applied sciences

MDPI

Article

Draw-a-Deep Pattern: Drawing Pattern-Based Smartphone User Authentication Based on Temporal Convolutional Neural Network

Junhong Kim and Pilsung Kang *

School of Industrial & Management Engineering, Korea University, Seoul 02841, Korea; junhongkim@korea.ac.kr
* Correspondence: pilsung_kang@korea.ac.kr; Tel.: +82-2-3290-3383

Abstract: Present-day smartphones provide various conveniences, owing to high-end hardware specifications and advanced network technology. Consequently, people rely heavily on smartphones for a myriad of daily-life tasks, such as work scheduling, financial transactions, and social networking, which require a strong and robust user authentication mechanism to protect personal data and privacy. In this study, we propose draw-a-deep-pattern (DDP)—a deep learning-based end-to-end smartphone user authentication method using sequential data obtained from drawing a character or freestyle pattern on the smartphone touchscreen. In our model, a recurrent neural network (RNN) and a temporal convolution neural network (TCN), both of which are specialized in sequential data processing, are employed. The main advantages of the proposed DDP are (1) it is robust to the threats to which current authentication systems are vulnerable, e.g., shoulder surfing attack and smudge attack, and (2) it requires few parameters for training; therefore, the model can be consistently updated in real-time, whenever new training data are available. To verify the performance of the DDP model, we collected data from 40 participants in one of the most unfavorable environments possible, wherein all potential intruders know how the authorized users draw the characters or symbols (shape, direction, stroke, etc.) of the drawing pattern used for authentication. Of the two proposed DDP models, the TCN-based model yielded excellent authentication performance with average values of 0.99%, 1.41%, and 1.23% in terms of AUROC, FAR, and FRR, respectively. Furthermore, this model exhibited improved authentication performance and higher computational efficiency than the RNN-based model in most cases. To contribute to the research/industrial communities, we made our dataset publicly available, thereby allowing anyone studying or developing a behavioral biometric-based user authentication system to use our data without any restrictions.

Keywords: mobile user authentication; behavioral biometrics; temporal convolution neural network; recurrent neural network; sequence modeling

Citation: Kim, J.; Kang, P. Draw-a-Deep Pattern: Drawing Pattern-Based Smartphone User Authentication Based on Temporal Convolutional Neural Network. *Appl. Sci.* **2022**, *12*, 7590. https://doi.org/10.3390/app12157590

Academic Editor: Javier Hernando

Received: 12 July 2022
Accepted: 26 July 2022
Published: 28 July 2022

1. Introduction

In the hyper-connected world of today, everything is equipped with connectivity. Since 2010, the smartphone has become a key device for networking, and its rapid evolution has provided us with improved quality of life. The smartphone's personal assistant feature can help users with almost everything in their daily lives. However, smartphones have access to a considerable amount of private information of its owner/user; therefore, data privacy and mobile security mechanisms have seen a manifold increase in order to protect the data and privacy of the owner/user against various security attacks attempted by unauthorized users [1]. To provide a safe and reliable mobile experience to users, numerous companies have invested a large amount of capital in both hardware and software development to enhance the security mechanisms for mobile devices.

Among the different layers of a mobile security system, user authentication is the first and most important because a successful unauthorized access to the device would be the worst possible compromise to the user's data and privacy. Consequently, various

studies have been conducted on user authentication for mobile devices [2]. Three well-known user authentication methods for mobile devices with touchscreens are personal identification number (PIN), password, and pattern lock [3]. These methods are known to be vulnerable to certain threat scenarios, such as shoulder surfing attack (SSA) and smudge attack (SA). For example, in SSA, the intruder snoops over the authorized user's shoulder while the user enters their PIN, and later unlocks the device with the same PIN [4]. To overcome the vulnerability of these static information-based user authentication methods, physical biometric-based, e.g., face-, fingerprint-, iris-, blood-vessel-, and voice-based user authentication has been studied [5]. Although physical biometric-based authentication methods reported excellent authentication performance in previous studies, they have limited expandability because they require expensive additional hardware to be installed to recognize the physical biometrics [6]. Moreover, it is well known that physical biometrics can be spoofed; previous studies reported that techniques, such as printed face forgery, copied fingerprint, and copied iris, were successful in gaining unauthorized access to an authorized user's device [7].

To address this problem, human behavioral biometric-based user authentication for mobile devices has been studied. These studies employed various features of the built-in smartphone sensors, such as pressure, velocity, acceleration, and keystroke [8]. However, they have several limitations as follows. First, the authentication performance in them relies considerably on handcrafted features. In other words, theirs is not an end-to-end authentication model that is optimized for raw sequential data. Therefore, information loss of human behavioral characteristics can occur [9]. Second, there are few studies for human behavioral biometric-based user authentication with deep learning methods, such as temporal convolution neural network (TCN) [10] and recurrent neural network (RNN) [11] that have demonstrated outstanding performance for sequential data analytics. Third, these studies demonstrated that the authentication performance decreased rapidly in the case of intrusion attacks, such as SSA and SA.

To overcome the aforementioned limitations, in this study, we propose a draw-a-deep-pattern (DDP) that is a deep learning-based end-to-end smartphone user authentication method using sequential data obtained from drawing a character or freestyle pattern on a smartphone touchscreen. Characteristics of the proposed DDP method are outlined below. First, we used drawing patterns, which can significantly increase the number of input attempts than the authentication types of PIN, 3×3 grid pattern lock [12], and drawing PIN number [13]. Second, we designed an end-to-end model that works with raw sequential data instead of handcrafted features. This model could reduce the loss of behavioral characteristics in each user's raw data. Third, the proposed TCN-based DDP not only yielded superior authentication performance but also significantly reduced both training and inference times than the RNN-based model. Fourth, DDP can be fully implemented through software. Therefore, it has less overheads in terms of physical space and costs than those of physical biometric-based methods that generally require additional hardware. Finally, based on our experimental results, it was observed that DDP can be extremely robust to attack/threat mechanisms, such as SA and SSA. Another contribution of this study is that we make our dataset publicly available to research/industrial communities. Therefore, we hope that our experimental results and dataset will provide an impetus to various research ideas in behavioral biometric-based user authentication in smartphones.

The remainder of this paper is organized as follows. In Section 2, we briefly review related studies. In Section 3, we present the data collection, basic statistic of the collected raw data, preprocessing methods, and authentication algorithms used in this study. In Section 4, we describe the experimental design and discuss the experimental results. Finally, we conclude our study by stating the limitations of the current study and future research directions in Section 5.

2. Related Work

Three well-known user authentication methods using the touchscreen on a smartphone are PIN, password, and pattern lock. These methods use a predefined key from an authorized user, such as n-digit PIN and character password. The main drawback of these methods is their inherent vulnerability to threats, such as SSA. Previous studies have demonstrated that the vision algorithm as a function of the human eye can SSA during the authentication process [14]. Moreover, it is possible to steal the predefined key even if the unauthorized user does not watch the login process. For example, the unauthorized user can obtain the trace of the authorized user's fingertip by illuminating the smartphone display [15]. Another drawback of these methods is the likely overhead of memorizing a complicated pattern that is set to increase the level of security. A simple key pattern can compromise the security during authentication within a small number of attempts, whereas a complex pattern can prove burdensome on the user's memory [16].

To solve the aforementioned problem, several previous studies have attempted to enhance the PIN-based authentication methods. Ref. [17] proposed "draw a secret" (DAS) that uses a freehand drawing on an N × N grid space, which is supposed to increase the number of input attempts for an unauthorized user to compromise the authentication in comparison with the typical PIN-based method. However, the innate problem in DAS is that if the fingertip moves to the grid corner or line, a fuzzy situation that cannot be accepted by the system occurs [18]. Ref. [19] proposed a shifted information-based PIN pad instead of the typical PIN pad, ref. [20] proposed a dictionary-based PIN method, and [21] proposed a spin dial-based PIN. Furthermore, ref. [22] proposed a multimodal-based user authentication method that uses audio information in addition to PIN to protect users from SA. However, these enhanced PIN-based methods have the following limitations. First, these methods increase the complexity of the login process in comparison with typical PIN methods. Second, they inherit the limitations of PIN-based methods, i.e., vulnerability to SSA; if an unauthorized user obtains the authorized user's PIN or rule, the authentication mechanism fails.

To overcome these limitations, physical biometrics, such as iris-, face-, and fingerprint-based user authentication methods, have been studied [13]. They are more robust to SSA than key-based authentication methods, and they provide satisfactory performance [23]. However, they have several drawbacks as follows. First, they can be operated only when the recognition hardware is installed, which increases the physical space and cost overheads. Second, they are vulnerable to a fake iris, face, and fingerprint. Several studies have demonstrated that spoofed physical biometrics can pass the physical biometric-based authentication login process [24]. Third, privacy issues can eventually arise because no one can change their own physical biometrics. Therefore, if an unauthorized user obtains the authorized user's physical biometric information through the aforementioned tricks, the authorized user's identity would be compromised [25].

