Journal of Theoretical and Applied Information Technology
31st January 2025. Vol.103. No.2
© Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

KEYSTROKE DYNAMICS ON MULTI-SESSION AND UNCONTROLLED SETTINGS USING CNN BI-LSTM

SEPTIO RAHMAN PUTRA[1], ANDRY CHOWANDA[2]
[1]Computer Science Department, BINUS Graduate Program - Master of Computer Science
[2]Computer Science Department, School of Computer Science
Bina Nusantara University
Jakarta 11480, Indonesia
[1]septio.putra@binus.ac.id, [2]achowanda@binus.edu

ABSTRACT

Keystroke dynamics is a behavioral biometric that identifies individuals based on their typing style and rhythm. It is a non-intrusive and low-cost authentication method that does not require additional hardware. This study addresses the challenge of continuous authentication using keystroke data collected from multiple sessions and long periods. A hybrid Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (Bi-LSTM) model was developed to capture both spatial and temporal dependencies in the keystroke data. The research process involved data collection from 10 users over six months, preprocessing to remove irrelevant data, and feature extraction to transform the data into a usable format. The model achieved impressive performance with Equal Error Rates (EERs) ranging from 0.009 to 0.127, demonstrating its effectiveness in continuous authentication scenarios.

Keywords. *Keystroke Dynamics, Biometric Authentication, CNN, Bi-LSTM,RNN, Classification, Performance Evaluation, EER.*

1. INTRODUCTION

In several years the are increasing cases of users who got attacked in cyber security risk. The highest source of cybersecurity risk comes from user behavior/actions, where most cases occur due to the culture of sharing passwords between users when using an account [1]. Based on research conducted [2], 2 out of 3 users of an information system have shared their account passwords with colleagues in an organization. This is very risky because providing account information it can give other people access to authenticate as genuine users into the system. Authentication is the process of confirming a user's identity before allowing a user to enter or access the system. This process generally involves several inputs, such as entering a name and password to confirm identity and validate user authenticity [3], [4]. When a user enters someone else's name and password, the user can enter the system with their account. This became a problem because no one can detect whether the user who has successfully authenticated is the actual user with access to the systems. Most systems utilize traditional one-time authentication methods, such as fingerprint, face, and voice recognition based on biometrics; however, these methods have some drawbacks because they require specific hardware devices, which can impact the user experience that leads to a less intuitive and seamless interaction with the system [5]. Along with it, keystroke dynamics is a promising form of behavioral characteristics that can be used in existing problems to achieve the purpose of the authentication process.

Keystroke dynamics is a multi-factor authentication approach using behavioral biometrics that uses a user's unique typing style and rhythm pattern to validate their identity [6]. This passive biometric does not require the installation of any additional hardware or software on the user's computer [7]. Detecting a user's keystroke habits can continuously verify the user's identity without affecting user input, information extracted from the habits in types such as the time interval between key presses or releases that are used to classify different people. This strategy offers several benefits, including uniqueness, simplicity of execution, and non-invasiveness [8]. Keystroke dynamics can be used in static and dynamic authentications [9]. Static authentication occurs at the user's initial interaction with the system, while dynamic authentication verifies that the current user is the same as the one initially granted access [10].

Several research studies on keystroke dynamics authentication have studied authentication mechanisms based on fixed texts (such as passwords and ID numbers) and free text (freely typing during regular interaction with the system) but are limited to the selection and improvement of algorithms. However, one significant challenge in implementing free-text keystroke dynamics is developing an effective deep learning model that can handle inter-session variations and maintain accuracy over extended periods. The current research on continuous authentication via free text has mainly used a small number of typing patterns where it expects that a user's typing style will not change. Nevertheless, it is often the case that the typing style of a user changes over a time period, and there remains a lack of in-depth studies on continuous authentication with keystroking patterns based on free text over a long period/several session.

For authentication purposes, the keystroke dynamics model needs to be robust and able to handle the flexibility and variability of free text. A model needs to collect different scenarios in the enrollment phase and recognize the pattern of users. This research aims to address the challenges by developing a dep learning model for continuous free-text authentication that can effectively adapt to and recognize evolving user typing patterns while maintaining high accuracy across different sessions. The proposed approach focuses on building a practical and accurate continuous authentication model using free text data from a several session in long period using deep learning that expected can contribute to the practical use of continuous user authentications.

The remaining content of this paper organizes as follows. Section II describes the related work and model performance comparisons. Section III presents the methodology used in this research. Section IV discusses results and evaluation. At last, section V present concluding remarks and suggestions for future work.

2. RELATED WORKS

2.1. Authentication

Authentication is defined as a process of verifying the identity of someone or something based on their claims [11]. Several authentication models have been proposed in the literature, like ownership or possession-based, knowledge-based, and inherent-based models [12], [13]. Each model has its strengths and weaknesses, such as knowledge-based authentication that relies on the user's knowledge (such as password, PIN code, lock pattern, etc.). This model is easy to implement and use but is also susceptible to brute-force and dictionary attacks [12]. The next model is the ownership or possession model, which refers to something that the user has (such as a smart card, NFC, RFID, etc.) [12], [13]. Ownership or possession authentication requires the user to have a physical device, such as a security token or a smart card, which adds an extra layer of protection and raises the risk of losing or forgetting the device [12].

