statistical results of the six baseline deep learning models on the five selected CASAS datasets. First, we introduce the evaluation metrics of each baseline deep learning model on five CASAS datasets. Second, we discuss the Friedman test results of the six baseline deep learning models.

4.2.1.1 Evaluation

This section, we show the evaluation results of the six baseline deep learning models on five CASAS datasets (Cairo, Kyoto7, Kyoto8, Kyoto11 and Milan).

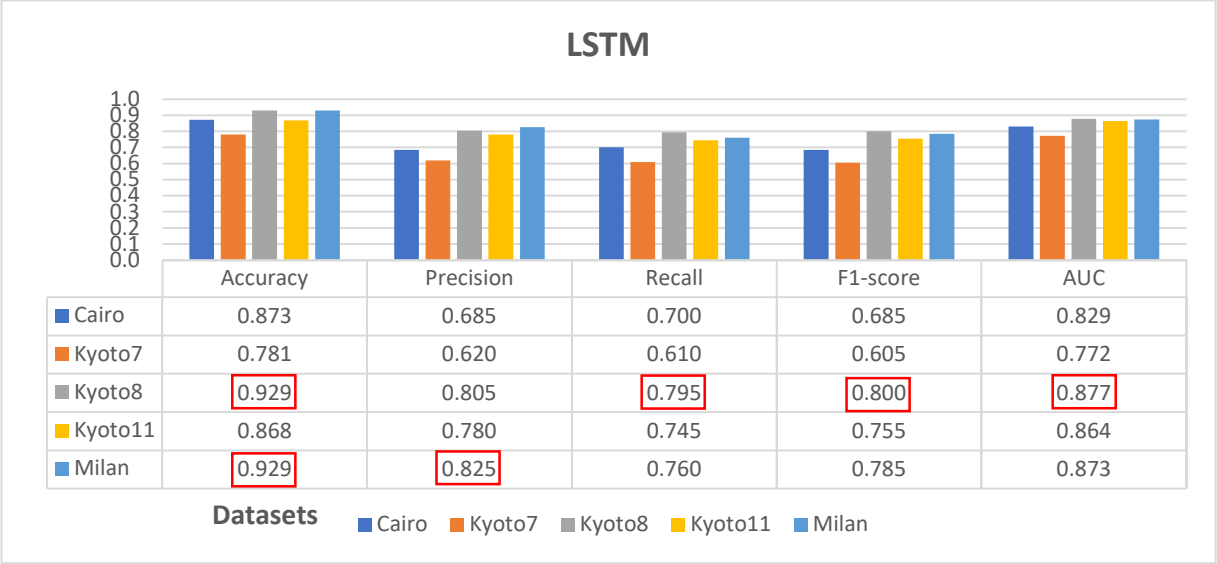


	Accuracy	Precision	Recall	F1-score	AUC
Cairo	0.957	0.425	0.370	0.385	0.636
Kyoto7	0.880	0.310	0.315	0.305	0.603
Kyoto8	0.859	0.280	0.275	0.275	0.550
Kyoto11	0.726	0.310	0.275	0.280	0.604
Milan	0.867	0.340	0.280	0.300	0.604

Figure 19. Evaluation results of CNN

Figure 19 shows the evaluation results of the CNN model on the five selected CASAS datasets (Cairo, Kyoto7, Kyoto8, Kyoto11 and Milan). The highest value of each evaluation measure is enclosed within a red rectangle. Based on the results, the following conclusions were drawn:

- CNN achieved the highest accuracy, precision, recall, F1-score and AUC on the Cairo dataset. The classification accuracy of CNN reached 0.957, the precision of CNN is 0.425, the recall of CNN is 0.370, the F1-score is 0.385 and the AUC of CNN achieve 0.636.
- CNN was less successful at classifying the Kyoto8 and Kyoto11 datasets. The recognition accuracy was lowest (0.726) on Kyoto11.
- The precision (0.280), recall (0.275), F1-score (0.275) and AUC (0.550) of CNN were lowest on the Kyoto8 dataset.
- CNN is the most suitable classifier of Cairo, and is unsuitable for classifying Kyoto8.

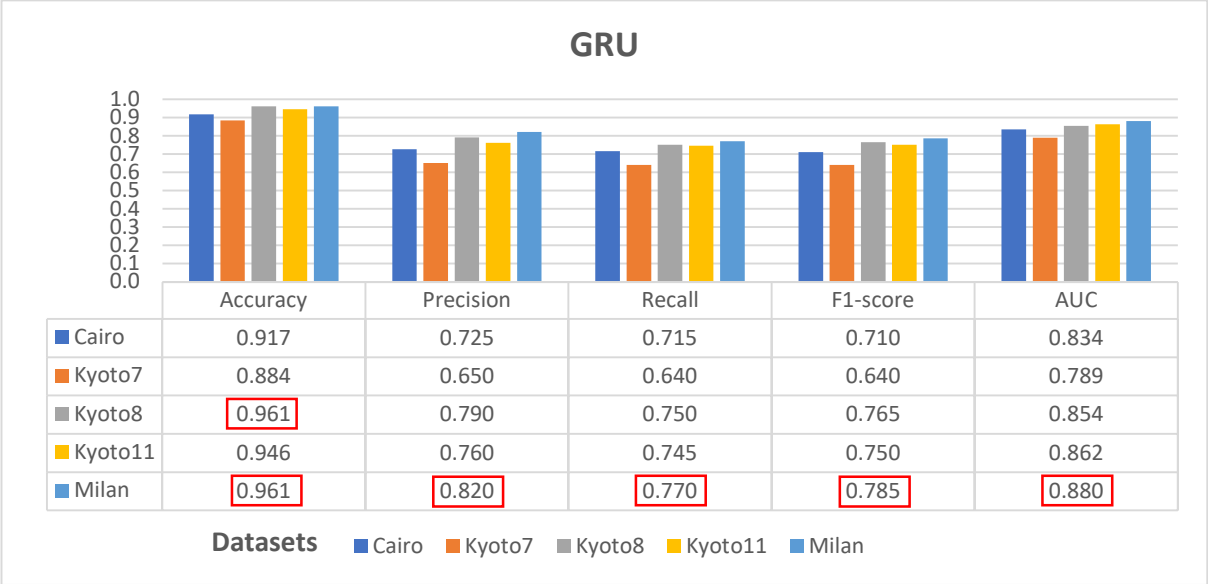


	Accuracy	Precision	Recall	F1-score	AUC
Cairo	0.873	0.685	0.700	0.685	0.829
Kyoto7	0.781	0.620	0.610	0.605	0.772
Kyoto8	0.929	0.805	0.795	0.800	0.877
Kyoto11	0.868	0.780	0.745	0.755	0.864
Milan	0.929	0.825	0.760	0.785	0.873

Figure 20. Evaluation result of LSTM

Figure 20 shows the performances of the LSTM classifiers on the five CASAS datasets. The highest value of each evaluation measure is enclosed within a red rectangle. It is observed that

- LSTM achieved the highest accuracy (0.929) on the Kyoto8 and Milan datasets. Although the precision (0.825) was also maximised on Milan, the recall , F1-score and AUC were lower on Milan than on Kyoto8. The recall (0.795), F1-score (0.800) and AUC (0.877) of LSTM were highest on the Kyoto8 dataset.
- The accuracy, precision, recall, F1-score and AUC of LSTM were lowest on the Kyoto7 dataset, the classification accuracy of LSTM is 0.781.
- The classification performance of LSTM was highest on the Kyoto8 dataset, the lowest performance of LSTM was classifying on Kyoto7.

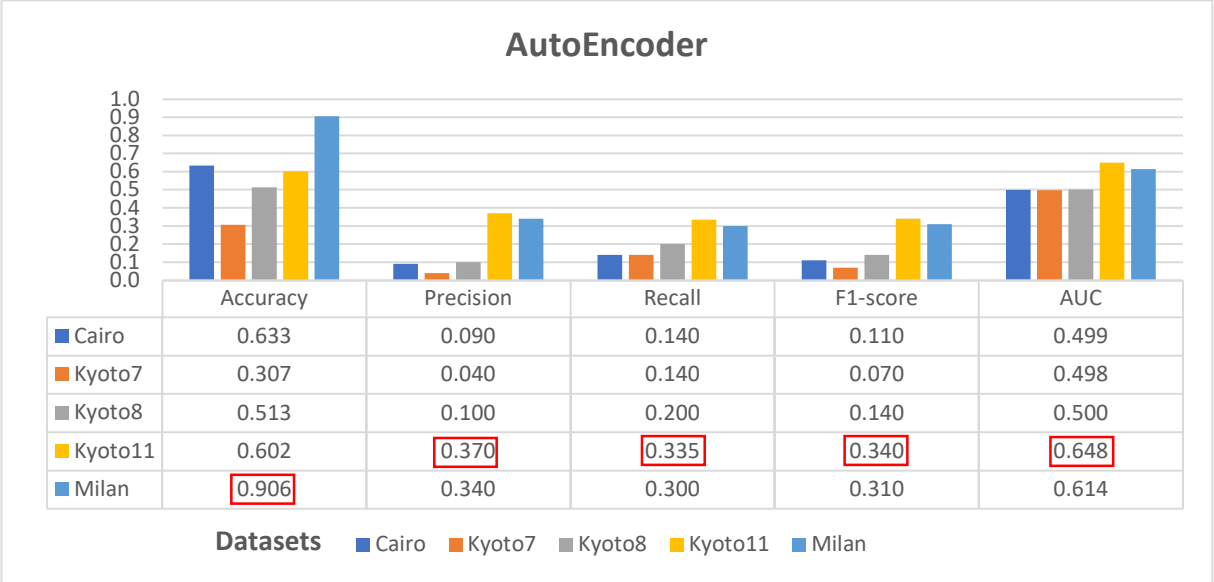


	Accuracy	Precision	Recall	F1-score	AUC
Cairo	0.917	0.725	0.715	0.710	0.834
Kyoto7	0.884	0.650	0.640	0.640	0.789
Kyoto8	0.961	0.790	0.750	0.765	0.854
Kyoto11	0.946	0.760	0.745	0.750	0.862
Milan	0.961	0.820	0.770	0.785	0.880

Figure 21. Evaluation results of GRU

Figure 21 shows the evaluation results of the GRU classifier on the five CASAS datasets. The highest value of each measure is enclosed within a red rectangle. From this figure, we observe that:

- GRU achieved the highest accuracy (0.961), precision (0.820), recall （0.770), F1-score (0.785) and AUC (0.880) on the Milan dataset.
- The accuracy on the Kyoto8 dataset was equal to that of Milan, but the precision, recall, F1-score and AUC were lower on Kyoto8 than on Milan.
- The classification performance of GRU was lowest on the Kyoto7 dataset. The accuracy, precision, recall, F1-score and AUC of GRU were lowest on the Kyoto7 dataset.
- The accuracy of GRU on five CASAS datasets was higher than 0.880. GRU performed better on the Milan dataset than on the other datasets.

