Human Posture Tracking with Flexible Sensors for Motion Recognition

Abstract

The integration of conventional clothes with flexible electronics is a promising solution as a future-generation computing platform. However, the problem of user authentication on this novel platform is still under-explored. This work uses flexible sensors to track human posture and achieves the goal of user authentication. We capture human movement pattern by four stretch sensors around the shoulder and one on the elbow. We introduce the Long Short-Term Memory Fully Convolutional Network (LSTM-FCN), which directly takes noisy and sparse sensor data as input and verifies its consistency with the user's pre-defined movement patterns. The method can identify a user by matching movement patterns even if there are large intra-personal variations. The authentication accuracy of LSTM-FCN reaches 98.0%, which is 10.7% and 6.5% higher than that of Dynamic Time Warping (DTW) and Dynamic Time Warping Dependent (DTW-D).

Keywords: user authentication, smart clothes, flexible sensor

1 Introduction

Smart clothes are gaining a wide range of interest due to the fast progress of flexible electronics. They are expected to empower people to inter-connect with the world while introducing minimal intervention to their daily activity. Before smart clothes serve as a personal device, user authentication arises as an open challenge. However, explicit authentication approaches (e.g., pattern locks) suffer from several limitations including the requirement of visual display. Therefore, an authentication mechanism for smart clothes users is in need.

Existing methods explored the use of a variety of biometric data, e.g., fingerprint,[1] iris,[2] face,[3] ECG.[4] However, specialized (often rigid and bulky) sensors are required to capture these biometric data. Our work proposes the use of flexible stretch sensors, as a low-cost and user-friendly solution, to track human posture. The captured sequence of postures is used for user authentication. However, different from finger stroke patterns on mobile phones, human postures in 3D present large intra-personal variation, i.e., multiple attempts of the same posture sequence may largely differentiate from each other.

Our work directly addresses the problem of user authentication on the platform of smart clothes. The authentication is defined as two key sub-problems: 1) to track human posture

with flexible sensors, 2) to verify the consistency between the current attempt of motion trajectory with the pre-stored ones. The contribution of this work is two-fold:

- We present a complete solution of both hardware and software for human posture tracking and trajectory authentication. The use of flexible sensors introduces minimal intervention to the user's activity and ensures the maximal comfort of user experience.
- We introduce LSTM-FCN which directly takes noisy and sparse sensor data as input and matches pre-defined movement patterns. Compared with the representative methods in the time-series analysis (DTW,[5] DTW-D[6]) and CNNs, our method shows advantages in both the authentication accuracy and alleviation of parameter tuning.

2 Related Work

2.1 Wearable Motion Tracking

The recent development of sensing technology has paved the way for using wearable sensors to monitor human motion. Mainstream systems used optical, acoustic, and electromagnetic sensors for this task. Readers could refer to an overview[7] of using different sensors and techniques to estimate human upper limb motion. Optical-based systems for human motion tracking[8] are relatively expensive, require adequate illumination, and suffer from the problem of occlusion. Acoustic-based systems[9] have poor real-time performance and are vulnerable to ambient interference. Electromagnetic-based systems[10] are prone to be interfered with by the magnetic field since the metal objects around the site will cause magnetic field distortion ,which seriously affects the accuracy.

Motion tracking using inertial measurement units (IMUs) has gradually attracted people's attention.[11] presented a method to estimate the human whole-body pose with application to bicycle riding using the fusion of gyroscopes, accelerometers and force sensors.[12] proposed a method that combines a single hand-held camera and a set of IMUs attached at the body limbs to estimate 3D poses in the wild. However, the two methods above have to adopt extra constraints or filtering algorithm to correct the drift of the IMU sensors. Another work[13] tried to fuse IMU data with Kinect to provide stable hand position information with no long-term drifts. Although the direct integration of the gyroscope signal can ensure the accuracy of the output angle in a short time, the output error also accumulates with time elapse.[13]

More recent works used soft sensors for motion tracking. Researchers used a wearable sensing suit with flexible sensors for motion capture of full-body,[14] elbow,[15] finger,[16] and upper body.[17] A recent work introduced silicone-based strain and force sensors composed of a novel biocompatible conductive liquid for motion capture.[18] Researchers proposed a stretch-sensing soft glove to interactively capture hand poses with high accuracy and without requiring an external optical setup.[16] Wearable soft sensors are non-intrusive and can accurately track human posture in an unrestricted environment. Currently, one of the common features of smart clothes is motion tracking. This accessible feature offers the direct solution of user authentication, i.e., to verify user identifies with the user-defined motion trajectory.

Our human motion tracking system shows the advantages of convenient wearing, unlimited movement space, and low cost. Since the flexible sensors can be seamlessly integrated

on the clothes, this offers users the maximum convenience. However, the problem of user authentication on smart clothes is yet to be explored.