In comparison, Inherent-based authentication uses the physiological or behavioral characteristics of the user, which is also called biometrics authentication[11]. Biometrics authentication is more secure and reliable, but it also faces challenges such as high costs and accuracy issues due to the use of sunglasses and surgery that may affect the authentication process [14]. In addressing the limitations of biometric authentication, keystroke dynamics is proposed as a viable alternative. Keystroke dynamics is a type of behavioral biometrics that measures how a user types on a keyboard, and it is one of the most cost-effective biometric authentications for personal computers [15].

2.2. Keystroke Dynamics

Keystroke dynamics is a behavioral biometric that captures a user's typing rhythm on a keyboard. It analyzes the quantitative features of the user's typing pattern, such as the duration and latency of keystrokes, to identify the user or verify their identity [16]. Studies by the National Science Foundation (NSF) and the National Institute of Standards and Technology (NIST) in the United States have confirmed that typing patterns are unique to each typist. Keystroke dynamics offers a simple and natural way to enhance computer security. There are two types of keystroke analysis: static and dynamic. The static analysis uses predetermined text for all the users, while dynamic analysis monitors the keystrokes continuously or periodically during and after the log-in session [11].

Keystroke dynamics consist of two main phases: enrollment and authentication/verification. In the enrollment phase, the user's keystroke biometric data is captured, processed, and stored as a reference file in a database. This file serves as a template for future authentication operations. In the authentication/verification phase, the user's keystroke biometric data is acquired and processed again. The system compares the new keystroke data with the stored reference templates and makes an authentication decision based on the similarity score. The authentication is successful if the score exceeds a predefined threshold [11]. For authentication purposes, the keystroke dynamics model must be robust and able to handle the accuracy issues of biometric authentication caused by potential limitations of certain biometric traits [14]. In the case of free-text keystroke dynamics, the emotion and time of day could affect the user's typing pattern [17]–[24]. It is difficult to determine the typing trend because it could change over time in a fluctuating manner [23]. Therefore, a robust model needs to collect different scenarios in the enrollment phase and recognize the pattern of a specific user in the authentication/verification phase.

2.3. Classification

Classification is a critical task in keystroke dynamics, especially for authentication. Traditional machine learning algorithms have been widely used for this task but often fail to capture the subtle timing features between keys [10]. The researcher has discussed using deep learning methods for classification in keystroke dynamics to overcome the challenge.

The research of [21] introduces TypeNet, a Siamese Recurrent Neural Network (RNN) that can authenticate users based on short and few keystroke sequences. The model evaluated on the Aalto University Dataset contains more than 136 million keystrokes from 168K users typing on physical keyboards. They achieved an Equal Error Rate (EER) of 4.8% using only five enrollment sequences and 1 test sequence per user with 50 keystrokes per sequence. Another research [25] presents a methodology for optimizing the parameters of a Bidirectional Recurrent Neural Network (BRNN) using a modified Dipper Throated Optimization (DTO) algorithm. The modified DTO algorithm assigns dynamic weights to three search leaders instead of one to improve the optimization results. The research evaluates the proposed methodology on two datasets and compares it with other optimization and machine learning techniques. The results show that the proposed optimized BRNN achieves the best classification accuracy of 99.02% and 99.32% for the two datasets. [22] propose a method that integrates key names and two kinds of timing features through an embedding mechanism. Using a Bidirectional Long Short-Term Memory (Bi-LSTM) network, the method captures the relationship between context keystrokes. The experimental results show that TKCA achieves state-of-the-art performance with an Equal Error Rate (EER) of 8.28% when using only 30 keystrokes and 2.78% of EER when using 190 keystrokes.

Another classification algorithm that can be used for keystroke dynamics is Convolutional Neural Network (CNN). The research [4] applies CNN to the CMU keystroke benchmark dataset and performs feature engineering techniques using Deep Feature Synthesis (DFS) algorithm and Synthetic Minority Oversampling Technique (SMOTE). The method achieves an average equal error rate (EER) of 0.009, significantly lower than the previous methods in the literature. Another research [10] proposes a method for continuous authentication by free-text keystroke based on CNN and recurrent neural networks (RNN). The method uses CNN to extract the sequential and combined features of the keystrokes and RNN to learn the individual keystroke features for identity authentication. The method also applies the sliding window and packet structure to realize continuous identity authentication and designs the intrusion decision condition to explore the practical continuous authentication mechanism.