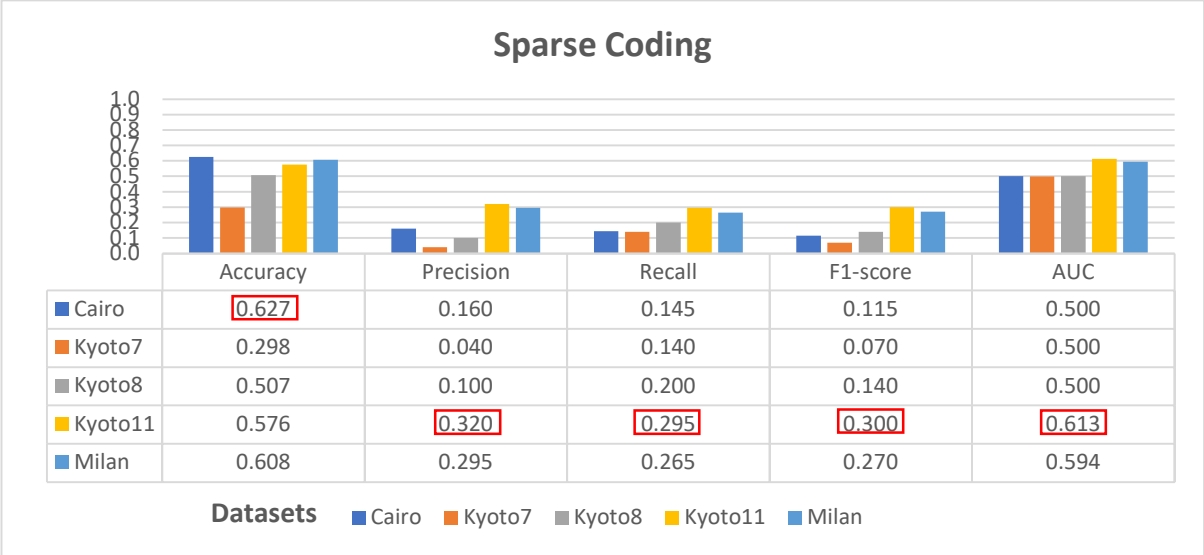


	Accuracy	Precision	Recall	F1-score	AUC
Cairo	0.633	0.090	0.140	0.110	0.499
Kyoto7	0.307	0.040	0.140	0.070	0.498
Kyoto8	0.513	0.100	0.200	0.140	0.500
Kyoto11	0.602	0.370	0.335	0.340	0.648
Milan	0.906	0.340	0.300	0.310	0.614

Figure 22. Evaluation results of Autoencoder

Figure 22 shows the evaluation results of Autoencoder on the five CASAS datasets. The highest value of each evaluation measure is enclosed by a red rectangle. From this figure, we conclude that:

- The recognition accuracy of Autoencoder was highest (0.906) on the Milan dataset.
- The accuracy of Autoencoder on the other datasets was lower than 0.65.
- The precision (0.370), recall (0.335), F1-score (0.340) and AUC (0.648) of Autoencoder were highest on Kyoto11. The accuracy, precision, recall, F1-score and AUC of GRU were lowest on the Kyoto7 dataset.
- The statistics indicate that Autoencoder accurately classifies the Kyoto 11 dataset, but performs comparatively poorly on Kyoto7.

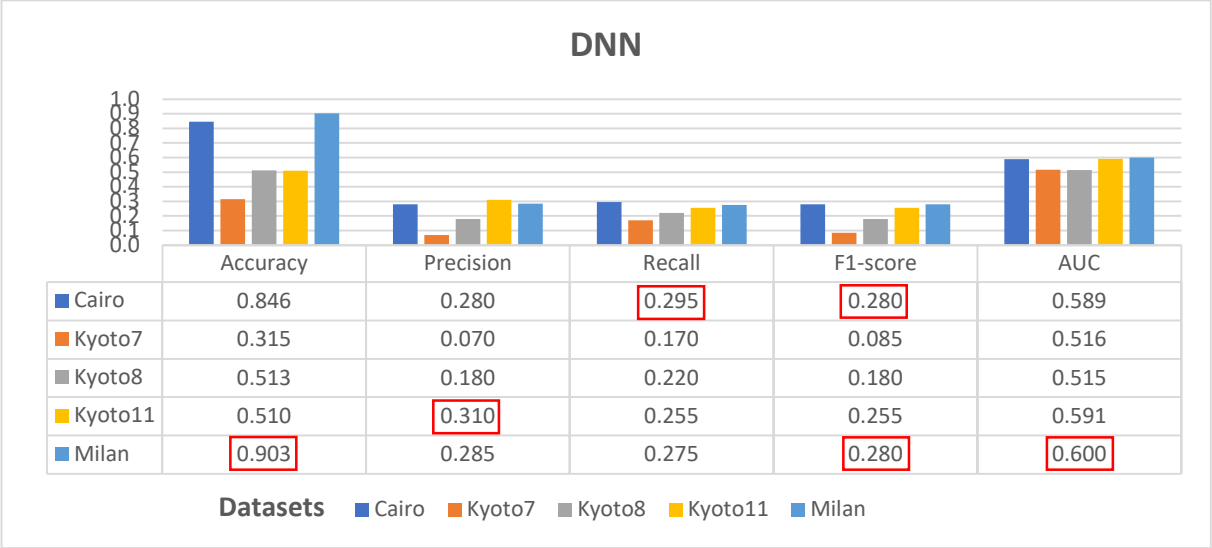


	Accuracy	Precision	Recall	F1-score	AUC
Cairo	0.627	0.160	0.145	0.115	0.500
Kyoto7	0.298	0.040	0.140	0.070	0.500
Kyoto8	0.507	0.100	0.200	0.140	0.500
Kyoto11	0.576	0.320	0.295	0.300	0.613
Milan	0.608	0.295	0.265	0.270	0.594

Figure 23. Evaluation results of Sparse Coding

Figure 23 shows the Sparse Coding evaluations on the five CASAS datasets. The highest value of each evaluation measure is enclosed within a red rectangle. This figure shows that:

- The recognition accuracy of Sparse Coding was highest (0.627) on the Cairo dataset, and lowest (0.298) on the Kyoto7 dataset.
- The precision (0.320), recall (0.295), F1-score (0.300) and AUC (0.613) were highest on the Kyoto11 dataset.
- On the Cairo, Kyoto7 and Kyoto8 datasets, the AUC was only 0.5. The accuracy, precision, recall and F1-score of sparse coding were lowest on Kyoto7.
- Sparse Coding performed better on the Kyoto11 dataset than on the other datasets.



	Accuracy	Precision	Recall	F1-score	AUC
Cairo	0.846	0.280	0.295	0.280	0.589
Kyoto7	0.315	0.070	0.170	0.085	0.516
Kyoto8	0.513	0.180	0.220	0.180	0.515
Kyoto11	0.510	0.310	0.255	0.255	0.591
Milan	0.903	0.285	0.275	0.280	0.600

Figure 24. Evaluation result of DNN

Figure 24 shows the DNN evaluation results on the five CASAS datasets. The highest value of each measure is enclosed within a red rectangle. We conclude that:

- DNN achieved the highest recall (0.295) and F1-score (0.280) on the Cairo dataset, the highest precision (0.310) on the Kyoto11 dataset, and the highest accuracy (0.903), AUC (0.280)and F1-score (0.600) on the Milan dataset (note that the F1-scores were identical on the Milan and Cairo datasets).
- The accuracy, precision, recall and F1-score of DNN were lowest on the Kyoto7 dataset.
- Overall, the classification performance of DNN was highest on the Milan dataset. DNN is unsuitable for classifying Kyoto7.

Table 4. Evaluation results of the six baseline deep-learning models on each dataset

Dataset	Model	Metric				
		Accuracy	Precision	Recall	F1-score	AUC
Cairo	CNN	0.957	0.425	0.370	0.385	0.636
	LSTM	0.873	0.685	0.700	0.685	0.829
	GRU	0.917	0.725	0.715	0.710	0.834
	Autoenco	0.633	0.090	0.140	0.110	0.499
	Sparsecod	0.627	0.160	0.145	0.115	0.500
	DNN	0.846	0.280	0.295	0.280	0.589
kyoto7	CNN	0.880	0.310	0.315	0.305	0.603
	LSTM	0.781	0.620	0.610	0.605	0.772
	GRU	0.884	0.650	0.640	0.640	0.789
	Autoenco	0.307	0.040	0.140	0.070	0.498
	Sparsecod	0.298	0.040	0.140	0.070	0.500
	DNN	0.315	0.070	0.170	0.085	0.516
kyoto8	CNN	0.859	0.280	0.275	0.275	0.550
	LSTM	0.929	0.805	0.795	0.800	0.877
	GRU	0.961	0.790	0.750	0.765	0.854
	Autoenco	0.513	0.100	0.200	0.140	0.500
	Sparsecod	0.507	0.100	0.200	0.140	0.500
	DNN	0.513	0.180	0.220	0.180	0.515
kyoto11	CNN	0.726	0.310	0.275	0.280	0.604
	LSTM	0.868	0.780	0.745	0.755	0.864
	GRU	0.946	0.760	0.745	0.750	0.862
	Autoenco	0.602	0.370	0.335	0.340	0.648
	Sparsecod	0.576	0.320	0.295	0.300	0.613
	DNN	0.510	0.310	0.255	0.255	0.591
milan	CNN	0.867	0.340	0.280	0.300	0.604
	LSTM	0.929	0.825	0.760	0.785	0.873
	GRU	0.961	0.820	0.770	0.785	0.880
	Autoenco	0.906	0.340	0.300	0.310	0.614
	Sparsecod	0.608	0.295	0.265	0.270	0.594
	DNN	0.903	0.285	0.275	0.280	0.600

Table 4 summarises the evaluation results of the six models on the five CASAS datasets. The highest evaluation results on each dataset are highlighted in red font. We conclude that:

- On the Cairo dataset, the highest recognition accuracy (0.957) was achieved by CNN, but the highest precisions (0.725), recall (0.715), F1-score (0.710) and AUC (0.834) were achieved by GRU. The top three performers on the Cairo

datasets were GRU, LSTM and CNN. Autoencoder has worst performance on the Cairo dataset.