To overcome the limitations of physical biometric-based authentication, human behavioral biometric-based user authentication has been studied. Ref. [26] proposed multiclass classification-based user authentication with 53 behavioral biometric variables that were collected using a sensor glove and touchscreen. It yielded values of 4.66 and 0.13% for the false acceptance rate (FAR) and false rejection rate (FRR), respectively. However, their method required the extra hardware of a sensor glove in addition to a smartphone. Moreover, they used the data of both authorized and unauthorized users for the multiclass classification method, which is far from a real-world situation. Ref. [27] proposed an enhanced pattern lock method with behavioral biometric features, such as pressure and velocity from a built-in sensor in the smartphone. However, their approach yielded values of 19 and 21% for FRR and FAR, respectively, which indicate a relatively low authentication performance. Ref. [9] proposed a gesture-based user authentication scheme (GEAT) that uses a multitouch gesture on a touchscreen. They obtained seven features that were derived from velocity, acceleration, and touch stroke. Subsequently, they preprocessed the collected sequential sensor data by using the root mean-squared error (RMSE). Finally, they used the radial

basis function-based support vector distribution estimation for the user-authentication model. The experimental results showed values ranging from 0.94 to 0.96 in terms of area under receiver operating characteristic (AUROC) in 10 gestures. Furthermore, up to 5% equal error rate (EER) was observed when they used 10 data samples. Although they experimented in a real-world situation, their method had several drawbacks. First, they used a statistical handcrafted feature derived from sequential data, which can lead to loss of behavioral characteristics in sequential data. Second, theirs is not an end-to-end authentication method that uses raw sequential data but uses handcrafted features instead. Ref. [8] proposed an enhanced PIN-based authentication method. They built an authentication model that used Gaussian estimation, z-score, and standard deviation drift using the collected n-graph-based time and touch features. According to the experimental results, the proposed method yielded 6.81 and 5.59% average EER, while using 50 and 150 16-digit data samples, respectively. Therefore, low authentication performance is a limitation of their study. Ref. [28] proposed keystroke dynamics-based mobile user authentication. They used accelerator and time-based features transformed by a di-graph and tri-graph. However, their method yielded values of 40 and 7% for FRR and FAR, respectively, which indicate low authentication performance. Ref. [13] proposed a two-stage drawing PIN-based authentication that comprises the following steps. In the first step, they built a digit classifier using k-nearest neighbors (k-NN) to recognize the predefined 4-digit PIN. The second step comprises a template match-based user authentication using behavioral biometrics, such as the coordinate, pressure, and size of the touch area. This can mitigate SSA, which is an inherent drawback of the PIN method. Their method yielded an average EER of 4.84% when an unauthorized user knew the predefined 4-digit PIN, which is commonly assumed in a PIN attack scenario. However, for authentication in the imitation scenario, in which an imposter knows both the predefined PIN and the hand movement, the EER increased to 14.11% on average, which is relatively lower than in the PIN attack scenario. Although they experimented in a real-world scenario, their study had several drawbacks. First, if the PIN numbers were misclassified in the first step, then an authentication failure occurred. Second, they used a median-based similarity score for transforming raw sequential data. This can lead to loss of behavioral characteristics in raw sequential data. Third, their method did not provide sufficiently high authentication performance in the imitation attack scenario.

The previous studies on human behavioral biometric-based authentication using a smartphone touchscreen had several drawbacks. First, their methods required a multistage user-authentication process. In other words, they used handcrafted features instead of raw sequential data, thereby causing loss of behavioral characteristics in the raw sequence data. Second, few of these studies are based on deep neural network models that yield excellent performance for sequential data. Third, their methods produced relatively low authentication performance during threat scenarios, such as SSA. Therefore, we propose the DDP method, which is a designed end-to-end authentication model using raw sequential data instead of handcrafted features. It can mitigate the loss of behavioral characteristics in raw sequential data. Second, we use the TCN-based anomaly detection that not only provides superior authentication performance but higher computational efficiency than the RNN-based model in most cases. Third, we demonstrate that the proposed DDP is robust to threat scenarios and suitable for real-world scenarios.

3. Data Collection and Preprocessing

As illustrated in Figure 1, we collected the experimental data based on the developed Android-based data collector. The experiment involved 40 participants, each of whom, in a sitting position, drew 13 predefined patterns 20 times with both hands by using the Samsung Galaxy S8 smartphone. The parameters of the experimental environment are listed in Table 1. As presented in Table 2, the experimented drawing patterns comprised five English, five freestyle, and three Korean patterns.

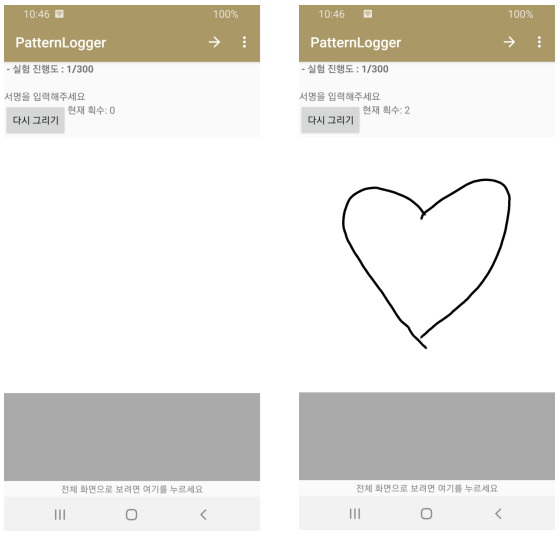


Figure 1. Android application-based data collector. Menus in the guideline in Korean is some supportive information when collecting keystroke data from each participant.

Table 1. Experimental environment.

Experimental Factor	Value
Total number of participants	40
Number of drawing patterns	13
Repetition of each drawing pattern	20
Smartphone device	Samsung Galaxy S8

Table 2. Details of drawing patterns used in this study.

Drawing Category	Number of Patterns
English	5
Freestyle	5
Korean	3

We set one of the most unfavorable experimental environments possible, for the authorized users during the authentication process, wherein all potential intruders know the authenticated user's drawing pattern (e.g., shape, direction, and number of strokes), as illustrated in Figure 2 (for complete information of the patterns used this study, refer to Appendix A). Consequently, we assumed threat scenarios, such as SSA and SA, for our experimental design.

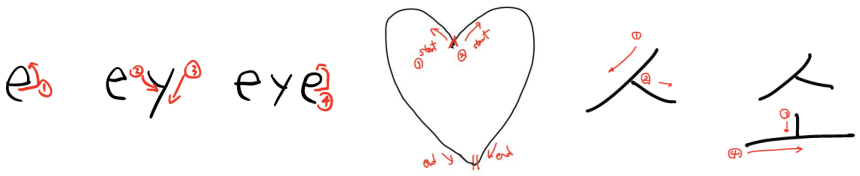


Figure 2. Details of English, Freestyle, and Korean patterns.

The seven variables in the collected raw data are described in Table 3. $Timestamp_t$ denotes the absolute time at which data were collected at time at which data were collected at time t. The unit of $Timestamp_t$ is nanosecond (ns). Act_t denotes the touch state at time t and takes values of D (down), M (move), and U (up). D and U represent the states of contact

and detachment, respectively, of the touch screen with the fingertip. If the fingertip moves on the touchscreen at time t, Act_t is represented as M. X_t and Y_t denote the coordinates of the x- and y-axes at time t, respectively. Ax_t, Ay_t, and Az_t denote the angular velocity for the x-, y-, and z-axes at time t, derived from the accelerometer sensor.

Table 3. Description of seven raw data variables.

Name	Description
$Timestamp_t$	Absolute time (Unit: ns) at time t
Act_t	Touch state at time t {D (Down), M (Move), U (Up)}
X_t	Coordinate of x-axis at time t
Y_t	Coordinate of y-axis at time t
Ax_t	Angular velocity of x-axis at time t
Ay_t	Angular velocity of y-axis at time t
Az_t	Angular velocity of z-axis at time t

Figure 3 illustrates a single line drawn by the participants on the touchscreen using a fingertip. The collected sequential raw data from the single line example are listed in Table 4. D and U are both collected once because of the action of drawing the line once. Table 5 lists the data that transform two consecutive records from Table 4. $T\_Diff_{(t,t-1)}$ is computed using Equation (1), and it refers to the time difference between t and $t+1$. We used the transformed data for building a user-authentication model.