3. METHODOLOGY

This section presents our methodology proposition for the problem of continuous authentication in free-text keystroke dynamics as shown in Figure 1. The first contribution is to propose mechanisms for dataset collection that provide enough data from different sessions. The data preprocessing is conducted to transform the keystroke sequence into a keystroke feature sequence to train the model. The classification process performs by using the combination of Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (Bi-LSTM) to train the keystroke data, which results in a robust keystroke model; The detailed process explains in Figure 1:

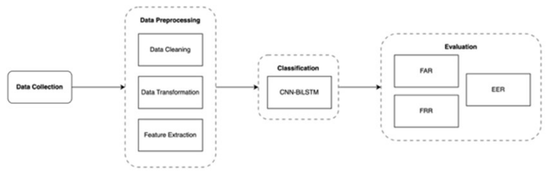


Figure 1: Methodology

3.1. Data Collection

Different methods have been used to collect keystroke data in previous studies. [25] used screen-recorded videos to capture typing process of users within an apllication by tracking and detecting a text-cursor object. [24] used a JavaScript API deployed in online computer systems. JavaScript provides a high-resolution time of up to 1 ms, essential for capturing the subtle features of keystroke dynamics. In this research, we also use a JavaScript API and store the data in JSON format. The API was embedded in the username login form, which was the most frequently used input form. The data was collected from a live website running in production with 80 active users. However, this study focused only on 10 users. The pseudocode for the data collection process is shown as follow:

Pseudocode 1: Data Collection

```
Create an empty array

function record_key_event(evt):

      array.append({code:evtwhich,evt.typ
e:current_time()})

      keystroke.value=JSON.stringify(arra
y)

on_keydown = record_key_event

on_keyup = record_key_event
```

To capture more time-sensitive data, we collected information on key codes, key press times, and key release times in milliseconds timestamps from each user. The amount of data collected per user varied depending on their intensity and frequency of app use. We did not impose any experimental settings during data collection. The only requirement was that the user must be genuinely logged in to the system. We did not restrict the device used, time, or demography of the user. This was done to understand the natural environment and spontaneous typing behavior of keystroke dynamics implementation in real life. However, this also meant that we had to perform further preprocessing to filter out noise and outliers and extract relevant keystroke analysis features.

3.2. Data Preprocessing

Data preprocessing involves various tasks such as data cleaning, transformation, and feature extraction. Because this study involved data collection, data preprocessing was important to ensure the performance of the model.

3.2.1. Data cleaning

Data cleaning was performed to remove noise, errors, and inconsistences from the data, such as missing values, outliers, and duplicates.

3.2.2. Data transformation

The Dataset resulting from the data cleaning was transformed into keystroke sequences. Keystroke sequences are sequences of keystroke vectors representing the input of a piece of text. The keystroke data were divided into fixed-length sequences and converted into keystroke vectors according to the time characteristics of the keystroke [10]. The format of the keystroke sequences used in this study is illustrated in Figure 2 Each keystroke sequence represents the typing of two consecutive characters in chronological order. The first $[X]_1$ and second is $[X]_2$. This pattern is repeated for the subsequent rows, where $[X]_n$ indicates the previous character of $[X]_{n+1}$.

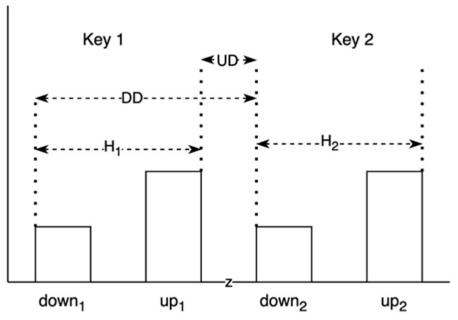


Figure 2: Keystroke sequences format

3.2.3. Feature extraction

The feature extraction process aims to capture the characteristics of the keystroke sequences. These features based on the relational features proposed by [12], which measure the duration and interval of each keystroke event. The features are defined as follow:

$$\begin{aligned} H_n &= up_n - down_n \\ DD_n &= down_{n+1} - down_n \\ UD_n &= down_{n+1} - up_n \\ H_{(n+1)-n} &= H_n - H_{n+1} \\ B_n &= H_n + UD_n + H_{n+1} \end{aligned} \quad (1)$$

3.3. Classification

The model use in this research is combination of Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (Bi-LSTM) to process keystroke data. The CNN model extracts relevant features from the keystroke data, while the RNN model captures the temporal dependencies in the keystroke sequences. This hybrid approach aims to improve the accuracy and robustness of the recognition system.

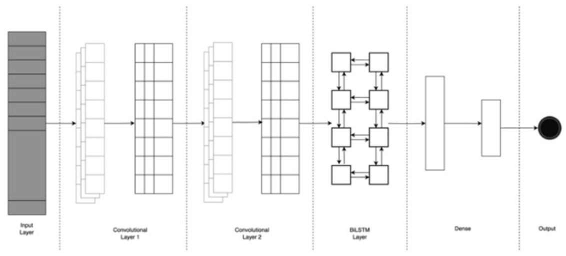


Figure 3: CNN+Bi-LSTM Model

3.3.1. CNN layer

CNN Automatically generates useful and discerning features from raw data [26]. A one-dimensional CNN is used as a local feature extractor. CNN adds a preprocessing stage and extends the Bi-LSTM neural network. In the processing stage, useful features are extracted from the original data, which improves the accuracy of subsequent predictions [26]. The activation function used in the convolutional layers is the rectified linear unit (ReLU), which defined as:

