Novel Human Activity Recognition and Recommendation Models for Maintaining Good Health of Mobile Users

XINYI ZENG[1], MENGHUA HUANG[1], HAIYANG ZHANG[2], ZHANLIN JI[3,*], IVAN GANCHEV[4,*]
[1]College of Artificial Intelligence,
North China University of Science and Technology,
Tangshan,
CHINA

[2]Department of Computing,
Xi'an Jiaotong–Liverpool University,
Suzhou,
CHINA

[3]College of Artificial Intelligence,
North China University of Science and Technology,
Tangshan,
CHINA

also with
Telecommunications Research Center (TRC),
University of Limerick, Limerick,
IRELAND

[4]University of Plovdiv "Paisii Hilendarski",
Plovdiv,
BULGARIA

also with
Institute of Mathematics and Informatics,
Bulgarian Academy of Sciences (IMI–BAS),
Sofia,
BULGARIA

also with
Telecommunications Research Center (TRC),
University of Limerick, Limerick,
IRELAND

**Corresponding Authors*

Abstract: - With the continuous improvement of the living standard, people have changed their concept from disease treatment to health management. However, most of the current health management software makes recommendations based on users' static information, with low updating frequency. The effect of targeted suggestions becomes weak with time, and it is hard for the recommendation effect to be satisfactory. Based on the use of smartphones for recognizing human activities in real-time, firstly, a novel 'CNN+GRU' model is proposed in this paper, utilizing both convolutional neural networks (CNNs) and gated recurrent units (GRUs). 'CNN+GRU' can improve the recognition speed and extract the features in sensor data more accurately by achieving in the conducted experiments an average accuracy of 91.27%, thus outperforming other models compared. Secondly, another model, named SimilRec, is proposed for physical activity recommendation to

users based on their health profile, the similarities between their current physical activity sequence, and the historical physical activity sequence of other (similar) users.

Key-Words: feature extraction; convolutional neural network (CNN); gated recurrent unit (GRU); human activity recognition (HAR); physical activity recommendation, recommendation system.

Received: July 15, 2022. Revised: October 18, 2023. Accepted: November 18, 2023. Published: January 23, 2024.

1 Introduction

Smartphones and smart wearable devices have become very popular recently. In addition, lots of personal health management software has been developed for such devices. However, the overall health-related recommendations and suggestions made are still not well personalized to individual users. A vast majority of wearable devices worn by users can only be used to record data that are easy to obtain, such as the step count or heart rate. Based on the average daily target of 10,000 steps made, most commercial software can only calculate the number of calories burnt by the user in the performed physical exercises and evaluate his/her health condition but cannot recognize the actual human activities and evaluate the real exercise volume of users. The inability of such software to provide personal health-related recommendations, that best suit the user, could badly affect his/her physical condition and/or behavioral habits.

Human activity recognition (HAR) can be considered a typical pattern recognition task. Decision trees, support vector machines (SVMs), adaptive boosting (AdaBoost), [1], and hidden Markov models are mainly used for modeling in the conventional pattern recognition methods, which have shown good progress and achieved satisfactory results, [2]. However, these methods are constrained by human domain knowledge in most daily activity recognition tasks, [3]. Additionally, these methods can only be used for learning shallow features, which may lead to a decline in the HAR performance.

Deep learning has shown excellent abilities in various fields in recent years. Different from conventional pattern recognition methods, it can greatly reduce the workload of feature design and help learn more advanced and meaningful features by training end-to-end neural networks. In addition, a deep network structure is more suitable for unsupervised learning. This makes it quite suitable for HAR. Basically, for this task, the data from multiple sensors are inputted into a convolutional neural network (CNN) or recurrent neural network (RNN) model to obtain time series data and capture the features therein to recognize human activities. As illustrated in Figure 1, for the HAR task, sensors embedded in smartphones and smart wearable devices (e.g., acceleration sensors, gyroscope sensors, etc.) can be greatly utilized, [4], to capture the movement specifics of the corresponding users and then extract the time series features to recognize the human activities performed. In the past, these features were extracted manually. The common features include statistical features, such as maximum, mean, and minimum values and variance, and frequency domain features, such as Fourier transforms, [5]. Nevertheless, the manually extracted features are highly dependent on the data sets used and very poor in generalization, which necessitates a second manual extraction of features upon replacement of data sets, which is both time- and effort-consuming. Additionally, the manual extraction of features is limited to the cognition of human experts performing this task. Generally, experts can extract shallow features but cannot obtain deep features in most cases. Thus, automatic extraction of features, e.g., using deep learning, is mainly used today, [6]. In line with this trend, this paper considers the use (single or combined) of artificial neural networks for HAR to find the best solution by utilizing the data supplied by sensors embedded in users' smartphones and/or smart wearable devices.

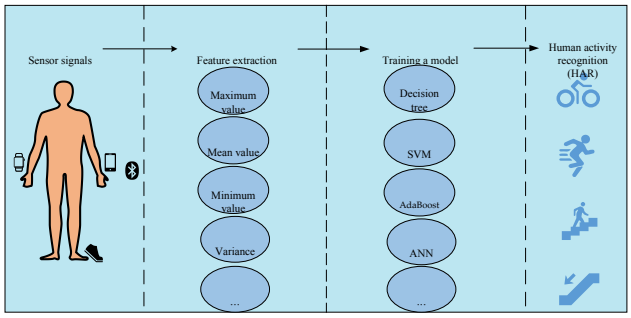

Fig. 1: An illustration of HAR steps

In their normal daily life, people usually follow a fixed lifestyle by performing various physical activities in the daytime and having a rest at night. Important here is the ability to mine users' life patterns based on their daily life records, then infer their subsequent activities based on effective

personal historical activity sequences, and recommend the most appropriate physical activities to users, so that they can maintain good living habits and keep their bodies in a good health condition, [7].

It is possible to recognize basic human activities, such as walking, running, going upstairs or downstairs, sitting, standing, etc., by utilizing models that can recognize human activities based on mobile phones' sensor data, [8]. But in daily health events, the aim is to analyze users' health condition, based on their eating, sleeping, and other similar activities, with high-level semantics. To this end, further analysis is required, based on experiments with data sets containing the daily activity data of many users, to analyze the semantic similarities of the users' activities, [9]. Then, the distance of users' historical activity sequence can be judged, and the required subsequent physical activities can be recommended, as to provide the users with proper suggestions for maintaining a good health condition.