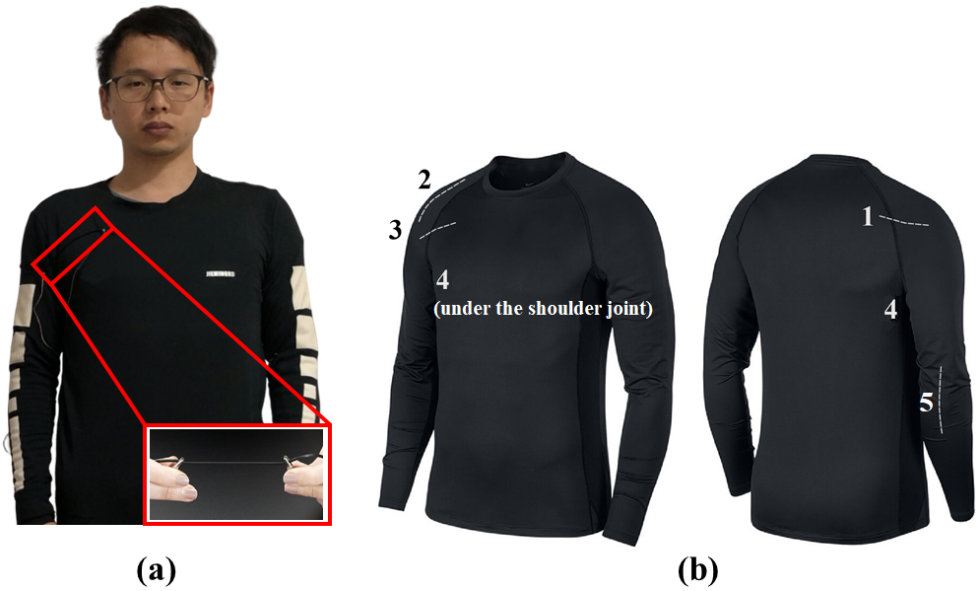

Figure 1: (a) The developed prototype and a close-up view of Sensor 3. (b) Sensor layout.

2.2 Biometric Authentication

Biometric authentication refers to the use of human physiological and/or behavioral characteristics for identification. At present, the mainstream methods for biometric recognition using physiological characteristics are: fingerprint,[1] iris,[2] palm,[19] facial,[3] finger[20] and others. A variety of behavioral features, including voice, handwriting,[21] keystroke[22] are also adopted by the mainstream methods. In addition to these conventional biometric technologies, recent years saw emerging modalities for authentication, such as ear imaging,[23] movements of arm,[24] head,[25] and gait.[26,27] Especially, Gait recognition has attracted extensive attention from institutes and researchers,[28,29] utilized the convolution neural network (CNN) to extraction feature expression and trained a classifier to authenticate human. Researchers first proposed to fuse coarse-grain minute-level physical activity (step counts) and physiological data (heart rate, calorie burn, and the metabolic equivalent of task) for the task of user authentication.[30] Although there is a large collection of works on body part movement recognition for authentication, the existing equipment is expensive or not small enough to be convenient.

Our work uses arm movement data to authenticate whether a movement matches the password (the pre-stored movement pattern). Our prototype is low-cost, convenient, and not affected by light or a cluttered background. The authentication introduces minimal intervention. Although motion tracking is similar to monitoring the stroke trajectory on the touchscreen, motion execution in an unconstrained 3D world presents significantly larger variations than pattern locks on a 2D screen.

3 Method

3.1 Hardware and Software Implementation

We developed a prototype of smart clothes (Fig. 1) with flexible sensors to complete this study. In life, we are used to using arms to manipulate and express. The movement of the arms mainly depends on the movement of elbow joint and shoulder joint. The prototype is integrated with five flexible stretchable sensors: four around the shoulder joint and one on the elbow joint. The sensors are manually sewed onto the garment with the technique of flat stitching. The clothes style is tight sportswear, which ensures that the sensors are sufficiently and consistently stretched at different attempts. The clothes fabric is made of 80% polyester and 20% polyurethane. We use the conductive rubber cord stretch sensor fabricated by Adafruit [1]. Each sensor is 10cm in length, 2mm in diameter and made of carbon-black impregnated rubber. In a relaxed state, the sensor resistance is about 350 Ohms per inch. The joint rotation is tracked by monitoring the change in the resistance value of the stretchable sensors. For example, when the user bends his/her elbow joint, the sensor is stretched and its resistance value increases accordingly. The sampling frequency of the stretch information is 32Hz.

The server computer receives the sensor data, authenticates it, and shows the results. The server computer is equipped with Intel Core i7 (6-core), 16G memory, and NVIDIA GTX 1080Ti. This server is also used for training and testing of the prediction model in the following text.

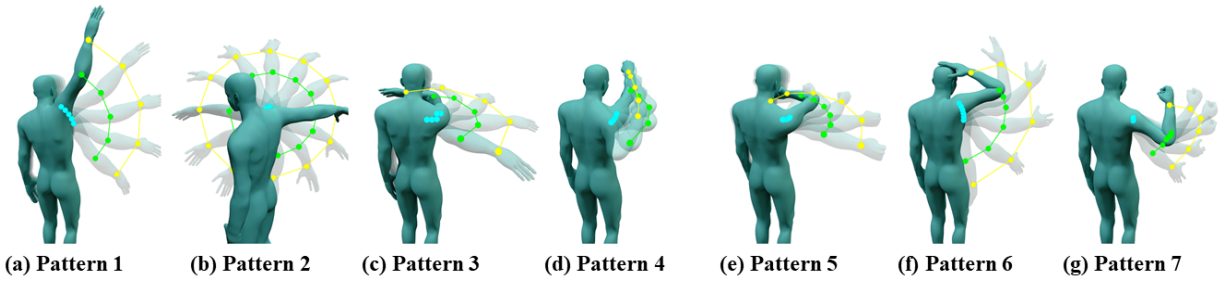

Figure 2: The collected seven movement patterns.

3.2 Data Collection

Participant: Experiments were performed on a volunteer whose age is 28, and this participant is a doctoral student. We explained the experimental design to this participant and obtained his written consent. He was given sufficient instructions before the experiment and became familiar with the experimental procedures.

Procedures: Before the experiment began, the participant was informed of the experiment purpose. After putting on the prototype system, he conducted a few moves to get familiarized with the system. Then, he can freely perform the movement according to his preferences. To indicate the start and stop of an attempt, the experimenter vocally informed the participant. These activities were repeated in a short period of time. Fig. 2 shows the seven patterns designed by the participant. If the participant is tired, the capture

[1]Product link: https://www.adafruit.com/product/519

process can be terminated at any time. The data collected from each session was automatically uploaded to our server and manually annotated with the corresponding pattern tags. A complete collection took about 1.5 hours, including the participant's breaks.

