IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 21, 2020, accepted January 25, 2021, date of publication February 3, 2021, date of current version February 10, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3056880

Multi-Model Long Short-Term Memory Network for Gait Recognition Using Window-Based Data Segment

LAM TRAN[1], THANG HOANG[2], THUC NGUYEN[3], HYUNIL KIM[4], AND DEOKJAI CHOI[1]

[1]Department of Artificial Intelligence Convergence, Chonnam National University, Gwangju 61186, South Korea
[2]Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33612, USA
[3]Department of Computer Science, Ho Chi Minh University of Science, Ho Chi Minh City 700000, Vietnam
[4]Department of Robotics Engineering, Daegu Gyeongbuk Institute of Science & Technology (DGIST), Daegu 42988, South Korea

Corresponding author: Lam Tran (halam189@gmail.com)

This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant by the Korean Government, Ministry of Science and ICT (MSIT) (Research on AI-Based Cryptanalysis and Security Evaluation) under Grant 2020-0-00126, and in part by the Vietnam National University (VNU-HCM) under Grant NCM2019-18-01.

ABSTRACT Inertial Measurement Units (IMUs)-based gait analysis is a promising and attractive approach for user recognition. Recently, the adoption of deep learning techniques has gained significant performance improvement. However, most existing studies focused on exploiting the spatial information of gait data (using Convolutional Neural Network (CNN)) while the temporal part received little attention. In this study, we propose a new multi-model Long Short-term Memory (LSTM) network for learning the gait temporal features. *First*, we observe that LSTM is able to capture the pattern hidden inside the gait data sequences that are out-of-synchronization. Thus, instead of using the gait cycle-based segment, our model accepts the gait cycle-free segment (*i.e.,* fixed-length window) as the input. By this, the classification task does not depend on the gait cycle detection task, which usually suffers from noise and bias. *Second*, we propose a new LSTM network architecture, in which, one LSTM is used for each gait data channel and a group of consecutive signals is processed in each step. This strategy allows the network to effectively handle the long input data sequence and achieve improved performance compared to existing LSTM-based gait models. In addition, besides using the LSTM alone, we extend it by combining with a CNN model to construct a hybrid network, which further improves the recognition performance. We evaluated our LSTM and hybrid networks under different settings using the whuGAIT and OU-ISIR datasets. The experiments showed that our LSTM network outperformed the existing LSTM networks, and its combination with CNN established new state-of-the-art performance on both the verification and identification tasks.

INDEX TERMS Gait authentication, gait recognition, wearable sensor data, recurrent neural network, LSTM network.

I. INTRODUCTION

Gait has been discovered to contain individual unique features that could be used for user recognition. The prior gait recognition researches relied on computer vision [1]–[5] or floor sensor techniques [6]–[8]. Such approaches are applicable for video surveillance or security access control in a specific area (*e.g.,* building entrance, airport checkpoint) [9], [10]. The recent evolution of Micro Electro Mechanical technique opens a new gait recognition method, which leverages the Inertial Measurement Units (IMUs) attached to the human body to obtain the gait data [11]. Different from the original methods, the IMUs-based approach is promising for user identification/authentication on mobile devices (*e.g.,* smartphone, smartwatch) which presents various positive features as low-cost, mobility, unobtrusive, light-weight, ubiquitous [9], [12]. Thus, IMUs-based gait recognition has become an active research topic over the last decade.

Despite the merit of existing researches, IMUs-based gait recognition is still a challenging and ongoing topic [9], [13]. Gait is a special biometric modality that contains both spatial and temporal information [14]. However,

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

major efforts in recent studies focused on using Convolutional Neural Network (CNN) [15] to extract the spatial information (*e.g.,* [16]–[20]), while there was little attention for the temporal part. Some latest studies adopted Long Short-term Memory (LSTM) network [21] to capture the temporal information from the gait data sequence and showed promising results [19], [22]. However, we observe that there are two limitations in these models. *First*, they used gait cycle-based segment as the input, where gait cycle is the time interval between two consecutive ground touching events of the same foot [12]. However, gait cycle detection is also a challenging task, which is noise-sensitive and device setting-dependent (*e.g.,* device position in user body, sensor model) [23]. In such models, an incorrect gait cycle detection usually leads to a false recognition decision. *Second*, they did not pay attention to the shortcoming of LSTM in the sequence classification task. Specifically, even though LSTM is able to capture the correlation between data points across a sequence, the classification task with LSTM (*i.e.,* many-to-one task) still suffers from the long input sequence [24].

In this study, we propose a new multi-model LSTM network for gait recognition which introduces two innovations comparing to existing works [19], [22]. *First*, instead of using gait cycle-based segments as the input, our model accepts fixed-length segments, which are extracted independently from the gait cycle. We observe that, unlike the handcrafted methods, LSTM can automatically detect and extract the hidden features from data segments that start at different walking phases. Thus, for the LSTM model, there is no need for the input segments to be well-aligned as using the handcrafted methods. *Second*, we propose a new LSTM network architecture which can effectively extract the meaningful features from long input data sequence. Specifically, instead of giving one signal data to each step of LSTM, we input a group of continuous signals and process them simultaneously (see Figure 2b). In this way, the number of LSTM steps could be reduced without losing the gait information. We show that, such architecture achieves improved performance comparing to processing one signal at each step.

In summary, the contributions of this study are:

- We propose a new LSTM network architecture for IMUs-based gait recognition (Section III). Our LSTM network uses free segment as the input to simplify the segmentation step, and be independent from the gait cycle detection task. Then, each gait data channel is proceeded separately to fully exploit the LSTM potential, and a group of signals is input at each step to overcome the difficulty of the long input sequence.
- We extend our network by combining with the state-of-the-art CNN network [18] to construct a hybrid architecture which achieves the improved performance comparing to using only LSTM or CNN (Section IV-D).
- We further extend the proposed network to build the verification model with triplet loss [25] and One-class Support Vector Machine (OCSVM) [26]. We show that our proposed LSTM network is also effective for the verification task (Section IV-E).
- We conduct comprehensive experiments using two public gait datasets (Section IV). The first dataset is OU-ISIR - the largest population gait dataset [27] which includes data of 744 users. The second dataset is whuGAIT, which comprises data of 118 users collected in the wild conditions [19]. We show that our LSTM network outperforms existing LSTM networks. In addition, its combination with CNN established new state-of-the-art performance on both user verification and identification tasks.

We organize the remaining parts of the paper as follows. We provide a review of the existing gait recognition researches in Section II. Then, we describe in detail the proposed LSTM model in Section III. The experiment and evaluation are presented in Section IV. Finally, we summarize this study by a conclusion in Section V.

II. RELATED WORK

In this section, we briefly review the gait recognition researches. First, we summary different gait recognition approaches in Section II-A. Then, we present in detail the use of LSTM for IMU-based approach in Section II-B.

A. GAIT RECOGNITION

Based on the data acquiring method, gait recognition researches could be categorized into 3 approaches as computer vision (*e.g.,* [1]–[5], [28], floor sensor (*e.g.,* [6]–[8]), and IMUs (*e.g.,* [11]–[13], [19]). The computer vision and floor sensor approaches were the active researches in the early time. In the computer vision model, a camera located on a specific place (*e.g.,* building's entrance) is used to capture a video of passing user. The acquired video is then used to identify a user with some computer vision techniques. The state-of-the-art researches on this approach were summarized in [29], [30]. On the other hand, the floor sensor-based approach uses the sensors placed on the floor to capture the foot pressures when the user walking on. Comparing to some common biometric traits (*e.g.,* face, fingerprint, iris), the computer vision- and floor sensor-based models could capture the users' data without their notice or interaction, thus enable transparent identification [9]. So, these methods are promising for the task of video surveillance or security access control in a specific area.

Recently, the evolution of Micro Electro Mechanical techniques allows the Inertial Measurement Units (*e.g.,* accelerometer, gyroscope) to be embedded in common mobile devices (*e.g.,* smartphone, smartwatch). This technology enables a new gait recognition approach, which uses the mobile devices (with the embedded sensors inside) attached to the human body to obtain the gait data [12]. This approach is promising for user authentication on mobile devices, which offers many favorable properties as low-cost, lightweight, portability, ubiquitous, and transparency.

The first IMUs-based gait recognition model was proposed by Alisto *et al.* [11] in 2005. The gait data sequence was captured by an accelerometer, and split into walking step-based segments. The data in X and Z axes of each segment were used to calculate the correlation between the probe and gallery segments. The model achieved the FAR of 6.4% and FRR of 5.4% for a dataset of 36 users. After that, this field received more research attention, and a large number of gait recognition models have been proposed in the literature [9], [13], [16]–[20], [31].

Most prior IMUs-based gait recognition models used handcrafted methods for feature extraction, which could be summarized in [12]. The later researches adopted deep learning techniques to automatically extract more discriminative and stable features, which showed the improved performance [17]–[20], [22]. In addition, some researches improved the recognition performance by combining multiples sensors [32], [33].