- On the Kyoto7 dataset, the highest values of all five metrics were achieved by GRU which, so the GRU is obviously the best classifier of this dataset. The top three of the six models on Kyoto7 were GRU, LSTM and CNN. Autoencoder and Sparse coding are unsuitable for classifying Kyoto7.
- On the Kyoto 8 dataset, the highest recognition accuracy (0.961) was achieved by GRU, but the precision (0.805), recall (0.795), F1-score (0.800) and AUC (0.877) were maximised by the LSTM classifier. LSTM also achieved the second highest accuracy (0.929) on Kyoto8. We conclude that LSTM is the best classifier of the Kyoto8 dataset. The top performers on the Kyoto8 datasets were GRU, LSTM and CNN. Sparse coding has worst performance on the Kyoto8 dataset.
- On the Kyoto11 dataset, the top performers were the LSTM and GRU classifiers. GRU achieved the highest accuracy (0.946) and the highest recall (0.745) among the classifiers, while LSTM maximised the precision (0.780), recall (0.745), F1-score (0.755) and AUC (0.864). The top three performers on the Kyoto11 dataset were GRU, LSTM and CNN. DNN has worst performance on the Kyoto11 dataset.
- On the Milan dataset, the highest accuracy (0.961), recall (0.770), F1-score (0.785) and AUC (0.880) were achieved by GRU. Meanwhile, the LSTM maximised the precision (0.825) and F1-score (0.785). In fact, the LSTM and GRU evaluation results were quite similar on this dataset. The CNN and DNN evaluation results were also comparatively similar. The top four performers on the Milan dataset were GRU, LSTM, CNN and DNN. Sparse coding has worst performance on the Milan dataset.
- Based on the above analyses of the CASAS datasets, the top three models among the six baseline deep learning models were identified as GRU, LSTM and CNN,

consistent with our assumptions in Section 3.5.2. The remaining three models are unsuitable for classifying CASAS datasets.

4.2.1.2 Statistical analysis and Comparison

Table 5 displays the Friedman test results of the six baseline deep-learning models classifier on the five CASAS datasets. The table of Friedman test consists of two parts. The mean ranks (upper part of Table 5) indicate the actual differences between the models. The test statistics (lower part of Table 5) indicate whether the mean ranks of the different evaluation methods are significantly different. The highest mean rank of each evaluation measure is highlighted in red font. From Table 5, we drew the following conclusions:

Table 5. Friedman test results of the six baseline deep-learning models

Friedman Test					
Models	**Mean Rank**				
	Accuracy	**Precision**	**Recall**	**F1-score**	**AUC**
CNN	4.2	3.4	3.4	3.4	3.4
LSTM	4.6	**5.6**	5.3	**5.5**	5.4
GRU	**5.8**	5.4	**5.7**	**5.5**	**5.6**
AutoEncoder	2.7	2.3	2.4	2.4	2.3
SparseCoding	1.2	2	1.8	1.8	1.9
DNN	2.5	2.3	2.4	2.4	2.4
Test Statistics[a]					
N	5	5	5	5	5
Chi-Square	20	19.23977	19.47674	19.36047	19.02299
df	5	5	5	5	5
Asymp. Sig.	0.00125	0.001734	0.001566	0.001646	0.001903

- The accuracy statistics yielded $\chi^2 = 20$ and p (Asymp.Sig) = 0.00125 (< 0.05), implying that the mean-rank accuracies of the six baseline deep learning models were statistically different. The highest mean rank of the accuracy (5.8) was obtained by the GRU and the lowest mean rank of the accuracy (1.2) was Sparse Coding.
- The precision statistics yielded $\chi^2 = 19.239$ and p (Asymp.Sig) = 0.00173 (< 0.05), implying that the mean-rank precisions of the six baseline deep learning models

were statistically different. The highest mean rank of the precision (5.6) was obtained by LSTM. The lowest mean rank of the precision (2) was Sparse Coding.

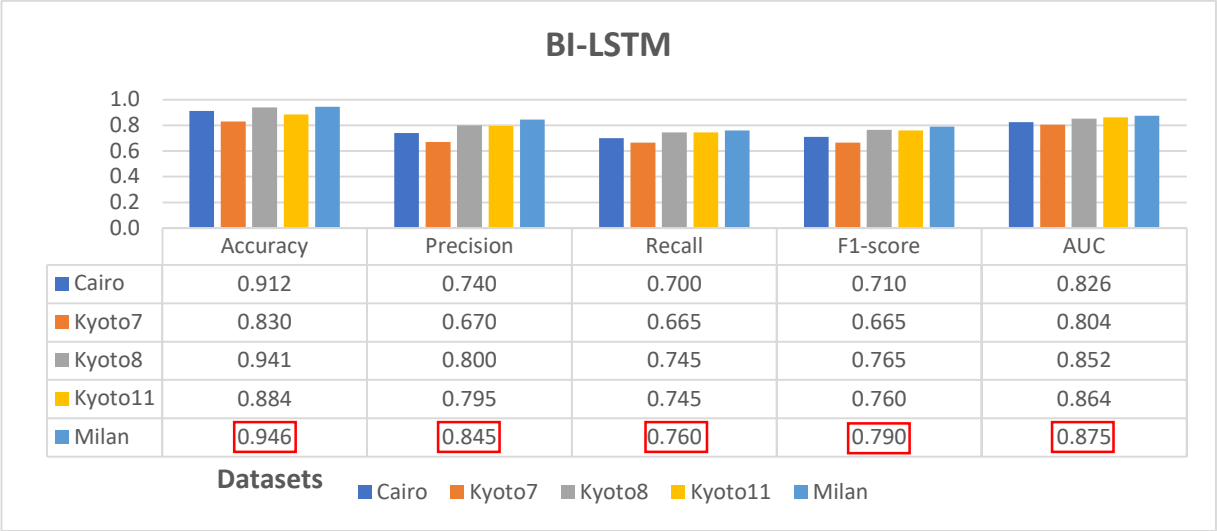
- The recall statistics yielded χ^2 =19.477 and p (Asymp.Sig) = 0.00157 (< 0.05), confirming significant differences among the mean-rank recalls of the six baseline deep learning models. The highest mean rank of the recall (5.7) was obtained by GRU. The lowest mean rank of the recall (1.8) was Sparse Coding.
- The F1-score statistics yielded χ^2 =19.360 and p (Asymp.Sig) = 0.00165 (< 0.05), implying significant differences among the mean-rank F-scores of the six deep learning models. The highest mean rank of the F1-score (5.5) was jointly achieved by LSTM and GRU. The lowest mean rank of the F1-score (1.8) was Sparse Coding.
- The AUC statistics yielded χ^2 =19.023 and p (Asymp.Sig) = 0.00190 (< 0.05), implying that the mean rank AUCs of the six models were significantly different. The highest mean tank of the AUC (5.6) was obtained by GRU. The lowest mean rank of the AUC (1.9) was Sparse Coding.
- The GRU model achieved the highest mean ranks for accuracy, recall, F1-score and AUC, whereas the LSTM model achieved the highest mean ranks for precision and F1-score. Overall, GRU outperformed LSTM in terms of mean rank. The Sparse Coding model obtained the worst mean-rank values among the baseline deep learning models.

4.2.2 Improved baseline models

In this section, we show and discuss the evaluation result and statistical result of the three improved baseline deep learning models on the five selected CASAS datasets. We first introduce the evaluation metric of each improved baseline deep learning model on five CASAS datasets. Next, we discuss the Friedman test result of the three improved baseline deep learning models.

4.2.2.1 Evaluation

This section, we show the evaluation results of each improved baseline deep learning model (BI-LSTM, BI-GRU, LSTM-CNN) on the five CASAS datasets (Cairo, Kyoto7, Kyoto8, Kyoto11 and Milan).

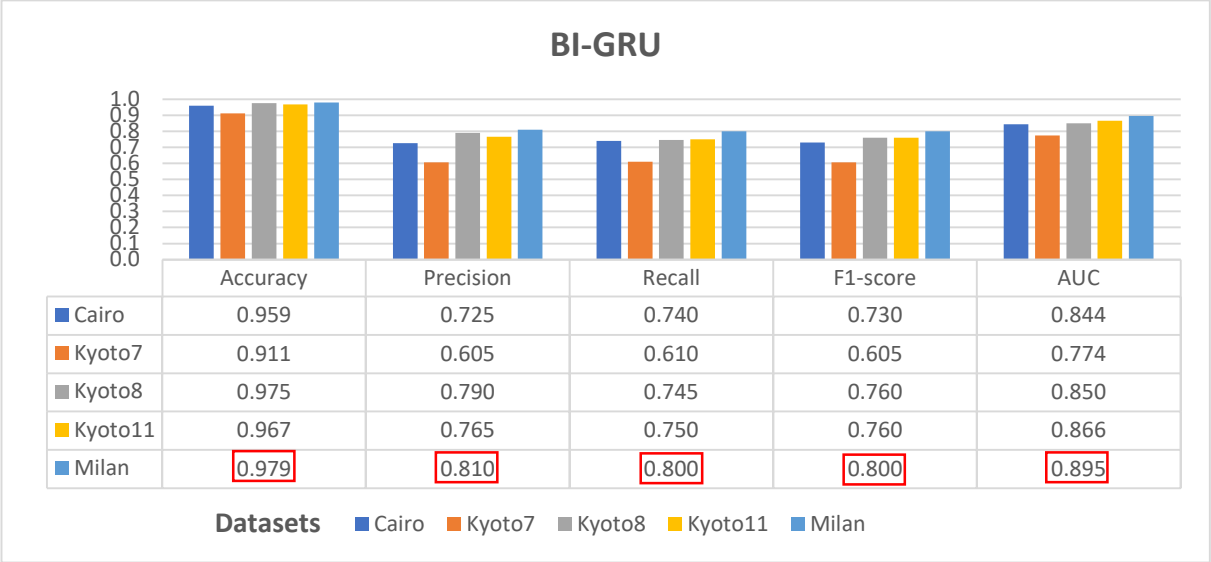


	Accuracy	Precision	Recall	F1-score	AUC
Cairo	0.912	0.740	0.700	0.710	0.826
Kyoto7	0.830	0.670	0.665	0.665	0.804
Kyoto8	0.941	0.800	0.745	0.765	0.852
Kyoto11	0.884	0.795	0.745	0.760	0.864
Milan	0.946	0.845	0.760	0.790	0.875

Figure 25. Evaluation results of BI-LSTM

Figure 25 presents the evaluation results of the BI-LSTM classifications on the five CASAS datasets (Cairo, Kyoto7, Kyoto8, Kyoto11 and Milan). The highest evaluations are enclosed in red rectangles. From Figure 25, we conclude the following:

- The accuracy, precision, recall, F1-score and AUC of BI-LSTM were highest on the Milan dataset. The accuracy and AUC of BI-LSTM on this dataset reached 0.946 and 0.875, respectively.
- The accuracy of BI-LSTM was lowest (0.830) on the Kyoto7 dataset.
- The accuracy, precision, recall, F1-score and AUC of BI-LSTM were lowest on the Kyoto7 dataset. BI-LSTM was a good classifier of the Milan dataset. The lowest performance of BI-LSTM was classifying on Kyoto7.