3.3 LSTM Fully Convolutional Network

In this work, we use the LSTM-FCN[31] as a two-class (positive or negative) classification tool. Each pattern is associated with a separate classifier. After our system has been configured with a password pattern, the network will judge whether the authentication is successful according to the classification result of the new movement input. The input is the human posture trajectory captured by flexible sensors and the output is whether the current trajectory belongs to the pattern to be authenticated.

LSTM-FCN is formed by a full convolution block and an LSTM block (Fig. 3). The input time series go through the convolution block and the LSTM block simultaneously. The full convolution block is composed of three stacked temporal convolution blocks, and

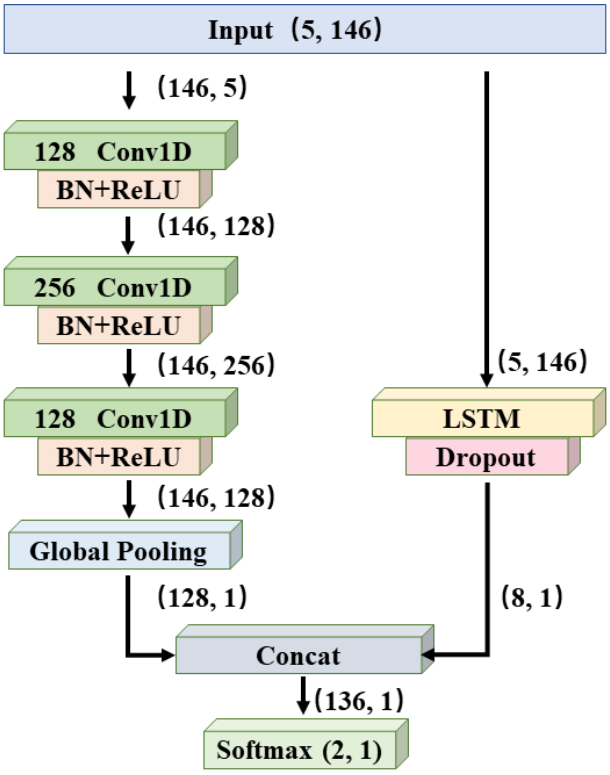

Figure 3: The LSTM-FCN architecture.

the size of the filter is 128, 256, and 128, respectively. Each convolutional block consists of a temporal convolution layer, batch normalization, and a ReLU activation function. The global average pooling is applied after the last convolution block. The LSTM block is composed of a conventional LSTM layer and a dropout layer. The outputs of the global pooling layer and the LSTM block are concatenated and passed to the softmax classification layer.

In convolutional layers, a set of 1D filters is applied to capture the evolution of input signals throughout the course of an action. The filters for each layer are parameterized by

the tensor $W^{(l)} \in \mathbb{R}^{L_l \times d \times L_{l-1}}$ and biases $b^{(l)} \in \mathbb{R}^{L_l}$. L is input feature vector length and is set to 146 in this work. $l \in \{1, \ldots, 3\}$ is the layer index and d is the filter duration. For the l-th layer and each time t, the i-th component of the (unnormalized) activation $\hat{\mathbf{C}}_t^{(l)} \in \mathbb{R}^{L_l}$ is a function of the incoming (normalized) activation matrix $C^{(l-1)} \in \mathbb{R}_{l-1}^{L} \times T_{l-1}$ from the previous layer:

$$\hat{\mathbf{C}}_{i,t}^{(l)} = f\left(b_i^{(l)} + \sum_{t'=1}^{d} \left\langle W_{i,t',.}^{(l)}, C_{.,t+d-t'}^{(l-1)} \right\rangle\right)$$

$f(\cdot)$ is a Rectified Linear Unit.

Data Processing and Flow: Our smart clothes have five sensors, and all data are padded to a length of 146. Therefore, one sample in the collected data is a time series of shape (146, 5). The fully convolutional block views the time series as a univariate time series with multiple time steps and receives the data in 146-time steps. And LSTM block receives the multivariate time series after a masking operation.

Min-Max scaling: One method of normalization is the so-called "min-max" scaling. Min-Max scaling scales data to a fixed range-usually 0 to 1. Compared with standardization, the cost of having this bounded range is that we will eventually get a smaller standard deviation, which can suppress the effect of outliers. Min-Max scaling is typically done via the following equation:

$$X_{norm} = \frac{X - X_{min}}{X_{\max} - X_{min}}.$$

The existing work[31] normalized the input data along each feature dimension using Min-Max scaling. We define this strategy as Feature Normalization (FN). Different from existing work, we normalize the input data at each time point and define this as Temporal Normalization (TN). TN does not require the prior knowledge of the complete dataset in order to compute the minimum an maximum values for each feature.

Training & Testing: The collected dataset contains $N_P = 7$ patterns and each pattern contains $N_S = 20$ samples. We train seven separate classifiers. For the classifier K ($K \in [1, N_P]$), it verifies whether the current motion attempt belongs to Pattern K, and returns true or false. To build the training dataset for classifier K, we randomly select N_{TR} samples from all patterns. N_{TR} samples from Pattern K are labeled as positive ones, while N_{TR}*(N_P-1) samples from the excluding patterns are labeled as negative ones. The rest samples ($(N_S - N_{TR}) * N_P$) are testing ones. Again, $N_S - N_{TR}$ samples from Pattern K are labeled as positive ones, while ($N_S - N_{TR}$)*(N_P-1) samples from the excluding patterns are labeled as negative ones. We choose $N_{TR} = 3$. The intuition is that to initialize the authentication system on smart clothes, the user needs to perform the password trajectory for three times. The repetition aims to maximally capture the intra-personal variation when the user conducts the same trajectory.

Training Hyperparameters: The number of training epochs is 1000. The batch size and initial learning rate are 128 and 1e-3, respectively. A high dropout rate of 80% is used after the LSTM layer to address the issue of overfitting. Our models are trained using the Keras library with the TensorFlow backend and the Adam optimizer. All convolution kernels are initialized with the approach proposed by He et al.[32] The learning rate was reduced by a factor of $1/\sqrt[3]{2}$ every 100 epochs if there is no improvement in the validation score, until its value reached 1e-4.

