Behavioral Biometrics: User Authentication using Keystroke Dynamics

Berrospi Rodriguez, Luis Alfonso
Computer Science
Universidad de Ingeniería y Tecnología
Lima, Perú
luis.berrospi@utec.edu.pe

***Keywords*—Behavioral biometrics, keystroke dynamics, machine learning, user authentication, cybersecurity, deep learning, CNN, RNN, GRU.**

***Abstract*—Keystroke dynamics, a biometric authentication technique, has gained significant attention due to its potential for secure user identification and verification. This paper presents a comprehensive review of the type of text used in keystroke dynamics, the features extracted from the data, and the classification algorithms used. Then two algorithms are proposed, one based on metric distance to compare data with a nearest neighbor algorithm and another based on a hybrid neural network architecture that use a CNN and a RNN. The algorithms where tested with public datasets from Universities. The results, according to the literature, are an Equal Error Rate (EER) of 8.4% for the nearest neighbor algorithm and 5.97% for the hybrid neural network.**

I. Introduction

Since the beginning of the digital age, cybersecurity has been developing and adapting to the new threats that arise in the digital world. One of the topics of great importance in this field is user authentication. Different methods have been used over the years, such as passwords, iris recognition, fingerprint matching, etc. However, most of these methods seek to guarantee the authentication only at the beginning of the session of interaction between a human and a device, but not through said session. This creates a problem: if a user leaves their already authenticated session open, anyone can access their information. For this reason, the use of behavioral biometrics has been proposed. These authentication methods allows the user to be authenticated not only at the begining of the session but also throughout it [2].

Keystroke Dynamics are behavioral biometrics that provides the authentication a user at the begining and throughout a session by analyzing the patterns and rhythm that they present when writing using a keyboard [3]. This techniques were suggested during World War II [26] when the idea of using the rhythm of the Morse code to identify the operator of a telegraph was proposed. But it was not until the develop of computers and the Internet studies started to encourage the use of this technique [27].

The main problem with the Keystroke Dynamics techniques as a method for user authentication is the different algorithms, data extraction methods and features used to authenticate a user. Different comparison techniques have been used through the years. Probabilistic methods [9], multiple machine learning techniques [4] and Neural Networks are some examples of these techniques [5], [12]. All these techniques have very different results, some of the best achieve an average FAR of 2% while others are between 8 and 27% [6]. In addition, it has been shown that precision can vary depending on what is being written. It has been possible to observe how the precision goes from 79% with text that is transcribed to 21% with text that is freely written [7] and how if a person is writing in a particular context, like programmers writing code, the algorithms can reach a precision of 98.6% [8].

The goal of this document is to implement two algorithms for user authentication through Keystroke Dynamics that have already been proposed in the literature. It is seeked to compare the results obtained by each of the algorithms, and determine which of them is the most appropriate for this type of authentication. In addition, this project also pursue to determine the behavior of each algorithm with respect to the type of data that is used, this is, if the algorithm is more accurate with free text or with fixed text.

II. Theoretical Foundations

In the following section, the theoretical foundations of keystroke dynamics are going to be explained. First, the type of texts that can be used for the feature extraction phase is going to be explained. Then, the features that are going to be extracted from the text are going to introduced. Finally, the different algorithms that are going to be used are going to be explained along with some characteristics of biometric systems that can be used to evaluate the performance of each algorithm.

A. Data extraction

It is possible to define two types of text that can be used for feature extraction:

- Fixed text: This type of text is the one that is written by the user using a fixed source. It is generally a short phrase or a word that is used to identify the user. A good

example of this type of text is the password and username that are employed to log into a computer or a website.
- Free text: This type of text is the one that is written by the user without any restriction. This kind of text is the most common in the real world but it is generally the most difficult to analyze because of the lack of restrictions and the introduction of important factors like, the user's level of concentration, thinking time, etc.