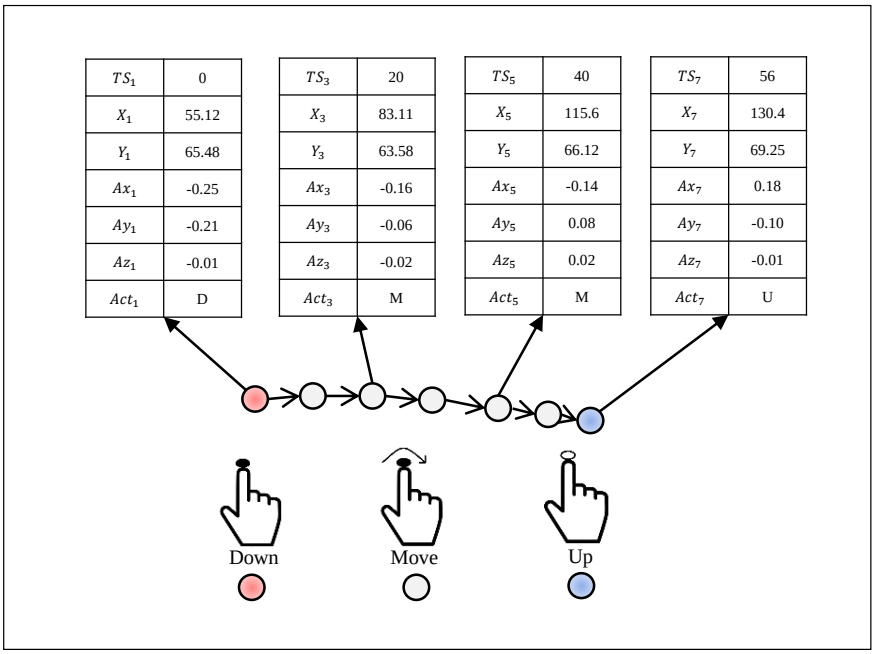


Figure 3. Example of data collection (TS denotes timestamp).

Table 4. Sample of the collected raw data.

Timestamp	*Act*	*X*	*Y*	*Ax*	*Ay*	*Az*
0	D	55.12	65.48	−0.25	−0.21	−0.01
10	M	67.15	63.58	−0.63	0.06	0.02
20	M	83.11	63.58	−0.16	−0.06	−0.02
30	M	95.54	65.48	0.18	−0.21	−0.01
40	M	115.6	66.12	−0.14	0.08	0.02
50	M	128.4	67.25	−0.16	−0.05	−0.02
56	U	130.4	69.25	0.18	−0.10	−0.01

Table 5. Data transformed from the collected raw data.

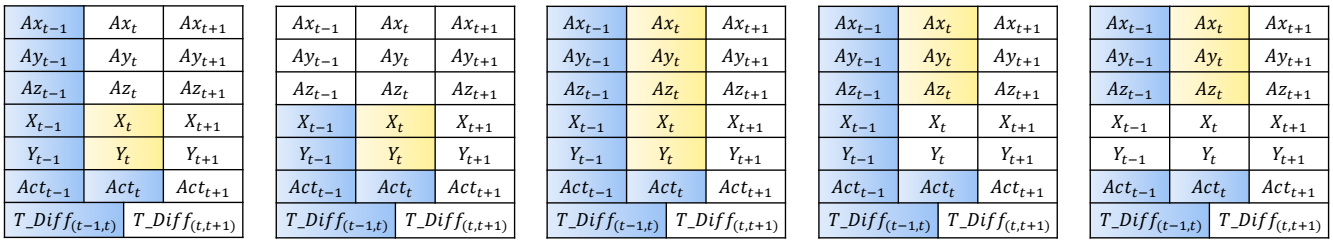
$T\_Diff_{(t,t+1)}$	Act_t	X_t	Y_t	Ax_t	Ay_t	Az_t	Act_{t+1}	X_{t+1}	Y_{t+1}	Ax_{t+1}	Ay_{t+1}	Az_{t+1}
10	D	55.12	65.48	−0.25	−0.21	−0.01	M	67.15	63.58	−0.63	0.06	0.02
10	M	67.15	63.58	−0.63	0.06	0.02	M	83.11	63.58	−0.16	−0.06	−0.02
10	M	83.11	63.58	−0.16	−0.06	−0.02	M	95.54	65.48	0.18	−0.21	−0.01
10	M	95.54	65.48	0.18	−0.21	−0.01	M	115.6	66.12	−0.14	0.08	0.02
10	M	115.6	66.12	−0.14	0.08	0.02	M	128.4	67.25	−0.16	−0.05	−0.02
6	M	128.4	67.25	−0.16	−0.05	−0.02	U	130.4	69.25	0.18	−0.10	−0.01

$$T\_Diff_{(t,t+1)} = Timestamp_{t+1} - Timestamp_t, \quad (1)$$

As presented in Table 6, we used five combinations of the variable set to investigate the effect of each variable category on authentication performance. To develop the sequential model, we used the following input variables: (i) past-to-previous-time variables, Ax, Ay, Az, X, and Y, and (ii) past-to-current-time variables, T_Diff and Act. Then, we used the following output variables from the current time: Ax, Ay, Az, X, and Y. Figures 4 and 5 depict the used variables based on the five combinations of the variable set for the forward and backward sequential structures, respectively. Coordinate-A/B, Angular-A/B, and full variable combination sets use the output variable coordinate-based feature (X,Y), angular velocity-based feature (Ax, Ay, Az), and both, respectively.

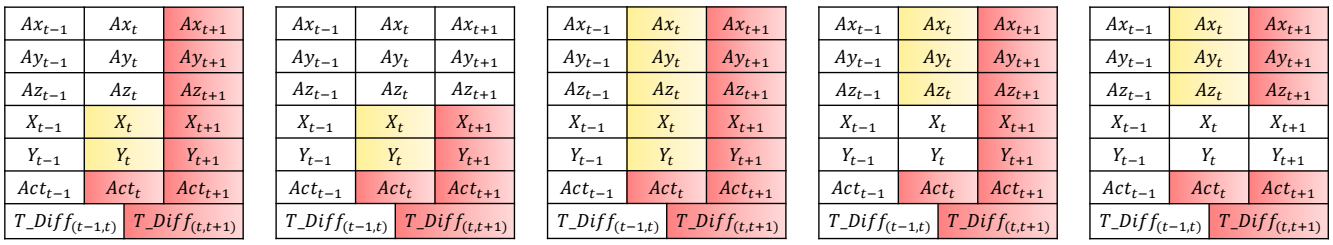
Table 6. Five combinations of variable sets used in the study.

No	Name	Input	Output
1	Coordinate-A	T_Diff, X, Y, Ax, Ay, Az, Act	X, Y
2	Coordinate-B	T_Diff, X, Y, Act	X, Y
3	Full	T_Diff, X, Y, Ax, Ay, Az, Act	X, Y, Ax, Ay, Az
4	Angular-A	T_Diff, Ax, Ay, Az, Act	Ax, Ay, Az
5	Angular-B	T_Diff, Ax, Ay, Az, Act	Ax, Ay, Az



(**a**) Coordinate-A (**b**) Coordinate-B (**c**) Full (**d**) Angular-A (**e**) Angular-B

Figure 4. Input and output variables at time t based on five combinations of variable sets for the forward sequence (blue and yellow denote input and output variables, respectively).



(**a**) Coordinate-A (**b**) Coordinate-B (**c**) Full (**d**) Angular-A (**e**) Angular-B

Figure 5. Input and output variables at time t based on five combinations of variable sets for the backward sequence (red and yellow denote input and output variables, respectively).

4. Data Partition and Modeling

4.1. Data Partition

We divided the collected raw data for each pattern into 10 sets of training and test data each. The average and standard deviation obtained from the length of each drawing pattern and the *p*-value obtained from the two-sided paired *t*-test are presented in Table 7. The *p*-value result indicates that there was no statistically significant difference between the lengths of the training and test datasets. In this study, we used the training data of 10 authorized users to build a user-authentication model. Then, we authenticated 400 test datasets based on the trained user-authentication model for each pattern, which comprised 10 authorized users' test data and 390 intruders' test data (10 test data from 39 participants each).

Table 7. Statistics pertaining to training and test data from each collected drawing dataset.

Pattern Name	Stroke Count	Avg. Sequence of Training Set	Std. of Training Set	Avg. Sequence of Test Set	Std. of Test Set	*p*-Value
English 1	4	59.63	11.37	59.52	11.88	0.88
English 2	4	75.94	13.12	75.98	13.33	0.96
English 3	5	61.27	10.43	61.16	10.36	0.87
English 4	4	69.65	10.51	69.38	10.29	0.68
English 5	4	69.19	11.83	69.01	12.21	0.81
Freestyle 1	4	82.54	16.88	82.78	17.23	0.82
Freestyle 2	2	52.88	13.47	52.85	13.31	0.98
Freestyle 3	5	92.58	21.47	92.68	21.00	0.94
Freestyle 4	2	67.78	14.46	67.77	14.75	0.99
Freestyle 5	6	124.70	25.87	124.69	26.24	1.00
Korean 1	4	41.47	6.99	41.34	6.97	0.78
Korean 2	5	70.23	14.16	70.22	14.03	1.00
Korean 3	5	60.90	10.00	61.03	10.24	0.84

4.2. Modeling

4.2.1. TCN

TCN has demonstrated excellent performance with sequential data in various studies [29]. Especially, it exhibited superior performance in sequence classification, such as vision, natural language processing (NLP), and music fields, in comparison with Vanilla RNN, gated recurrent unit, and long short-term memory (LSTM) [10]. In this study, we used dilated causal 1D convolution [30] for DDP to use the available information up to the current time. The 1D dilated causal convolution, which is computed by Equation (2), is illustrated in Figure 6, where s, d, and k denote 1D sequence input, dilation rate, and filter size, respectively. Furthermore, $F(t)$ denotes the result of the convolution operation at time t.