The activity log data of a user can be easily recorded if s/he uses a smartphone or a smart wearable device. Then, when the user is performing a specific activity, another related activity can be recommended to be performed next, by analyzing the user's historical activity data. The daily activity logs of some users may be similar, but the duration of executing each physical activity and the sequence of activities may be quite different. In addition, different activities may be executed under special circumstances. Thus, it is crucial to find possible influencing factors for executing such activities. In a daily life log, each activity can be recorded along with time stamps and other contextual information. Thus, a life log may be considered as a series of activity sequences with different features. Each activity may occur several times within a certain scope. The entire activity sequence, sorted by the time of execution, can be expressed as:

$$S =< s_1, s_2, \dots, s_n >, (1)$$

where S denotes a sequenced set of a series of activities $s_i\ (i = 1, 2, \dots, n)$ executed within a certain period. The different features of the i-th activity s_i, e.g., its start time, duration, place/location of execution, etc., can be expressed as:

$$s_i =< f_i^1, f_i^2, \dots, f_i^m >, (2)$$

where $f_i^j\ (j = 1, 2, \dots, m)$ denotes the j-th feature of activity s_i. The feature factors of the sequence need to be paid attention to when recommending subsequent activities according to historical activity records of users. For this, it is necessary to calculate the similarities between the current activity sequence and the historical activity sequences contained in the records. To this end, a similarity recommendation model, called SimilRec, is proposed and described further in Subsection 4.2 of this paper.

The main contributions of the paper could be summarized as follows:

1) For HAR, a novel 'CNN+GRU' model is proposed, based on a combined use of CNNs and Gated Recurrent Units (GRUs). The proposed model works with human activity data collected by sensors, embedded in the users' smartphones or smart wearable devices. For better performing the task, the proposed 'CNN+GRU' model is optimized in terms of the number of utilized convolution filters, the number of convolution kernels, and the loss function, in order to find a good compromise between stability and accuracy achieved. Results, obtained by experiments conducted on a public data set, demonstrate that the proposed 'CNN+GRU' model outperforms other similar models used for HAR, in terms of the average accuracy achieved.

2) For physical activity recommendation, a novel SimilRec model is elaborated, which recommends physical activity to a target user based on his/her health-related profile and the discovered similarities between the current physical activity sequence performed by that user and the historical physical activity sequence of other (similar) users.

2 Background

2.1 Artificial Neural Networks (ANNs)

ANN is a powerful computing system, consisting of many artificial neurons connected in a network, simulating this way various neurons in the human brain interconnected to perform diverse basic functions. Overall, each ANN can be divided into an input layer, a hidden layer, and an output layer. The hidden layer may have multiple sublayers, which convert an ANN into a deep neural network (DNN). Figure 2 shows a fully connected DNN with two hidden layers, where circles represent the neurons (a.k.a. activation functions in mathematics) through which information is transmitted to the next layer. Introducing more hidden layers into the network allows it to enhance its capability in performing different tasks. However, having lots of hidden layers does not always work well, because this not only increases the amount of computation but also leads to gradient explosion or gradient

disappearance, resulting in worse network performance compared to using only fewer layers, [10].

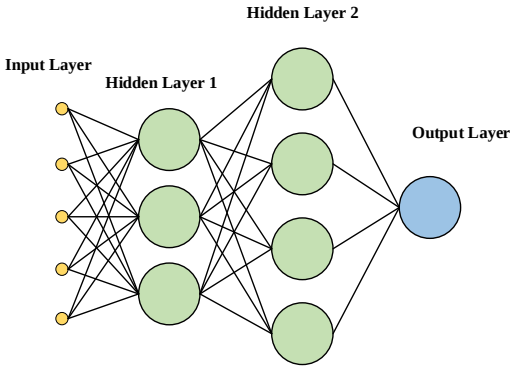

Fig. 2: A fully connected DNN with two hidden layers

The two main ANN types used in the models, proposed in this paper, are CNNs and RNNs, described in the following subsections.

2.1.1 Convolutional Neural Networks (CNNs)

CNNs were developed in response to image classification problems. A CNN model can process multi-dimensional sensor data series, extract features from such series, and map internal features to different activity types, [11]. The advantage of using CNNs in classification tasks relates to the direct extraction of features from the original series.

A typical one-dimensional (1D) CNN model with a sequential structure is depicted in Figure 3, [12]. The convolution layer is essentially a feature extraction layer. A hyperparameter F is set to specify how many feature extractors (i.e., filters that process the input data in parallel) are used. The flatten layer is used as a transition from the convolution layer to the fully connected layer to flatten multi-dimensional data into 1D data, [13]. The pooling layer is used to reduce the number of feature samples to a quarter of the original number, highlighting the most obvious features. The pooling layer performs dimensionality reduction operations on the features of the filter to form the final features. The dropout layer is used for data normalization. Due to the high learning speed of CNNs, the dropout layer is needed to help slow down the learning, prevent the model from overfitting, and improve the generalization ability of the model. Generally, a fully connected layer is used after that to complete the classification process. For the multitype task of HAR, the Adam optimization algorithm is usually used to optimize the network, and the loss function adopts the categorical cross-entropy loss.

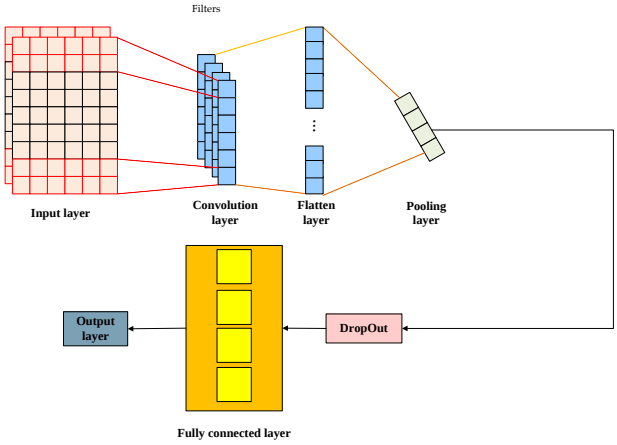

Fig. 3: A typical 1D CNN structure

2.1.2 Recurrent Neural Networks (RNNs)

RNNs are a special type of neural network, designed to process time series data. RNNs are widely used for speech recognition and machine translation, [14]. The continuous inputs of RNN are correlated with each other. However, there are many cumulative products, which may easily cause the problem of vanishing or exploding gradients. In addition, RNNs have higher requirements for the hardware used, and their real-time performance for pattern recognition is low. A typical RNN structure is shown in Figure 4. Note that removing the W layer converts the RNN into a fully connected neural network.