4 Results

4.1 Dataset Statistics

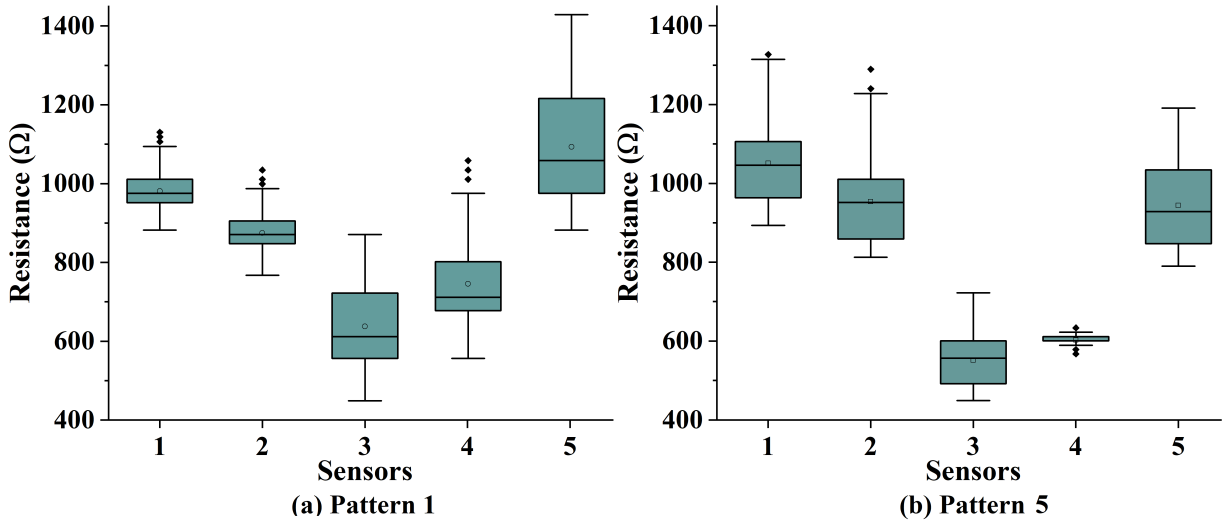

Figure 4: The resistance distribution of five sensors in Pattern 1 and Pattern 5.

Here, we visualize the distribution of the sensor resistance values across different sensors and patterns. The resistance distributions of the five sensors in Pattern 1 and Pattern 5 are shown in Fig. 4. Pattern 1 (Fig. 4 (a))is a movement of arm elevation in the coronal plane. Sensor 3 and 4 locate on the shoulder joint and demonstrate considerable variations caused by the shoulder rotation. This pattern also comes with a noticeable secondary movement on the elbow joint (Sensor 5). Pattern 5 (Fig. 4 (b))is a movement of arm forward bending. The resistances of Sensor 1, Sensor 2 (on the shoulder joint) and Sensor 5 have great changes due to arm forward bending. The sensor 4 (under the shoulder joint) is hardly stretched, and the resistance change is small. This figure confirms the large intra-personal variations when the user performs the same movement pattern.

The sample duration is between 2.788s and 4.998s, and the statistics for the duration of these samples are shown in Fig. 5. Even for the same pattern, the duration of different attempts could vary to a large extent of >10%. We address this issue by padding and converting the signal to segments of fixed durations.

The results confirm the large intra-personal variations of the sensor signal. The causes could be two-fold: the temporal factor is rooted in the varying duration for different attempts of the same pattern, and the spatial factor is due to the unconstrained 3D trajectory. These two factors are inherently correlated with the freedom on the user side when designing and performing the pattern. Our method proves to be effective in robustly handling this variation and does not require manual efforts of feature engineering.

4.2 Analysis of Authentication Results

We use the following three evaluation criteria to evaluate our experiment results:

- True Positive (TP): a true password passes the authentication.

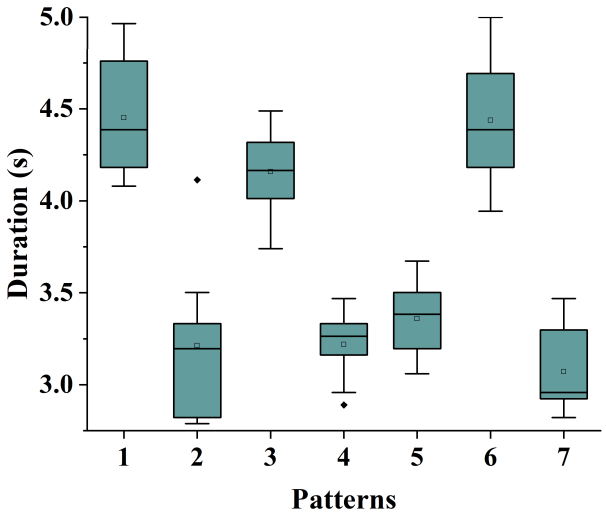

Figure 5: The statistics of movement duration in each pattern.

- True Negative (TN): a forged password fails to match.
- False negative (FN): a true password fails to match.
- False positive (FP): a forged password passes the authentication.
- Pattern Accuracy: (TP+TN)/(TP+TN+FN+FP) for Pattern i
- Overall Accuracy: (TP+TN)/(TP+TN+FN+FP) for all patterns.

		Predicted Password Pattern						
		1	2	3	4	5	6	7
Actual Password Pattern	1	15	0	0	0	0	0	0
	2	0	14	0	1	0	0	0
	3	0	1	13	0	0	0	0
	4	0	0	2	16	0	0	0
	5	0	0	0	0	14	0	0
	6	0	0	0	0	0	15	0
	7	0	0	0	0	0	0	17
Numbers of samples		17	17	17	17	17	17	17

Figure 6: Confusion matrix of authentication.