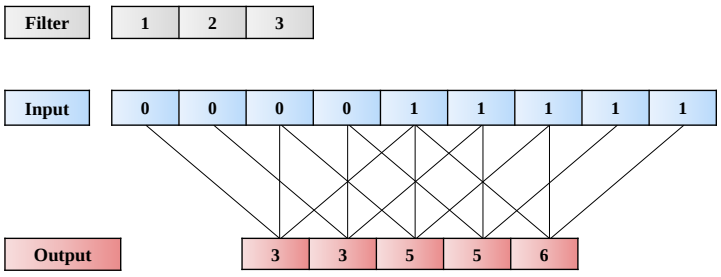


Figure 6. Example of 1D dilated causal convolution (filter size = 3, stride = 1, dilation rate = 2, zero padding = 4).

$$F(t) = (\mathbf{x} *_d f)(t) = \sum_{j=0}^{k-1} f(j) \cdot \mathbf{s}_{t-d \cdot j}. \tag{2}$$

The proposed TCN-based DDP model is illustrated in Figure 7. If we use the FULL variable combination set, the variables input ($I_{(t,t-1)}$) and output (O_t) at time t for the forward structure are represented by Equations (3) and (4), respectively.

$$I_{(t,t-1)} = \{Act_t, Act_{t-1}, X_t, Y_t, Ax_t, Ay_t, Az_t, T\_Diff_{(t,t-1)}\}, \tag{3}$$

$$O_{(t)} = \{X_t, Y_t, Ax_t, Ay_t, Az_t\}. \tag{4}$$

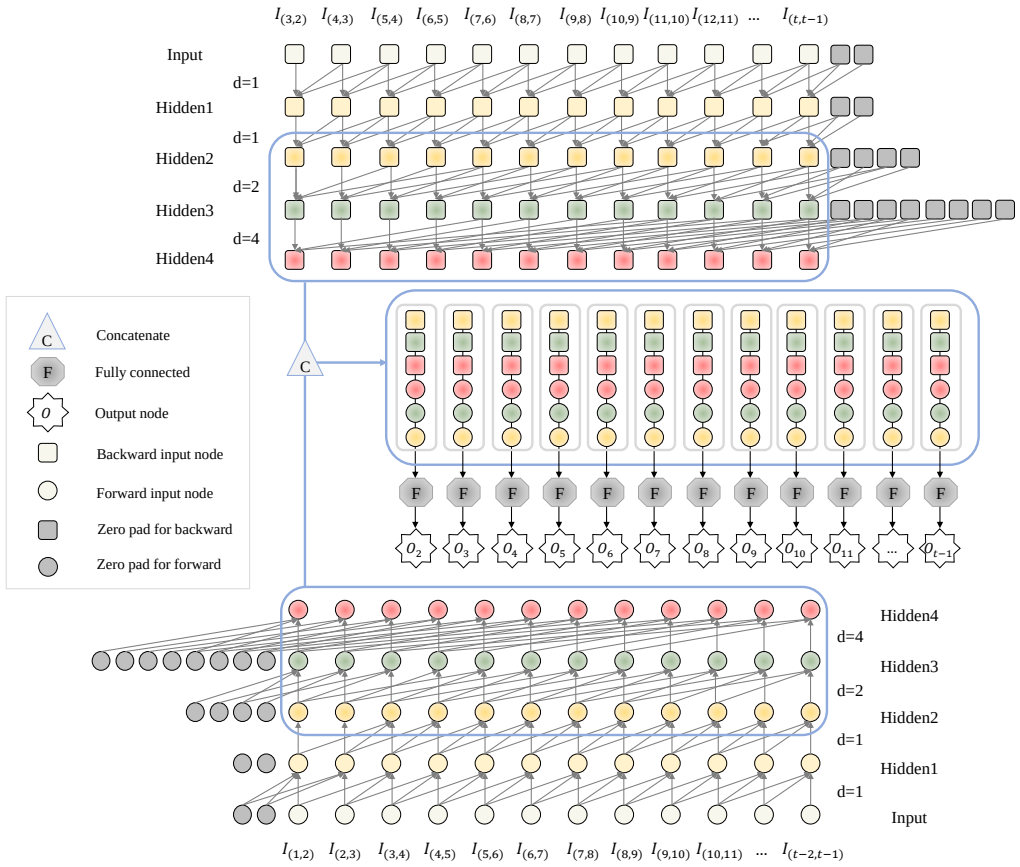


Figure 7. Proposed TCN-based DDP model.

The proposed TCN-based DDP model comprises forward and backward sequential structures. We set four convolution layers with dilation rates of 1, 1, 2, and 4 for the forward and backward structures. After each convolution layer, layer normalization [31] is performed and the rectified linear unit (ReLU) [32] is used as the activation function. The parameter details of this model are summarized in Table 8. The parameters kernel size, kernel depth, and fully connected (FC) hidden nodes are set to 3, 20, and 64, respectively. We use concatenated output features calculated from the second, third, and fourth hidden layers to use the FC input features. One of the characteristics of the calculated output features by dilated 1D convolution is that the output features from a higher hidden layer have a larger receptive field than those obtained from the lower hidden layer. Therefore, the designed concatenated output vector for FC input features can optimize both short- and long-term dependency simultaneously. The number of proposed TCN parameters for the five combinations of the variable set used are listed in Table 9, which confirms that the

proposed networks have about 17k parameters according to the number of input (nI) and output variables (nO).

Table 8. Network parameters for proposed TCN-based DDP model.

Forward		Backward
1D Conv3 20 (d = 1)		1D Conv3 20 (d = 1)
1D Conv3 20 (d = 1)		1D Conv3 20 (d = 1)
1D Conv3 20 (d = 2)		1D Conv3 20 (d = 2)
1D Conv3 20 (d = 4)		1D Conv3 20 (d = 4)
	Concatenate	
	FC-64	
	FC-nO	

Table 9. Number of TCN parameters according to five combinations of variable sets.

Method	Coordinate-A	Coordinate-B	Full	Angular-A	Angular-B
nI	12	9	12	12	10
nO	2	2	5	3	3
Total	16,674	16,314	16,869	16,739	16,499

4.2.2. RNN

RNN has provided excellent performance in various sequential data applications, such as NLP, voice, and time-series signals [33]. In this study, we used a basic LSTM cell for the authentication model because its performance is the same as that of the other variants of the RNN cell [34]. The LSTM cell is illustrated in Figure 8, and Equations (5)–(10) provide the computation of each component in the figure. The line on the top denotes the cell state, which is the internal memory, whereas that on the bottom indicates the hidden state. We denote the cell and the hidden state by c and h, respectively. The input gate, i, forget gate, f, output gate, o, and internal hidden state, g, are the distinctive features of LSTM, and they are designed to prevent the vanishing gradient problem [35]. During training, the weights and bias in each gate are trained. The forget gate, f_t, can control how much the previous hidden state, $h_{(t-1)}$, should be preserved in the current state, whereas the input gate, i_t, can control how much the current input, x_t, should be reflected in the current state. The output gate, o_t, regulates the amount of hidden state, $h_{(t-1)}$, in the next sequence and the internal hidden state, g_t, is calculated from the input, x_t, and the previous hidden state $h_{(t-1)}$.

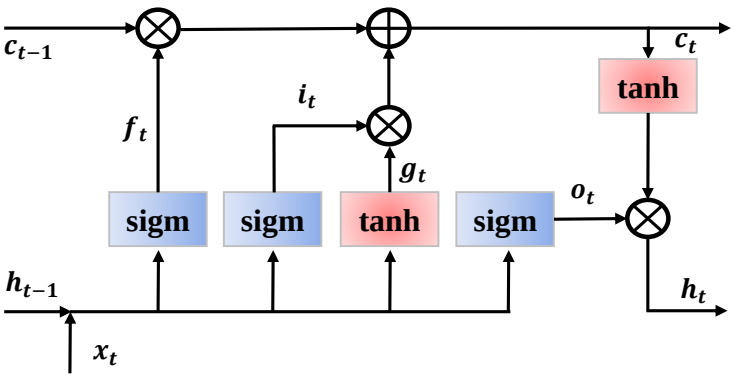


Figure 8. LSTM cell.

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f), \tag{5}$$
$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i), \tag{6}$$
$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o), \tag{7}$$
$$g_t = tanh(W_g * [h_{t-1}, x_t] + b_g), \tag{8}$$
$$c_t = (c_{t-1} \otimes f_t) \oplus (g_t \otimes i_t), \tag{9}$$
$$h_t = tanh(c_t) \otimes o_t. \tag{10}$$

The RNN-based DDP model used in this study is illustrated in Figure 9, and its parameter details are provided in Table 10. The number of hidden nodes of LSTM and FC are set to 33 and 64, respectively. The number of RNN parameters for the used five combinations of the variable set is presented in Table 11, which confirms that most networks have 17k parameters, which is similar to the case of TCN models.