B. IMUs-BASED GAIT RECOGNITION WITH LSTM

Although Recurrent Neural Network (RNN) techniques (*i.e.,* LSTM [21], GRU [34]) are powerful for sequential data, only a few studies applied them for gait recognition [19], [22]. The study [22] was the first one that adopted LSTM to extract the hidden features from a gait cycle's data (both accelerometer and gyroscope) to be used for user authentication. A grid search was used to determine the best parameters set. They evaluated their method on the OU-ISIR dataset and achieved the EER of 7.55% when using 1 gait cycle to verify the attempting user. In [19], the authors proposed 2 hybrid networks (*i.e.,* the combination of CNN and LSTM), one for user authentication and one for identification. The user identification network consisted of two branches (one CNN and one LSTM) processed in parallel. The features extracted from two branches were then concatenated, and used to identify the user by a fully connected (FC) network. In the user authentication network, CNN was used to transforms the gait cycle's data to some hidden features which were then used as the input of LSTM.

Both of those studies require the input as gait cycle-based segments, thus, they are strongly impacted by the adopted gait cycle detection algorithm. In addition, in each step of LSTM, data of all dimensions captured at a specific time were used as the input. This fact causes the learning process unstable when using a long input sequence. In this study, we propose a new LSTM network architecture, which accepts a gait cycle-free segment as the input, and can handle the long input sequence effectively. The experiment results showed that the performance of our proposed network surpassed the existing LSTM models.

III. LSTM-BASED GAIT RECOGNITION FRAMEWORK

In this section, we present the proposed multi-model LSTM network for IMU-based gait recognition. First, we describe in section III-A the data preprocessing and segmentation to form the network's input. Then, we explain in detail the LSTM network architecture in section III-B.

A. DATA PREPROCESSING AND SEGMENTATION

The input data for gait recognition are the sequences of signals generated by the accelerometer and gyroscope sensors. Each acceleration signal is recorded as a vector of 3-dimensional $\mathbf{a} = \left[a^X\ a^Y\ a^Z\right]$, where a^X, a^Y and a^Z are the acceleration forces acting along the X, Y, and Z axes, respectively. Similarly, each gyroscope signal $\mathbf{g}$ is a 3-dimensional vector $\mathbf{g} = \left[g^X\ g^Y\ g^Z\right]$, where g^X, g^Y, g^Z represent the rotation rates around the X, Y, and Z axes, respectively. Due to the asynchrony between accelerometer and gyroscope sensors, their signals may not be sampled simultaneously. Therefore, we apply the spline interpolation technique [35], to normalize the raw gyroscope sequence so that each interpolated gyroscope signal is yielded simultaneously with the corresponding acceleration signal. Then, the acceleration and gyroscope sequences are combined to one data stream of 6-channel

$$\mathbf{S} = \left[\mathbf{s}_1\ \mathbf{s}_2\ \ldots\ \mathbf{s}_n\right], \tag{1}$$

where each element $\mathbf{s}_j$ $(1 \leq j \leq n)$ is a 6-dimensional vector $\mathbf{s}_j = \left[a_j^X\ a_j^Y\ a_j^Z\ g_j^X\ g_j^Y\ g_j^Z\right]^\top$.

After that, $\mathbf{S}$ is split into fixed-length segments $\mathbf{S}^{(i)} \in \mathbb{R}^{6\times N_c}$, each segment contains N_c consecutive signals, where N_c is chosen so that each segment contains at least one gait cycle. Specifically, let f_w be the walking frequency of the user (*i.e.,* f_w is the number of steps performed in one second). Let f_s be the signal sampling frequency of the sensors. Then, N_c is chosen so that

$$N_c \geq 2\frac{f_s}{f_w}. \tag{2}$$

According to [36], people usually walk with the f_w between 1.8 and 2 Hz, thus, N_c could be determined as

$$N_c = \alpha f_s, \tag{3}$$

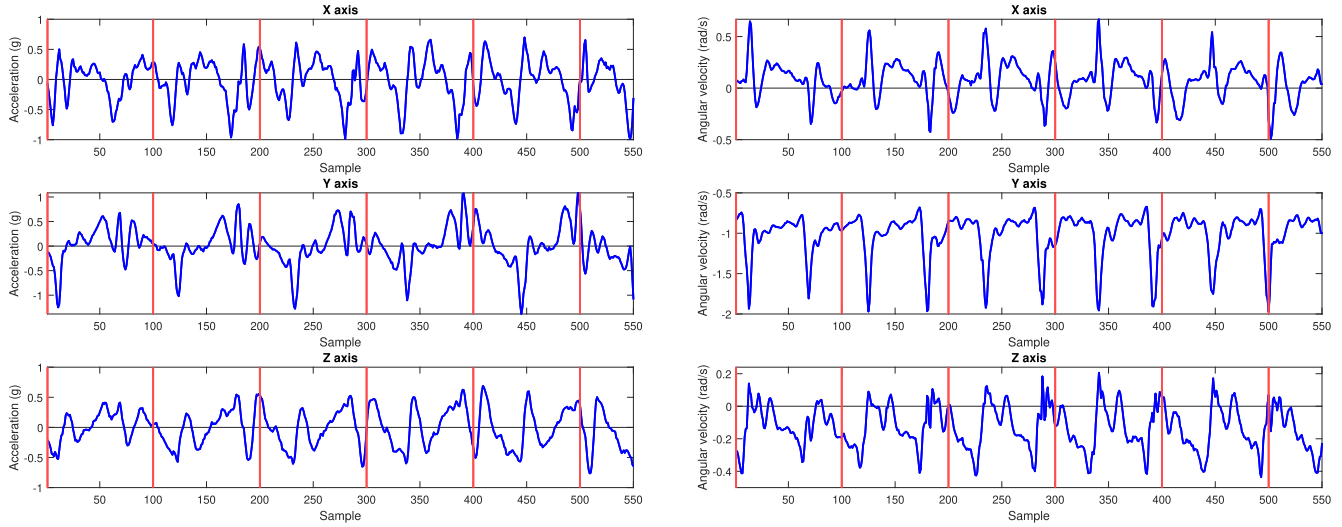
where $\alpha \geq 1.0$ is a user-selected parameter. Figure 1 illustrates an example of the accelerometer and gyroscope data sequences captured with the sampling rate of 100 Hz, where the red vertical lines denote the borders of the gait segments split with $\alpha = 1$. Note that, to increase the amount of training data, two consecutive training segments $\mathbf{S}^{(i)}$ and $\mathbf{S}^{(i+1)}$ overlap each other θ%. For the testing data, there is no overlapping portion between the segments.

From here, we denote $\mathbf{S}^{(i)}$ as the matrix consisting data of entire gait segment i. We mean $\mathbf{s}_j^{(i)}$ as the data vector of channel j of gait segment i. And $s_{j,k}^{(i)}$ represents the value of signal k of channel j in segment i.

B. THE MULTI-MODEL LSTM NETWORK

Overall, the network accepts a gait segment $\mathbf{S}^{(i)}$ as the input, and computes an output vector

$$\hat{\mathbf{y}}^{(i)} = \left[\hat{y}_0^{(i)}\ \hat{y}_1^{(i)}\ \ldots\ \hat{y}_{N_u-1}^{(i)}\right], \tag{4}$$



(a) Acceleration signals

(b) Gyroscope signals

FIGURE 1. An example of gait data signals captured with the sampling rate of 100 Hz. The red vertical lines denote the borders of gait segments (each one has 100 gait signals) which are determined independently with the gait cycle.

where N_u is the number of users, and $\hat{y}_u^{(i)}$ represents the probability of segment $\mathbf{S}^{(i)}$ belong to the user u, $(0 \leq u \leq N_{u-1})$. The architecture and computation of the network are described as follows.

1) THE OVERALL ARCHITECTURE

Figure 2 depicts the overall network architecture which could be divided into two parts:

- *The feature extractor*: It consists of 6 LSTMs, each processes a gait data channel $\mathbf{s}_j^{(i)}$ and outputs a vector $\mathbf{f}_j^{(i)}$ of length H. Then, the output of all channels are concatenated to form a unique vector $\mathbf{f}^{(i)} = \left[\mathbf{f}_1^{(i)}\ \mathbf{f}_2^{(i)}\ \mathbf{f}_3^{(i)}\ \mathbf{f}_4^{(i)}\ \mathbf{f}_5^{(i)}\ \mathbf{f}_6^{(i)}\right]$ which is then used as the input of the classifier. The detailed structure and processing of each LSTM network are described in Section III-B2.
- *The classifier*: It is a fully connected (FC) layer which predicts the user identity given the feature vector $\mathbf{f}^{(i)}$. Specifically, for a model of N_u users, the FC layer maps the feature vector $\mathbf{f}^{(i)}$ to the output vector $\hat{\mathbf{y}}^{(i)}$ of length N_u. Each element of $\hat{\mathbf{y}}^{(i)}$ is computed from all elements of $\mathbf{f}^{(i)}$, followed by the log softmax function as the activator. To avoid overfitting and improve the generalization, we employ dropout with the rate of 0.5 in this layer. Specifically, in the training phase, 50% nodes of $\mathbf{f}^{(i)}$ are dropped along with their connections to prevent co-adapting too much and help the remaining network parameters to be adjusted more effective.

2) THE LSTM STRUCTURE

Figure 2b depicts the detailed processing of each LSTM network, which accepts a gait channel $\mathbf{s}_j^{(i)}$ as the input, and outputs a feature vector $\mathbf{f}_j^{(i)}$ of length H $(1 \leq j \leq 6)$.