In Figure 4, X represents the vector in the input layer, S represents the vector of the intermediate hidden layer, and O represents the vector of the output layer. S can be calculated as follows:

$$S = X \times U + W \times S_{last}\,, (3)$$

where U denotes the weight matrix from the input layer to the hidden layer, V denotes the weight matrix from the hidden layer to the output layer in which $O = S * V$, W denotes the weight of the hidden layer, and S_{last} denotes the vector of the previous hidden layer. This way, the previous data are used in each cycle. However, if there are too many cycles, X will be multiplied several times. Then, this may lead to the problem of exploding or vanishing gradients, which can be solved by a new type of network, called long short-term memory (LSTM), which is formed by expanding the RNN structure, as depicted in Figure 5.

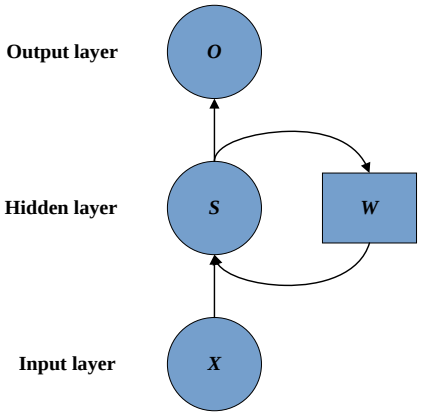

Fig. 4: A typical RNN structure

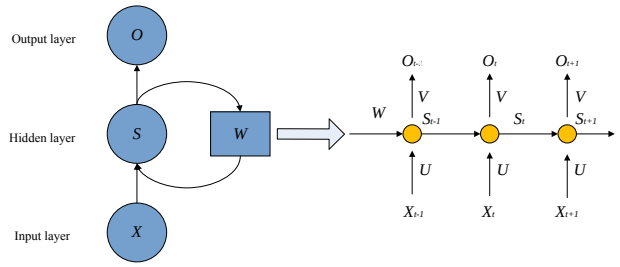

Fig. 5: The expansion of a RNN into a LSTM

2.1.3 LSTMs

With the improvement of technology, LSTMs have gradually evolved from RNNs, [15]. LSTMs are mainly used for natural language processing (NLP) and time series data processing. A LSTM model can not only avoid the problems of vanishing and exploding gradients, but also can keep the features in the time series sequence from getting lost, thanks to the three gates utilized, [16], as shown in Figure 6. The input gate is used to form the current input, using a *sigmoid* function, and add it to the value of the previous hidden state to maintain the long-term features of the time series sequence unchanged. The forget gate determines which information should be lost or retained. The output gate determines what the next hidden state is.

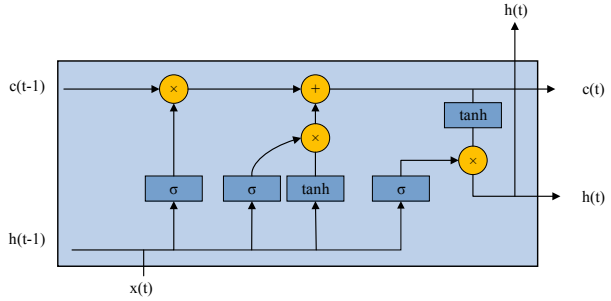

Fig. 6: A typical LSTM structure

2.1.4 Gated Recurrent Units (GRUs)

As a variant of the LSTM, the gated recurrent unit (GRU) combines the forget gate and the input gate into a single update gate, which allows to reduce the parameter calculation workload and shorten the model's training time, [17]. For both GRU and LSTM, gates can be used to retain important features. The selection of GRU or LSTM for use is generally determined by the specific task. Relatively, as shown in Figure 7, GRU has a simpler structure than LSTM, which means a smaller calculation workload. As a result, GRU maintains a fast calculation speed when a large amount of input data is used.

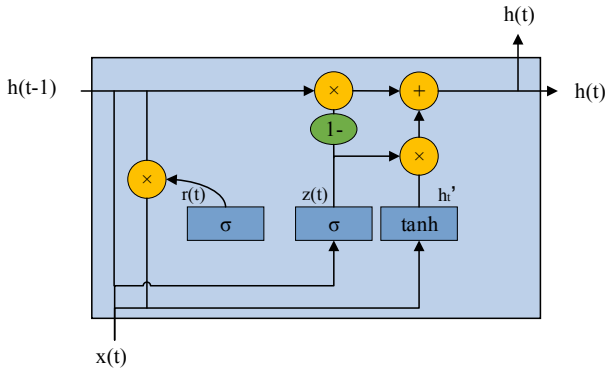

Fig. 7: A typical GRU structure.

In the proposed 'CNN+GRU' model, presented in Subsection 4.1, GRUs are used as units for extracting time series features from the collected sensor data. The experimental results, presented in Subsection 5.2, show that the combined use of CNN(s) and GRU(s), as utilized in the proposed model, allows to achieve better results in the HAR task, when compared to the combined use of CNN(s) and LSTM(s), or the single use of LSTM or GRU.

2.2 User Profiles

Health-status-related user profiles contain the users' basic health attributes and health preferences. The initial user profiles are usually created by utilizing the information entered by the respective users at the time of their registration in health-related information systems. Subsequently, these profiles are continuously updated in the process of using such systems by the users, either explicitly by utilizing the feedback (preferences, reviews, comments, tags, etc.) provided directly by the users (through various means) or implicitly by assessing the user behavior in using these systems. The updated user profiles are in turn utilized for the provision of more accurate recommendations of daily activities and/or physical exercises to the respective users. The process of creating and updating a user profile to recommend physical activities is shown in Figure 8.

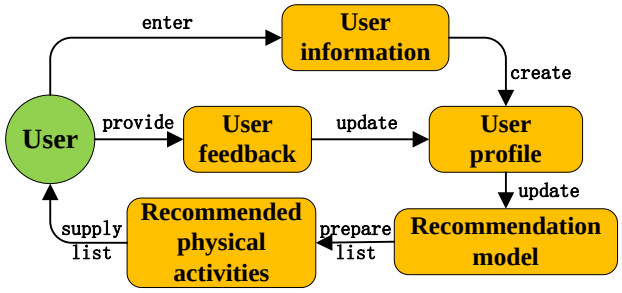

Fig. 8: The process of creating and updating a user's profile to recommend physical activities.

A user profile is initially created by inputting the user's personal information, such as gender, age, height, weight, athletic goal, and level, etc., considered as different features of that particular user, i.e.:

$$u_i = (p_{i1}, p_{i2}, \dots, p_{ij}, \dots), \quad (4)$$

where p_{ij} denotes the j-th feature of the i-th user u_i. One-hot coding is usually carried out in the selected areas for features, to reduce the data dimension and lessen the computation difficulty.