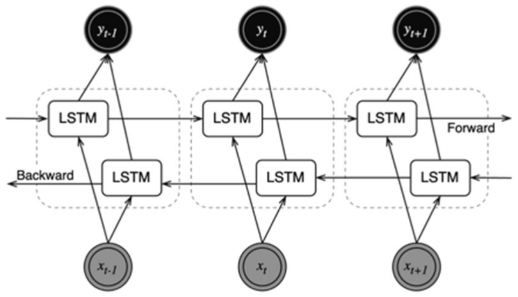
$$f(x) = \max(0, x) \tag{2}$$

the architecture is further extended by incorporating another convolutional layer, consisting of 128 filters, each convolutional layer has an additional maxpooling1d layer with 'same' padding to reduce the spatial size of the feature maps while preserving crucial information. L2 regularization is added to prevent overfitting by adding a penalty term to the loss function based on the squared magnitude of the model weights. L2 regularization formula, which defines the regularization term as the sum of the squares of all the feature weights:

$$L_2\, regularization = \|\omega\|_2^2 = \omega_1^2 + \omega_2^2 + \cdots + \omega_n^2 \tag{3}$$

3.3.2. Bi-LSTM layer

The Long Short-Term Memory (LSTM) model is a special form of Recurrent Neural Network (RNN) that provides feedback on each neuron. LSTM uses gates to avoid gradient issues in long sequences. Unlike traditional RNN and LSTM, Bi-LSTM has two sets of LSTM that learn forward and backward sequence information. This makes the prediction more comprehensive and reduces the lag problem. The structure of Bi-LSTM is shown in Figure 4, and the LSTM cell in Figure 4 which has three gates: input, output and forget. These gates control how much information is stored, updated, or discarded in the cell state [26].



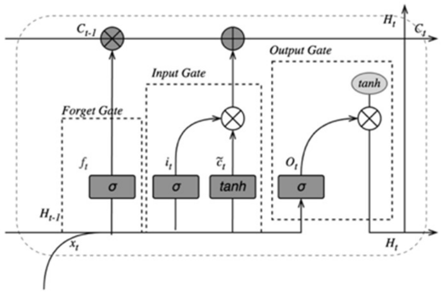


Figure 4: Bi-LSTM Structure Unit & LSTM Cell

The formulas for the gating units of the Bi-LSTM model are as follows:

$$\begin{aligned} i_t &= \sigma_t(x_t W_{xi} + h_{t-1} W_{hi} + b_i) \\ f_t &= \sigma_f\left(x_t W_{xf} + h_{t-1} W_{hf} + b_f\right) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \sigma_c(x_t W_{xc} + h_{t-1} W_{hc} + b_c) \\ \sigma_t &= \sigma_0(x_t W_{x0} + h_{t-1} W_{h0} + b_0) \\ h_t &= o_t \odot \sigma_h(c_t) \end{aligned} \tag{4}$$

3.3.3. Output

The output layer of this architecture is a fully connected layer with size 8,4 respectively in the final output of one neuron with a sigmoid function, which is suitable for binary classification. The sigmoid function maps any input value to a value between 0 and 1, representing the probability of belonging to the positive class. the sigmoid function defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

3.4. Evaluation

User identity verification is a crucial task in biometric security systems, which requires a reliable and effective evaluation metric. One such metric is Equal Error Rate (EER), which is widely used in one-class classification problems [27]. EER is defined as the point where the False Acceptance Rate (FAR) and the False Rejection Rate (FRR) are equal for a given acceptance threshold [28]. EER is defined as:

$$EER = FAR = FRR \tag{6}$$

FAR and FRR can be defined as:

$$FAR = \frac{FP}{FP + TN} \qquad FRR = \frac{FN}{FN + TP} \tag{7}$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

FAR is the percentage of how many illegal users successfully enter the system, while FRR is the percentage of how many genuine users fail to enter the system. EER provides a balance between security and convenience, as a lower EER indicates a higher accuracy and usability of the biometric system.

4. RESULT

4.1. Data Collection

The data stored in JSON format consists of 4671 raw records of keystroke events, such as key press and release times and key codes.. The data was collected over a period of 6 months from users who logged in to a website. There was no control over the device, login frequency, and login time of the users. The data was collected naturally without any intervention. The number of data collected per user varied depending on the user's intensity in using the website.

4.2. Data Preprocessing

4.2.1. Data cleaning

The first phase of data preprocessing is data cleaning where data is checked and corrected for errors and inconsistencies. in this research, the data cleaning involves two steps: removing the data that doesnt contain keystroke values and removing the data that does not contain key codes in the keystroke data. The first step elimininates of 57 data rows, reducing the dataset size to 4614 row for further processing. The second step is focus on removing any records that do not contain a key code. This cleaning process is significant, removing 2709 entries resulting in 1905 data row. The large number of data entries remove in this step is due to many users not typing the username because they use the save password feature on the browser. Despite the absence of keystroke data, the API continues to record these instances, leading to a surplus of irrelevant data points.

The current condition of the data can be visualized in Figure 5. This figure provides a clear representation of the remaining data's distribution. The decision to concentrate on these ten users is also influenced by the nature of the remaining data. This suggests that the keystroke data from the remaining users shows a flat distribution or minimal variation, which may not contribute significantly to the model's training or evaluation. In essence, the research focuses on the ten users with the most keystroke data to ensure sufficient variability of session for evaluation of model performance.