The LSTM network processes the input sequence through several steps, each step has L layers. In this study, instead of inputting one signal for each step as the existing methods [19], [22], we use a group of K $(N_c \mid K)$ consecutive signals. Specifically, given a gait channel sequence $\mathbf{s}_j^{(i)}$ of length N_c, the LSTM processes $\mathbf{s}_j^{(i)}$ in $T = \frac{N_c}{K}$ steps. In the step t $(1 \leq t \leq T)$, each layer l $(1 \leq l \leq L)$ of the LSTM cell gets the input $\mathbf{x}_t^{(l)}$ and the outputs of step $t-1$ (*i.e.,* the hidden state $\mathbf{h}_{t-1}^{(l)}$, the cell state $\mathbf{c}_{t-1}^{(l)}$) to compute the new hidden state $\mathbf{h}_t^{(l)}$ and the cell state $\mathbf{c}_t^{(l)}$. The new cell state $\mathbf{c}_t^{(l)}$ is updated as follows:

$$\begin{aligned}
\mathbf{f}_t^{(l)} &= \sigma(\mathbf{w}_{if}^{(l)}\mathbf{x}_t^{(l)} + \mathbf{b}_{if}^{(l)} + \mathbf{w}_{hf}^{(l)}\mathbf{h}_{t-1}^{(l)} + \mathbf{b}_{hf}^{(l)}),\\
\mathbf{i}_t^{(l)} &= \sigma(\mathbf{w}_{ii}^{(l)}\mathbf{x}_t^{(l)} + \mathbf{b}_{ii}^{(l)} + \mathbf{w}_{hi}^{(l)}\mathbf{h}_{t-1}^{(l)} + \mathbf{b}_{hi}^{(l)}),\\
\mathbf{g}_t^{(l)} &= \tanh(\mathbf{w}_{ig}^{(l)}\mathbf{x}_t^{(l)} + \mathbf{b}_{ig}^{(l)} + \mathbf{w}_{hg}^{(l)}\mathbf{h}_{t-1}^{(l)} + \mathbf{b}_{hg}^{(l)}),\\
\mathbf{c}_t^{(l)} &= \mathbf{f}_t^{(l)} \circ \mathbf{c}_{t-1}^{(l)} + \mathbf{i}_t^{(l)} \circ \mathbf{g}_t^{(l)},
\end{aligned} \tag{5}$$

where $\circ$ is the Hamadard product, and σ is a logistic sigmoid function. Then, the hidden state $\mathbf{h}_t^{(l)}$ is determined from the cell state $\mathbf{c}_t^{(l)}$

$$\begin{aligned}
\mathbf{o}_t^{(l)} &= \sigma(\mathbf{w}_{io}^{(l)}\mathbf{x}_t^{(l)} + \mathbf{b}_{io}^{(l)} + \mathbf{w}_{ho}^{(l)}\mathbf{h}_{t-1}^{(l)} + \mathbf{b}_{ho}^{(l)}),\\
\mathbf{h}_t^{(l)} &= \mathbf{o}_t^{(l)} \circ \tanh(\mathbf{c}_t^{(l)}).
\end{aligned} \tag{6}$$

In equations (5) and (6), the weight matrices (*i.e.,* $\mathbf{w}_{if}^{(l)}$, $\mathbf{w}_{hf}^{(l)}$, $\mathbf{w}_{ii}^{(l)}$, $\mathbf{w}_{hi}^{(l)}$, $\mathbf{w}_{ig}^{(l)}$, $\mathbf{w}_{hg}^{(l)}$, $\mathbf{w}_{io}^{(l)}$, $\mathbf{w}_{ho}^{(l)}$) and the bias (*i.e.,* $\mathbf{b}_{if}^{(l)}$, $\mathbf{b}_{hf}^{(l)}$, $\mathbf{b}_{ii}^{(l)}$, $\mathbf{b}_{hi}^{(l)}$, $\mathbf{b}_{ig}^{(l)}$, $\mathbf{b}_{hg}^{(l)}$, $\mathbf{b}_{io}^{(l)}$, $\mathbf{b}_{ho}^{(l)}$) are the parameters of layer l that need to be updated through the learning process. Here, when $t = 1$, the cell state $\mathbf{c}_0^{(l)}$ and $\mathbf{h}_0^{(l)}$ are initialized randomly. The input data $\mathbf{x}_t^{(l)}$ is determined by

$$\mathbf{x}_t^{(l)} = \begin{cases} \left[s_{j,\,K(t-1)+1}^{(i)} \ \cdots \ s_{j,\,Kt}^{(i)}\right] & \text{if } l = 1,\\ \mathbf{h}_t^{(l-1)} & \text{otherwise.} \end{cases} \tag{7}$$

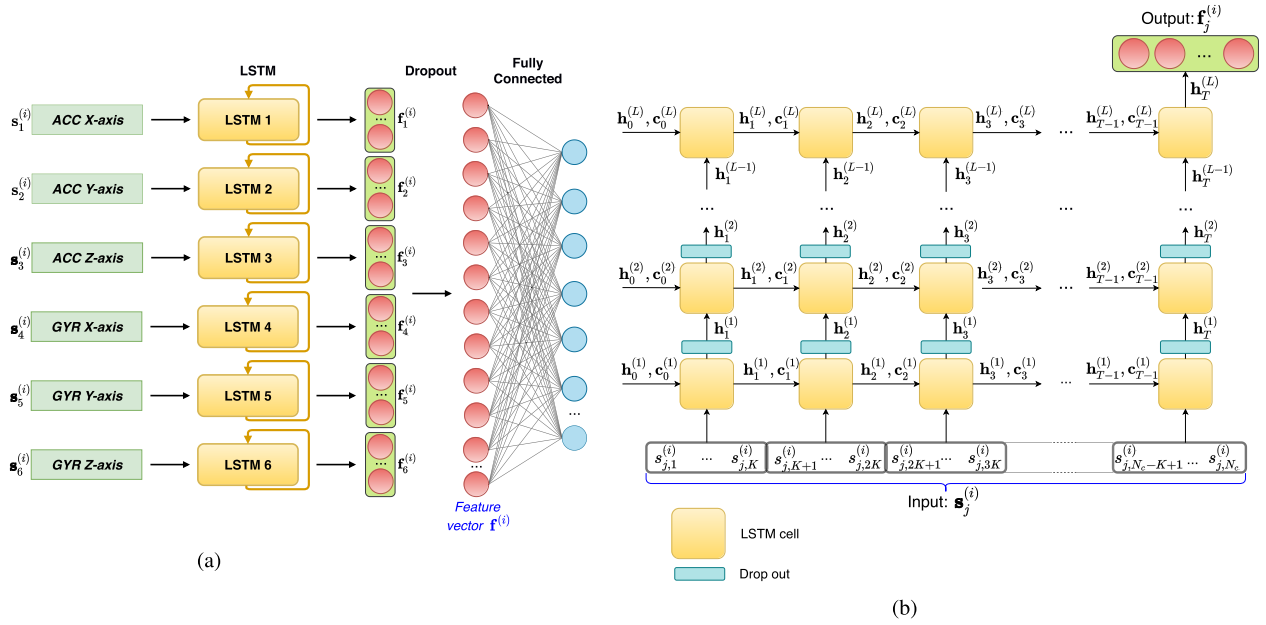


FIGURE 2. The multi-model LSTM gait recognition network. (a) The overall architecture of our gait recognition model consists of 6 LSTMs, each one extracts features from a gait data channel separately. (b) The detailed processing of each LSTM network.

That is, in the first layer of each step t, K consecutive elements of the gait segments are used as the input vector; in the l^{th} layer (*i.e.,* $1 < l \leq L$), the output vector of the lower layer (*i.e.,* $\mathbf{h}_t^{(l-1)}$) is input.

In case of multi-layer LSTM (*i.e.,* $L \geq 2$), we adopt dropout 50% for all the layers excepting the first one, to prevent the training model from overfitting [37]. Specifically, in the training phase, at each layer $l > 1$), 50% elements (selected randomly) of the hidden features $\mathbf{h}_t^{(l-1)}$ are zeroed out before inputting to layer l. Note that, in the testing phase, dropout is not applied, that means, all elements of the hidden features extracted from lower layer are input to the upper layer.

In this study, the LSTM network is used for the classification task, thus, it follows the many-to-one architecture. That means, the hidden state of the last layer outputted in the last step is used as the output of the network, *i.e.,* $\mathbf{f}_j^{(i)} = \mathbf{h}_T^{(L)}$.