3 Related Work

3.1 Human Activity Recognition (HAR)

Due to the rapid development of the Internet of Things (IoT) and ubiquitous computing, human activity recognition based on various sensors is becoming more and more popular. Many HAR examples can be found nowadays. For instance, bracelets with an integrated heart rate sensor, an acceleration sensor, and other sensors, can capture rich human motion data. Sensors in smartphones are even more abundant. Mobile crowd sensing and computing methods, utilizing the massive number of mobile sensing devices and personal communication devices (smartphones, tablets, smartwatches, etc.), are usually used for collecting the needed data and reducing the cost of data acquisition. However, due to the uneven data quality and reliability of different processing methods used, effective data analysis and mining should be carried out subsequently, e.g., by employing deep learning techniques. There are many methods to recognize human activities through mobile phones' sensors, and multiple corresponding applications exist, e.g., for fall detection, step count statistics, and so on.

3.2 Physical Activity Recommendation

In the case of everyday health events, the goal is to analyze the user's health status based on daily life events, such as eating, sleeping, etc., with high-level semantics. At present, recommendation systems can be roughly divided into the following categories:

1) *Recommendation systems based on user behavior*: These systems employ traditional collaborative filtering recommendation algorithms and matrix decomposition algorithms. Through regular user behavior analysis and model training, user characteristic information and model parameters are updated accordingly;

2) *Tag-based recommendation systems*: These systems do not need complex algorithms. For instance, in an e-commerce recommendation system, users can recommend resources they are interested in according to their associated tags; in social networks, users can find friends with the same hobbies through tags, etc.;

3) *Recommendation systems utilizing deep learning models*: Compared with traditional recommendation models, deep learning models generally have stronger feature expression and generalization ability, can effectively capture nonlinear and unusual relationships between users and resources, and support more complex abstractions as higher-level data representation. In addition, they can learn complex relationships in the available data, contextual text, and visual information.

3.3 Recommendation Approaches

The goal of computer algorithms, used by the respective recommendation systems, is to provide users with accurate recommendations as fast as possible, [18]. Different recommendation approaches exist, each with its pros and cons, due to different recommendation tasks and different types of data sources utilized. Generally, the recommendation approaches could be divided into two main groups – content-based filtering (CBF) and collaborative filtering (CF).

CBF, [19], depends on the resource portrait and user behavior. It can search for similar resources (e.g., items, services, physical activities, exercises, etc.) under the portrait information of resources in the user history and recommend them to the user. CBF is widely used in industry due to its simplicity and efficiency.

CF, however, is typically favored over CBF due to its overall better performance in predicting common behavior patterns and its ability to address data aspects that are usually difficult to profile using CBF, [20]. CF only requires user–resource interactions to make recommendations, meaning that it is easier to adapt it to real-world scenarios than CBF, [21]. As a result, CF has been more successful and more widely used than CBF, as it only relies on the past user's

behavior (e.g., previous user's transactions, reviews, ratings, tags, etc.) without requiring specific domain information.

A widely accepted taxonomy, [22], divides CF into two categories:

(1) *Memory-based CF*, in which recommendations are based on the assumption that users who share common interests have similar tastes, or resources with similar features have similar rating patterns;

(2) *Model-based CF*, in which various machine learning or data mining techniques are utilized to discover complex patterns in the user history data to make recommendations based on these, [18], [21].

Generally, the *memory-based CF* is simpler and easy to implement, while at the same time, it can obtain reasonably accurate results, [18]. Two of the most widely used methods in this category are based on the *k*-nearest neighbor (KNN) heuristic, [23], divided into: (i) user-based KNN, [24], that predicts the rating of resource *i* by user *u* by using the existing ratings given to *i* by the set of users that are most similar to *u*; and (ii) resource-based KNN, [25], [26] that predicts the rating of resource *i* by user *u* using the existing ratings given by *u* to the set of resources that are most similar to resource *i*. Both methods use the following two steps to make predictions:

(1) *Finding the k most similar neighbors to the target user/resource*: The most important part here is to compute the similarity between users/resources. The two most popular choices for similarity metrics are: (i) the Pearson's correlation coefficient, [23], which measures the extent to which two vectors are linearly related to each other; and (ii) the cosine similarity, [27], which measures the similarity between two vectors by computing the cosine of the angle between them, [28];

(2) *Aggregating the neighbors to generate the predictive score,* [18], [23]: The predicted rating is calculated based on the ratings of the *k*-nearest neighbors selected in the first step, e.g., as the weighted sum of the ratings of the same resource, provided by the neighbors of the target user, or as the average sum of other ratings.

Due to its simplicity and flexibility, [23], the nearest-neighbor-based CF has been extensively studied, including different similarity measures, [29] [30], [31], alternative strategies to select the neighbors, etc. Two drawbacks of the memory-based CF are: (i) the low efficiency since the computation of the similarity between users/resources is expensive (quadratic time complexity) as all users/resources need to be examined to make a single prediction, and (ii) the performance of recommendation heavily depends on the similarity measure, [32].

On the other side, the *model-based CF* can tackle the data sparsity and scalability issues that the memory-based CF struggles to cope with. In addition, the model-based CF can achieve better recommendation performance and coverage than the memory-based CF, because it trains a model based on global rating data, while the memory-based CF only focuses on the local rating information, [33]. Various machine learning and data mining algorithms have been elaborated by different researchers in the past for making recommendations, such as Restricted Boltzmann Machines, [34], regression-based models, [35] and latent factor models (mostly based on matrix factorization, [20], e.g., SVD [36], SVD++ [37]), etc.

The elaborated SimilRec model, presented in Subsection 4.2, utilizes model-based CF techniques. More specifically, it is based on the *word2vec* model, [38] and the Continuous Bag-of-Words (*CBOW*) model, [39].

4 Proposed Models

4.1 'CNN+GRU' (for Human Activity Recognition)

For HAR, a novel 'CNN+GRU' model is proposed here, based on a combined use of CNN and GRU. The elaborated 'CNN+GRU' model consists of three parts (Figure 9): the first part is used to extract features using two convolution layers; the second part is used to obtain the time series relationship existing in the collected sensor data through two GRU layers; and the third part is used to expand the data generated by GRU using a fully connected layer, then input all data into a *SoftMax* function, and finally get the classification result of human activities.

One important hyperparameter in the CNN part of the proposed model is the number of filters used, initially being set to 8. However, we experimented also with other values, such as 16, 32, 64, 128, and 256, to find the optimal value of this hyperparameter for the proposed model. The obtained results are shown in Figure 10. The presented box plot diagram shows that the highest median classification accuracy is achieved with 128 filters used; however, the stability then is not great. Thus, as a good compromise between stability and accuracy, the default value of convolution filters is set to 64 in the proposed 'CNN+GRU' model.