Fig. 6 shows the matching result of our method. If the actual pattern is classified to the same class as the predicted one, the match is successful and the count is increased by 1. The number in each grid is the matching result of samples in a pattern and the current

password. The numbers in the bottom row are the number of samples for each pattern in the testing dataset.

As depicted in Fig. 6, we get 15 FNs and four FPs in total. We tried 833 matches and failed 19 times. The overall accuracy of our method reaches 97.72%. Pattern 1, 2, 3, 4, 5, 6 are associated with two, three, four, one, three and two FNs, respectively. This indicates that the true password is identified as false. The potential reason is the large deviation from the training dataset. When Pattern 7 is the password, all the samples in the testing dataset are matched successfully. However, one sample in Pattern 2 are mis-identified as Pattern 4, one sample in Pattern 3 are mis-identified as Pattern 2, and two samples in Pattern 4 are mis-identified as Pattern 3, creating the defect of four FPs.

We also perform an inter-person authentication experiment. We collect data on seven patterns of three users, and each pattern was repeated 20 times by each user. The single-person authentication accuracy rate reaches 97.74%. While, when one of a user's patterns is set as password, we only select samples from this user for training, all actions attempts by the other users cannot pass the authentication.

4.3 Comparison with DTW/DTW-D

Dynamic time warping (DTW)[5] is one of the baseline methods for time-series signal processing. DTW-D[6] enhances the correlation of five features in the time series during matching and improves the matching accuracy. However, their performance is subject to threshold values, which will be further analyzed in Sec. 4.4. We here first compared our method with DTW and DTW-D of the highest matching accuracy.

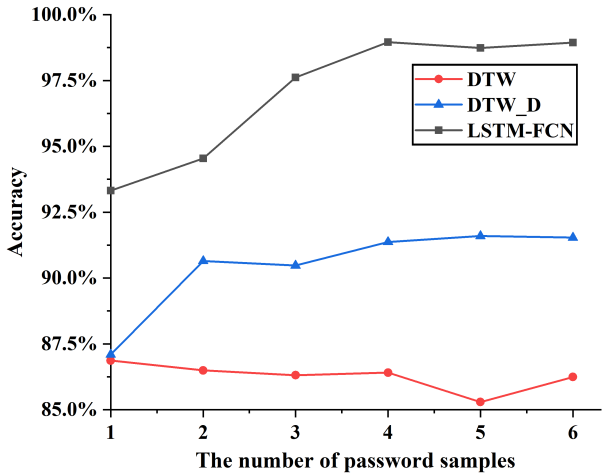

Figure 7: Accuracy of LSTM-FCN, DTW, and DTW-D.

Success condition for DTW & DTW-D: For DTW, if the distance of each channel to a sample in the password set is within a threshold, the match is successful. For DTW & DTW-D, once there are multiple passwords in a password set, the match is successful as long as the distance between the testing sample and one of the password set is less than the threshold.

Fig. 7 shows the comparison results of three methods: DTW, DTW-D and LSTM-FCN. Our LSTM-FCN model consistently outperforms DTW and DTW-D, regardless of the number of samples in the training dataset N_{TR}. Increasing the value of N_{TR} steadily

improves the authentication accuracy to 98.0%. More details of using different values of N_{TR} are discussed in Sec. 4.5.

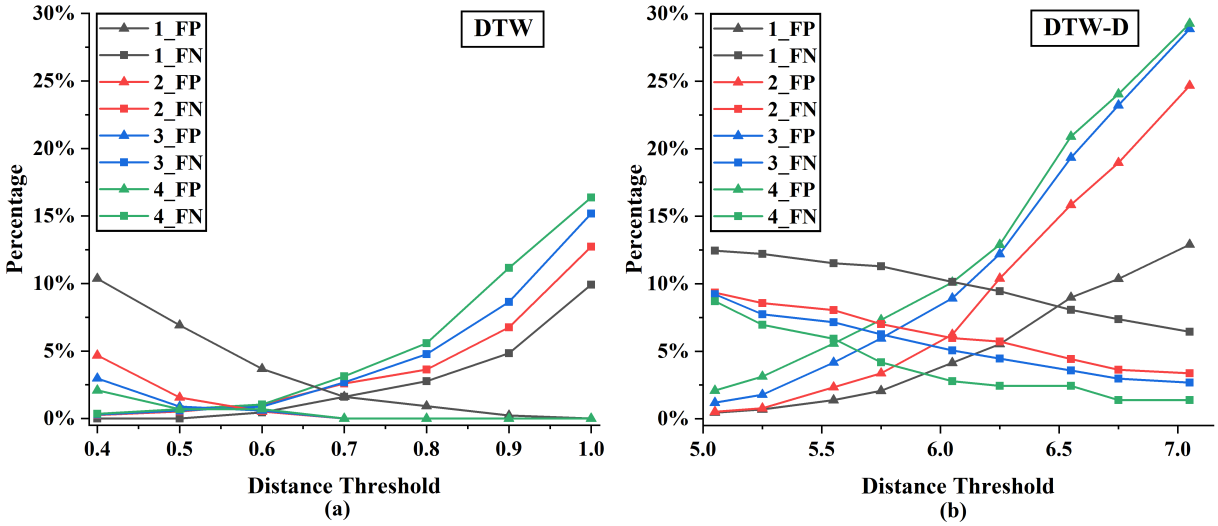

Figure 8: The performance sensitivity of DTW/DTW-D algorithms given different threshold values.

In terms of time performance, LSTM-FCN spent about 27.10s for training. However, it is trained only once and it costs 0.26s for authentication. DTW and DTW-D do not need offline training and cost 0.74s and 0.001s respectively for online authentication.