3) TRAINING AND OPTIMIZATION

We initialize the network's parameters using the Kaiming He's method [38], then train them with the gradient descent algorithm to adjust them iteratively. Let $\mathcal{L}$ be the set of training gait segments and N_l be the size of this set (*i.e.,* $N_l = |\mathcal{L}|$). In the training phase, each segment $\mathbf{S}^{(i)} \in \mathcal{L}$ is labeled with its user owner's identity $u^{(i)}$, $(0 \leq u^{(i)} \leq N_u - 1)$. The network is trained repeatedly through a number of epochs. In each epoch, the training set $\mathcal{L}$ is randomly shuffled and divided into several batches $\mathcal{B}$, each batch $\mathcal{B}$ has B segments (B is usually referred as batch size). With each batch $\mathcal{B}$, the network computes the set of output vectors $\mathcal{Y}$, where each vector $\hat{\mathbf{y}}^{(i)} \in \mathcal{Y}$ gives a prediction for the label of gait segment $\mathbf{S}^{(i)} \in \mathcal{B}$. The negative log-likelihood (NLL) loss is then computed over the batch $\mathcal{B}$ as follows

$$\text{NLL}_{(\mathcal{B})} = -\frac{1}{B} \sum_{\hat{\mathbf{y}}^{(i)} \in \mathcal{Y}} \log(\hat{y}_{u^{(i)}}^{(i)}), \tag{8}$$

where $\hat{y}_{u^{(i)}}^{(i)}$ is the element $u^{(i)}$ of the output vector $\hat{\mathbf{y}}^{(i)}$. Then, $\text{NLL}_{(\mathcal{B})}$ is used to update network parameters following the back propagation procedure [39]. When all the batches have been used to train the network, one training epoch is completed, then, the process is repeated with a new epoch until it meets a stopping condition (see Section IV-A).

IV. EXPERIMENTS

In this section, we present the experiment and evaluation for the proposed method. First, we describe the datasets and experimental settings. Then, we report the results analyzed under different parameters and network architectures (*i.e.,* pure LSTM, the combination of LSTM and CNN). Finally, we provide a comparison between our study and the existing methods.

A. DATASET AND SETTING

Our experiment was conducted using PyTorch 1.6.0 framework running on Python 3.6.9 and Ubuntu 18.04.3. The used computer was equipped with the Intel(R) Xeon(R) Gold 6126 (2.6 GHz, 8 cores) processor, 16 GB of RAM memory, and the NVIDIA Tesla V100 16 GB GPU. The proposed model was evaluated on two public datasets as the OU-ISIR [27], and whuGAIT datasets [19]. The experiment procedure for each dataset is performed as follows. All the source code and trained models were published in GitHub[1] to facilitate the future research.

[1] https://github.com/halam189/Gait_Recognition_LSTM.git

1) THE OU-ISIR DATASET

OU-ISIR is considered as the largest population IMUs-based gait dataset that has been published so far. This dataset is organized into 2 subsets. The first one contains data of 744 users, collected by one IMU placed in the middle of the user's back waist. The second subset consists of data of 408 users, acquired by 3 IMUs placed on the left, right and center of back waist. In this study, we used the first subset as we aimed to evaluate the proposed method on a large number of users.

For each user in the dataset, we divided his/her data into 2 parts, to be used for training and testing, respectively. Then, each sequence in each part was split into a set of gait segments with $\alpha = 1.0$, thus, each segment has $N_c = 100$ gait signals. For the training sequences, two consecutive segments overlapped each other $\theta = 97\%$. There was no overlapping between the testing segments. The segments extracted from the training sequence were used to form the training set $\mathcal{L}$. Some segments from the testing sequence (*i.e.,* 20% of each user) formed the validating set $\mathcal{V}$, and the remaining formed the testing set $\mathcal{T}$.

The network was trained by the Stochastic Gradient Descent algorithm with the learning rate $r = 0.15$, momentum $m = 0.9$, and batch size $B = 64$. The set $\mathcal{V}$ was used for early stopping when training the network. Specifically, when completing a training epoch, the loss $\mathsf{NLL}_{(\mathcal{V})}$ over the validation set $\mathcal{V}$ was computed. If $\mathsf{NLL}_{(\mathcal{V})}$ did not decrease during 15 epochs, we terminated the training process. The set $\mathcal{T}$ was used for evaluating the trained network's performance. Specifically, each segment $\mathbf{s}^{(i)}$ in $\mathcal{T}$ was input to the trained network to get the predicted label $\hat{u}^{(i)}$. Let C be the number of correct classifying cases (*i.e.,* $\hat{u}^{(i)} = u^{(i)}$). Then, the accuracy ACC was identified as:

$$ACC = \frac{100 \times C}{|\mathcal{T}|}. \tag{9}$$

2) THE whuGAIT DATASET

WhuGAIT is a dataset of 118 users collected in the wild conditions, conducted by Wuhan University. This dataset was organized into 8 subsets, each one was used for a specific experiment in the original paper. In this study, we used the subset #3, which comprises of training and testing sets, each one has data of all 118 users. In the original work, the gait data sequences were split into fixed-length segments, in which, each segment consists of gait signals of 2.68 seconds (*i.e.,* 128 signals), and 2 consecutive segments overlap 1.28 seconds (*i.e.,* 64 signals). Thus, our first step is to concatenate the given segments according to the order of acquisition time, then remove the overlap to reform the raw gait sequence.

After that, the experimental procedure was similar to the OU-ISIR dataset, excepting some differences in the data segmentation and dataset dividing as follows. In the segmentation step, we split the reformed sequence into gait segments so that each one contained 80 gait signals (*i.e.,* $\alpha = 1.6$). The value of α used in this dataset was higher than in OU-ISIR because the sampling rate for collecting OU-ISIR dataset was 100 Hz, higher than in whuGAIT dataset (50 Hz). Due to the low sampling rate, some information may be missed in each gait cycle, thus, to complement the missing parts, we increased the value of α. To facilitate the comparison, we used the training/testing set divided from the original study for training/testing our networks. The training segments (extracted from the training set) also overlapped each other 97%, and the segments for testing had no overlapping portion. For each user in the dataset, we randomly selected 1500 segments extracted from the training set, to be used for training the model. The remaining segments were used to form the validating set $\mathcal{V}$, to terminate the training process. All the segments extracted from the testing set were used to evaluate the model's accuracy.

TABLE 1. The identification accuracy (%) of the LSTM network when using optimal parameters.

Dataset	*Group size*	*Hidden size*	*Number of layers*	*Accuracy*
OU-ISIR	10	40	2	78.92 %
whuGAIT	4	40	2	93.14 %

B. RESULT

The optimal identification accuracy experienced in each dataset and the used parameters are presented in Table 1. The model achieved the best accuracy of 78.92% for the OU-ISIR dataset, and 93.14% for the whuGAIT dataset. For both of the datasets, the optimal setting of LSTM layer number (*i.e., L*) was 2, and the optimal hidden size was 40. However, for the input group size, different settings were used (*i.e.,* $K = 10$ for OU-ISIR dataset, and $K = 4$ for whuGAIT dataset). The reason for this inconsistency is the difference of sampling rates f_s used in each dataset (*i.e.,* 100 Hz in OU-ISIR, 50 Hz in whuGAIT dataset). Specifically, with a higher sampling rate, a larger number of signals will be acquired in a certain time period, thus the input gait data sequence gets longer. So, the value of K needs to be increased accordingly to adjust the number of steps T to an optimal value.

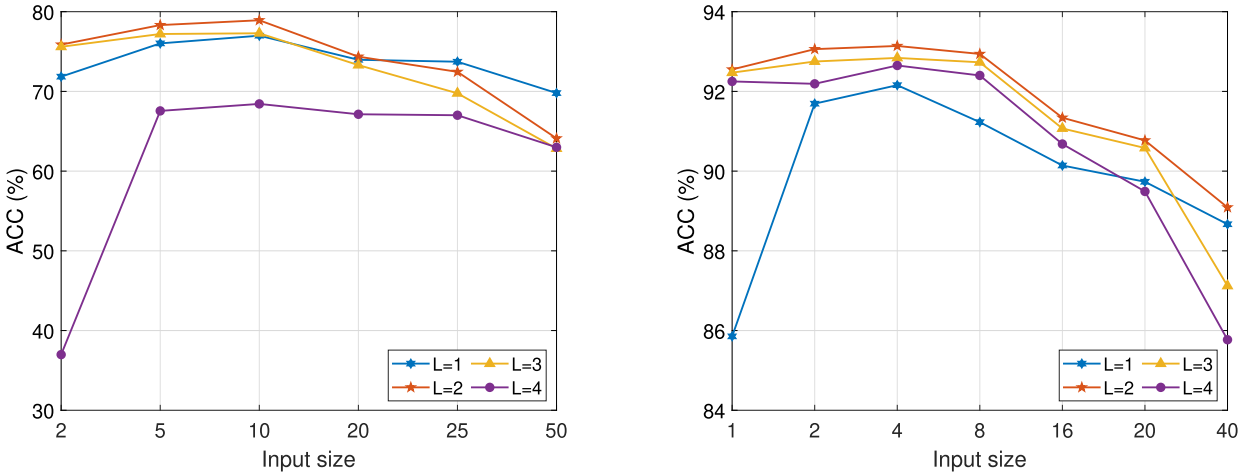
C. DISCUSSION

In this section, we analyze the impacts of the input data (*i.e.,* gait cycle-based or fixed-length segment) and the important parameters as the input group size K, the hidden size H, and the number of LSTM layer L to the recognition accuracy.

1) SEGMENTATION METHOD

We utilized the datasets #1 and #3 in the study [19] to analyze the impact of segmentation method (*i.e.,* fixed-length window or gait cycle). We experimented our LSTM model under three cases:

- *(i) Cycle-based segmentation (overlapping 50%)*: The model was trained with the dataset #1. This dataset contained gait segments of 118 users, in which, each segment had 118 signals of two continuous gait cycles



(a) OU-ISIR dataset

(b) WhuGAIT dataset

FIGURE 3. The identification accuracy under different input group sizes and number of LSTM layers when the hidden size is 40.

such that two consecutive segments overlapped one cycle.