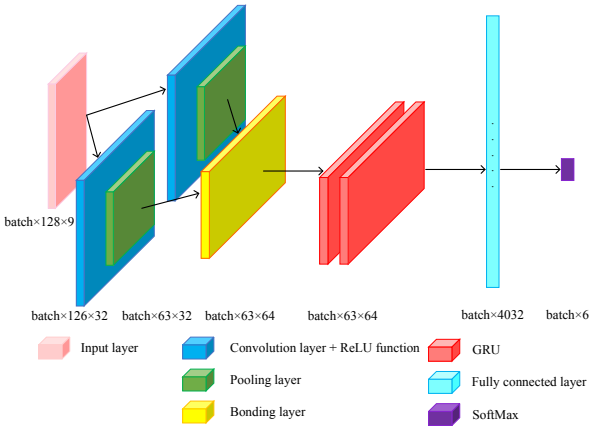

Fig. 9: The structure of the proposed 'CNN+GRU' model for human activity recognition

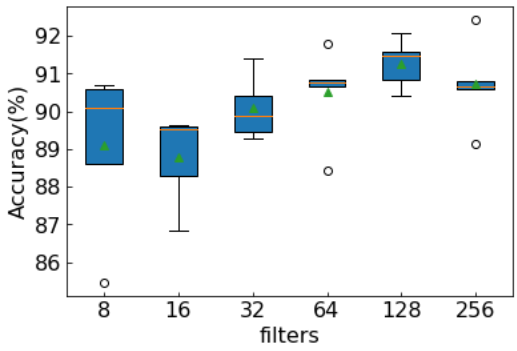

Fig. 10: The box plot diagram of accuracy vs. the number of convolution filters used by the proposed 'CNN+GRU' model

Another important hyperparameter of the proposed model is the number of convolution kernels, used to control the time step for computation each time the input sequence is read and then mapped to filters through convolution. A bigger number of kernels can embrace a wider data range for processing, thus achieving higher accuracy. However, the increase in the number of kernels leads to instability. To obtain the optimal value of this hyperparameter, experiments were conducted with different numbers of kernels, namely 2, 3, 5, 7, and 11. The obtained results are shown in Figure 11. According to the presented box plot diagram, the highest accuracy is achieved with 11 kernels; however, the stability then is not good. Thus, as a compromise between stability and accuracy, the default value of convolution kernels is set to 5 in the proposed 'CNN+GRU' model.

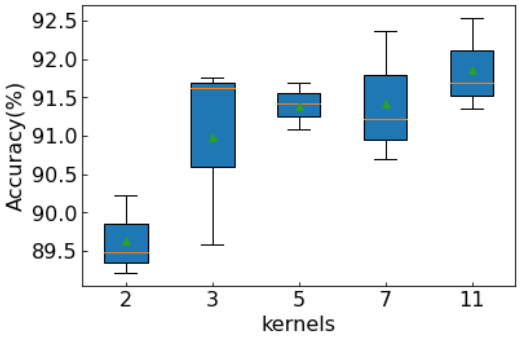

Fig. 11: The box plot diagram of accuracy vs. the number of convolution kernels used by the proposed 'CNN+GRU' model

The proposed model works with human activity data collected by sensors, embedded in the smartphones or smart wearable devices of users. These data are first converted into proper time series. By inputting N types of sensor data, the length of each sensor data sequence becomes T. So, the input to the convolution layers is:

$$x = (x_t)_{t=1}^{T}, \quad (5)$$

where $x_t \in \mathbb{R}^N$ denotes the time series corresponding to the N types of sensor data at time t. k 1D convolution layers, each adopting a ReLU as the activation function, are used. This allows the model to learn more complex features, which can be subdivided further so that the combined data may have more complex feature structures. The result of the convolution is:

$$h_{i,k} = x(t) * w_k + c_b, \quad (6)$$

where w_k denotes the weight of each filter and c_b denotes the relevant offset value. After the convolution, the data are processed by the pooling layer. The pooling process helps process data with relatively large fluctuations, which allows the model to more accurately retain signal fluctuations while extracting features. Then, the pooled data structures are merged and inputted into the two GRU layers, which are used to learn the correlations existing between the sensor sequence data. Each GRU layer is composed of E basic GRUs. The output of the e-th GRU $(e = 1, 2, \ldots, E)$ at time t is:

$$h_e(t) = GRU\big(h_c(t), h_e(t-1)\big), \quad (7)$$

where $h_c(t)$ denotes the output of the convolution layer at time t, and $h_e(t-1)$ denotes the output of the e-th GRU at time $(t-1)$.

In order to prevent the model from overfitting after passing the two-layer GRU, dropout is used to suspend the operation of some units, thus making the model more generalized. Using two GRU layers by the proposed 'CNN+GRU' model is sufficient as having more GRU layers would lead to an exponential growth of memory overhead and time overhead. Moreover, the vanishing gradient problem and the dilemma of local optimality may also occur between different GRU layers, [40].

After passing the second GRU layer, the features of the time series are further enhanced, and more accurate features can be extracted. Then, the GRU output values are expanded and inputted into the fully connected layer. The learned features are mapped to the labeled space to form a 1D array. Finally, the final classification result is obtained by applying a *SoftMax* function to the output as follows:

$$y^* = softmax\big(c(i)\big)\,, (8)$$

where $c(i)$ denotes the output values of the fully connected layer.

The proposed 'CNN+GRU' model is optimized by using the cross-entropy loss function:

$$Loss = -c_j\, log(y^*)\,, (9)$$

where c_j denotes the true value in the sample class j.

4.2 SimilRec (for Physical Activity Recommendation)

The SimilRec model, proposed in this paper for physical activity recommendation, belongs to the model-based CF category. To recommend a physical activity to a target user based on his/her health profile (containing among other things the physical activity records of that user), the similarities between the current physical activity sequence of the user and the historical physical activity sequence of other (similar) users is calculated first in three steps:

(1) ***Calculating the correlation between (historical) physical activity sequences*** *(of the target user and all other users).*

Each physical activity sequence, containing different physical activities performed daily by the corresponding user, can be regarded as a paragraph of statements composed of a certain number of words. By utilizing the *word2vec* model, [38] and the *CBOW* model, [39], each physical activity sequence is transformed into an activity vector. First, the continuous word bag concept of *word2vec* is used to model the physical activity sequences. Then, the *CBOW* model is used to predict the central word according to the context words. The *CBOW* model consists of three layers – an input layer, a hidden layer, and an output layer –, as depicted on Figure 12.