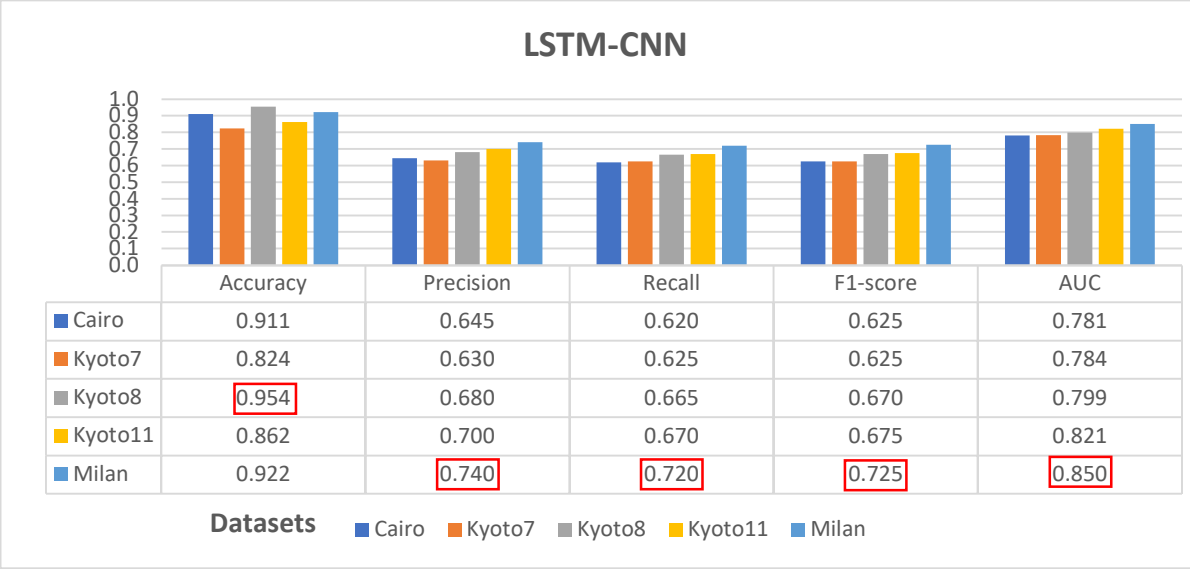


	Accuracy	Precision	Recall	F1-score	AUC
Cairo	0.959	0.725	0.740	0.730	0.844
Kyoto7	0.911	0.605	0.610	0.605	0.774
Kyoto8	0.975	0.790	0.745	0.760	0.850
Kyoto11	0.967	0.765	0.750	0.760	0.866
Milan	0.979	0.810	0.800	0.800	0.895

Figure 26. Evaluation results of BI-GRU

Figure 26 shows the evaluation results of BI-GRU on the five CASAS datasets. The highest evaluation results are enclosed in red rectangles. It can be concluded that

- BI-GRU achieved its highest accuracy, precision, recall, F1-score, and AUC on the Milan dataset. The accuracy of BI-GRU reached 0.979 on this dataset,the precision of BI-GRU is 0.810, the recall of BI-GRU is 0.800, the F1-score is 0.800 and the AUC of BI-GRU achieve 0.895.
- The accuracy of BI-GRU exceeded 0.9 on all five datasets. The lowest accuracy was 0.911 on the Kyoto7 dataset.
- The accuracy, precision, recall, F1-score and AUC of BI-GRU were lowest on the Kyoto7 dataset. The BI-GRU classification was most successful on the Milan dataset. The lowest performance of BI-GRU was classifying on Kyoto7.



	Accuracy	Precision	Recall	F1-score	AUC
Cairo	0.911	0.645	0.620	0.625	0.781
Kyoto7	0.824	0.630	0.625	0.625	0.784
Kyoto8	0.954	0.680	0.665	0.670	0.799
Kyoto11	0.862	0.700	0.670	0.675	0.821
Milan	0.922	0.740	0.720	0.725	0.850

Figure 27. Evaluation results of LSTM-CNN

Figure 27 shows the evaluation results of LSTM-CNN on the five CASAS datasets. The highest evaluation results are enclosed in red rectangles. From this figure, we draw the following conclusions:

- The accuracy of LSTM-CNN was highest (0.954) on the Kyoto8 dataset. However, the precision (0.740), recall (0.720), F-score (0.725) and AUC (0.850) were highest on the Milan dataset.
- The lowest accuracy, precision, F1-score and AUC were on the Kyoto7 dataset, and the lowest recall was on the Cairo dataset.
- The classification performance of LSTM-CNN was highest on the Milan dataset, the lowest performance of LSTM-CNN was classifying on Kyoto7.

Table 6. Evaluation results of the three improved baseline deep-learning models on each dataset

Dataset	Model	Metric				
		Accuracy	Precision	Recall	F1-score	AUC
Cairo	Bi-LSTM	0.912	0.740	0.700	0.710	0.826
	Bi-GRU	0.959	0.725	0.740	0.730	0.844
	LSTM+CNN	0.911	0.645	0.620	0.625	0.781
Kyoto7	Bi-LSTM	0.830	0.670	0.665	0.665	0.804
	Bi-GRU	0.911	0.605	0.610	0.605	0.774
	LSTM+CNN	0.824	0.630	0.625	0.625	0.784
Kyoto8	Bi-LSTM	0.941	0.800	0.745	0.765	0.852
	Bi-GRU	0.975	0.790	0.745	0.760	0.850
	LSTM+CNN	0.954	0.680	0.665	0.670	0.799
Kyoto11	Bi-LSTM	0.884	0.795	0.745	0.760	0.864
	Bi-GRU	0.967	0.765	0.750	0.760	0.866
	LSTM+CNN	0.862	0.700	0.670	0.675	0.821
Milan	Bi-LSTM	0.946	0.845	0.760	0.790	0.875
	Bi-GRU	0.979	0.810	0.800	0.800	0.895
	LSTM+CNN	0.922	0.740	0.720	0.725	0.850

Table 6 presents the evaluation results of the three improved models on the five CASAS datasets (Cairo, Kyoto7, Kyoto8, Kyoto11 and Milan). The highest evaluation values of each dataset are highlighted in red font. The results of the individual datasets are described below.

- On the Cairo dataset, the BI-GRU achieved the highest accuracy (0.959), recall (0.740), F1-score (0.730) and AUC (0.844), and BI-LSTM achieved the highest precision which is 0.740. Although LSTM-CNN was the worst performer on this dataset, its recognition accuracy still exceeded 0.9. Among the three improved models, BI-GRU has best performance classifier on the Cairo dataset.
- On the Kyoto7 dataset, BI-GRU achieved the highest accuracy (0.911), but BI-LSTM exhibited the highest precision (0.670), recall (0.665), F1-score (0.665) and AUC (0.804). On this dataset, Bi-LSTM and LSTM+CNN were the most and least effective classifiers, respectively.
- On the Kyoto8 dataset, BI-LSTM and BI-GRU outperformed LSTM+CNN. BI-GRU achieved the highest accuracy (0.975), and BI-LSTM yielded the highest precision (0.800), F-score (0.765) and AUC (0.852). The highest recall score

(0.745) was shared by BI-LSTM and BI-GRU. In general, BI-LSTM and LSTM+CNN were the most and least suitable classifiers of Kyoto8, respectively.

- On the Kyoto11 dataset, BI-LSTM achieved the highest precision (0.795), however BI-GRU yielded the highest accuracy (0.967), recall (0.750) and AUC (0.866). BI-LSTM and BI-GRU conjointly achieved the highest F1-score which is 0.760. Overall, Bi-GRU was the most suitable classifiers of the Kyoto11 dataset and LSTM+CNN was least suitable classifiers of the Kyoto11 dataset.
- On the Milan dataset, Bi-LSTM achieved the highest precision (0.845), however BI-GRU yielded the highest accuracy (0.979), recall (0.800), F-score (0.800) and AUC (0.895). Overall, BI-GRU and LSTM+CNN were the best and worst classifiers of the Milan dataset, respectively.
- Comparing the performances of the improved models on the various datasets, the BI-GRU achieved the highest evaluation score on the Milan dataset, meaning that BI-GRU is the best classifier of Milan. BI-GRU and BI-LSTM provided good performance on all five CASAS datasets, whereas LSTM + CNN performed comparatively poorly.

4.2.2.2 Statistical analysis and Comparison

Table 7 presents the Friedman test results of the three improved baseline deep learning models. Table 7 is divided into two parts: the different mean ranks of the related deep-learning models (upper part), and the test statistics indicating whether the differences were statistically significant (lower part). The highest mean rank in each evaluation is highlighted in red font. From Table 7, we observe that:

Table 7. Friedman test results of the three improved baseline deep-learning models

Friedman Test					
Models	Mean Rank				
	Accuarcy	Precision	Recall	F1-score	AUC
BI-LSTM	1.8	3	2.3	1.2	2.4
BI-GRU	3	1.8	2.5	2.5	2.4
LSTM-GRU	1.2	1.2	1.2	2.3	1.2
Test Statistics[a]					
N	5	5	5	5	5
Chi-Square	8.4	8.4	5.15789474	5.157894737	4.8
df	2	2	2	2	2
Asymp. Sig.	0.014996	0.014996	0.075854	0.075854	0.090718

- The accuracy statistics yielded χ^2 = 8.4 and p (Asymp.Sig) = 0.0149 (< 0.05), implying that the mean-rank accuracies of the three improved models were significantly different. The highest mean rank of the accuracy (3) was obtained by the BI-GRU and the lowest mean rank of the accuracy (1.2) was LSTM+CNN.
- The precision statistics yielded χ^2 = 8.4 and p (Asymp.sig) = 0.0149 (< 0.05), indicating that the mean-rank precisions of the three improved models were significantly different. The highest mean rank of the precision (3) was obtained by the BI-LSTM and the lowest mean rank of the precision (1.2) was LSTM+CNN.
- The recall statistics yielded χ^2 = 5.158 and p (Asymp.sig) = 0.0759 (> 0.05), indicating no significant differences among the mean-rank recalls of the three improved models. The highest mean rank of the recall (2.5) was obtained by the BI-GRU and the lowest mean rank of the recall (1.2) was LSTM+CNN.
- The F-score statistics yielded χ^2 = 5.158 and p (Asymp.sig) = 0.0759 (> 0.05), indicating no significant differences among the mean-rank F-scores of the three improved models. The highest mean rank of the F-score (2.5) was obtained by the BI-GRU and the lowest mean rank of the recall (1.2) was BI-LSTM.
- The AUC statistics yielded χ^2 = 4.8 and p (Asymp.Sig) = 0.0907 (> 0.05), indicating no significant differences among the mean-rank AUCs of the three

improved models. The highest mean rank of the AUC (2.4) were obtained by the BI-LSTM and BI-GRU and the lowest mean rank of the AUC (1.2) was LSTM-CNN.

- Comparing the Friedman test result of three improved baseline deep learning models, the BI-LSTM model achieved the highest mean ranks of precision and AUC, but BI-GRU obtained the highest mean ranks of accuracy, recall, F-score and AUC. Overall, BI-GRU outperformed BI-LSMT in terms of mean rank.
- LSTM-CNN obtained the lowest mean ranks among the three improved models.