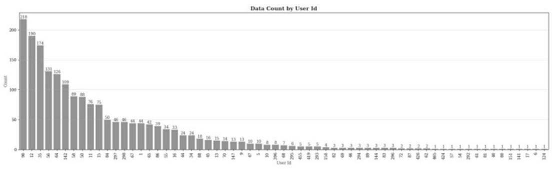


Figure 5: Distribution of data

4.2.2. Data transformation

In the initial phase of data transformation, we begin by deconstructing JSON into individual keystroke values, generating a distinct column for each key press and release. This procedure produces a total of 118 columns, suggesting that the longest character string a user has inputted into this dataset is 59 characters.

Moving forward, we perform a minor cleaning of the dataset, specifically eliminating any character that does not have a corresponding pair of key press and key release. The existence of unpaired keystrokes indicates possible errors during data collection, as each key press should ideally have a matching key release. This step resulted in the removal of 691 data entries, leaving us with 1,214 data points.

Subsequently, we proceed with a process known as 'melting' the data, in which each row contains a single key event, either a key press or a key release. This transformation led to an expanded dataset of 36,708 data points, indicating that 18,354 characters were typed into this dataset. Next, we merge the datasets so that each row contains the key code information, key press time, and key release time. Again, we validate the data to ensure that each key code has a corresponding pair of key press and key release times. Any session with unpaired data is entirely removed. After this step, the dataset was reduced to 8,719 rows.

Finally, we transform the data into sequences of two-character keystrokes. This final transformation results in a dataset with 7,506 rows. Hence, from the initial breakdown of JSON to the final two-character keystroke sequences, the dataset has undergone a significant transformation.

4.3. Feature Extraction

The next phase of this research is feature extraction, where relevant features are derived from the transformed data. The detail of these featue are shown in

Table 1.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Feature Extraction

Feature	Description
ID1	Key code of first character
ID2	Key code of second character
H1	Hold time of key 1
H2	Hold time of key 2
DD	digraph latency of the key 1 and key 2
UD	flight time of key 1 and key 2
H2-1	The difference of hold time key 1 and key 2
B	Total duration of bigram key 1 and key 2

Following the feature extraction process, the data is divided into ten distinct datasets, each representing one user. Given that this research uses binary classification to classify genuine and impostor users, we must define negative data needed by binary classification. This negative data is essential for training the model to identify unauthorized or anomalous keystroke patterns, enhancing the model's predictive accuracy.

To obtain this negative data, we adopt a cross-user strategy, extracting it from the datasets of other users. This approach is based on the assumption that a user's keystroke dynamics are distinctive, and data from other users can thus serve as 'impostor' data. Furthermore, to maintain balance between the genuine and impostor data, we take a proportion of the negative data that is equivalent to the total genuine data of each user, divided by nine (the other nine user), distribution of data for each user can be seen in Figure 6. This process ensures that our dataset is balanced, promoting a more effective and unbiased classification model.

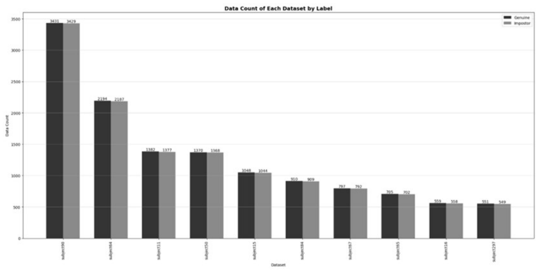


Figure 6: Distribution of data for each user

The dataset collected for this study consists of various keystroke events from different users. The statistic description of the dataset IS PRESENTED in Table 2. The data collected is very varied in terms of the number of characters typed and the number of sessions per user. The number of characters typed reflects the amount of data collected from each user, which ranges from 603 to 3817. The number of sessions per user reflects the frequency and duration of data collection, which ranges from 29 to 193. The standard deviation of hold time is another measure of variation in the dataset, which indicates how consistent the users are in their typing style and rhythm. The standard deviation of hold time ranges from 26.52 to 197.72. The hold time is the duration between a key press and a key release, which is one of the features used for keystroke analysis. The hold time has a wide range from 1 ms to 1267 ms in the dataset.

Table 2: Dataset Statistic Description

Subject	Number of Char	Number of session	mean	std	min	max	median
90	3817	193	147.51	80.34	32	768	128
64	2434	121	113.29	74.48	31	632	91
11	1528	73	114.74	90.46	1	775	93
50	1508	62	117.13	64.34	17	593	100
15	1172	69	122.68	39.34	58	286	112
84	1010	50	133.81	48.45	16	280	120
67	879	41	136.49	89.06	28	940	110
65	789	42	103.17	26.52	50	235	99
16	617	29	162.73	197.72	44	1504	117
297	603	30	112.21	116.53	16	1267	84

4.4. Classification

The classification model used in this research is a combination of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), where RNN uses long short-term memory (LSTM). This combination of CNN and LSTM represents a potential tool for sequential data, as it can capture both spatial (CNN) and temporal (RNN) dependencies in the data. The model consists of several layers including Conv1D, MaxPooling1D, Dropout, BatchNormalization, Bi-LSTM, and Dense layers. The first Conv1D layer has 64 filters, a kernel size of 2, an activation function of 'relu', and a kernel regularizer of l2 with a value of 0.5. The first MaxPooling1D layer has a pool size of 2 and uses 'same' padding. The second Conv1D layer has 128 filters, a kernel size of 2, an activation function of 'relu', and a kernel regularizer of l2 with a value of 0.5. The second MaxPooling1D layer has a pool size of 2 and uses 'same' padding. The Dropout layer has a rate of 0.02. The Bi-LSTM layers have 128 units each. The Dense layers have 8, 4, and 1 units respectively with the last one having an activation function of 'sigmoid'. The optimizer used is Adam with a learning rate of 0.0001 and the loss function is binary cross-entropy.