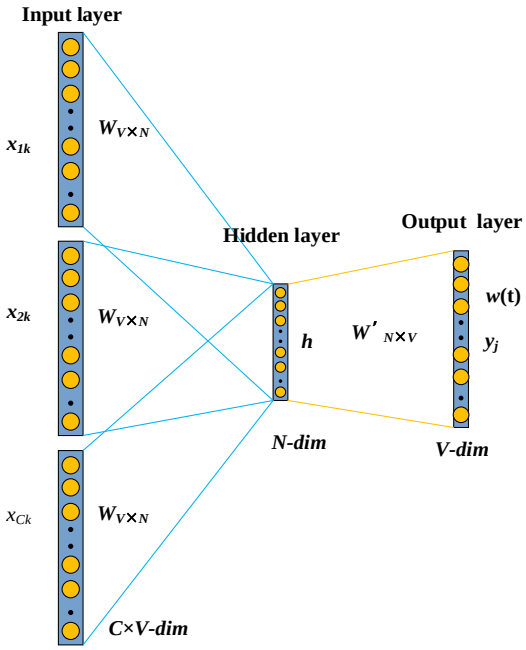

Fig. 12: The *CBOW* model structure

Given the number of contextual words C, the dimension of the word vector space V, an input layer with one-hot codes for C contextual words, the initial input weight matrix $W_{V\times N}$, and the output weight matrix $W'_{N\times V}$, the output h of the hidden layer can be calculated as:

$$h = \frac{1}{C} W_{V\times N}\left(\sum_{i=1}^{C} x_i\right), (10)$$

where x_i denotes the input of each node. Then, the output vector y_j can be calculated as:

$$y_j = softmax(hW'_{N\times V})\,. (11)$$

The input weight matrix $W_{V\times N}$ and the output weight matrix $W'_{N\times V}$ are repeatedly updated with the decrease in the subsequent gradient. The word vector of the one-hot code can be obtained by multiplying the vector of the code with the input weight matrix. This way, the activity vectors of all users can be obtained, based on their physical activity sequences. Then, the cosine similarity $cos(\theta)$ is used to measure the correlation between

the activity vector of the target user and the activity vector of each other user, as follows:

$$cos(\theta) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} {x_i}^2}\sqrt{\sum_{i=1}^{n} {y_i}^2}}\ , \qquad (12)$$

where x_i and y_i denote the corresponding components of the activity vectors of the two users. The larger the cosine similarity obtained, the stronger the correlation between the physical activity sequences of the two users (within the period considered).

(2) ***Calculating the edit distance between physical activities*** *(contained in the historical physical activity sequences of the target user and all other users)*.

Each physical activity has features such as name, start time, duration, place/location where it has been carried out by the corresponding user, etc. The edit distance between two physical activities a_1 and a_2 can be calculated by counting the minimum number of steps required to transform a_1 into a_2 by changing (editing) the features of a_1 (one feature per step). If the edit distance between two physical activities is relatively low, the correlation between these activities is high.

(3) ***Calculating the Levenshtein distance between physical activity sequences*** *(of the target user and all other users)*.

The Levenshtein distance, [41], is a string metric for measuring the difference between two strings (sequences). For instance, the Levenshtein distance between two words is represented by the minimum number of single-character edits (insertions, deletions, substitutions) required to convert one word into the other. A lower Levenshtein distance indicates a bigger similarity between two strings. The Levenshtein distance $lev_{a,b}(i,j)$ between the first *i* characters in string a and the first *j* characters in string b is defined as:

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & if\ min(i,j) = 0, \\ min\begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & otherwise. \end{cases} \qquad (13)$$

Based on the Levenshtein distance, similar physical activity sequences S_1 and S_2 can be discovered in the historical records of two users (i.e., the target user and another user) by counting the minimum number of operations required to transform S_1 into S_2. In this process: (i) a physical activity is inserted into S_1 with distance d_{insert}; (ii) an activity in S_1 is substituted with another activity with distance $d_{substitute}$; and (iii) an activity is deleted from S_1 with distance d_{delete}. Then, the activity distance between two physical activity sequences S_1 and S_2 can be expressed as:

$$d_{activity}(S_1, S_2) = \sum_{i=1}^{x} w_a \times d_{insert} + \sum_{j=1}^{y} w_a \times d_{delete} + \sum_{k=1}^{z} w_a \times d_{substitute}\ , \qquad (14)$$

where x, y, and z denote how many times a physical activity was inserted into S_1, deleted from S_1, and substituted in S_1 with another activity, respectively.

After obtaining the activity distance $d_{activity}(S_j, S_c)$ between the current activity sequence S_c of the target user (discovered in real time by a HAR technique) and each sequence S_j (found in the historical record of some other user), the score $Score(a_{rec}^j)$ for an activity a_{rec} (carried out by that other user), which is a candidate for recommendation to the target user, is calculated as:

$$Score(a_{rec}^j) = 1 - \frac{d_{activity}(S_j, S_c)}{\underset{T_p \in \ell}{MAX}\ d_{activity}(S_p, S_c)}\ , \qquad (15)$$

where $\underset{T_p \in \ell}{MAX}\ d_{activity}(S_p, S_c)$ denotes the maximum activity distance among all distances existing between the current activity sequence S_c (of the target user) and each other activity sequence (contained in the historical records of users). This is then repeated to each other user, who is different to the target user.

A score list of physical activities, which are candidates for recommendation to the target user, is prepared at the end, based on (15). Then, the average score for all same-name physical activities is calculated, followed by preparing a ranked list of all candidate activities, based on their average score. Finally, the first *N* activities in this list are recommended to the target user.

5 Experiments and Results

5.1 Data Set

The data set used in the conducted experiments was a public data set for human activity recognition using smartphones, [42], available from the machine learning repository of the University of California Irvine (UCI) [43], called here UCI data set for short.

This data set contains data of 30 adult volunteers performing activities of daily living while carrying a waist-mounted smartphone with embedded sensors. Through six types of activities (i.e., walking horizontally, walking upstairs, walking downstairs, sitting, standing, and lying down), the 3-axial linear acceleration and 3-axial angular velocity were captured by the smartphone's accelerometer and gyroscope of each participant at a constant rate of 50 Hz. The sensor signals were pre-processed using noise filters and then sampled in 128 time-step sliding windows of 2.56 sec with 50% overlap. The sensor acceleration signal was separated using a Butterworth low-pass filter into body acceleration and gravity components. The data set was randomly partitioned into two sets – a *training* set, containing 70% of the volunteers' data, and a *test* set, containing the rest of the data.