4.2.3 Comprehensive comparison

In this section, we show and discuss the evaluation result and statistical result of the nine baseline deep learning models on the five selected CASAS datasets. First, we show training times of each baseline deep learning model classifying five CASAS datasets. Second, we discuss the statistical analysis and comparison of the evaluation result of the nine baseline deep learning models.

4.2.3.1 Training time

Table 8 compares the training times of the nine deep learning methods on the five CASAS datasets. The training time is the total time of training the deep learning models on all five datasets. All experiments were run on an Intel Core i7-4710 CPU 2.50 GHz with 16 GB of RAM, and were assessed by 5-fold cross-validation. Therefore, the data in Table 8 are the summed training times of two experiments which are improved models (upper part) and baseline models (lower part). We observe that:

Table 8. Training times of the deep learning models

Models Types	Deep Learning models	Times
Improved baseline Models	BI-LSTM	150 hours
	BI-GRU	126 hours
	LSTM+CNN	29 hours
Baseline Models	LSTM	25 hours
	GRU	23 hours
	CNN	52 mins
	Sparse Coding	15 mins
	Autoencoder	5 mins
	DNN	5 mins

- Compare the training time of Baseline models, the LSTM required the longest training time (25 hours). The BI-LSTM required the longest training time (150 hours) in nine deep learning models, whereas Autoencoder and DNN required only a very short training time (five minutes).
- The improved baseline models required a longer training time than the baseline models, therefore, we conclude that the training time increased with increasing complexity of the model.

4.2.3.2 Statistical analysis and Comparison of the evaluation results

This section, we first show summary of improved result (Table 9) for three baseline deep learning models and three improved baseline deep learning models. Then we show the summary of improved result (Table 10), evaluated by the Friedman test. Next, we compare the classification evaluation results (Table 11) of the nine deep learning models on the five CASAS datasets. Finally, shows the Friedman test results (Table 12) of the nine deep learning models.

Table 9. Summary of the improved results

Dataset	Model		Metric				
			Accuracy	Precision	Recall	F1-score	AUC
Cairo	Base	CNN	0.957	0.425	0.370	0.385	0.636
	Improved	LSTM+CNN	0.911	0.645	0.620	0.625	0.781
	Base	LSTM	0.873	0.685	0.700	0.685	0.829
	Improved	Bi-LSTM	0.912	0.740	0.700	0.710	0.826
	Base	GRU	0.917	0.725	0.715	0.710	0.834
	Improved	Bi-GRU	0.959	0.725	0.740	0.730	0.844
Kyoto7	Base	CNN	0.880	0.310	0.315	0.305	0.603
	Improved	LSTM+CNN	0.824	0.630	0.625	0.625	0.784
	Base	LSTM	0.781	0.620	0.610	0.605	0.772
	Improved	Bi-LSTM	0.830	0.670	0.665	0.665	0.804
	Base	GRU	0.884	0.650	0.640	0.640	0.789
	Improved	Bi-GRU	0.911	0.605	0.610	0.605	0.774
Kyoto8	Base	CNN	0.859	0.280	0.275	0.275	0.550
	Improved	LSTM+CNN	0.954	0.680	0.665	0.670	0.799
	Base	LSTM	0.929	0.805	0.795	0.800	0.877
	Improved	Bi-LSTM	0.941	0.800	0.745	0.765	0.852
	Base	GRU	0.961	0.790	0.750	0.765	0.854
	Improved	Bi-GRU	0.975	0.790	0.745	0.760	0.850
Kyoto11	Base	CNN	0.726	0.310	0.275	0.280	0.604
	Improved	LSTM+CNN	0.862	0.700	0.670	0.675	0.821
	Base	LSTM	0.868	0.780	0.745	0.755	0.864
	Improved	Bi-LSTM	0.884	0.795	0.745	0.760	0.864
	Base	GRU	0.946	0.760	0.745	0.750	0.862
	Improved	Bi-GRU	0.967	0.765	0.750	0.760	0.866
Milan	Base	CNN	0.867	0.340	0.280	0.300	0.604
	Improved	LSTM+CNN	0.922	0.740	0.720	0.725	0.850
	Base	LSTM	0.929	0.825	0.760	0.785	0.873
	Improved	Bi-LSTM	0.946	0.845	0.760	0.790	0.875
	Base	GRU	0.961	0.820	0.770	0.785	0.880
	Improved	Bi-GRU	0.979	0.810	0.800	0.800	0.895

As shown in Table 9, we compared the evaluation results of three baseline models (CNN, LSTM and GRU) and three improved baseline models (LSTM+CNN, BI-LSTM and BI- GRU). The best measures on each dataset are highlighted in red font. We first compared the metrics of the baseline models and their improved versions on each dataset. Second, we compared the overall metrics of each model, and highlighted the model with the better overall evaluation result. The findings are summarised below.

- On the Cairo dataset, the performances of LSTM+CNN, BI-LSTM and BI-GRU were generally improved over their base models, but the accuracy of CNN decreased from 0.957 to 0.911 in LSTM+CNN. However, the performances of

LSTM +CNN were dropped compare with LSTM, although the accuracy of the LSTM+CNN was improved from 0.873 to 0.911 in LSTM.

- On the Kyoto7 dataset, the performances of CNN and LSTM rose in their respective improved models. However, the performance of the GRU model dropped in the improved model, although the accuracy increased from 0.884 to 0.911. The precision, recall, F1-score and AUC all decreased in the improved GRU model. The performances of LSTM+CNN were improved compare with LSTM, all five metrics are increased.
- On the Kyoto8 dataset, only the performance of CNN increased in the improved model. Although the accuracies of the LSTM and GRU increased in their respective improved versions, the precision, recall, F1-score and AUC of these base models dropped or remained unchanged after improvement. The performances of LSTM +CNN were dropped compare with LSTM, although the accuracy of the LSTM+CNN was improved from 0.929 to 0.954 in LSTM.
- On the Kyoto11 dataset, the performances of all baseline models were increased in their improved models, although the recall and AUC of the LSTM model were not improved in BI-LSTM. The performances of LSTM +CNN were dropped compare with LSTM, all five metrics are decreased.
- On the Milan dataset, the performances of all baseline models were increased in their improved versions, but the precision of GRU dropped from 0.820 to 0.810 in BI-GRU, and the recall of LSTM was not improved in BI-LSTM. The performances of LSTM +CNN were dropped compare with LSTM, all five metrics are decreased.
- Overall, the performances of the improved models (LSTM+ CNN, Bi-LSTM and Bi-GRU) on the Cairo, Kyoto11 and Milan datasets were significantly improved after improvement. However, the performances of BI-GRU and BI-LSTM on the Kyoto7 and Kyoto8 datasets were reduced from those of the GRU and LSTM

baseline models, respectively. In addition, the performances of LSTM +CNN were rose compare with LSTM on the Kyoto7.

Table 10. Summary of the improved results, evaluated by the Friedman test

Friedman Test					
Models	Mean Ranks				
	Accuracy	Precision	Recall	F1-score	AUC
CNN	5	3.4	3.4	3.4	3.4
LSTM-CNN	5.4	5.4	5.4	5.4	5.4
LSTM	5	7.4	6.9	6.8	6.9
BI-LSTM	6.4	8.8	7.1	8.1	7.3
GRU	7.8	7	7.8	7.1	7.6
BI-GRU	9	6.4	7.8	7.6	7.8

As shown in Table 10, we compared the improved results evaluated by the Friedman test of three baseline models (CNN, LSTM and GRU) and three improved baseline models (LSTM+CNN, Bi-LSTM and Bi- GRU). The best measures on each dataset are highlighted in red font. The results are summarised below.

- The performances of LSTM+CNN are generally improved compare with CNN. However, the performances of LSTM +CNN were dropped compare with LSTM, although the accuracy of the LSTM+CNN was improved in LSTM.
- The performances of BI-LSTM are significantly improved compare with LSTM, the mean ranks of all five metrics are increased.
- The performances of BI-GRU are generally improved over GRU, but the precision of GRU dropped from 7 to 6.4 in BI-GRU, and the recall of GRU was not improved in BI-GRU.

Table 11. Evaluation results of the nine deep learning models

Dataset	Model	Metric				
		Accuracy	Precision	Recall	F1-score	AUC
Cairo	CNN	0.957	0.425	0.370	0.385	0.636
	LSTM	0.873	0.685	0.700	0.685	0.829
	GRU	0.917	0.725	0.715	0.710	0.834
	Autoencoder	0.633	0.090	0.140	0.110	0.499
	Sparsecoding	0.627	0.160	0.145	0.115	0.500
	DNN	0.846	0.280	0.295	0.280	0.589
	Bi-LSTM	0.912	0.740	0.700	0.710	0.826
	Bi-GRU	0.959	0.725	0.740	0.730	0.844
	LSTM+CNN	0.911	0.645	0.620	0.625	0.781
Kyoto7	CNN	0.880	0.310	0.315	0.305	0.603
	LSTM	0.781	0.620	0.610	0.605	0.772
	GRU	0.884	0.650	0.640	0.640	0.789
	Autoencoder	0.307	0.040	0.140	0.070	0.498
	Sparsecoding	0.298	0.040	0.140	0.070	0.500
	DNN	0.315	0.070	0.170	0.085	0.516
	Bi-LSTM	0.830	0.670	0.665	0.665	0.804
	Bi-GRU	0.911	0.605	0.610	0.605	0.774
	LSTM+CNN	0.824	0.630	0.625	0.625	0.784
Kyoto8	CNN	0.859	0.280	0.275	0.275	0.550
	LSTM	0.929	0.805	0.795	0.800	0.877
	GRU	0.961	0.790	0.750	0.765	0.854
	Autoencoder	0.513	0.100	0.200	0.140	0.500
	Sparsecoding	0.507	0.100	0.200	0.140	0.500
	DNN	0.513	0.180	0.220	0.180	0.515
	Bi-LSTM	0.941	0.800	0.745	0.765	0.852
	Bi-GRU	0.975	0.790	0.745	0.760	0.850
	LSTM+CNN	0.954	0.680	0.665	0.670	0.799
Kyoto11	CNN	0.726	0.310	0.275	0.280	0.604
	LSTM	0.868	0.780	0.745	0.755	0.864
	GRU	0.946	0.760	0.745	0.750	0.862
	Autoencoder	0.602	0.370	0.335	0.340	0.648
	Sparsecoding	0.576	0.320	0.295	0.300	0.613
	DNN	0.510	0.310	0.255	0.255	0.591
	Bi-LSTM	0.884	0.795	0.745	0.760	0.864
	Bi-GRU	0.967	0.765	0.750	0.760	0.866
	LSTM+CNN	0.862	0.700	0.670	0.675	0.821
Milan	CNN	0.867	0.340	0.280	0.300	0.604
	LSTM	0.929	0.825	0.760	0.785	0.873
	GRU	0.961	0.820	0.770	0.785	0.880
	Autoencoder	0.906	0.340	0.300	0.310	0.614
	Sparsecoding	0.608	0.295	0.265	0.270	0.594
	DNN	0.903	0.285	0.275	0.280	0.600
	Bi-LSTM	0.946	0.845	0.760	0.790	0.875
	Bi-GRU	0.979	0.810	0.800	0.800	0.895
	LSTM+CNN	0.922	0.740	0.720	0.725	0.850

Table 11 compares the classification evaluation results of the nine deep learning models on the five CASAS datasets (Cairo, Kyoto7, Kyoto8, Kyoto11 and Milan). The highest evaluation values of each dataset are highlighted in red font. The results are summarised below.