4.4 Threshold Sensitivity of DTW/DTW-D

DTW algorithm matches each of the five channels in a sample separately, while the DTW-D algorithm directly matches a sample using the five channels at the same time. However, both methods face the challenge of determining the appropriate threshold value. The experimental results are shown in Fig. 8.

Both DTW and DTW-D show a similar trend when increasing the threshold value: FP decreases and FN increases. Once the threshold is small, negative samples cannot be matched easily. As a result, the FP is relatively small. At the same time, positive samples which are in the password pattern fails to pass the authentication successfully, and the FN will be large. As the threshold increases, FP continues to increase, while FN continues to decrease. This is equivalent to reduce the authentication stringency so that samples with larger deviations could pass. Therefore, choosing an appropriate threshold value is the prerequisite for using DTW and DTW-D.

When using the different values of N_{TR}, the threshold chosen to get the balance of FP and FN changed irregularly. This is particularly true for DTW-D, where the curves of FP and FN cross at different threshold values for different values of N_{TR}. This implies the difficulty to use these two algorithms in practice. For the best performance (N_{TR}=3), DTW and DTW-D use a distance threshold of 0.6 and 5.75, respectively.

4.5 Number of Password Samples

The number of samples in the training dataset N_{TR} impacts the authentication capability and this subsection investigates the performance of our method when N_{TR} increases from

one to six. We selected the threshold value of DTW and DTW-D with the highest matching accuracy during the matching process above to compare with LSTM-FCN. Fig. 9 illustrates the changes of FP and FN with the value of N_{TR}.

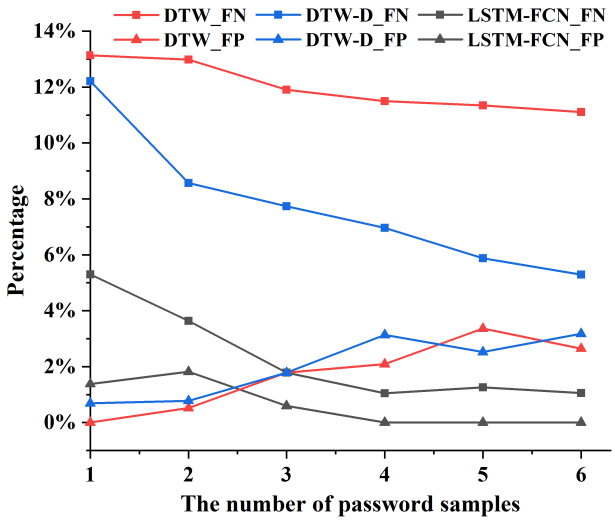

Figure 9: The relationship between the percentage of FP and FN and the number of samples in the training dataset

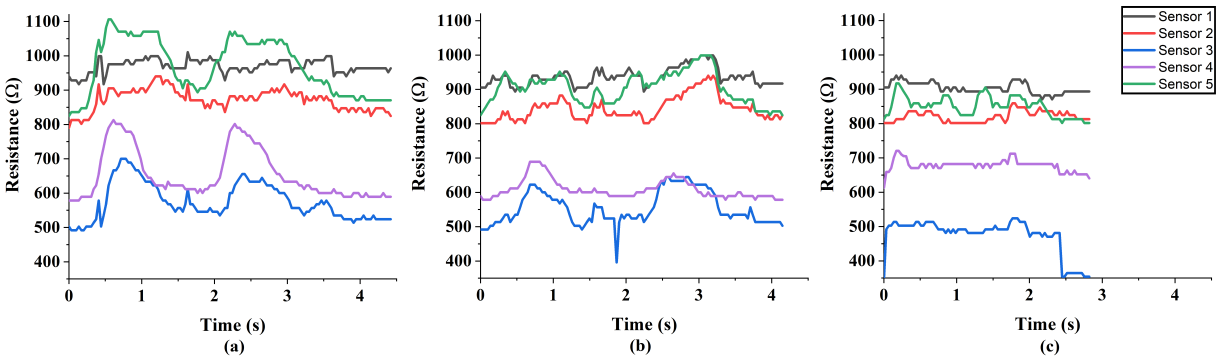

Figure 10: The trajectories of P6S1, P6S10, and P7S4. P6S1 indicates Pattern 6 Sample 1.

From the results in Fig. 9, the FN of DTW and DTW-D becomes smaller as N_{TR} increases. This means that more positive samples in the pattern can be correctly authenticated. The reason should be that the increase of training samples enlarges the state space and included the sample which was previously excluded from this space. However, this leads to the side effect: the FP increases accordingly, i.e., samples that are not belonging to this pattern are also matched incorrectly. This is consistent with the finding from Fig. 7: The increase of N_{TR} does not contribute much to the authentication accuracy for DTW and DTW-D. As a matter of fact, when increasing N_{TR} from 4 to 5, the accuracy of DTW drops by a moderate extent (around 2%). The results show that increasing the number of training samples is not an effective strategy to improve the accuracy of DTW and DTW-D.

In comparison, both FP and FN of LSTM-FCN are decreasing when N_{TR} increases. This indicates that our LSTM-FCN can effectively extract the consistent latent features from the newly-added samples without introducing outlier information. This ensures that the

positive samples are authenticated while the negative ones are denied. This performance boost is significant when increasing N_{TR} from 1 to 3, and this boost is moderate when N_{TR} is 4 and over. Compared with DTW and DTW-D, LSTM-FCN produces a higher authentication accuracy when only one sample is used as the password. The advantage will further improve with the increase of N_{TR} and the accuracy of LSTM-FCN reaches more than 98%. Unlike DTW and DTW-D, we do not need to choose the value of N_{TR} in order to find the trade-off between FP and FN.