- *(ii) Window-based segmentation (overlapping 50%)*: We used the dataset #3 for training the model. This dataset also contained gait segments of 118 users, however, each segment consisted of 128 gait signals that were collected in 2.56 seconds (*i.e.,* approximate to the time of two gait cycles), and two continuous segments overlapped each other 50% (*i.e.,* 64 signals).
- *(iii) Window-based segmentation (overlapping 97%)*: We also used the dataset #3 for this case. However, we created a larger training dataset by increasing the overlapping between two continuous segments as $\theta = 97\%$ (see Section IV-A2).

Under three cases of data segmentation, we used the same setting for the model as input group size $K = 4$, hidden size $H = 40$ and LSTM network of 2-layer.

Table 2 summarizes the identification accuracy of 3 cases above. We could see that under the same amount of training data, the cycle-based segmentation achieved the accuracy of 92.19%, which is a little higher comparing to window-based segmentation. However, with the fixed length window segmentation, it is easy to increase the training dataset (by increasing the overlapping portion), and the model could reach higher accuracy comparing to cycle-based segmentation. Note that, this experiment assumed a good gait cycle detection algorithm (*i.e.,* having manual checking after a heuristic algorithm) [19]. Without manual checking, the number of incorrect gait cycles may increase, and the overall accuracy could reduce. Thus, with the fixed-length window segmentation, the model could reach higher accuracy by creating larger amount of training data, and does not depend on the gait cycle detection algorithm.

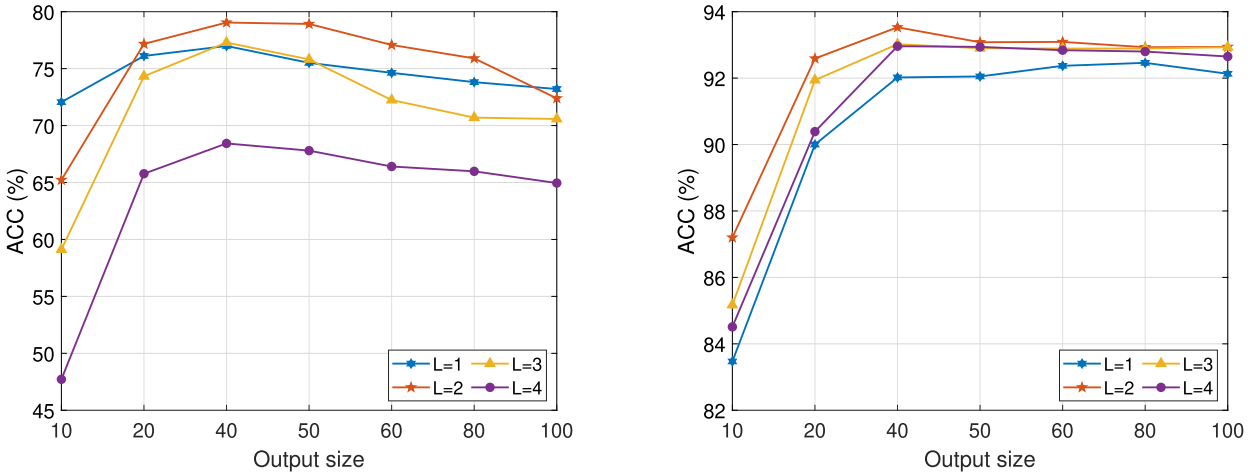
2) THE INPUT GROUP SIZE

In Figures 3, we present the recognition performance under different settings of the input group size (K) and the number of LSTM layers (L). It could be seen that, selecting the right value for input group size K is important for the model to achieve optimal performance. From the figure, increasing K (from 1) will improve the recognition performance. However, upon reaching a certain value, continuing to increase K will decrease the accuracy. For example, with the OU-ISIR dataset and using the 1-layer LSTM model, the recognition performance (ACC) was 71.85% when K was 2. Increasing the input group size K to 5 and 10 gradually improved the accuracy to 76.02% and 76.99%, respectively. However, when K reached 20 and 25, the accuracy was degraded to 73.96 and 73.72%, respectively. With the whuGAIT dataset, the 1-layer LSTM achieved the accuracy of 85.85% when the input group size K was 1. When increasing K to 2 and 4, the accuracy was also improved to 91.69% and 92.16%; then, gradually degraded to 91.23% and 90.14% when K was increased to 8 and 16, respectively. Similar observations were obtained when using other settings for LSTM layer (see Figure 3).

TABLE 2. The identification accuracy (%) of the LSTM network under different settings of data segmentation, measured on whuGAIT dataset.

Settings	*(i)* Cycle ($\theta = 50\%$)	*(ii)* Window ($\theta = 50\%$)	*(iii)* Window ($\theta = 97\%$)
Accuracy	92.19%	91.98%	93.14 %

The reasons behind that are as follows. When K increases, the number of LSTM steps for processing entire input sequence are decreased (see Section III-B2). Thus, it is easier for the LSTM network to handle the entire gait sequence when increasing K (*i.e.,* LSTM usually has difficulty when dealing with a long input sequence in the classification task [24]). However, LSTM is powerful for learning the hidden information between inputs at different steps, and unable to extract the correlation hidden inside elements of the input vector at each step. Thus, when K is increased,



(a) OU-ISIR dataset (b) WhuGAIT dataset

FIGURE 4. The identification accuracy under different settings of output size and LSTM layers when using optimal setting for input group size.

the input vector at each step becomes larger, a higher amount of correlation information will be missed and the recognition performance is degraded.

Note that, in Figure 3a, when $K = 1$, the network could not converge when training (*i.e.,* the recognition accuracy was below 10 %). Thus, we do not present the case of $K = 1$ in this figure.

3) THE OUTPUT SIZE

The output vector $\mathbf{f}_j^{(i)}$ contains the features extracted from a gait channel j of sequence i. In general, if the size of $\mathbf{f}_j^{(i)}$ (*i.e.,* H) is too small, the model could suffer from underfitting which results to low accuracy. However, too large H would make the feature space too big for the classifier to effectively handle due to the curse of dimensionality problem [40].

Figures 4 display the model accuracy under different output sizes H when using the optimal input group size (*i.e.,* $K = 10$ for OU-ISIR dataset, $K = 4$ for whuGAIT dataset). For the OU-ISIR dataset and 2-layer LSTM model, when using a small output size as 10, the model had low recognition accuracy as 65.23%. Then, increasing the output size to 20 and 40 improved the performance to 77.17% and 79.04%, respectively. Upon reaching the optimal value as 40, continuing increasing the output size caused overfitting and degraded the performance (*e.g.,* 78.92% with $H = 50$, and 77.08% with $H = 60$). Similarly, for the whuGAIT dataset and using 2-layer LSTM, the model had low recognition accuracy as 87.2% when using a small output size as 10. When increasing the output size to 20 and 40, the recognition accuracy was also improved to 92.59% and 93.53%, respectively. However, when the output size was beyond 40, the performance gradually decreased (*e.g.,* 93.08% with $H = 50$, 93.09% with $H = 60$). Similar observations could be obtained when using different settings for number of LSTM layers.

4) THE NUMBER OF LSTM LAYERS

The number of LSTM layers L defines the number of parameters used in the LSTM model. A higher value of L, a larger number of parameters. So, in general, a small value of L leads to a simple and underfitting LSTM model which is inflexible and has low recognition performance. On the other hand, too large value of L results in an overfitting model, which loses the generalization and also has low recognition performance. Typically, this parameter is determined based on how much training data is available.

As we could see in the Figures 3 and 4, the model achieved optimal performance in two datasets when using 2-layer LSTM network. However, with the OU-ISIR dataset, the 4-layer model has the lowest performance comparing to other settings of L. The reason for this is the data scarcity in the OU-ISIR dataset (*i.e.,* each user just walked 2 sessions of 9 m). Thus, with this dataset, there is not enough data for training, and it is easy for the model to be overfitting when using a network with large number of parameters as 4-layer LSTM.

On the other hand, with the whuGAIT dataset, even though the 4-layer model did not achieve the optimal performance, it still had better performance comparing to 1-layer LSTM. The reason for this is, each user in the whuGAIT dataset has much more training data comparing to the user in the OU-ISIR dataset. Thus, the model is easy to be suffered from underfitting than overfitting.

D. THE COMBINATION OF CNN AND LSTM

LSTM is strong in learning the temporal relations hidden inside sequential data. On the other hand, CNN is great for extracting the spatial information. The fusion of CNN and LSTM is potential to be superior as it can provide a richer set of features than using one technique. In this section, we present an extended model which combines our LSTM network and an existing CNN network [18]. We show that,

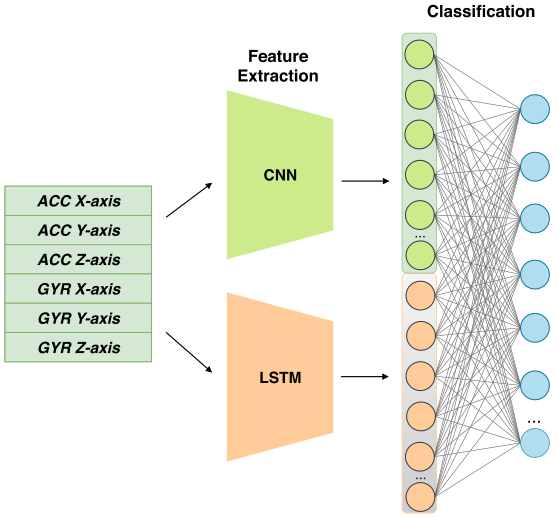


FIGURE 5. The hybrid deep network architecture which combines our LSTM model and the CNN model [18].

the hybrid model outperforms the existing networks and establishes new state-of-the-art performance.