- On the Cairo dataset, BI-LSTM achieved the highest precision (74.00%), but BI-GRU obtained the highest Accuracy, Recall, F1-score and AUC (95.90%, 74.00%, 73.00% and 84.41%, respectively). The BI-GRU model optimised the classification of the Cairo dataset.
- On the Kyoto7 dataset, BI-LSTM obtained the highest precision, recall, F1-score and AUC (67.00%, 66.5%, 66.5% and 80.40%, respectively), but BI-GRU maximise the accuracy (91.10%). The optimal deep learning model on the Kyoto7 dataset is BI-LSTM.
- On the Kyoto8 dataset, LSMT achieved the highest precision, recall, F1-score and AUC (80.50%, 79.5%, 80.00% and 87.70%, respectively). However, BI-GRU yielded the highest accuracy (97.50%). LMST was the most suitable deep learning model for classifying the Kyoto8 dataset.
- On the Kyoto 11 dataset, BI-GRU achieved the highest accuracy, recall, F1-score (first equal with BI-LSTM) and AUC (96.70%, 75.00%, 76.00% and 86.60%, respectively), although BI-LSMT maximised the precision (79.5%). BI-GRU was the most suitable deep learning model for classifying the Kyoto11 dataset.
- On the Milan dataset, BI-LSMT achieved the highest precision (84.50%). However, BI-GRU delivered the highest accuracy, recall, F1-score and AUC (97.90%, 80.00%, 80.00% and 89.50%, respectively). BI-GRU optimised the deep learning on the Milan dataset.

Table 12. Friedman test results of the nine deep learning models

Friedman Test					
Models	Mean Ranks				
	Accuracy	Precision	Recall	F1-score	AUC
BI-LSTM	6.4	**8.8**	7.1	**8.1**	7.3
BI-GRU	**9**	6.4	**7.8**	7.6	**7.8**
LSTM-CNN	5.4	5.4	5.4	5.4	5.4
CNN	5	3.4	3.4	3.4	3.4
LSTM	5	7.4	6.9	6.8	6.9
GRU	7.8	7	**7.8**	7.1	7.6
Autoencoder	2.7	2.3	2.4	2.4	2.3
SparseCoding	1.2	2	1.8	1.8	1.9
DNN	2.5	2.3	2.4	2.4	2.4
Test Statistics[a]					
N	5	5	5	5	5
Chi-Square	34.684	35.327	34.020	34.064	33.365
df	8	8	8	8	8
Asymp. Sig.	3.05E-05	2.33E-05	4.03E-05	3.96E-05	5.29E-05

Table 12 shows the Friedman test results of the nine deep learning models. The upper part of this table displays the mean ranks showing how the results differed among the models. The lower part displays the test statistics, which determine whether the differences were statistically significant. The highest rank in each evaluation method is highlighted in red font. The results are explained below.

- The accuracy statistics yielded $\chi^2 = 34.684$ and p (Asymp.Sig) = 3.05E-05 (much less than 0.05). Therefore, we can confidently assume significant differences between the mean-rank accuracies of the deep learning models. The highest mean rank of the accuracy was 9.0, obtained by BI-GRU.
- The precision statistics yielded $\chi^2 = 35.327$ and p (Asymp.Sig) = 2.33E-05 (much less than 0.05). Therefore, we can confidently assume significant differences between the mean-rank precisions of the deep learning models. The highest mean rank of the precision was 8.8, obtained by BI-LSTM.
- The recall statistics yielded $\chi^2 = 34.020$ and p (Asymp.Sig) = 4.03E-05 (much less than 0.05). Therefore, the mean-rank recalls of the nine deep learning models

are significantly different. The highest mean rank of the recall was 7.8, jointly achieved by BI-GRU and GRU.

- The F-score statistics yielded $\chi^2 = 34.064$ and p (Asymp.Sig) = 3.96E-05 (much less than 0.05). Therefore, the mean-rank F-scores of the nine deep learning models are significantly different. The highest mean rank of the F-score was 8.1, achieved by BI-LSTM.
- The AUC statistics yielded $\chi^2 = 33.365$ and p (Asymp.Sig) = 5.29E-05 (much less than 0.05), implying that the mean-rank AUCs are significantly different among the nine deep learning models. The highest mean rank of the AUC was 7.8, obtained by the BI-GRU model.
- The GRU model maximised the recall, whereas Bi-LSTM maximised the precision and F1-score. However, the BI-GRU model obtained the highest mean-rank values of the accuracy recall, and AUC. In a comprehensive mean-ranking, BI-GRU outperformed the other deep learning models.
- Sparse Coding obtained the lowest mean rank values among the baseline deep learning models.

4.3 Discussion

In this thesis, we evaluated the classification performances of nine deep learning models on five HAR datasets. To verify the performances of these models, we cycled through all training sequences using 5-fold cross-validation and reported the mean evaluation results. As the evaluation metrics, we selected the accuracy, precision, recall, F-score and AUC. Considering all of these metrics, the improved baseline deep-learning model BI-GRU exhibited the highest classification performance on the five CASAS datasets, especially on the Milan dataset, where the accuracy reached 97.90%. The Friedman test results confirmed that BI-GRU and BI-LSTM were the best choices for classifying the CASAS datasets. BI-GRU obtained the highest mean ranks of the accuracy, recall and AUC measures, whereas BI-LSTM achieved the highest mean-rank

precision and F1-score. Note that the performance gap between BI-GRU and BI-LSTM is not very obvious.

The BI-GRU model extends the unidirectional GRU model by adding a second layer. For each training sequence, the forward and backward GRUs of the BI-GRU are connected to an output layer. The BI-GRU structure provides complete past and future contextual information at each point in the input sequence of the output layer. The model uses this past and future information to classify data, which may explain (at least partly) its high performance in identifying human activities. Meanwhile, BI-LSTM extends the unidirectional LSTM network pair, also by introducing a second layer. For each training sequence, the forward and backward LSTMs are connected to an output layer. Like BI-GRU, the BI-LSTM model classifies data based on the past and future information. However, the LSTM model is more complicated than the GRU model and requires more parameters, so takes longer to iterate. Specifically, the training time of the BI-LSTM model (on five datasets) was 150 hours, versus 126 hours for BI-GRU. Give its high performance and higher efficiency than BI-LSTM, we conclude that BI-GRU is the best model for classifying the five CASAS datasets.

In Chapter 3.5. we hypothesize that the bidirectional and hybrid architectures would improve the performances of these deep learning models (CNN, LSTM, GRU). Comparing the improved result, the performances of the improved models are on the Cairo, Kyoto11 and Milan datasets were significantly improved after improvement. The improved result by the Friedman test confirmed that the bidirectional architectures can significantly improve the performance of LSTM and GRU, the hybrid architectures can significantly improve the performance of CNN. However, the performance of the improved model (LSTM + CNN) on CASAS datasets reduced from LSTM baseline models. Overall, the bidirectional architecture can effectively improve the performance of deep learning models, the effect of hybrid architectures on improved models requires more experiments to verify.

4.4 Summary

In this chapter, we reported the evaluation results and analysed them by statistical methods. We divided the models into three groups for comparison. The first group comprised the six baseline deep-learning models. In a comparison study of this group, the GRU and LSTM demonstrated excellent performance on the five CASAS datasets. The second group comprised the three improved baseline models. Among these models, the BI-GRU delivered excellent performance on the CASAS datasets. The third group consisted of all nine deep learning models (the six baseline models and the three improved models). A comprehensive performance comparison of the nine deep learning models confirmed that BI-GRU is most suitable for HAR, and the bidirectional architecture can effectively improve the performance of deep learning models.

Chapter 5
Conclusions and Future Work

In this thesis, we separately evaluated the performances of nine deep learning models. A Friedman test confirmed that the improved baseline model BI-GRU was the most suitable classifier of HAR. This chapter summarises the research contributions, mentions the limitations of the study, and suggests ideas for future improvements.

5.1 Summary of Contributions

In recent years, deep learning technology has made significant progress. Many research teams have begun to focus on human activity recognition based on deep learning models， especially in smart environments, Human activity recognition based on sensor data and activity recognition technology . In this thesis, we propose an empirical evaluation of six baseline deep learning models (Convolution Neural Network, Long Short-Term Memory, Gated Recurrent Units, Deep Neural Network, Autoencoder, Sparse Coding) and three improved baseline deep learning models (Bi-directional LSTM, Bi-directional GRU, LSTM and CNN) for five resident activity recognition datasets with evaluation methods and statistical test.

In Chapter 1, we introduced the research background, research motivation, research questions, contribution and thesis structure. In Chapter 2, we summarize a large number of research thesis on human activity recognition and deep learning models which provide an essential theoretical basis for research design. We also study the related work of evaluation method, statistical test and datasets. In this process, we found a large number of baseline deep learning models and improved methods. This provided us with evidence for the evaluation and performance improvement of deep learning models, and guided our subsequent research work. In Chapter 3, we introduced the study design, datasets selection, data preprocessing, and our hypotheses. Firstly, we hypothesize that the performance of CNN, LSTM and GRU is optimal for the five resident activity recognition datasets. We improve baseline models is based on CNN, LSTM and GRU. Secondly, we hypothesize that the bi-directional models and hybrid model can improved performance. Then the improved baseline deep learning models are Bi-directional LSTM, Bi-directional GRU, LSTM and CNN. In Chapter 4, we evaluation the performance of six baseline deep learning models and three improved baseline deep learning models for five resident activity datasets with evaluation methods and statistical test. We find the performance of Bi-directional GRU is optimal and bi-directional models can effectively

improve model's performance. In Chapter 5, we found the limitations of the research, Meanwhile, we also propose future research work.

The overall contribution of this thesis is to explore the classification performance of multiple baseline deep learning models in human activity recognition and methods to improve model performance. We used a total of six baseline deep learning models and three improved baseline deep learning models. By comparing the experimental results of different deep learning models, we found that BI-GRU has better performance in classify activity recognition datasets. The bi-directional models can effectively improve the performance of models.

5.2 Limitations

In this research, we compared and analysed the performances of several baseline deep- learning models and their improved versions. Although the deep learning models were successfully evaluated on HAR, we must be aware of four limitations: 1) The datasets record only the simple daily activities collected from embedded sensors; 2) More comprehensive data preprocessing is required; 3) The training of complex deep learning models is time-consuming; 4) The tested deep learning algorithms are unsophisticated. Each limitation is discussed below.

- The HAR datasets record only simple daily activities.