Robustness, or the ability to perform well on unknown data and resistance to both overfitting and underfitting, is a crucial need for this classification model. When a model performs very well on training data but poorly on test data, overfitting has occurred because the model has learnt the training data including its noise too well. Underfitting, on the other hand, occurs when the model is unable to capture the fundamental patterns in the data, leading to subpar performance on both the training and test datasets.

The performance of the model, particularly concerning training and validation loss, is illustrated in Figure 7. Training loss provides insights into how well the model is learning from the training data, whereas validation loss shows how well the model generalizes to unseen data. A good fit across all datasets, as indicated in the figure, suggests that the model is not overfitting. This means that the model has effectively learned the patterns in the training data and can generalize these patterns to new, unseen data, which is an indicator of a robust and well-performing model.

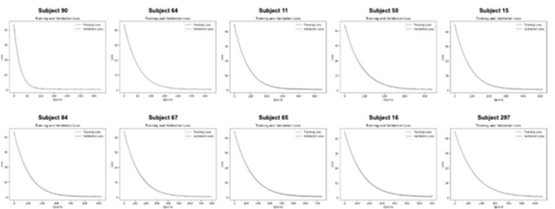


Figure 7: Traning and validation loss

4.5. Performance Evaluation

We evaluated the performance of then diiferents models in binary classification task. The models were trained and tested using 75:25 train-test split. The effectiveness of each model was assessed using the Equal Error Rate (EER), a measure where both the false acceptance rate and false rejection rate are balanced. a lower EER signifies best model performance.

Table 3: Performance of the model

Subject	FAR	FRR	EER
90	0.0349	0.0006	0.009
64	0.1639	0.0697	0.071
11	0.1242	0.1035	0.096
15	0.0733	0.0800	0.080
50	0.1839	0.0583	0.058
84	0.1418	0.0667	0.066
67	0.0648	0.0667	0.061
65	0.1333	0.1231	0.096
16	0.1205	0.0120	0.012
297	0.0725	0.1449	0.127

Table 3 shows that the model performed well on all variants of the number of sessions. The best performance was achieved by subject 90 with the lowest EER value of 0.009, and the worst performance was by subject 297 with the highest EER value of 0.127. The number of sessions did not affect the model performance significantly, as shown by user 16 who had the least number of sessions but still achieved a low EER value of 0.012.

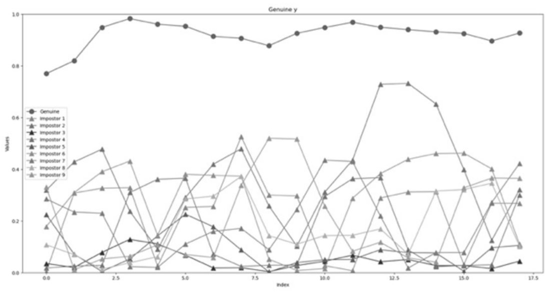


Figure 8: Probability of model recognition

Continuous authentication is carried out by simulation using sliding windows, as shown in Figure 8. The simulation results show that the authentication process is going well. The model can predict the 'real' user with a high probability of 0.98. Meanwhile, prediction results varied widely for 'impostor' users, but the probability of 'impostor' users being deemed 'genuine 'stayed below 0.5. The highest probability for an 'impostor' user is 0.32, and the lowest is 0.00. the results of the prediction details can be seen in

Table 4.

Table 4: Probability of Model Recognition

Subject	Genuine	Impostor								
		1	2	3	4	5	6	7	8	9
90	0.92	0.06	0.04	0.14	0.05	0.08	0.06	0.10	0.11	0.17
64	0.98	0.05	0.27	0.25	0.08	0.14	0.07	0.11	0.17	0.11
11	0.89	0.23	0.31	0.07	0.13	0.23	0.09	0.11	0.18	0.32
15	0.83	0.03	0.01	0.12	0.07	0.00	0.02	0.02	0.13	0.02
50	0.91	0.28	0.20	0.17	0.26	0.26	0.19	0.16	0.32	0.18
84	0.83	0.12	0.21	0.26	0.25	0.20	0.26	0.16	0.26	0.29
67	0.84	0.03	0.23	0.11	0.23	0.13	0.11	0.11	0.05	0.08
65	0.81	0.14	0.07	0.05	0.26	0.19	0.20	0.04	0.10	0.03
16	0.95	0.14	0.08	0.12	0.11	0.09	0.10	0.23	0.18	0.14
297	0.77	0.09	0.07	0.03	0.09	0.06	0.09	0.10	0.12	0.11

5. DISCUSSION

This research presents a hybrid CNN-Bi-LSTM model for keystroke dynamics authentication, achieving Equal Error Rates (EERs) ranging from 0.009 to 0.127. These results suggest the model's potential for use in continuous authentication systems. The approach taken in this study offers several important distinctions from previous research, which could enhance its applicability in real-world scenarios.