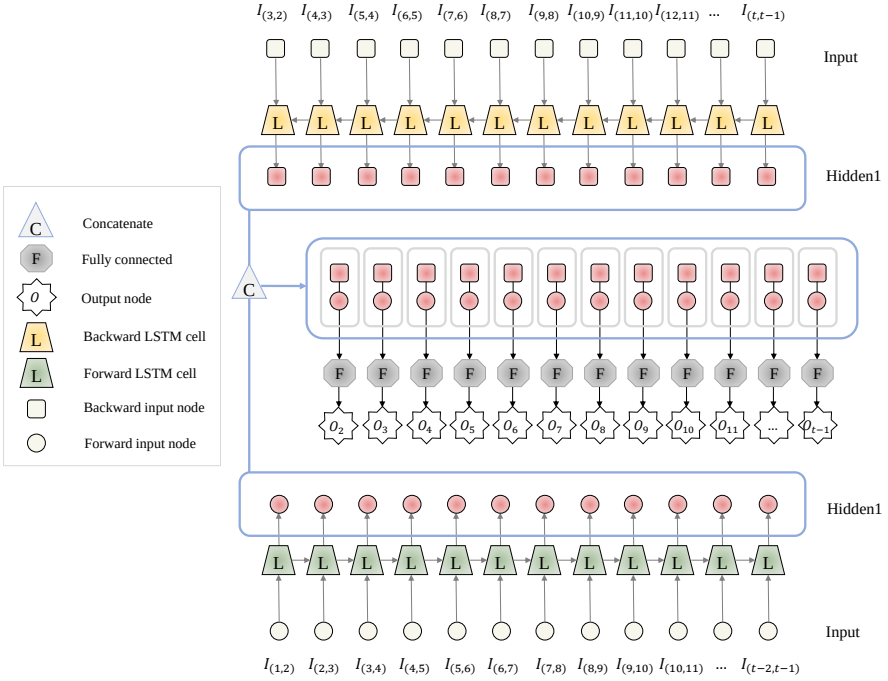


Figure 9. Structure of RNN-based DDP model.

Table 10. Network parameters for the proposed RNN-based DDP model.

Forward		Backward
LSTM-33		LSTM-33
	Concatenate	
	FC-64	
	FC-nO	

Table 11. Number of RNN parameters according to five combinations of variable sets.

Method	Coordinate-A	Coordinate-B	Full	Angular-A	Angular-B
nI	12	9	12	12	10
nO	2	2	5	3	3
Total	16,434	15,642	16,629	16,499	15,971

4.2.3. Training Details

We use standardization for the continuous variables based on the training dataset; additional training guide details are provided in Table 12. We used Huber loss [36] with $\delta = 1$ for the loss function because it is less sensitive to outliers than the mean squared error (MSE). The Huber loss equation is expressed as Equation (11). The learning rate starts at 10^{-1} and decays by 10^{1} for every 100 iterations with the RMSProp optimization. We set the variables batch size, total number of iterations, and weight initialization method to 10, 300, and He initialization [32], respectively.

Table 12. Training guide details.

Loss	Batch Size	Learning Rate	Optimization Method	Initialization Method	Number of Iterations
Huber loss	10	10^{-2}–10^{-5}	RMSProp	He	300

$$L_\delta(y, f(x)) \begin{cases} \frac{1}{2}(y - f(x))^2, & \text{if } |y - f(x)| \le \delta, \\ \delta|y - f(x)| - \frac{1}{2}\delta^2, & \text{otherwise.} \end{cases} \tag{11}$$

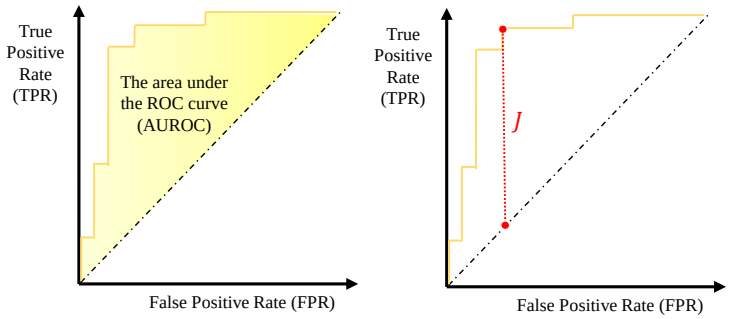
4.2.4. Novelty Score and Performance Measure

As shown by Equation (12), the novelty score that we used is the average Huber loss of all sequences:

$$Novelty\ score = \frac{\sum_{t=1}^{T} L_\delta(y_t, f(x_t))}{T}. \tag{12}$$

To compare the performance of the TCN- and RNN-based DDP models, we used AUROC [37]. AUROC, which is depicted in Figure 10a, is commonly adopted as a cut-off-independent performance metric. To obtain the true positive rate (TPR) and the false positive rate (FPR), we use the best cut-off value when the maximum Youden index is obtained [38]. The Youden index, J, which is expressed by Equation (13), is illustrated in Figure 10b.

$$J = TPR_{cut-off} - FPR_{cut-off}. \tag{13}$$



(**a**) (**b**)

Figure 10. Two types of performance indices. (**a**) AUROC; (**b**) Youden index (J) by threshold.

5. Experimental Results

The average and standard deviation of AUROC for TCN and RNN are presented in Tables 13 and 14, respectively. The following observations can be made from these tables. First, the Coordinate-A-based model (estimating the x and y coordinates based on the available time, x and y coordinates, and angular information) produced the best result for both TCN and RNN for all cases except one (Korean 3). Furthermore, the TCN model

based on the FULL method produced average AUROC of 0.001 greater than that of the Coordinate-A-based TCN model. In the case of Korean 3, the difference in the performance of the aforementioned models is extremely small. Second, by comparing the performance of the two models, Coordinate-B, i.e., excluding the angular velocity-based features, and Angular-B, i.e., excluding the coordinate-based features, we confirmed that the former exhibited relatively superior performance for all cases.

$$\mu_{TCN} > \mu_{RNN} \tag{14}$$

Table 13. Average AUROC result obtained by the TCN model. Numbers in parentheses denote standard deviation and the text in bold denotes the best performance for each drawing pattern. E, F, and K denote English, freestyle, and Korean patterns, respectively.

Model	Feature	E1	E2	E3	E4	E5	F1	F2	F3	F4	F5	K1	K2	K3
TCN	Coordinate-A	**0.997 (0.008)**	**0.997 (0.007)**	**0.997 (0.006)**	**0.990 (0.021)**	**0.995 (0.02)**	**0.989 (0.025)**	**0.989 (0.022)**	**0.997 (0.007)**	**0.992 (0.015)**	**0.992 (0.018)**	**0.990 (0.017)**	**0.996 (0.011)**	0.993 (0.015)
TCN	Coordinate-B	0.985 (0.028)	0.990 (0.017)	0.986 (0.022)	0.976 (0.036)	0.988 (0.025)	0.979 (0.035)	0.978 (0.030)	0.989 (0.019)	0.981 (0.024)	0.976 (0.040)	0.978 (0.033)	0.980 (0.032)	0.984 (0.028)
TCN	Full	0.995 (0.015)	0.990 (0.020)	0.991 (0.022)	0.988 (0.023)	0.991 (0.021)	0.984 (0.025)	0.983 (0.030)	0.982 (0.036)	0.980 (0.039)	0.972 (0.044)	0.986 (0.030)	0.994 (0.015)	**0.994 (0.013)**
TCN	Angular-A	0.983 (0.037)	0.965 (0.082)	0.981 (0.041)	0.968 (0.066)	0.957 (0.094)	0.961 (0.066)	0.967 (0.050)	0.957 (0.083)	0.964 (0.064)	0.941 (0.112)	0.965 (0.103)	0.981 (0.044)	0.988 (0.021)
TCN	Angular-B	0.947 (0.111)	0.938 (0.143)	0.950 (0.089)	0.940 (0.103)	0.929 (0.147)	0.925 (0.141)	0.925 (0.143)	0.934 (0.100)	0.942 (0.112)	0.919 (0.144)	0.951 (0.116)	0.960 (0.124)	0.961 (0.114)

Table 14. Average AUROC result obtained by RNN model. Numbers in parentheses denote standard deviation and the text in bold denotes the best performance for each drawing pattern. E, F, and K denote English, freestyle, and Korean patterns, respectively.