  In applications, HAR must detect not only simple activities (eating, working, and sleeping), but also more complex activities. Therefore, when evaluated on these simple datasets, the deep learning models cannot obtain comprehensive results. The models should be evaluated on more sophisticated datasets in the future.

- The data preprocessing methods are conventional methods

  The data preprocessing in this thesis proceeded in three steps: 1) remove the errors and any lost data, 2) reclassify the activity categories, and 3) convert the file formats. These data preprocessing methods are conventional methods. The lack of any particular data preprocessing may have caused the poor performance

of some models in the evaluation results. Improving the performance of the deep learning models is another future task.

- Complex deep learning models require a long training time

  The training times of the discriminative and hybrid deep-learning models exceeded 24 hours. Complex deep learning models are flexible, requiring a large number of architectures and node types to classify a large number of datasets. Therefore, they are time-consuming and demand high-end computer hardware. We trained the models on a CPU, which extended the training time. Reducing the training time in future will be important for improving the efficiency of the evaluation.

- The deep learning algorithms were uncomplicated.

  The baseline deep-learning models and their improved versions were oversimplified for HAR. These uncomplicated deep learning models should be replaced with more complex deep learning models in future experiments.

5.3 Future Work

In this thesis, we evaluated the classification results of the baseline deep-learning models and their improved versions on residential living activities. This research provides a deep understanding of each model. However, the research questions need to be refined and upgraded in future work.

The limitations of the research were discussed in the previous section. Based on these limitations, this section presents some research avenues requiring further exploration. Open to discussion are evaluations on complex high-level activity datasets, more sophisticated data pre-processing, reducing the training time, and employing more complex deep learning models. The research directions on these critical themes are suggested below:

- More complex high-level activity datasets

In this study, behaviour recognition was learned on simple daily activity datasets acquired from embedded sensors. In future work, behaviours should be learned from complex high-level activity datasets collected from hybrid sensors and from mobile and wearable devices. The deep learning algorithms should also be evaluated on mobile and wearable devices, which collect complex datasets. To improve the evaluation results, we need to adjust the parameters of the deep learning model.

- Data pre-processing

  Data preprocessing is only one of the important steps in HAR. In future work, we should evaluate the effect of data preprocessing on the deep learning algorithm. This analysis will reveal the impact of the data preprocessing method on the calculation time, classification accuracy and learning method performance. For this purpose, we should try different preprocessing methods such as normalisation, standardisation and different dimensionality reduction methods.

- Reduce the training time

  The training time is one performance criterion of deep learning models. This thesis recorded the training time of each model. The complex deep learning models required an unrealistically long training time. In future work, the training efficiency should be improved by improving the hardware, optimising the model, and simplifying the datasets.

- Evaluation of more complex deep learning models

  More complex deep learning models, such as Fast R-CNN and Faster R-CNN, should be evaluated for HAR in future work.

References

Ajmal, M., Ahmad, F., Naseer, M., & Jamjoom, M. (2019). Recognizing Human Activities From Video Using Weakly Supervised Contextual Features. *IEEE Access* .

Akkad, K., & He, D. (2019). A Hybrid Deep Learning Based Approach for Remaining Useful Life Estimation. *IEEE International Conference on Prognostics and Health Management (ICPHM)*.

Amroun, H., Temkit, M. H., & Ammi, M. (2017). Recognition of Human Activity Using Paired Connected Objects. *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*.

Amroun, H., Temkit, M., & Ammi, M. (2017). DNN-Based Approach for Recognition of Human Activity Raw Data in Non-Controlled Environment. *2017 IEEE International Conference on AI & Mobile Services (AIMS)*.

Aztiria, A., Augusto, C. J., Basagoiti, R., Izaguirre, A., & Cook, J. D. (2013). Learning Frequent Behaviors of the Users in Intelligent Environments. *IEEE Transactions on Systems*.

Badem, H., Caliskan, A., Basturk, A., & Yuksel, M. E. (2016). Classification of human activity by using a Stacked Autoencoder. *2016 Medical Technologies National Congress (TIPTEKNO)*.

Benmansour, A., Bouchachia, A., & Feham, M. (2015). Multioccupant Activity Recognition in Pervasive Smart Home Environments. *ACM Computing Surveys*.

Bharkad, A. A. (2013). *Survey of currency recognition system using image processing. International Journal of Computational Engineering Research.*

Bhat, G., Tuncel, G., An, S., Lee, G., & Ogras, Y. (2019). An Ultra-Low Energy Human Activity Recognition Accelerator for Wearable Health Applications. *ACM Trans. Embed. Comput. Syst*.

Bhattacharya, S., & Lane, D. (2016). From smart to deep: Robust activity recognition on smartwatches using deep learning. *IEEE*.

Bulling, A., Blanke, U., & Schile, B. (2014). A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys*.

Cao, L., Wang, Y., Zhang, B., Jin, Q., & Vasilakos, V. (2018). GCHAR:An efficient Group-based Context–aware human activity recognition on smartphone. *Journal of Parallel and Distributed Computing*.

Carfi, A., Motolese, C., Bruno, B., & Mastrogiova, F. (2018). *Online Human Gesture Recognition using Recurrent Neural Networks and Wearable Sensors.*

Chang, M.-C., Krahnstoever, N., Lim, S., & Yu, T. (2010). Group level activity recognition in crowded environments across multiple cameras. *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*.

Chen, L., Hoey, J., Nugent, D., Cook, J., & Yu, Z.-w. (2012). Sensor-Based Activity Recognition. *IEEE Transactions on Systems*.

Chen, W.-H., Baca, C. A., & Tou, C.-H. (2017). LSTM-RNNs combined with scene information for human activity recognition. *IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*.

Chen, Z., Yeo, C. K., Lee, B. S., & Lau, C. T. (2018). *Autoencoder-based network anomaly detection.* IEEE.

Cheng, W.-Y., Scotland, A., Lipsmeier, F., & Kilchenman, T. (2017). Human Activity Recognition from Sensor-Based Large-Scale Continuous Monitoring of Parkinson's Disease Patients. *IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*.

Compagnon, P., Lefebvre, G., Duffner, S., & Garcia, C. (2019). Personalized Posture and Fall Classification with Shallow Gated Recurrent Units. *IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*.

Cook, J. (2010). Learning Setting-Generalized Activity Models for Smart Spaces. *IEEE Intell Syst*.

Cook, J., & Das, K. (2014). *Smart Environments: Technology, Protocols, and Applications.* Wiley.

De-La-Hoz-Franco, E., Ariza-Colpas, P., Quero, M., & Espinilla, M. (2018). Sensor-Based Datasets for Human Activity Recognition – A Systematic Review of Literature. *IEEE Access*.

Deng, Y., Wang, L., Jia, H., Tong, X., & Li, F. (2018). *A Sequence-to-Sequence Deep Learning Architecture Based on Bidirectional GRU for Type Recognition and Time Location of Combined Power Quality Disturbance.* IEEE.

Deng, Y., Wang, L., Jia, H., Tong, X., & Li, F. (2019). *A Sequence-to-Sequence Deep Learning Architecture Based on Bidirectional GRU for Type Recognition and Time Location of Combined Power Quality Disturbance.* IEEE.

Diego, F., Reichinnek, S., Both, M., & Hamprecht, F. A. (2013). Automated identification of neuronal activity from calcium imaging by sparse dictionary learning. *2013 IEEE 10th International Symposium on Biomedical Imaging*.

Duffner, S., Berlemont, S., Lefebvre, G., & Garcia, C. (2014). 3D Gesture Classification with convolution neural networks. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Erfani, M., Rajasegarar, S., Karunasekera, S., & Leckie, C. (2016). High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition*.

Fahad, G., Tahir, F., & Rajarajan, M. (2015). Feature selection and data balancing for activity recognition in smart homes. *2015 IEEE International Conference on Communications (ICC)*.

Fang, H., & Hu, C. (2014). Recognizing human activity in smart home using deep learning algorithm. *Proceedings of the 33rd Chinese Control Conference*.

Fang, H., Srinivasan, R., & Cook, J. (2012). Feature selections for human activity recognition in smart home environments. *International Journal of Innovative*.

Feng, W., Chen, Z., & Fu, Y. (2018). Autoencoder Classification Algorithm Based on Swam Intelligence Optimization. *2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*.

Fournier, Q., & Aloise, D. (2019). Empirical Comparison between Autoencoders and Traditional Dimensionality Reduction Methods. *IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*.

Fraiwan, L., & Lweesy, K. (2017). Neonatal sleep state identification using deep learning autoencoders. *IEEE 13th International Colloquium on Signal Processing & its Applications (CSPA)*.

Gao, J., Yang, J., Wang, G., & Li, M. (2016). A novel feature extraction method for scene recognition based on Centered Convolutional Restricted Boltzmann Machines. *Neurocomputing*.

Gu, F., Khoshelham, K., Valaee, S., Shang, J., & Zhang, R. (2018). Locomotion Activity Recognition Using Stacked Denoising Autoencoders. *IEEE Internet of Things Journal*.

Gumaei, A., Hassan, M. M., Alelaiwi, A., & Alsalman, H. (2019). A Hybrid Deep Learning Model for Human Activity Recognition Using Multimodal Body Sensing Data. *IEEE Access*.

Ha, S., & Choi, S. (2016). Convolutional Neural Networks for Human Activity Recognition using Multiple Accelerometer and Gyroscope Sensors. *2016 International Joint Conference on Neural Networks (IJCNN*.

Ha, S., Yun, J.-M., & Choi, S. (2015). *Multi-modal Convolutional Neural Networks for Activity Recognition.*

Hammerla, Y. N., Halloran, S., & Ploetz, T. (2016). Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables.

Hammerla, Y., Fisher, J., Andras, P., Rochester, L., Walker, R., & Plötz, T. (2015). PD Disease state assessment in naturalistic environments using deep learning. *AAAI*.

Hao, Y., Sheng, Y., & Wang, J. (2019). Variant Gated Recurrent Units With Encoders to Preprocess Packets for Payload-Aware Intrusion Detection. *IEEE Access*.

Hayashi, T., Nishida, M., Kitaoka, N., & Takeda, K. (2015). Daily activity recognition based on DNN using environmental sound and acceleration signals. *2015 23rd European Signal Processing Conference (EUSIPCO)*.

He, B., Guan, Y., & Dai, R. (2018). Convolutional Gated Recurrent Units for Medical Relation Classification. *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*.

Janod, K., Morchid, M., Dufour, R., Linarès, G., & Mori, R. D. (2017). Denoised Bottleneck Features From Deep Autoencoders for Telephone Conversation Analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Jobanputra, C., Bavishi, J., & Doshi, N. (2019). Human Activity Recognition: A Survey. *Procedia Computer Science*.

Katsageorgiou, V.-M., Zanotto, M., Tucci, V., Murino, V., & Sona, D. (2017). Data-driven study of mouse sleep-stages using Restricted Boltzmann Machines. *2017 International Joint Conference on Neural Networks (IJCNN)*.