B. Features

In general, the features extracted from the keystroke dynamics are based on the study and analyzes data involving digraphs and trigraphs. In the keystroke dynamics literature a digraph is a set of two consecutive characters that represents two keystrokes, that can be letters, numbers or symbols and other keys that do not necessarily represent a language character like the space bar, shift key or enter key. Logically, a trigraph is a set of three consecutive characters that represents three keystrokes [24]. Some of the most common features that can be extracted from a single keystroke, a digraphs or a trigraphs are the following [27], [4]:

- Tap timestamp: Defined as the timestamp of the key press.
- Release timestamp: Defined as the timestamp of the key release.
- Time between keystrokes: This feature is commonly defined as the time that elapses between the key press of the first keystroke and the key release of the second keystroke of a digraph or trigraph.
- Hold time: This is defined as the time that elapses between the key press and the key release of a keystroke.
- Words per minute (WPM): This is the avarage number of words that a user can type in a minute.
- Percentage of usage of special keys: This feature is defined as the percentage of times that a user uses keys like the shift key or CapsLock key.

C. Classification techniques

Once the features are extracted, the next step is to use algorithms to authenticate the user. Most of these techniques can be classified into two groups.

The first group is based on the usage of distance metrics. These set of techniques are take use of the calculation of the distance between the features of the user that is trying to be identified and the features of the users that are already registered in the system. Features are displayed as a vector of multiple dimensions and the distance between vectors is calculated using a distance metric. Some metrics that can be used are the Manhattan distance and the Mahalanobis distance.

On one hand, Manhattan distance for n dimensions can be defined as the sum of the absolute differences between the coordinates of the vectors, As represented in the following formula:

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i| \tag{1}$$

Where x and y are vectors of n dimensions and x_i and y_i are the ith coordinates of the vectors x and y respectively.

On the other hand Mahalanobis distance can be defined as the distance between two points that depends on the probabiliy distribution of all the points [14]. The squared version of this distance can be described by the following formula:

$$d(x, y) = (x - y)^T S^{-1} (x - y) \tag{2}$$

Where x and y are vectors of n dimensions and S is the covariance matrix of the set of points.

These distances are commonly used in clustering algorithms like K-nearest neighbors. This algorithm stores the multidimensional feature vectos such that when a vector is received it can be classified by the majority of the classes of the nearest neighbors using a distance metric. In the user authentication problem, the user can be identified by finding the single nearest neighbor and using a threshold to determine if the user is the same as the one that is trying to be identified. A simple knn exaplme for 2 dimensional features can be shown in figure 1.

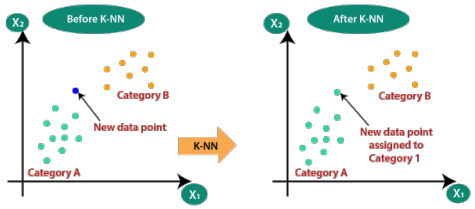

Fig. 1. K-nearest neighbors algorithm [15]

The second group of techniques is based on the use of deep learning techniques.

One of the most important parts of this approach is the use of algorithms to vectorize the text, such that said text can be used as input for the neural network. Some of the most common vectorization techniques try to represent text as numbers, assigning numeric values to words, characters or more complex structures like tokens. Other techniques try to transform images into matrices that contain RGB or grayscale values of the pixels. Finally there are techinques that have numbers as input and try to normalize and standardize the data to perform better in the neural network.

This techniques are characterized by the use of neural networks. These networks try to simulate the human brain by being composed of multiple layers of neurons that are

connected to each other. Each neuron is a mathematical function that receives multiple inputs and produces one or more outputs.

The neural networks that are going to be used in this project are the following:

- Convolutional Neural Networks: CNNs are a type of neural network architecture that have been successful in tasks such as image classification, object detection, image segmentation, by feature extraction. [20]. They are designed to automatically learn and extract hierarchical patterns and features from input data. CNNs use convolutional layers, pooling layers, and fully connected layers [21]. Convolutional layers apply filters to the input data, capturing spatial dependencies and detecting local patterns. Pooling layers reduce the spatial dimensions of the data. Fully connected layers are usaly placed at the end of the network and are used to get the final output.
- Recurrent Neural Network: RNNs are neural network architectures suitable for sequential data such as time series, text, and speech[19]. Unlike traditional networks, RNNs contain loops that allow information to persist and be processed across different time steps [18]. The presence of a hidden state, the internal memory of the network, grants RNNs the ability to capture temporal dependencies and contextual information within the data.
- Gated Recurrent Unit: GRU is a variation of RNN designed to address the vanishing gradient problem and improve the modeling of long-term dependencies. GRU incorporates gatelike mechanisms that regulate the flow of information within the network. GRU units consist of an update gate and a reset gate, which control the operations of information update and reset in the hidden state [16]. The update gate determines how much of the previous hidden state to retain, while the reset gate decides how much of the previous state to forget. GRUs have demonstrated superior performance over traditional RNNs in various tasks such as language modeling, machine translation, and speech recognition [17].

