PPG-based Biometric Identification: Discovering and Identifying a New User

Yalan Ye¹, Guocheng Xiong¹, Zhengyi Wan¹, Tongjie Pan¹, Ziwei Huang¹

Abstract—The convenience of Photoplethysmography (PPG) signal acquisition from wearable devices makes it becomes a hot topic in biometric identification. A majority of studies focus on PPG biometric technology in a verification application rather than an identification application. Yet, in the identification application, it is an inevitable problem in discovering and identifying a new user. However, so far few works have investigated this problem. Existing approaches can only identify trained old users. Their identification model needs to be retrained when a new user joins, which reduces the identification accuracy. This work investigates the approach and performance of identifying both old users and new users on a deep neural network trained only by old users. We used a deep neural network as a feature extractor, and the distance from the feature vector to discover and identify a new user, which avoids retraining the identification model. On the BIDMC data set, we achieved an accuracy of more than 99% for old users, an accuracy of more than 90% for discovering a new user, and an average accuracy of about 90% for identifying a new user. Our proposed approach can accurately identify old users and has feasibility in discovering and identifying a new user without retraining in the identification application.

I. INTRODUCTION

Generally, PPG-based identification systems are used for two applications: the verification application (one-to-one matching), is used to prove someone’s identity by matching the acquired features only by the person’s stored template, whereas the identification application (one-to-many matching) is used to identify a user by comparing his/her template with all the stored templates [1].

For the verification application, Gu et al. [2] first proposed that PPG signals can be used for verification. Fourier analysis, fuzzy logic analysis, and linear discriminant analysis (LDA) [3] were used to process PPG signals, classification, and yielded an accuracy of more than 90%. For the identification application, the features extracted by discrete wavelet transform (DWT) were combined with support vector machine (SVM) to classify users in [4]. The study in [5] obtained three features by calculating the autocorrelation coefficients of raw PPG signal and its derivatives. K. Wang et al. [6] proposed a Crow Search Algorithm (CSA) to optimize the SVM classification model. Some researches [7] even extracted up to 40 features. The handcrafted feature extraction process is complicated and tedious.

Recently, an approach based on convolutional neural network (CNN) in conjunction with long and short-term memory (LSTM) neural network, proposed by L. Everson et al. [8], [9], was employed to move away from the complexity and instability of extracting handcrafted features. Based on this network structure, Dae Yon Hwang et al. [10] proposed PPSNet, achieve an average accuracy of 96% in a single session verification. They then evaluated the performance in two-sessions scenarios [11]. However, these approaches based on deep neural network learning have not been employed in the identification application. In addition, a trained neural network can only identify the trained users. When a new user joins, a model must be retrained for it.

Consequently, we envisaged an approach that can discover and identify a new user without retraining for the identification application. Fig. 1 shows this identification scenario. Templates of old users are added to the database during the training stage, and the model can discover a new user by matching the existing templates.

In this paper, our contribution is that we first proposed to use a deep network as a feature extractor, combined with the distance similarity to the cluster center to discover...
and identify a new user, which avoids the high time cost of retraining the model. Some similarities and differences between our work and previous studies are given in Table I. The performance of our work demonstrates that our proposed approach can accurately identify old users, and has feasibility in discovering and identifying a new user without retraining in the identification application.

II. METHOD

The overview of our proposed approach is given in Fig. 2. At the training stage, k users is selected as old users to train the deep network feature extractor, seek their cluster centers through clustering algorithm, so as to construct feature vectors and store them. Then, at the testing stage, features of a remaining new user are extracted through feature extractor. They are transformed into a feature vector according to their distance to cluster centers. Finally, similarity of two feature vectors is compared to identify whether it is an old user or a new user. More details are introduced below.

A. Preprocessing

A second-order Butterworth bandpass filter with a cutoff frequency 0.5Hz-5Hz was employed to eliminate noises. Then, we normalized the raw PPG signal to zero mean and unit variance. Systolic peak detection was applied to segment the raw PPG signal into individual heartbeats. Following this, the Pan Tompkins algorithm [13] is used to seek the correct peak value of a ppg signal. We utilized cubic spline interpolation to scale it to 128 samples and form a unified input format for deep neural networks.