Kaur, A., & Kaur, I. (2018). An empirical evaluation of classification algorithms for fault prediction in open source projects. *Journal of King Saud University - Computer and Information Sciences*.

Kolekar, H. M., & Dash, P. D. (2016). Hidden Markov Model based human activity recognition using shape and optical flow based features. *IEEE Region 10 Conference (TENCON)*.

Lara, D. O., & Labrador, A. (2013). A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys & Tutorials* .

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *NATURE*, 436-444.

Lee, D.-G., Kim, P.-S., & Lee, S.-W. (2017). Local group relationship analysis for group activity recognition. *2017 17th International Conference on Control, Automation and Systems (ICCAS)*.

Lee, K.-S., Chae, S., & Park, H.-S. (2019). Optimal Time-Window Derivation for Human-Activity Recognition Based on Convolutional Neural Networks of Repeated Rehabilitation Motions. *IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*.

Lei, P., & Todorovic, S. (2016). Modeling human-skeleton motion patterns using conditional deep Boltzmann machine. *23rd International Conference on Pattern Recognition (ICPR)*.

Li, S., Wang, Q., Liu, X., & Chen, J. (2018). *Low Cost LSTM Implementation based on Stochastic Computing for Channel State Information Prediction.* IEEE.

Li, X., He, Y., Yang, Y., Hong, Y., & Jing, X. (2019). LSTM based Human Activity Classification on Radar Range Profile. *IEEE International Conference on Computational Electromagnetics (ICCEM)*.

Li, X., Zhang, Y., Zhang, J., Chen, S., Marsic, I., Farneth, A. R., & Burd, S. R. (2017). Concurrent Activity Recognition with Multimodal CNN-LSTM Structure.

Li, Z., Shi, Z., Li, Z., & Shi, Z. (2009). *Image Classification Using Structural Sparse Coding Model.* IEEE.

Liciotti, D., Bernardini, M., Romeo, L., & Frontoni, E. (2019). A sequential deep learning application for recognising human activities in smart homes. *Neurocomputing*.

Liu, C., Ying, J., Han, F., & Ruan, M. (2018). Abnormal Human Activity Recognition using Bayes Classifier and Convolutional Neural Network. *IEEE 3rd International Conference on Signal and Image Processing (ICSIP)*.

Liu, G., Liang, J., Lan, G., Hao, Q., & Chen, M. (2016). *Convolution neutral network enhanced binary sensor network for human activity recognition.* IEEE.

Liu, J., Qiu, Y., Ma, Z., & Wu, Z. (2019). Autoencoder based API Recommendation System for Android Programming. *2019 14th International Conference on Computer Science & Education (ICCSE)*.

Liu, W., Zha, Z.-J., Wang, Y., Lu, K., & Tao, D. (2016). p-Laplacian Regularized Sparse Coding for Human Activity Recognition. *IEEE Transactions on Industrial Electronics*.

M.Sarigui, B.M.Ozyildirim, & M.Avci. (2019). Differential convolutional neural network. *NCBI*.

Mohamad, S., Sayed-Mouchaweh, M., & Bouchachia, A. (2019). Online active learning for human activity recognition from sensory. *Neurocomputing*.

Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2018). Deep Learning for IoT Big Data and Streaming. *IEEE Communications Surveys & Tutorials*.

Moin, M. S., & Parviz, M. (2009). Exploring AUC Boosting Approach in Multimodal Biometrics Score Level Fusion. *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*.

Nweke, H. F., Teh, Y. W., Al-garadi, M. A., & Alo, U. R. (2018). Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems wirh Applications*, 233-261.

Oda, T., Obukata, R., Yamada, M., Hiyama, M., Barolli, L., & Takizawa, M. (2016). A Neural Network Based User Identification for Tor Networks: Comparison Analysis of Activation Function Using Friedman Test. *19th International Conference on Network-Based Information Systems (NBiS)*.

Okeyo, G., Chen, L., & Wang, H. (2014). Combining ontological and temporal formalisms for composite activity modelling and recognition in smart homes. *Future Generation Computer Systems*.

Olshausen, A., & Field, J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*.

Onofri, L., Soda, P., Pechenizkiy, M., & Iannello, G. (2016). A survey on using domain and contextual knowledge for human activity recognition in video streams. *Expert Systems with Applications*.

Ordóñez, J. F., & Roggen, D. (2016). Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition.

Paola, D., Naso, D., Milella, A., Cicirelli, G., & Distante, A. (2008). Multi-sensor surveillance of indoor environments by an autonomous mobile robot. *2008 15th International Conference on Mechatronics and Machine Vision in Practice*.

Park, J., Jang, K., & Yang, S.-B. (2018). Deep neural networks for activity recognition with multi-sensor data in a smart home. *IEEE 4th World Forum on Internet of Things (WF-IoT)*.

Pham, T., Tran, T., Phung, D., & Venkatesh, S. (2017). Predicting healthcare trajectories from medical records: A deep learning approach. *Journal of Biomedical Informatics*.

Phan, N., Dou, D., Piniewski, B., & Kil, D. (2015). Social restricted Boltzmann Machine: Human behavior prediction in health social network. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*.

Plötz, T., Hammerla, Y., & Olivier, P. (2011). Feature learning for activity recognition in ubiquitous computing. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*.

Poppe, R. (2010). A survey on vision-based human action recognition. *Image and Vision Computing*, 976-990.

Ranasinghe, S., Machot, A., & Mayr, C. (2016). A review on applications of activity recognition systems with regard to performance and evaluation.

Röcker, C., Ziefle, M., & Holzinger, A. (2011). Social Inclusion in Ambient Assisted Living Environments: Home Automation and Convenience Services for Elderly User.

Rogge, D., Calatroni, A., Rossi, M., Holleczek, T., Förster, K., Tröster, G., & al., e. (2010). Collecting complex activity datasets in highly rich networked sensor environments. *2010 Seventh International Conference on Networked Sensing Systems (INSS)*.

Sarma, N., Chakraborty, S., & Banerjee, D. S. (2019). Activity Recognition through Feature Learning and Annotations using LSTM. *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*.

Schrauwen, M. H. (2013). Training and analysing deep recurrent neural networks. *Advances in Neural Information Processing Systems 26 (NIPS 2013)*.

Shoaib, M., Bosch, S., Incel, O. D., Scholten, H., & Havinga, P. J. (2016). Complex Human Activity Recognition Using Smartphone and Wrist-Worn Motion Sensors.

Singla, G., Cook, J. D., & Schmitter-Edgecombe, M. (2010). Recognizing independent and joint activities among multiple residents in smart environments. *NCBI*.

Twomey, N., Diethe, T., Craddock, I., & Flach, P. (2017). Unsupervised learning of sensor topologies for improving activity recognition in smart environments. *Neurocomputing*.

Umakanthan, S., Denman, S., Fookes, C., & Sridharan, S. (2015). Class-specific sparse codes for representing activities. *2015 IEEE International Conference on Image Processing (ICIP)*.

Umphred, A. D., Lazaro, T., Roller, L., & Burton, U. (2013). *Neurological Rehabilitation.* The Netherlands:Elsevier.

van Kasteren, T. L., Englebienne, G., & Kröse, B. J. (2011). Human activity recognition from wireless sensor network data: Benchmark and software in Activity Recognition in Pervasive Intelligent Environments. *Activity Recognition in Pervasive Intelligent Environments* .

Wang, J., Chen, Y., Hao, S., Peng, X., & Hu, L. (2018). Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*.

Wang, S., & Zhou, G. (2015). A review on radio based activity recognition. *Digital Communications and Networks*, Pages 20-29.

Wang, X., & Wang, W. (2018). MRI Brain Image Classification Based on Improved Topographic Sparse Coding. *IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*.

Wang, Y., Cang, s., & Yu, H. (2019). A survey on wearable sensor modality centred human activity recognition in health care. *Expert systems with applications*, 167-190.

Wiretunga, N., & Cooper, K. (2017). Learning Deep and Shallow Features for Human Activity Recognition. *International Conference on Knowledge Science, Engineering and Management, Springer*, 469-482.

WSU CASAS Datasets. (2009). Retrieved from CASAS: http://casas.wsu.edu/datasets/

Wu, Y., Zheng, B., & Zhao, Y. (2019). *Dynamic Gesture Recognition Based on LSTM-CNN.* IEEE.

Xu, Q., Wu, Z., Yang, Y., & Zhang, L. (2017). The difference learning of hidden layer between autoencoder and variational autoencoder. *2017 29th Chinese Control And Decision Conference (CCDC)*.

Yang, J., Nguyen, N., San, P., Li, L., & Krishnaswamy, S. (2015). *Deep Convolutional Neural Networks On Multichannel Time Series For Human Activity Recognition.*

Ye, Q., Yang, X., Chen, C., & Wang, J. (2019). River Water Quality Parameters Prediction Method Based on LSTM-RNN Model. *Chinese Control And Decision Conference (CCDC)*.

Ye, X., Wang, L., Xing, H., & Huang, L. (2015). *Denoising hybrid noises in image with stacked autoencoder.* IEEE.

Yinghao Chu, C. H. (2018). Multilayer Hybrid Deep-Learning Method for Waste Classification and Recycling. *Computational Intelligence and Neuroscience in Neurorobotics*.

Yu, S. (2018). Residual Learning and LSTM Networks for Wearable Human Activity Recognition Problem. *37th Chinese Control Conference (CCC)*.

Yu, S., & Qin, L. (2018). Human Activity Recognition with Smartphone Inertial Sensors Using Bidir-LSTM Networks. *2018 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*.

Zeng, M., Nguyen, T., Yu, B., Mengshoel, J., Zhu, J., Wu, P., & Zhang, J. (2014). Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors. *6th International Conference on Mobile Computing, Applications and Services*.

Zhang, H., Liu, C., Inoue, N., & Shinoda, K. (2018). Multi-Task Autoencoder for Noise-Robust Speech Recognition. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Zhang, L., & Ma, C. (2012). Low-rank, sparse matrix decomposition and group sparse coding for image classification. *2012 19th IEEE International Conference on Image Processing*.

Zhang, L., Wu, X., & Luo, D. (2015). *Human activity recognition with HMM-DNN model.* IEEE.

Zhang, S., Pan, X., Cui, Y., Zhao, X., & Liu, L. (2019). Learning Affective Video Features for Facial Expression Recognition via Hybrid Deep Learning. *IEEE Access*.

Zhou, Y., Amimeur, A., Jiang, C., Dou, D., Jin, R., & Wang, P. (2018). Density-aware Local Siamese Autoencoder Network Embedding with Autoencoder Graph Clustering. *2018 IEEE International Conference on Big Data (Big Data)*.

Zolfaghari, S., Zall, R., & Keyvanpour, R. (2016). Sonar: Smart ontology activity recognition framework to fulfill semantic web in smart homes. *Second International Conference on Web Research (ICWR)*.