D. benchmark

The follwing factors are used to evaluate the performance of biometric systems [22]:

- False Acceptance Rate (FAR): FAR is a metric used in biometric system benchmarking to measure the likelihood of the system incorrectly granting access to unauthorized individuals. It represents the rate at which the system mistakenly identifies impostors as genuine users. A lower FAR indicates a higher level of security, as it means the system is less likely to mistakenly accept unauthorized individuals. This characteriztic can be represented by the following formula:

$$FAR = \frac{\text{False Acceptances}}{\text{Total Impostor Attempts}} \tag{3}$$

- False Rejection Rate (FRR): FRR is a metric used to assess the probability of the biometric system incorrectly rejecting legitimate users. It represents the rate at which the system fails to recognize or authenticate genuine users. A lower FRR indicates better user convenience, as it means the system is less likely to deny access to legitimate users. This metric can be represented by the following formula:

$$FRR = \frac{\text{False Rejections}}{\text{Total Genuine Attempts}} \tag{4}$$

- Equal Error Rate (EER): EER is a benchmark point where the FAR and FRR are equal as seen in Figure 2. It signifies the threshold or decision boundary at which the system achieves a balance between false acceptances and false rejections. The EER provides a summary measure of the system's performance and is commonly used for comparing different biometric systems. A lower EER indicates better overall performance, as it reflects a balanced trade-off between security and user convenience.

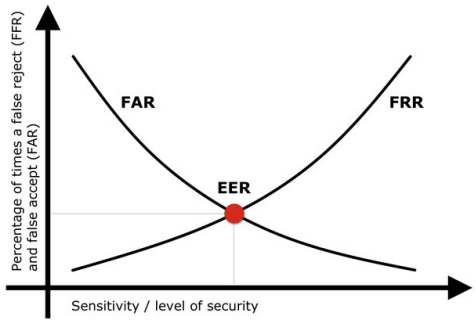

Fig. 2. Equal Error Rate [23]

III. Related Work

Multiple apporaches have been proposed to solve the problem of user authentication through keystroke dynamics. It is possible to classify said approaches into two main categories: those that focus on what data is obtained and those that focus on the classification algorithm. In the first category, work have been done comparing and analyzing results with free text and transcribed text [7], [8] and with memorized text like usernames and passwords [9], [5]. In the second category, work have been done using different classification algorithms, such as distance metrics [3], multiple machine learning techniques [4] and Neural Networks [12], [13].

Previos work has shown that in some cases simple distance based clustering algorithms can acomplish good performance [10], [11], [14], [8] this have been explained by the fact that the there are some distances that handle effectively characteristics of the features present in keystroke dynamics, like

the correlations and good interaction between the features. Some examples of these distances are Mahalanobis which takes into account the covariance matrix of the features for better differentiation and Manhattan distance which is more robust, it is not affected by the scale of the features [3].

It also have been shown that the selection of the features is a key factor for the performance of the models and that the algoritms used for the classification are less important [5]. This can be seen in the continious integration of new features thoughout the development of the techniques. At the beginning, the data was obtained from the time between 2 keystrokes [9] and then the pressing time was added [10]. Later more sofisticated features were added like the edit time [4], normalized data so that the emotional state of the user does not affect the results and the introduction of thrigraphs [11].

In recent years the use of deep learning techniques has been studied and proposed to solve the problem of user biometric recognition through keystroke dynamics. A hybrid model involving a CNN and a RNN was proposed in [12] using feature transformation to adapt the input data to a image-like format for better results. More simple models have also been proposed [13], using CNN an fixing the input size with a 6 digit pin on mobile devices.

IV. Techniques

In this section, we describe the 2 techniques that are going to be implemented in this project. The first one proposed by Yu Zhong et al [3], which use a common technique with distance metrics for user authentication. The second one is the technique proposed by Lu, Xiaofeng, et al [12], which uses a hybrid model of a CNN and a RNN.