1) NETWORK STRUCTURE AND OPTIMIZATION

Figure 5 depicts the architecture of the hybrid network which combines LSTM and CNN. The network architecture could be divided into two parts as feature extractor and classifier. The feature extractor comprises of the LSTM network (described in Section III) and CNN network (described in [18]) processing in parallel to independently extract the spatial and temporal features from the gait segment. The classifier is a fully connected layer which maps the feature vector extracted by LSTM and CNN to the output vector of size N_u, where N_u is the number of users.

The hybrid model was trained in two steps. In the first step, the original LSTM network (Figure 2) and CNN network [18] were trained separately. Then, we discarded the last fully connected layer (*i.e.,* the classifier) in each network, and used the remaining parts to form the feature extractor of the hybrid network. In the second step, we trained the classifier of the hybrid network. Specifically, entire hybrid network processed the data to obtain the output, which was used to update the parameters of the classifier iteratively using the Stochastic Gradient Descent algorithm. Note that, in this step, the parameters of the feature extractor (*i.e.,* the LSTM and CNN networks) were kept unchanged.

2) RESULT

Table 3 summarizes the identification performance of the hybrid network comparing to using CNN or LSTM alone, measured on the OU-ISIR and whuGAIT datasets. On both datasets, the hybrid model showed improved performance comparing to using only CNN or LSTM. Specifically, on the OU-ISIR dataset, the hybrid model achieved the identification accuracy of 89.79% while it was 78.92% for LSTM and 84.91% for CNN. Similarly, for the whuGAIT dataset, the accuracy was 94.15% with the hybrid network while it was 93.14% with LSTM and 93.64% with CNN.

TABLE 3. The identification accuracy (%) of the hybrid network comparing to LSTM and CNN alone.

Dataset	*LSTM*	*CNN* [18]	*Hybrid model*
OU-ISIR	78.92%	84.91%	**89.79%**
whuGAIT	93.14%	93.64%	**94.15%**

E. VERIFICATION

We extend the proposed models and analyze them on the verification task. In this task, the model will solve the problem of binary classification which validates whether a given gait segment belongs to a claimed identity or not. In addition, unlike the user identification task, verification requires the model to handle the data of unknown users (*i.e.,* the users that do not participate on the training phase). For that requirement, we combine the proposed network with One-class Support Vector Machine (OCSVM) [26] to construct the verification model as follows.

1) VERIFICATION MODEL

The overall architecture of verification model could be divided into two parts as the feature extractor and the classifier. The feature extractor is one of the above deep networks (*i.e.,* LSTM (section III), CNN [18], or hybrid (section IV-D)) which accepts a gait segment and extracts a feature vector to be used for classification. However, for this task, instead of using NLL loss, we use *triplet loss* for training the deep networks [25], thus, they could extract a unique feature vector for each user. Then, OCSVM is used as the classifier, which accepts the extracted feature vector as the input and decides whether it belongs to the genuine user or impostor. With OCSVM, the model could be trained using only data of the enrolled user, yet, could effectively handle the unknown impostor. Specifically, OCSVM gets a set of feature vectors $\mathbf{f}^{(i)}$ from the enrolled user u to learn a boundary $\mathsf{g}(.)$ that covers all $\mathbf{f}^{(i)}$ and separates them from the other users. This is performed by mapping the feature vectors $\mathbf{f}^{(i)}$ to a suitable space corresponding to the kernel, then a hyperplane $\mathsf{g}(.)$ for separating them from the origin with maximum margin is determined. Given a feature vectors $\mathbf{f}^{(i)}$, the user is classified as genuine if $\mathsf{g}(\mathbf{f}^{(i)}) \geq 0$, and impostor otherwise.

2) EXPERIMENT PROCEDURE

Figure 6 depicts the experiment procedure for the verification task, which could be divided into 3 steps as training the deep network, training the OCSVM, and testing. A given dataset was divided into 2 subsets as training and testing. In the first step, entire training set was used for training the deep network

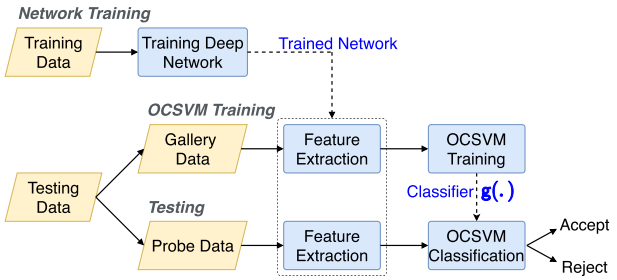


FIGURE 6. The verification model which uses the trained network as the feature extractor and OCSVM as the classifier.

TABLE 4. The Equal Error Rate (EER) of the verification models, evaluated on OU-ISIR and whuGAIT datasets.

Dataset	*Setting*	*LSTM*	*CNN*	*LSTM & CNN*
OU-ISIR	520/224	6.63%	3.74%	3.36%
	744/744	5.35%	3.05%	2.78%
whuGAIT	98/20	5.82%	5.07%	4.52%

with triplet loss. After that, the trained network was used as a feature extractor. For each user u in the testing set, half amount of data of u was used as the gallery data, to train the OCSVM boundary (*i.e,* the OCSVM training step). All remaining data in the testing set was used for evaluating the accuracy of the trained boundary (*i.e.,* testing step). Note that, steps 2 and 3 were repeated for all users in the testing set.

For the whuGAIT dataset, we used all the data of 98 users as the training set, and data of 20 users as the testing set. With each user u in the testing users, 50% data were used to construct the OCSVM boundary $\mathsf{g}(.)$, and all remaining data were used for evaluating the performance.

The OU-ISIR dataset has been used to evaluate in many studies [16], [18], [22], [41]–[43]. However, each study used different setting for their experiment. Thus, we evaluated our method with two settings, to facilitate the comparison to existing works. For the first setting, similar to the study [22], we used data of 520 users as the training set, for training the deep network, and the remaining 224 users for constructing the OCSVM and evaluating the verification performance. For the second setting, each user in the dataset participated in both the training and evaluating, similar to the studies [18], [41]–[43]. Specifically, 50% data of 744 users were used to train the deep network. Then, the remaining data were used to construct the OCSVM model and evaluate the performance. In addition, in the second setting, we combined the decisions of all segments to obtain the decision examined in entire sequence with majority voting. Specifically, if more than half number of segments extracted from the sequence were classified as positive, this sequence was decided belonging to the genuine user, otherwise, an impostor.

The verification performance was evaluated by the False Rejection Rate (FRR) and False Acceptance Rate (FAR). Specifically, let N_G be the number of attempts performed by the enrolled user, and F_G be the times being classified as an impostor. The FRR was determined by

$$\text{FRR} = \frac{F_G \times 100}{N_G}. \tag{10}$$

The FAR was calculated by

$$\text{FAR} = \frac{F_I \times 100}{N_I}, \tag{11}$$

where N_I is the number of attempts performed by the impostors, and F_I is the times being classified as enrolled user.

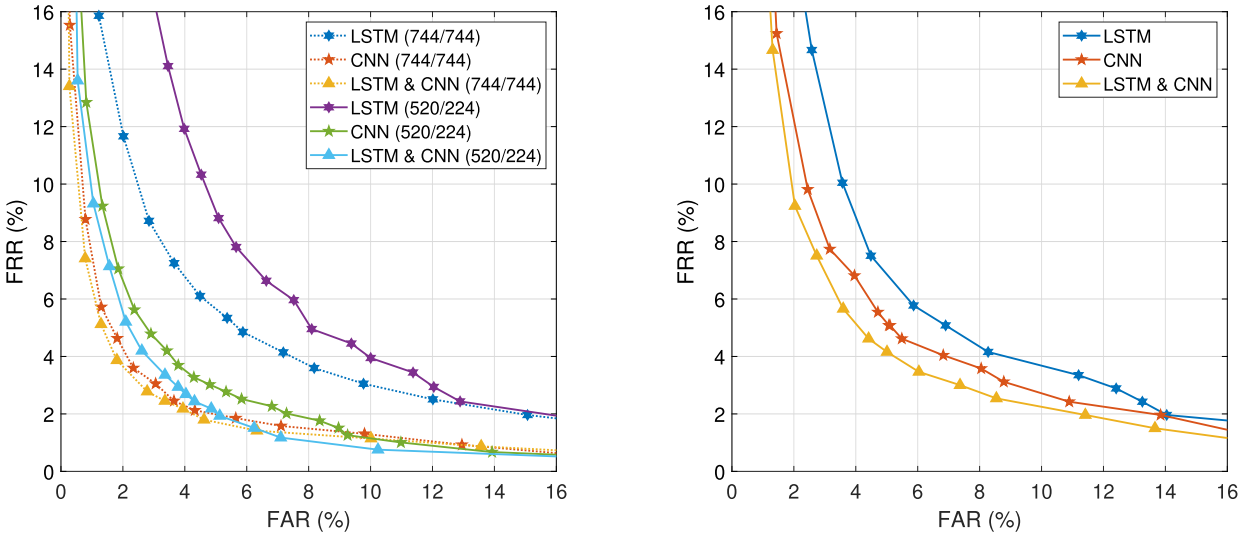
To provide a flexible trade-off between FAR and FRR for OCSVM, we introduced a threshold ϕ to separate the genuine and impostor. Specifically, for a specific ϕ, the user is classified as genuine if $\mathsf{g}(\mathbf{f}^{(i)}) \geq \phi$, and impostor if $\mathsf{g}(\mathbf{f}^{(i)}) < \phi$. Beside FAR and FRR, we additionally determined the Equal Error Rate (EER), which is the average of FAR and FRR when they are equal or approximate to each other, obtained by adjusting ϕ.