A key difference in the proposed model is its ability to handle data from multiple sessions and uncontrolled environments over extended periods. Many earlier studies focus on controlled settings or smaller datasets, which may limit the applicability of these models to real-world systems. This research addresses these challenges by maintaining accuracy despite variations in user behavior over time, making it a potentially valuable tool for applications such as online banking, e-commerce, and secure access to sensitive information.

6. CONCLUSION

This study has successfully demonstrated the potential of a hybrid Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (Bi-LSTM) model for continuous authentication using keystroke dynamics data collected from multiple sessions and uncontrolled environments. The research followed a rigorous process of data collection, preprocessing, feature extraction, and classification, and evaluated the performance of the model. The data collection phase resulted in 4671 raw records of keystroke data over six months, which were then preprocessed to remove irrelevant data and transform the data into a more usable format. This preprocessing phase highlighted the importance of data cleaning in ensuring the quality and relevance of the data used for model training.

The performance evaluation of the model showed that it performed well across all session variants, with the lowest Equal Error Rate (EER) value being 0.009 and the highest 0.127. These results indicate that the hybrid CNN-Bi-LSTM model is robust and capable of handling variations in typing patterns over multiple sessions and long periods. The study's objectives of achieving high accuracy and robustness in user authentication were met, demonstrating the model's effectiveness in continuous authentication scenarios.

In conclusion, this research has shown that a hybrid CNN-Bi-LSTM model can effectively handle keystroke dynamics authentication over several sessions in a period of six months. Future research could explore other factors that may influence the model's performance, such as the diversity of keystroke patterns, device types, user emotions, and the complexity of the tasks performed by the users. These additional factors could further enhance the model's robustness and applicability in real-world scenarios.

REFERENCES

[1] J. J. Cebula, M. E. Popeck, and L. R. Young, "A Taxonomy of Operational Cyber Security Risks Version 2:," Defense Technical Information Center, Fort Belvoir, VA, May 2014. doi: 10.21236/ADA609863.

[2] R. Manning, "The 2019 State of Password and Authentication Security Behaviors Report," *Yubico Releases the 2019 State of Password and Authentication Security Behaviors Report*, Jan. 28, 2019.

https://www.yubico.com/blog/yubico-releases-the-2019-state-of-password-and-authentication-security-behaviors-report/ (accessed Dec. 10, 2022).

[3] P. Baynath, K. M. S. Soyjaudah, and M. H.-M. Khan, "Machine Learning Algorithm on Keystroke dynamics Fused pattern in biometrics," in *2019 Conference on Next Generation Computing Applications (NextComp)*, Mauritius: IEEE, Sep. 2019, pp. 1–6. doi: 10.1109/NEXTCOMP.2019.8883621.

[4] N. Altwaijry, "Keystroke Dynamics Analysis for User Authentication Using a Deep Learning Approach," *International Journal of Computer Science and Network Security*, vol. 20, no. 12, pp. 209–216, Dec. 2020, doi: 10.22937/IJCSNS.2020.20.12.23.

[5] M. Handosa, A. Dasgupta, M. Manuel, and D. Gračanin, "Rethinking user interaction with smart environments—a comparative study of four interaction modalities," in *Distributed, Ambient and Pervasive Interactions: 8th International Conference, DAPI 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings 22*, Springer, 2020, pp. 39–57.

[6] P. Kasprowski, Z. Borowska, and K. Harezlak, "Biometric Identification Based on Keystroke Dynamics," *Sensors*, 2022, doi: 10.3390/S22093158.

[7] S. J. Quraishi and S. S. Bedi, "Keystroke Dynamics Biometrics, A tool for User Authentication–Review," in *2018 International Conference on System Modeling & Advancement in Research Trends (SMART)*, Moradabad, India: IEEE, Nov. 2018, pp. 248–254. doi: 10.1109/SYSMART.2018.8746932.

[8] P. S. Teh, A. B. J. Teoh, and S. Yue, "A Survey of Keystroke Dynamics Biometrics," *The Scientific World Journal*, vol. 2013, pp. 1–24, 2013, doi: 10.1155/2013/408280.

[9] P. H. Pisani and A. C. Lorena, "A systematic review on keystroke dynamics," *J Braz Comput Soc*, vol. 19, no. 4, pp. 573–587, Nov. 2013, doi: 10.1007/s13173-013-0117-7.

[10] X. Lu, S. Zhang, P. Hui, and P. Liò, "Continuous authentication by free-text keystroke based on CNN and RNN," *Computers & Security*, 2020, doi: 10.1016/J.COSE.2020.101861.

[11] M. Karnan, M. Akila, and N. Krishnaraj, "Biometric personal authentication using keystroke dynamics: A review," *Applied Soft Computing*, vol. 11, no. 2, pp. 1565–1573, Mar. 2011, doi: 10.1016/j.asoc.2010.08.003.