A. Distance Metrics

The first technique proposed by Yu Zhong et al [3] is based on the use of distance metrics to compare the data of a user with the data of the rest of the users. The data used in this technique comes from the CMU Keystorke Dynamics Benchmark and it is made up by the time between keystrokes and the hold time of each keystroke. This dataset contains 51 users, each one with 400 samples (feature vectos) of a fixed password written in English.

After a state of the art analysis, the authors realized that the 2 more successful distance metrics used in the literature were the Manhattan distance and the Mahalanobis distance. So they decided to combine them in a new distance metric so that the new metric could take advantage of the benefits of both. The new metric is defined as follows:

Firstly, the authors apply the following linear transform to the data like the Mahalanobis distance, so that the features become uncorrelated with equal variations.

$$x' = \Phi x \tag{5}$$

Where Φ is the principle of the square root of the covariance matrix S of the data such that $S^{-1} = \Phi \cdot \Phi^T$.

Then the Manhattan distance is measured between the transformed data.

$$d(x', y') = \sum_{i=1}^{n} |x'_i - y'_i| \tag{6}$$

Then, the authors apply the Manhattan distance to the transformed data. Finally, this new distance is used to compare the data of a user with the other data with a standard Nearest Neighbor classifier.

The authors also used an outlayer detection technique in the training of the model to improve the results. This technique is based on the use of the median μ and the standard deviation σ of the features so that only the feature vectors with its ith feature in the range $[\mu_i - k\sigma_i, \mu_i + k\sigma_i]$ are used in the training of the model.

This technique achieved an EER of 8.4% This result improved the results preiously obtaned [25] of the Manhattan distance (EER of 9.6%) and the Mahalanobis distance (EER of 10%).

B. Deep Learning

The authors of [12] propose a hybrid model of a CNN and a RNN to identify the user. The model is based on the use of a CNN to extract the features from the keystroke dynamics and a RNN to identify the user. The model used the Clarkson II and Buffalo datasets for keystroke dynamics. These datasets contained the timestamp of each keystroke, recolected from 103 user in the Clarkson II dataset and 157 in the Buffalo dataset. Booth of this datasets contained free text written by the users, but the Clarkson II dataset also contained data from the users writing a fixed text.

First, the data had to be vectorized. The vector data format was defined by the authors as can be seing in the following table I.

ID[1]	ID[2]	H[1]	H[2]	D[1]H[2]	D[1]D[2]

TABLE I
VECTOR DATA FORMAT PROPOSED BY THE AUTHORS OF [12]

Where D[n] is the timestamp of the nth keystroke tap, H[n] is the hold time of the nth keystroke and ID[n] is the key identifier of the nth keystroke keystroke, U[n] refeers to the timestamp of the nth key release. In addition, D[n]U[m] is the UD characteristic of the nth and the m keys and D[n]D[m] is the DD characteristic of the nth and the m keys.

Following this format, a vector of 6 features can be created from text. As can be shown in the image 3.

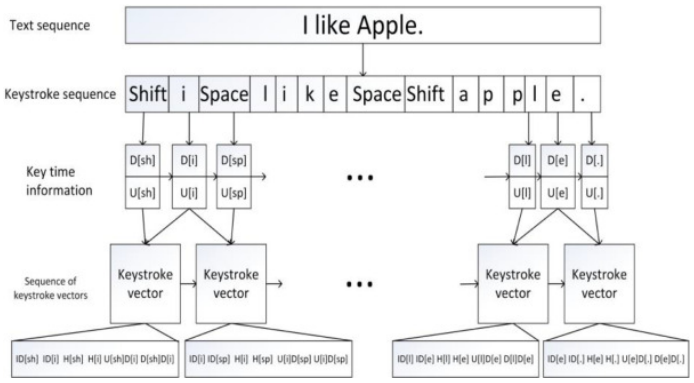

Fig. 3. Vector data format transform by the authors of [12]

Then, the authors proposed multiple configurations of the neural network architecture. However, the more effecive can be see in the image 4.