B. Feature Extraction

A completely data-driven approach base on CNN in conjunction with LSTM, proposed by [8], [9], was adopted and modified by ourselves to automatically extract features from the ppg cycle sequence. Instead, they used it as a binary classifier for the verification application, while our idea is to train it as a feature extractor for the identification application. The implementation details are introduced below:

1) Model Design: A heartbeat cycle signal with 128 samples is served as the input of the network. Following the input layer are two 1-D convolutional layers consisted of a batch norm layer, a max-pooling layer, a dropout layer and a Rectified Linear Unit (ReLU) activation function. Two LSTM layers are then used to capture the temporal properties. Finally, a dense layer with 32 neurons is used to retain the most discriminate features for clustering.

2) Optimization: Specially, we train the model by adding center loss $L_c$ [14] with a balance parameter $\lambda$ to traditional softmax loss $L_s$. The new loss function $L = L_s + \lambda L_c$ is used to update the parameters. Traditional softmax loss $L_s$ is defined by the following equation:

$$L_s = -\frac{1}{m} \sum_{i=1}^{m} \log \frac{e^{W^T_{yi} x_{yi} + b_{yi}}}{\sum_{j=1}^{n} e^{W^T_{yj} x_{yj} + b_{yj}}}$$

(1)

where $x_i \in \mathbb{R}^d$ denotes the $i$th deep feature, belonging to the $y_i$th subject. $d$ is the feature dimension. $W_j \in \mathbb{R}^d$ denotes the $j$th column of the weights $W \in \mathbb{R}^{d \times n}$ in the last fully connected layer and $b \in \mathbb{R}^n$ is the bias term. The size of mini-batch and the number of class is $m$ and $n$, respectively. Center loss function $L_c$ is defined by the following equation:

$$L_c = \frac{1}{2} \sum_{i=1}^{m} \| x_i - c_{y_i} \|_2^2$$

(2)

where $c_{y_i} \in \mathbb{R}^d$ denotes the $y_i$th subject’s center of deep features. Root Mean Square Propagation (RMSProp) with the default hyperparameters is used to train the CNN network.

C. Feature Vector Construction

1) Clustering: The K-Means clustering algorithm was employed to fit the data. The center can be considered as a good representative of the class associated with each user in the feature space. A feature vector will be obtained at the output of the CNN for each user’s input PPG signal.

2) Transformation: The position of a sample in the $k$-dimensional space is determined by the vector consisting of its distance to the $k$ centers. Mahalanobis Distance was used as a distance metric to eliminate the interference of correlation between different features. The distance $D(x_j, c_{y_j})$ from each sample $x_j$ to the $y_j$th user center $c_{y_j}$ is calculated by the following equation:

$$D(x_j, c_{y_j}) = \sqrt{(x_j - c_{y_j})^T S_{y_j}^{-1} (x_j - c_{y_j})},$$

(3)

where $S_{y_j}^{-1}$ denotes the covariance matrix of the $y_j$th user.

The transformed feature vector $z_j$ is given by:

$$z_j = (D(x_j, c_{y_1}), D(x_j, c_{y_2}), \ldots, D(x_j, c_{y_k})).$$

(4)
D. Identification

A user-specific feature vector composed of the average distance of all samples was used for identification. The average distance $\bar{z}_i$ from the trained samples in the $y_i$ user to all centers $c_{y_k}$ is calculated as the feature vector of the class by:

$$\bar{z}_i = \left( \frac{1}{m_j} \sum_{j=1}^{m_j} D(z^{y_j}, c_{y_i}), \cdots, \frac{1}{m_j} \sum_{j=1}^{m_j} D(z^{y_j}, c_{y_k}) \right),$$

(5)

where $z^{y_j}$ denotes the $j$th sample in the $y_i$th user. $m_j$ denotes the total number of the $y_i$th user. All $\bar{z}_i$ will be included in the database at the training stage. The transformed feature vector from the test samples $\bar{z}$ is also calculated by equation 5. The adjusted cosine similarity is utilized to quantify the similarity between $\bar{z}$ and $\bar{z}_i$, which is defined by:

$$\cos(\bar{z}, \bar{z}_i) = \frac{(\bar{z} - \bar{\mu}) \cdot (\bar{z}_i - \bar{\mu})}{\|\bar{z} - \bar{\mu}\| \times \|\bar{z}_i - \bar{\mu}\|},$$

(6)

where $\bar{\mu}$ denotes the average vector of all $\bar{z}^{y_i}$ in the database. The cosine similarity has a range of $[-1, 1]$, where perfect label assignments produce a value of 1. We set 0.99 as the threshold value. The performance of this threshold is discussed in the next section. When the maximum similarity between a certain $\bar{z}_y$ and $\bar{z}$ exceeds the threshold, this sample will be identified as the $y_i$th user. If there is no such $\bar{z}_y$, the sample will be identified as a new user.