Figure 13, Figure 14, and Figure 15 depict sample time series of the X-axis linear acceleration, extracted from this data set, corresponding to three recorded human activities, respectively. The presented data have obvious differences of course. In the case of walking activity (Figure 13), the data fluctuates violently with quite a large amplitude; the waveform, however, is quite regular. In the case of standing activity (Figure 14), there are only occasional fluctuations in data; the overall curve hardly moves, indicating that the participant is in a static state. In the case of the walking upstairs activity (Figure 15), although the data fluctuations are similar to that in the walking activity case, the overall regularity is not the same, indicating that this is a different activity indeed.

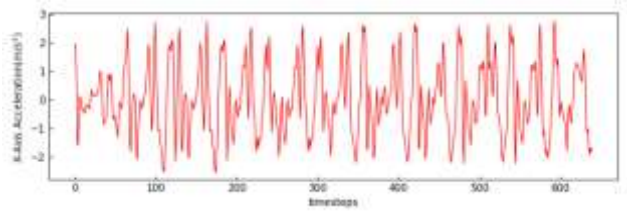

Fig. 13: Sample time series of the X-axis linear acceleration corresponding to walking activity (based on the public UCI data set)

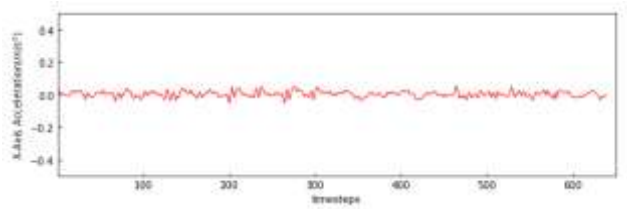

Fig. 14: Sample time series of the X-axis linear acceleration corresponding to standing activity (based on the public UCI data set)

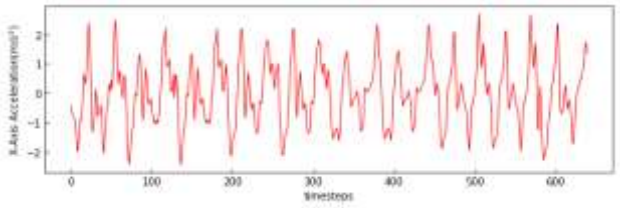

Fig. 15: Sample time series of the X-axis linear acceleration corresponding to walking upstairs activity (based on the public UCI data set)

5.2 Results

The public UCI data set was used to conduct performance comparison experiments with four models, shown in Table 1. The models were compared in terms of the average accuracy achieved by each of them in the HAR task. The obtained results (Table 1), demonstrate that the 'CNN+GRU' model, proposed in this paper, outperforms all other models, i.e., LSTM, GRU, and 'CNN+LSTM'. In addition, it is evident from the results that combining multiple types of neural networks has a better effect on HAR than using a single neural network. In addition, the results show that combining CNN(s) with GRU(s) is better for HAR than combining CNN(s) with LSTM(s).

Table 1. The HAR performance of compared models

Model	Average accuracy (%)
LSTM	88.62
GRU	88.33
'CNN+LSTM'	89.25
'CNN+GRU' (*proposed*)	**91.27**

6 Conclusion

This paper has presented a combined use of convolutional neural networks (CNNs) and gated recurrent units (GRUs) for building a novel model, named 'CNN+GRU', for human activity recognition. Multiple CNNs were adopted to extract sensor data features and capture more detailed information. Next, GRUs were used to extract the time series relationship between data features. Compared with the traditional single, simple neural networks, the recognition accuracy has been improved, the values of model parameters have been reduced, and both training and recognition speeds have been increased. As a result, the user can upload the mobile phone's sensor data to a server, generate the activity vector, and calculate the correlation

between his/her (historical) physical activity sequences and those of other users by using the second model, proposed in this paper, called SimilRec. A score list of physical activities, which are candidates for recommendation to the target user, is prepared at the end and the average score for all same-name physical activities is calculated by SimilRec, followed by the preparation of a ranked list of candidate activities, based on their average score. Finally, the first *N* activities in this list are recommended to the target user.

References:

[1] S. Wu and H. Nagahashi, "Parameterized AdaBoost: Introducing a Parameter to Speed Up the Training of Real AdaBoost," *IEEE Signal Processing Letters,* vol. 21, no. 6, pp. 687-691, 2014, doi: 10.1109/LSP.2014.2313570.

[2] K. Chen, L. Yao, D. Zhang, X. Wang, X. Chang, and F. Nie, "A Semisupervised Recurrent Convolutional Attention Model for Human Activity Recognition," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 31, no. 5, pp. 1747-1756, 2020, doi: 10.1109/TNNLS.2019.2927224.

[3] S. Angerbauer, A. Palmanshofer, S. Selinger, and M. Kurz, "Comparing Human Activity Recognition Models Based on Complexity and Resource Usage," *Applied Sciences,* vol. 11, no. 18, 2021, doi: 10.3390/app11188473.

[4] B. Fu, N. Damer, F. Kirchbuchner, and A. Kuijper, "Sensing Technology for Human Activity Recognition: A Comprehensive Survey," *IEEE Access,* vol. 8, pp. 83791-83820, 2020, doi: 10.1109/ACCESS.2020.2991891.

[5] H. Ku, P. Zhou, X. Cai, H. Yang, and Y. Chen, "Person re-identification method based on CNN and manually-selected feature fusion," in *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 29-31 July 2017 2017, pp. 93-96, doi: 10.1109/FSKD.2017.8393401.

[6] C. Ding, Y. Jia, G. Cui, C. Chen, X. Zhong, and Y. Guo, "Continuous Human Activity Recognition through Parallelism LSTM with Multi-Frequency Spectrograms," *Remote Sensing,* vol. 13, no. 21, 2021, doi: 10.3390/rs13214264.

[7] C. T. Yen, J. X. Liao, and Y. K. Huang, "Human Daily Activity Recognition Performed Using Wearable Inertial Sensors Combined With Deep Learning Algorithms," *IEEE Access,* vol. 8, pp. 174105-174114, 2020, doi: 10.1109/ACCESS.2020.3025938.

[8] M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar, "A Survey on Activity Detection and Classification Using Wearable Sensors," *IEEE Sensors Journal,* vol. 17, no. 2, pp. 386-403, 2017, doi: 10.1109/JSEN.2016.2628346.

[9] L. Dhammi and P. Tewari, "Classification of Human Activities using data captured through a smartphone using deep learning techniques," in *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, 13-14 May 2021 2021, pp. 689-694, doi: 10.1109/ICSPC51351.2021.9451772.

[10] K. D. Apostolidis and G. A. Papakostas, "A Survey on Adversarial Deep Learning Robustness in Medical Image Analysis," *Electronics,* vol. 10, no. 17, 2021, doi: 10.3390/electronics10172132.