4.6 Failure Case

When Pattern 6 is set to the password, Pattern 6 Sample 10 (short-named as P6S10 in Fig. 10 (b), this convention applies to the following paragraphs) can not be matched correctly regardless of the number of training samples. In order to find the reason, we changed our two-class network into a seven-class network. When the number of training samples for each movement pattern increased from one to five, P6S10 was classified as Pattern 7 in 4 out of 5 cases. From the visual demonstration (Fig. 2 (f) and (g)), Pattern 6 is numerically similar to Pattern 7 if the shoulder is not vertically elevated to a sufficient extent. We also used DTW-D to measure the distance between the P6S10 and the rest samples in Pattern 6, and all the samples in Pattern 7. The sample with the minimal distance to P6S10 is P7S4 (Fig. 10 (c)). The distance (7.99) of P6S10-P7S4 is less than the average distance value (9.36) of P6S10 with its intra-pattern samples. This finding actually proves that our method can robustly identify the outliers in a password set.

4.7 Comparison with Learning-based Methods

Recently, some works use CNN for authentication, such as[27] (Ryohei-CNN) and[28] (Matteo-CNN). However, these CNNs cannot obtain a higher classification accuracy under the condition of a small number of samples. We compared our method with these two CNNs using collected data.(Fig. 11).

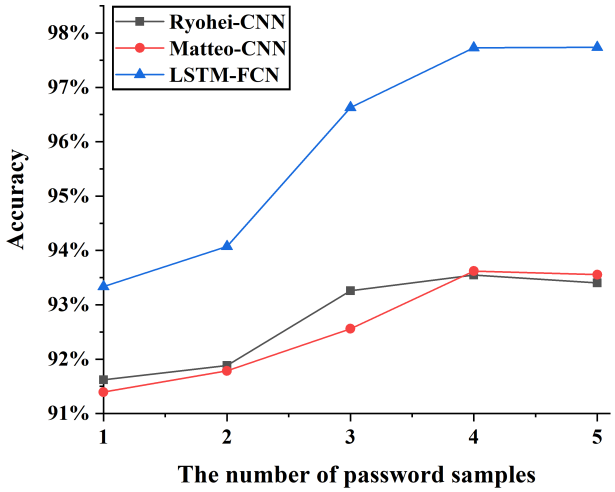

Figure 11: Accuracy of Ryohei-CNN, Matteo-CNN, and LSTM-FCN.

From the results in Fig. 11, when only one sample is configured as a password for training, LSTM-FCN can achieve better classification accuracy than CNNs. And with the increase

of training samples, the accuracy of LSTM-FCN has improved significantly. While, the classification accuracy of CNNs has not improved much. This finding shows that the fully convolutional network with LSTM module can get better classification accuracy in this experiment.

4.8 Comparison of two type data normalization methods

We compared two data normalization methods Feature Normalization (FN), and Temporal Normalization (TN). The result is shown in Fig. 12. From the comparison result, the classification accuracy of TN is much higher than that of FN. In this method, temporal normalization instead of feature normalization can enable our network to obtain a higher classification accuracy.

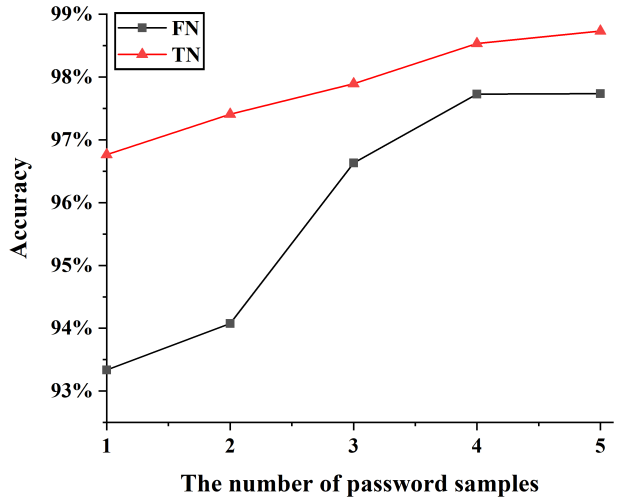

Figure 12: Accuracy of FN, TN.

5 Conclusion

Our work developed a hardware prototype of smart clothes with flexible stretchable sensors. The sensors monitor human joint motion when people are wearing smart cloth and conducting specific movements. We introduce the Long Short-Term Memory Fully Convolutional Network (LSTM-FCN), which directly takes noisy and sparse sensor data as input and matches the user's pre-defined movement patterns. The experimental results confirm the advantage of our method surpasses the representative time-series classification methods (DTW and DTW-D) and CNNs.

This work opens up a few directions for our future work. The first is to deploy our system in a real-world scenario, such as unlocking a mobile phone with our system. our current implementation only deals with the problem of motion pattern recognition, and does not consider identifying a specific person. Achieving the goal of authentication by combining pattern-specific and person-specific information could further improve the security of our system. The other is to validate our method for more subtle activities (e.g., the finger movement). This may require more complex sensor set-up and more powerful sensing algorithms. These applications could further validate the efficacy of our method in practical scenarios of human-computer interaction.

6 Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] Thorepadi Kannappan Thivakaran, SVVNC Padira, Ayyanar Sanjeevi Kumar, and Sureddy Sivar Reddy. Fusion based multimodel biometric authentication system using ear and fingerprint. International Journal of Intelligent Engineering and Systems, 12(1):62–73, 2019.

[2] Neural network approach to iris recognition in noisy environment. Procedia Computer Science, 78:675 – 682, 2016. 1st International Conference on Information Security & Privacy 2015.

[3] Hachim El Khiyari and Harry Wechsler. Face recognition across time lapse using convolutional neural networks. Journal of Information Security, 07(3):141–151, 2016.

[4] Steffen Peter, Bhanu Pratap Reddy, Farshad Momtaz, and Tony Givargis. Design of secure ecg-based biometric authentication in body area sensor networks. Sensors, 16(4):570, 2016.

[5] Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar, and Eli Woods. tslearn: A machine learning toolkit dedicated to time-series data, 2017.