3) VERIFICATION RESULT

Figures 7 display the verification performance (*i.e.,* FAR and FRR) measured on OU-ISIR and whuGAIT datasets. The EERs under different settings are summarized in Table 4, where the column "Setting" provides the number of users participating in the training and testing phases. In different settings, although LSTM has lower performance than CNN, the combination of LSTM and CNN achieves an improved performance comparing to using CNN alone. Specifically, on the OU-ISIR dataset, with the first setting (*i.e.,* 520 users for training, 224 users for testing), the EERs of LSTM and CNN were 6.63% and 3.74%, and their combination could reduce the EER to 3.36%. Similarly, with the second setting (*i.e.,* all users participated in both training and testing), the combination of LSTM and CNN achieved the EER of 2.78%, while it was 5.35% and 3.05% when using only LSTM or CNN. This efficiency could be additionally confirmed on the whuGAIT dataset. When using CNN alone, the EER was 5.07%, however, with the combination of LSTM, the EER was reduced to 4.52%. Those results confirm that, LSTM could extract from the gait segment some features that could not be obtained by using CNN. And those features could be used to improve the recognition performance in both the tasks of user identification and verification.

F. COMPARISON WITH EXISTING WORKS

In this section, we provide a comparison between our method and existing studies. Note that, as each study was experimented in a different setting, it is hard to make a fair comparison. We consider the performance measured on two settings as segment-based and session-based. For the segment-based, the performance was determined from the result of single



(a) OU-ISIR dataset (b) WhuGAIT dataset

FIGURE 7. The verification performance (FAR, FRR) of different model architectures.

TABLE 5. The comparison on identification performance (ACC) between our study and existing researches.

Dataset	*Study*	*Method*	*Accuracy*	
			Segment	*Session*
whuGAIT	Zou *et al.*, 2020 [19]	LSTM	91.88%	–
		LSTM & CNN	93.52%	–
	Our study	LSTM	93.14%	–
		LSTM & CNN	**94.15%**	–
OU-ISIR	Ngo *et al.*, 2014 [27]	DTW	70.2%	–
	Wei *et al.*, 2015 [44]	RBF network	83.8%	–
	Delgado *et al.*, 2018 [18]	CNN	–	94.8%
	Zou *et al.*, 2020 [19]	LSTM	72.32%	–
	Tran *et al.*, 2020 [20]	CNN	82.53%	–
	Our study	LSTM	78.92%	94.23%
		LSTM & CNN	**89.79%**	**98.67%**

segment. On the other hand, the session-based result was combined from the results of all its segments.

1) IDENTIFICATION

Table 5 provides a comparison on identification performance between our method and existing studies which used whuGAIT and OU-ISIR datasets as the benchmark. WhuGAIT dataset was conducted by the authors in [19], and used to evaluate their deep learning-based gait recognition models. In that study, their LSTM model achieved the identification accuracy of 91.88% while the combination of LSTM and CNN reached 93.52%. In this dataset, our LSTM model showed the improved performance when it can achieve the accuracy of 93.14% and its combination with CNN reaches 94.15%. OU-ISIR dataset was first introduced in [27]. In that study, the authors used it to confirm the performance of existing methods published so far, and the best accuracy as 70.2% was reported when measured in each segment. After that, it has been used to evaluate the recognition performance in many studies [18]–[20], [44]. On this dataset, our LSTM network achieved higher identification performance comparing to the methods used in studies [19], [20], [27]. In addition, our proposed hybrid network over-performed the method used in [44]. The CNN network in [18] achieved a high accuracy as 94.8% when using entire session for each test trial. Under the same setting, our LSTM network achieved an approximate result as 94.23%. Furthermore, the combination of our LSTM network and their CNN showed a clear improvement, which reached the accuracy of 98.67%.

TABLE 6. The comparison on verification performance (EER) between our study and existing methods.

Dataset	*Study*	*Method*	*Equal Error Rate*	
			Segment	*Session*
whuGAIT	(*)Zou *et al.*, 2020 [19]	LSTM	7.5%	–
		LSTM & CNN	6.5%	–
	(*)Our study	LSTM	5.82%	–
		LSTM & CNN	**4.52%**	–
OU-ISIR	Ngo *et al.*, 2014 [27]	DTW	13.5%	–
	Zhong *et al.*, 2014 [41]	RBF network	–	5.6%
	Sprager *et al.*, 2015 [42]	Handcraft	10.07%	–
	(*)Nguyen *et al.*, 2017 [16]	CNN	10.43%	–
	Subramanian *et al.*, 2018 [43]	Handcraft	> 6%	–
	Delgado *et al.*, 2018 [18]	CNN	–	1.1%
	(*)Fernandez *et al.*, 2019 [22]	LSTM	7.55%	–
	Tran *et al.*, 2020 [20]	CNN	4.49%	–
	Our study	(*)LSTM	6.63%	–
		(*)LSTM & CNN	**3.36%**	–
		LSTM	5.35%	2.65%
		LSTM & CNN	**2.78%**	**0.94%**

For the cases (*), some users in the dataset were used for training and the remaining users were used for testing.

2) VERIFICATION

We summarized in Table 6 the EER of our verification models comparing to existing researches. On the whuGAIT dataset, our LSTM model showed higher verification accuracy comparing to the LSTM network used in [19]. And our hybrid network also outperformed their hybrid architecture.

On the OU-ISIR dataset, our methods achieved improved performance under different experimental settings comparing to existing studies. Specifically, for the first setting (*i.e.,* 520 users for training, and the remaining 224 users for evaluation), our LSTM network achieved the EER of 6.63% while it was 7.55% in [22]. The authors in study [16] conducted a different experiment setting comparing to other studies. Data of a subset of users were used for training, and entire users participated on the validation phase. They reported the EER of 10.43%, however, it is hard to make a performance comparison here. For all remaining studies (*i.e.,*, [18], [20], [27], [41]–[43]), all users in the dataset participated on both training and testing phase, however, different portion of data was used in each study. Among them, the authors in [20], [27], [43] used a same amount of data for each testing portion, which was 1 gait cycle, and the EER of 13.5%, 6% and 4.49% were reported, respectively. On this setting, our LSTM network achieved the EER of 5.35%, and the hybrid network reached 2.78%. When using entire walking session for each testing trial, the EER of our LSTM network 2.65%, which was higher than the method in [41]. Comparing to the CNN network in [18], which achieved the EER of 1.1% measured on entire session, our LSTM model had lower performance. However, the combination of our network and that CNN model showed better performance, which achieved the EER of 0.94%.

V. CONCLUSION

In this study, we proposed a novel multi-model LSTM network for IMUs-based gait recognition. First, unlike the existing LSTM-based gait approaches, our model accepts the input as fixed-length segments, which are independent from the phase of gait cycle. This alteration allows the recognition model to be released from the gait cycle detection task which is sensitive to noise and bias. Second, we designed a new LSTM network architecture, in which, one LSTM is used for each data channel, and a group of consecutive signals is processed at each step to effectively handle the gait data sequence. Furthermore, we extended the LSTM network to construct a hybrid network by combining with CNN. The user verification model was also constructed by extending the existing network with OCSVM. The evaluation on OU-ISIR and whuGAIT datasets showed that our method outperformed the existing LSTM gait models. Although LSTM still could not reach the equal performance with CNN, it can improve the CNN model and achieve new state-of-the-art performance in both the tasks of identification and verification.

REFERENCES

[1] S. A. Niyogi and E. H. Adelson, "Analyzing and recognizing walking figures in XYT," in *Proc. CVPR*, vol. 94, Jun. 1994, pp. 469–474.