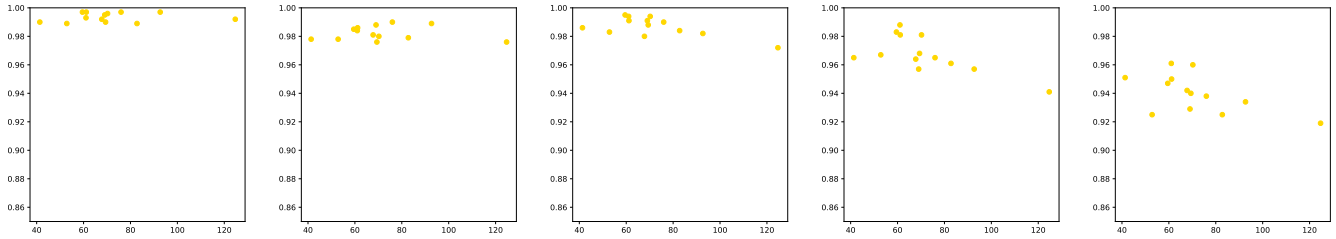
Model	Feature	E1	E2	E3	E4	E5	F1	F2	F3	F4	F5	K1	K2	K3
RNN	Coordinate-A	**0.998 (0.005)**	**0.994 (0.013)**	**0.996 (0.010)**	**0.987 (0.022)**	**0.993 (0.019)**	**0.992 (0.011)**	**0.988 (0.024)**	**0.992 (0.012)**	**0.986 (0.027)**	**0.990 (0.014)**	**0.990 (0.022)**	**0.996 (0.009)**	0.990 (0.020)
RNN	Coordinate-B	0.977 (0.034)	0.982 (0.025)	0.977 (0.029)	0.970 (0.036)	0.978 (0.039)	0.972 (0.027)	0.971 (0.034)	0.978 (0.025)	0.963 (0.039)	0.958 (0.049)	0.963 (0.044)	0.968 (0.035)	0.975 (0.034)
RNN	Full	0.988 (0.026)	0.982 (0.030)	0.986 (0.028)	0.980 (0.034)	0.974 (0.054)	0.967 (0.047)	0.959 (0.070)	0.972 (0.040)	0.966 (0.052)	0.960 (0.055)	0.980 (0.040)	0.990 (0.023)	**0.990 (0.018)**
RNN	Angular-A	0.981 (0.041)	0.956 (0.090)	0.980 (0.037)	0.960 (0.065)	0.951 (0.093)	0.943 (0.083)	0.946 (0.091)	0.956 (0.063)	0.946 (0.087)	0.941 (0.076)	0.974 (0.049)	0.974 (0.050)	0.982 (0.036)
RNN	Angular-B	0.956 (0.087)	0.934 (0.139)	0.949 (0.083)	0.938 (0.077)	0.915 (0.155)	0.919 (0.111)	0.901 (0.163)	0.934 (0.063)	0.910 (0.133)	0.896 (0.111)	0.941 (0.135)	0.952 (0.130)	0.961 (0.091)

Based on these results, it can be concluded that coordinate-based features are more suitable than angular velocity-based features for user authentication. Third, although all participants knew the authenticated user's drawing pattern (e.g., shape, direction, and number of strokes) in our experimental design, the Coordinate-A-based model yielded higher than 0.990 average AUROC regardless of the algorithm or pattern type used in this study. Therefore, it can be confirmed that the proposed DDP is robust to threat scenarios, such as SSA and SA. Moreover, we can expect superior user-authentication performance in real-world scenarios than that shown by our experimental results, because the authenticated user's drawing pattern may not be available to the intruders in the former. Fourth, in most cases, the TCN-based model exhibited better authentication performance than the RNN-based model. Table 15 presents the p-value of the paired t-test for the hypothesis shown in Equation (14). In most cases, the p-value is less than or equal to 0.05, which indicates that the authentication performance improvement of the TCN-based model and the RNN-based model is statistically significant in general.

Table 15. *p*-value of one sided paired *t*-test of average AUROC of TCN and RNN models. Text in bold denotes *p*-value less than or equal to 0.05. E, F, and K denote English, freestyle, and Korean patterns, respectively.

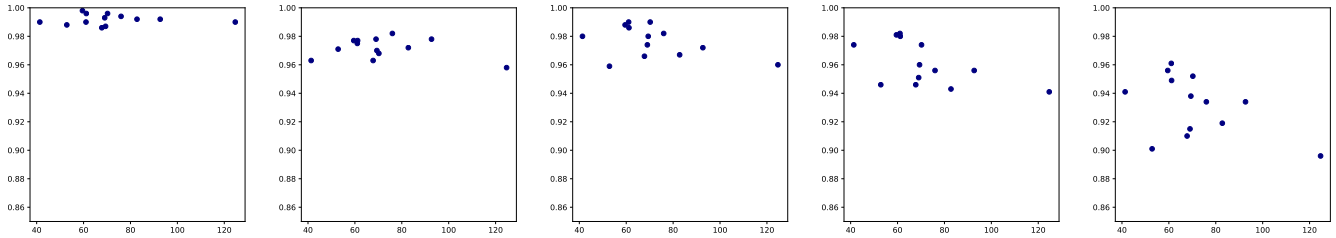
Feature	E1	E2	E3	E4	E5	F1	F2	F3	F4	F5	K1	K2	K3
Coordinate-A	0.60	**0.03**	0.23	0.22	0.07	0.80	0.39	**0.00**	**0.05**	0.30	0.46	0.59	0.07
Coordinate-B	**0.00**	**0.00**	**0.00**	**0.05**	**0.00**	0.06	**0.01**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**
Full	**0.01**	**0.00**	**0.00**	**0.00**	**0.01**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.01**	**0.01**	**0.01**
Angular-A	0.20	**0.01**	0.44	**0.04**	0.14	**0.00**	**0.01**	0.47	**0.02**	0.49	0.77	0.09	0.07
Angular-B	0.87	0.21	0.48	0.43	**0.02**	0.29	**0.00**	0.49	**0.00**	0.10	**0.00**	**0.00**	0.52

The average AUROC for the TCN- and RNN-based models according to the lengths of the used 13 patterns and for the 5 combinations of the variable sets are depicted in Figures 11 and 12, respectively. It can be observed that for both TCN and RNN, the authentication performance of the angular feature-based models (Angular-A and Angular-B) deteriorates when the pattern length increases. In contrast, the coordinate-based models were relatively less affected by this trend. This is because the gyroscope in the smartphone that measured the angular velocity is relatively less precise that the touch sensor that captured the coordinate features. Hence, angular-based features were more likely to be affected by the noise when the sequence length increased. Among them, the Coordinate-A-based model exhibits the best user-authentication performance and the best robustness to pattern length in comparison with other variable combination sets.



(**a**) Coordinate-A (**b**) Coordinate-B (**c**) Full (**d**) Angular-A (**e**) Angular-B

Figure 11. Average AUROC according to sequence length for each variable combination set by TCN (x-axis: sequence length, y-axis: AUROC).



(**a**) Coordinate-A (**b**) Coordinate-B (**c**) Full (**d**) Angular-A (**e**) Angular-B

Figure 12. Average AUROC according to sequence length for each variable combination set by RNN (x-axis: sequence length, y-axis: AUROC).

The training and inference time consumption while using Coordinate-A-based TCN and RNN models are presented in Tables 16 and 17, respectively. The main difference between these two models is that TCN conducts convolutional operations simultaneously but RNN conducts the LSTM operation sequentially. Therefore, when longer sequential data are used, RNN required considerably more time than TCN to process the entire data. Therefore, in all experimental cases, both the training and inference times for TCN are relatively smaller than those for RNN. Based on the experimental results, the minimum and maximum values of the ratio of RNN's consumption of training time to that of TCN are

4.66 and 11.89, respectively. Furthermore, the minimum and maximum values of the ratio of RNN's consumption of inference time to that of TCN are 5.25 and 12.66, respectively.

Table 16. Time consumption during 300 iterations for training each algorithm based on Coordinate-A method (unit of time: seconds, batch size = 10, measurement by RTX 2080-Ti GPU). E, F, and K denote English, freestyle, and Korean patterns, respectively.

	E1	E2	E3	E4	E5	F1	F2	F3	F4	F5	K1	K2	K3
Sequence length	59.63	75.94	61.27	69.65	69.19	82.54	52.88	92.58	67.78	124.70	41.47	70.23	60.90
TCN	4.58	4.57	4.58	4.66	4.57	4.62	4.55	4.66	4.61	4.61	4.55	4.69	4.54
RNN	30.10	36.70	30.44	33.99	32.80	37.35	25.72	40.71	31.07	54.88	21.19	31.46	28.30
Time ratio RNN over TCN	6.58	8.03	6.65	7.29	7.18	8.08	5.65	8.73	6.73	11.89	4.66	6.71	6.23

Table 17. Time consumption during an iteration of inference for each algorithm based on Coordinate-A method (unit of time: milliseconds, batch size = 10, measurement by RTX 2080-Ti GPU. E, F, and K denote English, freestyle, and Korean patterns, respectively).

	E1	E2	E3	E4	E5	F1	F2	F3	F4	F5	K1	K2	K3
Sequence length	59.52	75.98	61.16	69.38	69.01	82.78	52.85	92.68	67.77	124.69	41.34	70.22	61.03
TCN	7.46	7.92	7.01	7.16	7.35	8.51	7.56	6.56	6.76	7.28	7.11	7.46	7.50
RNN	53.87	72.80	57.92	63.31	56.20	64.73	46.81	64.26	59.61	92.18	37.33	52.17	50.20
Time ratio RNN over TCN	7.23	9.19	8.27	8.84	7.64	7.61	6.19	9.80	8.82	12.66	5.25	7.00	6.69

The bubble plot in Figure 13 illustrates the time consumption for different patterns. It can be observed that the processing time based on the TCN structure is not significantly affected by the pattern length; however, that based on the RNN structure is highly dependent on the pattern length—the longer the pattern, the longer the processing time. This independence on the patten length of TCN can be a significant advantage when an authentication algorithm is implemented in an edge device with an insufficient computational capacity. Figure 14 illustrates the time consumption ratio of RNN to TCN with respect to the average sequence length of a pattern. It is clearly seen that the processing time for RNN rapidly increases in comparison with that of TCN; therefore, the ratio on the y-axis increases when the pattern length increases.