[6] Ali Javed. Multivariate time series dynamic time warping using euclidean distance, November 2019.

[7] Muhammad Yahya, Jawad Ali Shah, Kushsairy Abdul Kadir, Zulkhairi M Yusof, Sheroz Khan, and Arif Warsi. Motion capture sensing techniques usedin human upper limb motion: a review. Sensor Review, 2019.

[8] Hongsheng He, Yan Li, Yong Guan, and Jindong Tan. Wearable ego-motion tracking for blind navigation in indoor environments. IEEE Transactions on Automation Science and Engineering, 12(4):1181–1190, 2015.

[9] Yang Liu, Wuxiong Zhang, Yang Yang, Weidong Fang, Fei Qin, and Xuewu Dai. Pamt: Phase-based acoustic motion tracking in multipath fading environments. pages 2386–2394, 2019.

[10] Hai Yan Song, Jianguo Zhang, and Fang Wang. Study on motion measurement of human upper limb based on electromagnetic tracking system during the activities of daily living. Applied Mechanics and Materials, pages 684–687, 2013.

[11] Yizhai Zhang, Kuo Chen, Jingang Yi, Tao Liu, and Quan Pan. Whole-body pose estimation in human bicycle riding using a small set of wearable sensors. IEEE/ASME Transactions on Mechatronics, 21(1):163–174, 2015.

[12] Timo von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In Proceedings of the European Conference on Computer Vision (ECCV), pages 601–617, 2018.

[13] Yushuang Tian, Xiaoli Meng, Dapeng Tao, Dongquan Liu, and Chen Feng. Upper limb motion tracking with the integration of imu and kinect. Neurocomputing, 159:207–218, 2015.

[14] Dooyoung Kim, Junghan Kwon, Seung Hyun Han, Yonglae Park, and Sungho Jo. Deep full-body motion network for a soft wearable motion sensing suit. IEEE-ASME Transactions on Mechatronics, 24(1):56–66, 2019.

[15] Ruibo Liu, Qijia Shao, Siqi Wang, Christina Ru, Devin Balkcom, and Xia Zhou. Reconstructing human joint motion with computational fabrics. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., 3(1), March 2019.

[16] Oliver Glauser, Shihao Wu, Daniele Panozzo, Otmar Hilliges, and Olga Sorkine-Hornung. Interactive hand pose estimation using a stretch-sensing soft glove. ACM Trans. Graph., 38(4), July 2019.

[17] Mohammad Iman Mokhlespour Esfahani and Maury A Nussbaum. A "smart" undershirt for tracking upper body motions: Task classification and angle estimation. IEEE Sensors Journal, 18(18):7650–7658, 2018.

[18] Siyi Xu, Daniel M. Vogt, Wen-Hao Hsu, John Osborne, Timothy Walsh, Jonathan R. Foster, Sarah K. Sullivan, Vincent C. Smith, Andreas W. Rousing, and Eugene C. and Goldfield. Biocompatible soft fluidic strain and force sensors for wearable devices. Advanced Functional Materials, 29(7):1807058.1–1807058.14, 2019.

[19] Gayathri Rajagopal and Senthil Kumar Manoharan. Personal authentication using multifeatures multispectral palm print traits. The Scientific World Journal, 2015:861629–861629, 2015.

[20] Yi Liu, Jie Ling, Zhusong Liu, Jian Shen, and Chongzhi Gao. Finger vein secure biometric template generation based on deep learning. Soft Computing, 22(7):2257–2265, 2018.

[21] Duo Lu, Kai Xu, and Dijiang Huang. A data driven in-air-handwriting biometric authentication system. In 2017 IEEE International Joint Conference on Biometrics (IJCB), pages 531–537. IEEE, 2017.

[22] Issa Traoré, Youssef Nakkabi, Sherif Saad, Bassam Sayed, Julibio D Ardigo, and Paulo Magella de Faria Quinan. Ensuring online exam integrity through continuous biometric authentication. In Information Security Practices, pages 73–81. Springer, 2017.

[23] Ajay Kumar and Chenye Wu. Automated human identification using ear imaging. Pattern Recognition, 45(3):956–968, 2012.

[24] Yong Kwang Kim and Jong Sub Moon. User authentication using accelerometer sensor in wrist-type wearable device. KIPS Transactions on Computer and Communication Systems, 6(2):67–74, 2017.

[25] Sugang Li, Ashwin Ashok, Yanyong Zhang, Chenren Xu, Janne Lindqvist, and Macro Gruteser. Whose move is it anyway? authenticating smart wearable devices using unique head movement patterns. In 2016 IEEE International Conference on Pervasive Computing and Communications (PerCom), pages 1–9. IEEE, 2016.

[26] Kuo-Hui Yeh, Chunhua Su, Wayne Chiu, and Lu Zhou. I walk, therefore i am: continuous user authentication with plantar biometrics. IEEE Communications Magazine, 56(2):150–157, 2018.

[27] Ryohei Shioji, Shin ichi Ito, Momoyo Ito, and Minoru Fukumi. Personal authentication based on wrist emg analysis by a convolutional neural network. 2017.

[28] M. Gadaleta, L. Merelli, and M. Rossi. Human authentication from ankle motion data using convolutional neural networks. In 2016 IEEE Statistical Signal Processing Workshop (SSP), pages 1–5, 2016.

[29] Xiuhui Wang and Jiajia Zhang. Gait feature extraction and gait classification using two-branch cnn. Multimedia Tools and Applications, 79(3):2917–2930, 2020.

[30] Sudip Vhaduri and Christian Poellabauer. Wearable device user authentication using physiological and behavioral metrics. In 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pages 1–6. IEEE, 2017.

[31] F. Karim, S. Majumdar, H. Darabi, and S. Chen. Lstm fully convolutional networks for time series classification. IEEE Access, 6:1662–1669, 2018.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. pages 1026–1034, 2015.