[2] L. Wang, T. Tan, H. Ning, and W. Hu, "Silhouette analysis-based gait recognition for human identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1505–1518, Dec. 2003.
[3] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 2, pp. 316–322, Feb. 2006.
[4] Z. Liu and S. Sarkar, "Improved gait recognition by gait dynamics normalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 6, pp. 863–876, Jun. 2006.
[5] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.
[6] K. Nakajima, Y. Mizukami, K. Tanaka, and T. Tamura, "Footprint-based personal recognition," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 11, pp. 1534–1537, Nov. 2000.
[7] J. Jenkins and C. Ellis, "Using ground reaction forces from gait analysis: Body mass as a weak biometric," in *Proc. Int. Conf. Pervasive Comput.* Berlin, Germany: Springer, 2007, pp. 251–267.
[8] T. C. Pataky, T. Mu, K. Bosch, D. Rosenbaum, and J. Y. Goulermas, "Gait recognition: Highly unique dynamic plantar pressure patterns among 104 individuals," *J. Roy. Soc. Interface*, vol. 9, no. 69, pp. 790–800, 2012.
[9] C. Wan, L. Wang, and V. V. Phoha, "A survey on gait recognition," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–35, 2018.
[10] J. P. Singh, S. Jain, S. Arora, and U. P. Singh, "Vision-based gait recognition: A survey," *IEEE Access*, vol. 6, pp. 70497–70527, 2018.
[11] H. J. Ailisto, M. Lindholm, J. Mantyjarvi, E. Vildjiounaite, and S.-M. Makela, "Identifying people from gait pattern with accelerometers," *Proc. SPIE*, vol. 5779, pp. 7–14, Mar. 2005.
[12] S. Sprager and M. B. Juric, "Inertial sensor-based gait recognition: A review," *IEEE Sensors J.*, vol. 15, no. 9, pp. 22089–22127, Sep. 2015.
[13] M. D. Marsico and A. Mecca, "A survey on gait recognition via wearable sensors," *ACM Comput. Surv.*, vol. 52, no. 4, pp. 1–39, 2019.
[14] O. Costilla-Reyes, R. Vera-Rodriguez, A. S. Alharthi, S. U. Yunas, and K. B. Ozanyan, "Deep learning in gait analysis for security and healthcare," in *Deep Learning: Algorithms and Applications*. Cham, Switzerland: Springer, 2020, pp. 299–334.
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[16] K.-T. Nguyen, T.-L. Vo-Tran, D.-T. Dinh, and M.-T. Tran, "Gait recognition with multi-region size convolutional neural network for authentication with wearable sensors," in *Proc. Int. Conf. Future Data Secur. Eng.* Cham, Switzerland: Springer, 2017, pp. 197–212.
[17] M. Gadaleta and M. Rossi, "IDNet: Smartphone-based gait recognition with convolutional neural networks," *Pattern Recognit.*, vol. 74, pp. 25–37, Feb. 2018.
[18] R. Delgado-Escaño, F. M. Castro, J. R. Cózar, M. J. Marín-Jiménez, and N. Guil, "An end-to-end multi-task and fusion CNN for inertial-based gait recognition," *IEEE Access*, vol. 7, pp. 1897–1908, 2019.
[19] Q. Zou, Y. Wang, Q. Wang, Y. Zhao, and Q. Li, "Deep learning-based gait recognition using smartphones in the wild," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3197–3212, 2020.
[20] L. Tran and D. Choi, "Data augmentation for inertial sensor-based gait deep neural network," *IEEE Access*, vol. 8, pp. 12364–12378, 2020.
[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[22] P. Fernandez-Lopez, J. Liu-Jimenez, K. Kiyokawa, Y. Wu, and R. Sanchez-Reillo, "Recurrent neural network for inertial gait user recognition in smartphones," *Sensors*, vol. 19, no. 18, p. 4054, 2019.
[23] V. N. Bobić, M. D. Djurić-Jovièić, S. M. Radovanović, N. T. Dragaević, V. S. Kostić, and M. B. Popović, "Challenges of stride segmentation and their implementation for impaired gait," in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2018, pp. 2284–2287.
[24] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*. [Online]. Available: http://arxiv.org/abs/1803.01271
[25] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.
[26] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 12, 2000, pp. 582–588.
[27] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi, "The largest inertial sensor-based gait database and performance evaluation of gait-based personal authentication," *Pattern Recognit.*, vol. 47, no. 1, pp. 228–237, 2014.
[28] G. Batchuluun, H. S. Yoon, J. K. Kang, and K. R. Park, "Gait-based human identification by combining shallow convolutional neural network-stacked long short-term memory and deep convolutional neural network," *IEEE Access*, vol. 6, pp. 63164–63186, 2018.
[29] T. K. M. Lee, M. Belkhatir, and S. Sanei, "A comprehensive review of past and present vision-based techniques for gait recognition," *Multimedia Tools Appl.*, vol. 72, no. 3, pp. 2833–2869, 2014.
[30] C. Prakash, R. Kumar, and N. Mittal, "Recent developments in human gait research: Parameters, approaches, applications, machine learning techniques, datasets and challenges," *Artif. Intell. Rev.*, vol. 49, no. 1, pp. 1–40, Jan. 2018.
[31] T. Hoang, D. Choi, and T. Nguyen, "On the instability of sensor orientation in gait verification on mobile phone," in *Proc. 12th Int. Joint Conf. E-Bus. Telecommun. (ICETE)*, vol. 4, Jul. 2015, pp. 148–159.
[32] G. Giorgi, F. Martinelli, A. Saracino, and M. Sheikhalishahi, "Try walking in my shoes, if you can: Accurate gait recognition through deep learning," in *Proc. Int. Conf. Comput. Saf., Rel., Secur.* Cham, Switzerland: Springer, 2017, pp. 384–395.
[33] O. Dehzangi, M. Taherisadr, and R. ChangalVala, "IMU-based gait recognition using convolutional neural networks and multi-sensor fusion," *Sensors*, vol. 17, no. 12, p. 2735, 2017.
[34] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078
[35] I. J. Schoenberg, *Cardinal Spline Interpolation*. Philadelphia, PA, USA: SIAM, 1973.
[36] T. Ji and A. Pachi, "Frequency and velocity of people walking," *Struct. Eng.*, vol. 84, no. 3, pp. 36–40, 2005.
[37] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1019–1027.
[38] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
[39] I. Goodfellow, Y. Bengio, and A. Courville, "6.5 back-propagation and other differentiation algorithms," *Deep Learn.*, vol. 1, no. 2, pp. 200–220, 2016.
[40] R. Hanka and T. P. Harte, "Curse of dimensionality: Classifying large multi-dimensional images with neural networks," in *Computer Intensive Methods in Control and Signal Processing*. Boston, MA, USA: Springer, 1997, pp. 249–260.
[41] Y. Zhong and Y. Deng, "Sensor orientation invariant mobile gait biometrics," in *Proc. IEEE Int. Joint Conf. Biometrics*, Oct. 2014, pp. 1–8.
[42] S. Sprager and M. B. Juric, "An efficient HOS-based gait authentication of accelerometer data," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 7, pp. 1486–1498, Jul. 2015.
[43] R. Subramanian and S. Sarkar, "Evaluation of algorithms for orientation invariant inertial gait matching," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 304–318, Feb. 2019.
[44] Z. Wei, W. Qinghui, D. Muqing, and L. Yiqi, "A new inertial sensor-based gait recognition method via deterministic learning," in *Proc. 34th Chin. Control Conf. (CCC)*, Jul. 2015, pp. 3908–3913.

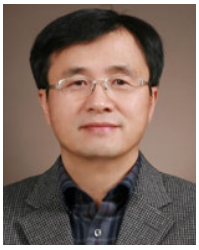
LAM TRAN received the B.S. degree in computer science from Ho Chi Minh University of Science, Ho Chi Minh City, Vietnam, in September 2012, and the M.E. degree in computer science from Chonnam National University, Gwangju, South Korea, in August 2017, where he is currently pursuing the Ph.D. degree with the Department of Artificial Intelligence Convergence. His research interests include inertial sensor-based gait biometrics, privacy enhancing for biometric systems, and machine learning.

THANG HOANG received the B.S. degree in computer science from Ho Chi Minh University of Science, Ho Chi Minh, Vietnam, in 2010, and the M.E. degree in computer science from Chonnam National University, Gwangju, South Korea, in 2014, and the Ph.D. degree from the University of South Florida, in August 2020.

He is currently a Postdoctoral Fellow hosted by Prof. Elaine Shi at Carnegie Mellon University (CMU), and a Research Associate hosted by Prof. Attila A. Yavuz at the University of South Florida (USF). His research interests include applied cryptography and machine learning, with special interests in privacy-enhancing technologies, zero-knowledge proofs, multi-party computation, oblivious RAM, and searchable encryption.

THUC NGUYEN received the B.S. degree from the Faculty of Information Technology, Vietnam National University-Ho Chi Minh City, (VNU-HCMC), Vietnam, in 1990, and the Ph.D. degree from VNU-HCMC, in 2000. He is currently an Associate Professor with the Department of Knowledge Engineering, Faculty of Information Technology, VNU-HCMC. His research interests include concrete structures and models, cryptography, database security, and sensor network security.

HYUNIL KIM received the B.S. degree in applied mathematics, and the M.S. and Ph.D. degrees in information security from Kongju National University, South Korea, in 2014, 2016, and 2019, respectively. He is currently a Postdoctoral Researcher with the Daegu Gyeongbuk Institute of Science & Technology (DGIST), Daegu, South Korea. His research interests include AI-based cryptanalysis, artificial intelligence, deep learning, blockchain, and federated learning.



DEOKJAI CHOI received the B.S. degree from the Department of Computer Engineering, Seoul National University, in 1982, the M.S. degree from the Department of Computer Science, Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 1984, and the Ph.D. degree from the Department of Computer Science and Telecommunications, University of Missouri-Kansas City, USA, in 1995. He is currently a Full Professor with the Department of Artificial Intelligence Convergence, Chonnam National University, South Korea. His research interests include mobile security, human activity profile, biometric authentication, and network security.