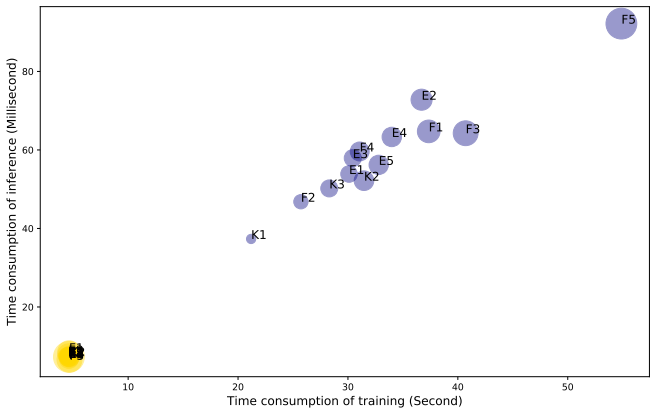


Figure 13. Training and inference time consumption for each length of each drawing pattern. Bubble size corresponds to the length of sequence. E, F, and K denote English, freestyle, and Korean patterns, respectively.

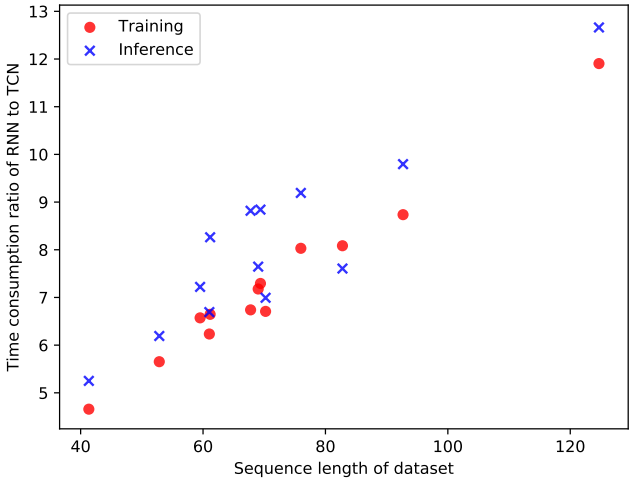


Figure 14. Training and inference time ratios of RNN to TCN by sequence length.

The average FAR and FRR values obtained by the proposed TCN-based DDP model for the used 13 patterns are listed in Table 18. The average FAR and FRR values of this model are in the ranges of 0.67–2.38% and 0.50–2.25%, respectively, and the total average FAR and FRR values are 1.41 and 1.23%, respectively. Therefore, this model exhibits fairly good user-authentication performance, even if all potential intruders know the authenticated user's drawing pattern. Therefore, it can prove to be robust to threat scenarios, such as SSA and SA.

Table 18. Average FAR and FRR values obtained from proposed TCN-based DDP model for each pattern based on the Coordinate-A method (E, F, and K denote English, freestyle, and Korean patterns, respectively, and AVG denotes average value).

Valid Index	E1	E2	E3	E4	E5	F1	F2	F3	F4	F5	K1	K2	K3	AVG
FAR	0.81	0.76	2.02	0.93	1.15	1.73	1.86	1.08	2.38	1.22	2.77	0.93	0.67	1.41
FRR	0.50	0.75	0.50	2.25	0.75	1.25	2.25	0.75	1.00	1.75	1.75	0.75	1.75	1.23
AVG	0.66	0.76	1.26	1.59	0.95	1.49	2.06	0.92	1.69	1.49	2.26	0.84	1.21	1.32

In conclusion, the proposed TCN-based DDP model provides statistically superior authentication performance and higher computational efficiency than the RNN-based model in most cases. Moreover, the time consumption ratios from RNN to TCN for training and inference increase linearly with the sequence length. Therefore, the TCN-based DDP model is more suitable than the RNN-based DDP model for smartphone user authentication using drawing patterns. From the perspective of computational time for real-world implementation, the proposed model requires a training time of less than 5 s and inference time of less than 9 ms, which is acceptable in a real-time service scenario.

6. Conclusions

In this study, we propose DDP—a deep learning-based end-to-end smartphone user authentication method using sequential data obtained from drawing a character or freestyle pattern on a smartphone touchscreen. The proposed method uses raw sequential human behavioral data instead of handcrafted features. The behavioral data are transformed using a statistical method. We collected raw data from 40 participants who drew 13 patterns each 20 times. The patterns comprised five English, five freestyle, and three Korean patterns. We set one of the most unfavorable experimental environments possible for the user-authentication process, wherein all potential intruders know the authenticated user's drawing pattern. In other words, we assumed threat scenarios such as SSA and SA for our experimental design.

Based on the experimental results, it was observed that the Coordinate-A-based user-authentication model yielded the best authentication performance in most cases. Additionally, the Coordinate-A-based TCN model produced excellent user-authentication performance, i.e., 0.99%, 1.41%, and 1.23% in terms of average AUROC, FAR, and FRR, respectively. We expect superior user-authentication performance in real-world user-authentication scenarios than that shown by our experimental results, because in the former, the authenticated user's drawing pattern may not be known to the intruder. In other words, the unauthorized user's login attempts can be blocked if the number of attempts for the pattern strokes exceeds its maximum allowable threshold. Based on the result, it can be seen that the Coordinate-A-based TCN model exhibited superior user-authentication performance and robustness for threat scenarios, such as SSA and SA.

Next, we compared the authentication performance between the TCN- and RNN-based models. The experimental results demonstrated that the proposed TCN-based DDP model provided statistically superior authentication performance than the RNN-based model in most cases. Additionally, both training and inference time consumption for the TCN-based model were relatively smaller than those for the RNN-based model. Therefore, the proposed TCN-based DDP model not only exhibited superior authentication performance but proved to be more computationally efficient than the RNN-based model in most cases. Finally, we made our dataset publicly available to research and industrial communities. Therefore, we hope that the results of our study and the shared data will contribute to further research in behavioral biometric-based user authentication for smartphones.

Despite the promising experimental results, the current study faces the following limitations that lead us to future research directions. First, although the proposed model has a sufficiently small number of parameters, it is beneficial to compress the network without performance loss. Second, this study used 10 training datasets for building an authentication model. Therefore, whether high performance can be preserved even with fewer training data for the purpose of user convenience must be studied. Third, we used the coordinates of the touchscreen, angular velocity, and time-based features. Therefore, whether performance improvement can be achieved when data from other built-in sensors are used must be studied.

Author Contributions: J.K. initiated the research idea and carried out the experiment. He also wrote the draft of the paper. P.K. wrote and finalized the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2022R1A2C2005455). This work was also supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00471, Development of Autonomous Control Technology for Error-Free Information Infrastructure Based on Modeling & Optimization).

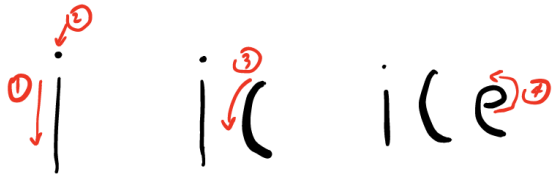
Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Figures A1–A3 show the drawing patterns of 'English', 'Freestyle', and 'Korean' used in our study.



(**a**) English 1

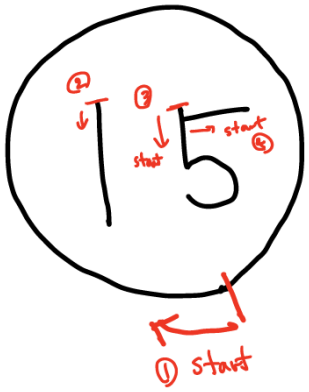
(**b**) English 2

(**c**) English 3

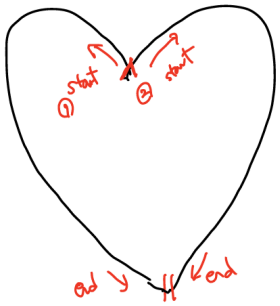
(**d**) English 4

(**e**) English 5

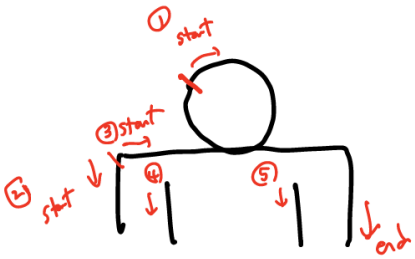
Figure A1. Drawing patterns of the 'English' drawing pattern used in this study.