III. RESULTS AND DISCUSSION

The dataset used in this work, which is called BIDMC, was obtained from [15], [16]. The 53 recordings containing physiological signals and physiological parameters were collected respectively for 8 minutes at a sampling rate of 125 Hz and 1 Hz. The first $k$ recordings from 53 users were selected to train the feature extractor. For each user, 80% was randomly selected for training and the remaining 20% was used for testing. We conducted the following experiments in Python using Pytorch.

<table>
<thead>
<tr>
<th>TABLE II</th>
<th>COMPARE IDENTIFICATION RESULT ON OLD USERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref</td>
<td>Subjects</td>
</tr>
<tr>
<td>[8]</td>
<td>12</td>
</tr>
<tr>
<td>[5]</td>
<td>38</td>
</tr>
<tr>
<td>[6]</td>
<td>50</td>
</tr>
<tr>
<td>Ours</td>
<td>12</td>
</tr>
</tbody>
</table>

A. Testing stage for identifying old users

We trained the network to get the feature extractor in the case of $k = 12$ and examined its performance. Table II compares identification results in old users among our approach and state of the art works. We achieved the average accuracy of 99.72% for these 12 users based on the clustering algorithm and the distance similarity. Fig. 3 (a) visualizes the results of clustering after dimensionality reduction through t-SNE. The adjusted cosine similarity matrix between the feature vector on the training set and the test set is given in Fig. 3 (b), which indicates that our approach can achieve accurate recognition on old users.

B. Testing stage for discovering a new user

We extracted 3, 6, or the remaining 41 new users’ features to evaluate the ability to discover a new user in our approach. Fig. 4 (a) and (b) respectively introduce the t-SNE feature distribution of 3 and 6 new users extracted by the feature extractor trained by 12 old users. Although the performance in identifying 6 new users is worse than that in 3 new users (affected by the trained feature extractor), this result proves that the feature extractor has the ability to distinguish features of a new user from old users, which makes it possible to discover a new user.

C. Discussion

Five deep neural networks ($k$=6, 9, 12, 15, 18) were trained to explore the factors that impact the performance of our approach. The results in Table III show the accuracies of 5 networks on the test set, their respective training time, the accuracies of discovering a new user and identifying a
new user. Training with more users will greatly increase the training time but can’t yield a promising performance. We then evaluated the performance of different thresholds on the same deep neural network. Fig. 5 shows the performance of different thresholds when there are 12 old users \((k=12)\). The 0.99 used in this work is the value that yielded the best performance in the experiment.

IV. CONCLUSIONS

In this paper, a semi-supervised approach combining deep learning, clustering algorithm and distance similarity was proposed to discover and identify a new user. We trained the CNN+LSTM network as the feature extractor, identified old users and new users based on Mahalanobis distance and cosine similarity, which can avoid retraining the feature extractor when a new user joins. We evaluated the performance of discovering and identifying a new user on different feature extractor models. The results demonstrate that our proposed approach can accurately identify old users, and has feasibility in discovering and identifying a new user without retraining in the identification application. Further exploration is to automatically select the most suitable trained old user \(k\) and classification threshold for better performance.

REFERENCES


TABLE III

<table>
<thead>
<tr>
<th>Number of Old Users((k))</th>
<th>Training Time(min)</th>
<th>Old Users</th>
<th>Discover A New User</th>
<th>Identify A New User</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>18</td>
<td>99.664</td>
<td>93.62</td>
<td>87.32</td>
</tr>
<tr>
<td>9</td>
<td>73</td>
<td>99.443</td>
<td>100</td>
<td>93.18</td>
</tr>
<tr>
<td>12</td>
<td>177</td>
<td>97.162</td>
<td>95.12</td>
<td>92.69</td>
</tr>
<tr>
<td>15</td>
<td>324</td>
<td>91.196</td>
<td>100</td>
<td>94.74</td>
</tr>
<tr>
<td>18</td>
<td>589</td>
<td>88.907</td>
<td>97.14</td>
<td>91.43</td>
</tr>
</tbody>
</table>
* The values in the last three columns are percentages.
* “Discover A New User” denotes the percentage of an untrained user discovered as a new user.
* “Identify A New User” denotes the average accuracy of reidentifying a new user on the expanded database. He/she was regarded as a trained user since his/her template has already been added to the database.