[11] I. H. Hsieh, H.-C. Cheng, H.-H. Ke, H.-C. Chen, and W.-J. Wang, "A CNN-Based Wearable Assistive System for Visually Impaired People Walking Outdoors," *Applied Sciences,* vol. 11, no. 21, 2021, doi: 10.3390/app112110026.

[12] H. Cheng, Z. Xie, Y. Shi, and N. Xiong, "Multi-Step Data Prediction in Wireless Sensor Networks Based on One-Dimensional CNN and Bidirectional LSTM," *IEEE Access,* vol. 7, pp. 117883-117896, 2019, doi: 10.1109/ACCESS.2019.2937098.

[13] S. Mekruksavanich and A. Jitpattanakul, "A Multichannel CNN-LSTM Network for Daily Activity Recognition using Smartwatch Sensor Data," in *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*, 3-6 March 2021 2021, pp. 277-280, doi: 10.1109/ECTIDAMTNCON51128.2021.9425769.

[14] S. Zhou and T. Gao, "Brain Activity Recognition Method Based on Attention-Based RNN Mode," *Applied Sciences,* vol. 11, no. 21, 2021, doi: 10.3390/app112110425.

[15] A. Shrestha, H. Li, J. L. Kernec, and F. Fioranelli, "Continuous Human Activity Classification From FMCW Radar With Bi-LSTM Networks," *IEEE Sensors Journal,* vol. 20, no. 22, pp. 13607-13619, 2020, doi: 10.1109/JSEN.2020.3006386.

[16] L. Zhang, C. Xu, Y. Gao, Y. Han, X. Du, and Z. Tian, "Improved Dota2 lineup recommendation model based on a bidirectional LSTM," *Tsinghua Science and Technology,* vol. 25, no. 6, pp. 712-720, 2020, doi: 10.26599/TST.2019.9010065.

[17] T. Ergen and S. S. Kozat, "Unsupervised Anomaly Detection With LSTM Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 31, no. 8, pp. 3127-3141, 2020, doi: 10.1109/TNNLS.2019.2935975.

[18] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi, "Evaluating Collaborative Filtering Recommender Algorithms: A Survey," *IEEE Access,* vol. 6, pp. 74003-74024, 2018, doi: 10.1109/ACCESS.2018.2883742.

[19] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The adaptive web*: Springer, 2007, pp. 325-341.

[20] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer,* vol. 42, no. 8, pp. 30-37, 2009, doi: 10.1109/MC.2009.263.

[21] J. Bobadilla, F. Ortega, A. Hernando, and A. GutiéRrez, "Recommender systems survey," *Know.-Based Syst.,* vol. 46, pp. 109-132, 2013, doi: 10.1016/j.knosys.2013.03.012.

[22] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," presented at the *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '08*, Las Vegas, Nevada, USA, 2008.

[23] X. Ning, C. Desrosiers, and G. Karypis, "A Comprehensive Survey of Neighborhood-Based Recommendation Methods," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira Eds. Boston, MA, US.: Springer, 2015, ch. Chapter 2, pp. 37-76.

[24] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," presented at the *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '99*, Berkeley, California, USA, 1999.

[25] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," presented at the Proceedings of the tenth *International Conference on World Wide Web - WWW '01*, Hong Kong, 2001.

[26] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing,* vol. 7, no. 1, pp. 76-80, 2003, doi: 10.1109/mic.2003.1167344.

[27] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," presented at the P*roceedings of the 1994 ACM conference on Computer supported cooperative work - CSCW '94*, Chapel Hill, North Carolina, USA, 1994.

[28] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1986.

[29] G. Guo, J. Zhang, and N. Yorke-Smith, "A Novel Evidence-Based Bayesian Similarity Measure for Recommender Systems," *ACM Transactions on the Web,* vol. 10, no. 2, pp. 1-30, 2016, doi: 10.1145/2856037.

[30] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Inform Sciences,* vol. 178, no. 1, pp. 37-51, 2008, doi: 10.1016/j.ins.2007.07.024.

[31] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," *Know.-Based Syst.,* vol. 56, no. C, pp. 156-166, 2014.

[32] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative Filtering beyond the User-Item Matrix: A Survey of the State of the Art and Future Challenges," (in English), *ACM Comput. Surv.,* vol. 47, no. 1, pp. 1-45, 2014, doi: 10.1145/2556270.

[33] G. Guo, J. Zhang, and D. Thalmann, "Merging trust in collaborative filtering to alleviate data sparsity and cold start," *Knowledge-Based Systems,* vol. 57, pp. 57-68, 2014, doi: 10.1016/j.knosys.2013.12.007.

[34] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," presented at the Proceedings of the 24th international conference on Machine learning - ICML '07, Corvalis, Oregon, USA, 2007.

[35] S. Vucetic and Z. Obradovic, "Collaborative Filtering Using a Regression-Based Approach," *Knowledge and Information Systems,* vol. 7, no. 1, pp. 1-22, 2005, doi: 10.1007/s10115-003-0123-8.

[36] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality

reduction in recommender system-a case study," Minnesota Univ Minneapolis Dept of Computer Science, 2000.

[37] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 426-434.

[38] Y. Chen, S. Huang, H. Lee, Y. Wang, and C. Shen, "Audio Word2vec: Sequence-to-Sequence Autoencoding for Unsupervised Learning of Audio Segmentation and Representation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 9, pp. 1481-1493, 2019, doi: 10.1109/TASLP.2019.2922832.

[39] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," 2013.

[40] Y. Deng, L. Wang, H. Jia, X. Tong, and F. Li, "A Sequence-to-Sequence Deep Learning Architecture Based on Bidirectional GRU for Type Recognition and Time Location of Combined Power Quality Disturbance," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 8, pp. 4481-4493, 2019, doi: 10.1109/TII.2019.2895054.

[41] V. I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals," *Soviet Physics Doklady*, vol. 10, p. 707, 1966.

[42] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition using Smartphones," in *ESANN*, 2013.

[43] J. Reyes-Ortiz, D. Anguita, A. Ghio, L. Oneto, X. Parra, UC Irvine Repository. "Human Activity Recognition Using Smartphones," [Online]. https://archive.ics.uci.edu/dataset/240/human+activity+recognition+using+smartphones (Accessed Date: January 12, 2024).

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed to the presented research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

This publication has emanated from joint research conducted with the financial support of the National Key Research and Development Program of China under Grant No. 2017YFE0135700 and the Bulgarian National Science Fund (BNSF) under the Grant No. KP-06-IP-CHINA/1 (КП-06-ИП-КИТАЙ/1).

Conflict of Interest

The authors have no conflicts of interest to declare.