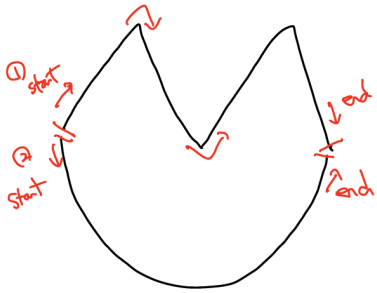
(**a**) Freestyle 1



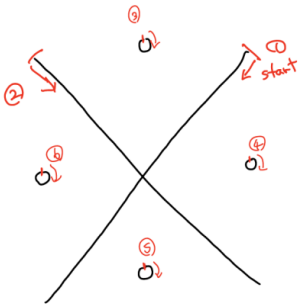
(**b**) Freestyle 2



(**c**) Freestyle 3

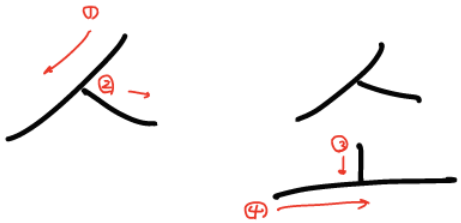


(**d**) Freestyle 4



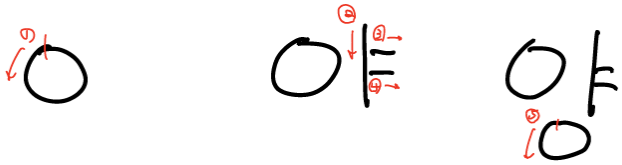
(**e**) Freestyle 5

Figure A2. Drawing patterns of the 'Freestyle' drawing pattern used in this study.



(**a**) Korean 1

(**b**) Korean 2



(**c**) Korean 3

Figure A3. Drawing patterns of the 'Korean' drawing pattern used in this study.

References

1. Kim, J.; Kang, P. Analyzing International Collaboration and Identifying Core Topics for the "Internet of Things" Based on Network Analysis and Topic Modeling. *Int. J. Ind. Eng.* **2018**, *25*, 349–369.
2. Ibrahim, T.M.; Abdulhamid, S.M.; Alarood, A.A.; Chiroma, H.; Al-garadi, M.A.; Rana, N.; Muhammad, A.N.; Abubakar, A.; Haruna, K.; Gabralla, L.A. Recent advances in mobile touch screen security authentication methods: A systematic literature review. *Comput. Secur.* **2019**, *85*, 1–24. [CrossRef]
3. Furnell, S.; Clarke, N.; Karatzouni, S. Beyond the pin: Enhancing user authentication for mobile devices. *Comput. Fraud. Secur.* **2008**, *2008*, 12–17. [CrossRef]
4. Sae-Bae, N.; Ahmed, K.; Isbister, K.; Memon, N. Biometric-rich gestures: A novel approach to authentication on multi-touch devices. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Austin, TX, USA, 5–10 May 2012; pp. 977–986.
5. Orrù, G.; Marcialis, G.L.; Roli, F. A novel classification-selection approach for the self updating of template-based face recognition systems. *Pattern Recognit.* **2020**, *100*, 107121. [CrossRef]
6. Jain, A.K.; Ross, A.; Pankanti, S. Biometrics: A tool for information security. *IEEE Trans. Inf. Forensics Secur.* **2006**, *1*, 125–143. [CrossRef]
7. Ratha, N.K.; Connell, J.H.; Bolle, R.M. Enhancing security and privacy in biometrics-based authentication systems. *IBM Syst. J.* **2001**, *40*, 614–634. [CrossRef]
8. Teh, P.S.; Zhang, N.; Teoh, A.B.J.; Chen, K. TDAS: A touch dynamics based multi-factor authentication solution for mobile devices. *Int. J. Pervasive Comput. Commun.* **2016**, *12*, 127–153. [CrossRef]
9. Shahzad, M.; Liu, A.X.; Samuel, A. Secure unlocking of mobile touch screen devices by simple gestures: You can see it but you can not do it. In Proceedings of the 19th Annual International Conference on Mobile Computing & Networking, Miami, FL, USA, 30 September–4 October 2013; pp. 39–50.
10. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
11. Su, B.; Lu, S. Accurate recognition of words in scenes without character segmentation using recurrent neural network. *Pattern Recognit.* **2017**, *63*, 397–405. [CrossRef]
12. Andriotis, P.; Oikonomou, G.; Mylonas, A.; Tryfonas, T. A study on usability and security features of the android pattern lock screen. *Inf. Comput. Secur.* **2016**, *24*, 53–72. [CrossRef]
13. Van Nguyen, T.; Sae-Bae, N.; Memon, N. DRAW-A-PIN: Authentication using finger-drawn PIN on touch devices. *Comput. Secur.* **2017**, *66*, 115–128. [CrossRef]

14. Shukla, D.; Kumar, R.; Serwadda, A.; Phoha, V.V. Beware, your hands reveal your secrets! In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3–7 November 2014; pp. 904–917.
15. Aviv, A.J.; Gibson, K.L.; Mossop, E.; Blaze, M.; Smith, J.M. Smudge Attacks on Smartphone Touch Screens. *Woot* **2010**, *10*, 1–7.
16. Lindemann, R. The evolution of authentication. In *ISSE 2013 Securing Electronic Business Processes*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 11–19.
17. Jermyn, I.; Mayer, A.; Monrose, F.; Reiter, M.K.; Rubin, A.D. *The Design and Analysis of Graphical Passwords*; USENIX Association: Berkeley, CA, USA, 1999.
18. Dunphy, P.; Yan, J. Do background images improve draw a secret graphical passwords? In Proceedings of the 14th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 28–31 October 2007; pp. 36–47.
19. Yan, Q.; Han, J.; Li, Y.; Zhou, J.; Deng, R.H. Designing leakage-resilient password entry on touchscreen mobile devices. In Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, Hangzhou, China, 8–10 May 2013; pp. 37–48.
20. Staneková, L.; Stanek, M. Analysis of dictionary methods for PIN selection. *Comput. Secur.* **2013**, *39*, 289–298. [CrossRef]
21. Aly, Y.; Munteanu, C.; Raimondo, S.; Wu, A.Y.; Wei, M. Spin-lock gesture authentication for mobile devices. In Proceedings of the 18th International Conference on Human–Computer Interaction with Mobile Devices and Services Adjunct, Florence, Italy, 6–9 September 2016; pp. 775–782.
22. Lee, M.K.; Nam, H.; Kim, D.K. Secure bimodal PIN-entry method using audio signals. *Comput. Secur.* **2016**, *56*, 140–150. [CrossRef]
23. Yager, N.; Dunstone, T. The biometric menagerie. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *32*, 220–230. [CrossRef] [PubMed]
24. Gupta, P.; Behera, S.; Vatsa, M.; Singh, R. On iris spoofing using print attack. In Proceedings of the 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; pp. 1681–1686.
25. Prabhakar, S.; Pankanti, S.; Jain, A.K. Biometric recognition: Security and privacy concerns. *IEEE Secur. Priv.* **2003**, *1*, 33–42. [CrossRef]
26. Feng, T.; Liu, Z.; Kwon, K.A.; Shi, W.; Carbunar, B.; Jiang, Y.; Nguyen, N. Continuous mobile authentication using touchscreen gestures. In Proceedings of the 2012 IEEE Conference on Technologies for Homeland Security (HST), Waltham, MA, USA, 13–15 November 2012; pp. 451–456.
27. De Luca, A.; Hang, A.; Brudy, F.; Lindner, C.; Hussmann, H. Touch me once and i know it's you!: Implicit authentication based on touch screen patterns. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Austin, TX, USA, 5–10 May 2012; pp. 987–996.
28. Corpus, K.R.; Gonzales, R.J.D.; Morada, A.S.; Vea, L.A. Mobile user identification through authentication using keystroke dynamics and accelerometer biometrics. In Proceedings of the 2016 IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft), Austin, TX, USA, 14–22 May 2016; pp. 11–12.
29. Geng, Y.; Su, L.; Jia, Y.; Han, C. Seismic Events Prediction Using Deep Temporal Convolution Networks. *J. Electr. Comput. Eng.* **2019**, *2019* . [CrossRef]
30. Lea, C.; Flynn, M.D.; Vidal, R.; Reiter, A.; Hager, G.D. Temporal convolutional networks for action segmentation and detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 156–165.
31. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
32. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
33. Zhang, X.; LeCun, Y. Which encoding is the best for text classification in chinese, english, japanese and korean? *arXiv* **2017**, arXiv:1708.02657.
34. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 2222–2232. [CrossRef] [PubMed]
35. Kim, J.; Kang, P. Recurrent neural network-based user authentication for freely typed keystroke data. *arXiv* **2018**, arXiv:1806.06190.
36. Niu, J.; Chen, J.; Xu, Y. Twin support vector regression with Huber loss. *J. Intell. Fuzzy Syst.* **2017**, *32*, 4247–4258. [CrossRef]
37. DeLong, E.R.; DeLong, D.M.; Clarke-Pearson, D.L. Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics* **1988**, *44*, 837–845. [CrossRef]
38. Hilden, J.; Glasziou, P. Regret graphs, diagnostic uncertainty and Youden's Index. *Stat. Med.* **1996**, *15*, 969–986. [CrossRef]