[12] M. H. Barkadehi, M. Nilashi, O. Ibrahim, A. Zakeri Fardi, and S. Samad, "Authentication systems: A literature review and classification," *Telematics and Informatics*, vol. 35, no. 5, pp. 1491–1511, Aug. 2018, doi: 10.1016/j.tele.2018.03.018.

[13] I. Velásquez, A. Caro, and A. Rodríguez, "Authentication schemes and methods: A systematic literature review," *Information and Software Technology*, vol. 94, pp. 30–37, Feb. 2018, doi: 10.1016/j.infsof.2017.09.012.

[14] S.-N. Cheong, H.-C. Ling, and P.-L. Teh, "Secure Encrypted Steganography Graphical Password scheme for Near Field Communication smartphone access control system," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3561–3568, Jun. 2014, doi: 10.1016/j.eswa.2013.10.060.

[15] R. Giot, B. Dorizzi, and C. Rosenberger, "A review on the public benchmark databases for static keystroke dynamics," *Computers & Security*, vol. 55, pp. 46–61, Nov. 2015, doi: 10.1016/j.cose.2015.06.008.

[16] P. Campisi and A. Neri, "Keystroke Dynamics," in *Encyclopedia of Cryptography and Security*, H. C. A. Van Tilborg and S. Jajodia, Eds., Boston, MA: Springer US, 2011, pp. 692–695. doi: 10.1007/978-1-4419-5906-5_877.

[17] L. Yang and S.-F. Qin, "A Review of Emotion Recognition Methods from Keystroke, Mouse, and Touchscreen Dynamics," *IEEE Access*, vol. 9, pp. 162197–162213, 2021, doi: 10.1109/ACCESS.2021.3132233.

[18] A. F. M. N. H. Nahin, J. M. Alam, H. Mahmud, and K. Hasan, "Identifying emotion by keystroke dynamics and text pattern analysis," *Behaviour & Information Technology*, vol. 33, no. 9, pp. 987–996, Sep. 2014, doi: 10.1080/0144929X.2014.907343.

[19] A. Kolakowska, "A review of emotion recognition methods based on keystroke dynamics and mouse movements," in *2013 6th International Conference on Human System Interactions, HSI 2013*, Gdansk, Sopot, 2013, pp. 548–555. doi: 10.1109/HSI.2013.6577879.

[20] R. Toosi and M. A. Akhaee, "Time–frequency analysis of keystroke dynamics for user authentication," *Future Generation Computer Systems*, vol. 115, pp. 438–447, Feb. 2021, doi: 10.1016/j.future.2020.09.027.

[21] A. Acien, A. Morales, R. Vera-Rodriguez, J. Fierrez, and J. V. Monaco, "TypeNet: Scaling up Keystroke Biometrics," *International Journal of Central Banking*, 2020, doi: 10.1109/IJCB48548.2020.9304908.

[22] L. Yang, C. Li, R. You, B. Tu, and L. Li, "TKCA: a timely keystroke-based continuous user authentication with short keystroke sequence in uncontrolled settings," *Cybersecurity*, vol. 4, no. 1, 2021, doi: 10.1186/s42400-021-00075-9.

[23] P. Kang, S. Hwang, and S. Cho, "Continual Retraining of Keystroke Dynamics Based Authenticator," in *Advances in Biometrics*, S.-W. Lee and S. Z. Li, Eds., in Lecture Notes in Computer Science, vol. 4642. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1203–1211. doi: 10.1007/978-3-540-74549-5_125.

[24] R. Giot, B. Dorizzi, and C. Rosenberger, "Analysis of template update strategies for keystroke dynamics," in *2011 IEEE Workshop on Computational Intelligence in Biometrics and Identity Management (CIBIM)*, Paris, France: IEEE, Apr. 2011, pp. 21–28. doi: 10.1109/CIBIM.2011.5949216.

[25] E.-S. M. El-Kenawy, S. Mirjalili, A. A. Abdelhamid, A. Ibrahim, N. Khodadadi, and M. M. Eid, "Meta-Heuristic Optimization and Keystroke Dynamics for Authentication of Smartphone Users," *Mathematics*, vol. 10, no. 16, p. 2912, Aug. 2022, doi: 10.3390/math10162912.

[26] M. Pi, N. Jin, D. Chen, and B. Lou, "Short-Term Solar Irradiance Prediction Based on Multichannel LSTM Neural Networks Using Edge-Based IoT System," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–11, Jan. 2022, doi: 10.1155/2022/2372748.

[27] S. Roy *et al.*, "A Systematic Literature Review on Latest Keystroke Dynamics Based Models," *IEEE Access*, vol. 10, pp. 92192–92236, 2022, doi: 10.1109/ACCESS.2022.3197756.

[28] A. Sulavko, A. Samotuga, and I. Kuprik, "Personal identification based on acoustic characteristics of outer ear using cepstral analysis, Bayesian classifier, and artificial neural networks," *IET Biometrics*, vol. 10, no. 6, pp. 692–705, Nov. 2021, doi: 10.1049/bme2.12037.