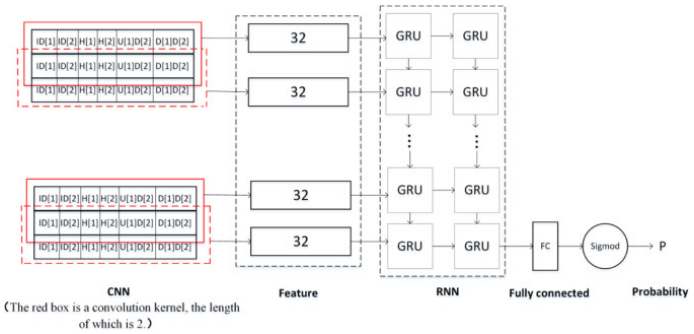

Fig. 4. Neural network architecture proposed by the authors of [12]

As it can be seen, all the vectors extracted from a given text with specifict lenght in terms of keystrokes are arrange as a matrix where each row is a vector. Then, a one dimensional CNN with a convolutional kernel with lenght 2 (feature vectors) is applied to the matrix to extract 32 the features that are going to be used by the RNN, a double layer GRU. Finally, the output of the RNN connected to the fully-connected layer and then to the output layer, which is a sigmoid function that outputs the probability of the user being the user that the model is trying to identify. The authors also added droutout layers to each layer of the model to avoid overfitting.

This approach obtained a FRR of 2.07%, a FAR of 3.26% and an EER of 2.67% using the Clarkson II dataset and a FRR of 6.61%, a FAR of 5.31% and an EER of 5.97% using the Buffalo dataset.

References

[1] A. V. García. (2021) A brief history of user verification & online identity. [Online]. Available: https://www.arengu.com/blog/a-brief-history-of-user-verification-online-identity

[2] M. Sultana, P. P. Paul, and M. Gavrilova, "Social behavioral biometrics: An emerging trend," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 29, no. 08, p. 1556013, 2015.

[3] Y. Zhong, Y. Deng, and A. K. Jain, "Keystroke dynamics for user authentication," in *2012 IEEE computer society conference on computer vision and pattern recognition workshops*. IEEE, 2012, pp. 117–123.

[4] A. Alsultan, K. Warwick, and H. Wei, "Non-conventional keystroke dynamics for user authentication," *Pattern Recognition Letters*, vol. 89, pp. 53–59, 2017.

[5] K. Revett, F. Gorunescu, M. Gorunescu, M. Ene, S. Magalhaes, and H. Santos, "A machine learning approach to keystroke dynamics based user authentication," *International Journal of Electronic Security and Digital Forensics*, vol. 1, no. 1, pp. 55–70, 2007.

[6] A. Peacock, X. Ke, and M. Wilkerson, "Typing patterns: A key to user identification," *IEEE Security & Privacy*, vol. 2, no. 5, pp. 40–47, 2004.

[7] F. Monrose and A. Rubin, "Authentication via keystroke dynamics," in *Proceedings of the 4th ACM Conference on Computer and Communications Security*, 1997, pp. 48–56.

[8] K. Longi, J. Leinonen, H. Nygren, J. Salmi, A. Klami, and A. Vihavainen, "Identification of programmers from typing patterns," in *Proceedings of the 15th Koli Calling conference on computing education research*, 2015, pp. 60–67.

[9] S. Bleha, C. Slivinsky, and B. Hussien, "Computer-access security systems using keystroke dynamics," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 12, pp. 1217–1222, 1990.

[10] J. A. Robinson, V. Liang, J. M. Chambers, and C. L. MacKenzie, "Computer user verification using login string keystroke dynamics," *IEEE transactions on systems, man, and cybernetics-part a: systems and humans*, vol. 28, no. 2, pp. 236–241, 1998.

[11] F. Bergadano, D. Gunetti, and C. Picardi, "User authentication through keystroke dynamics," *ACM Transactions on Information and System Security (TISSEC)*, vol. 5, no. 4, pp. 367–397, 2002.

[12] L. Xiaofeng, Z. Shengfei, and Y. Shengwei, "Continuous authentication by free-text keystroke based on cnn plus rnn," *Procedia computer science*, vol. 147, pp. 314–318, 2019.

[13] E. Maiorana, H. Kalita, and P. Campisi, "Deepkey: Keystroke dynamics and cnn for biometric recognition on mobile devices," in *2019 8th European Workshop on Visual Information Processing (EUVIP)*. IEEE, 2019, pp. 181–186.

[14] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1–18, 2000.

[15] "K-nearest neighbor(knn) algorithm for machine learning," https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning, accessed: 2023-05-16.

[16] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[17] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Light gated recurrent units for speech recognition," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 92–102, 2018.

[18] S. Dupond, "A thorough review on the current advance of neural network

structures," *Annual Reviews in Control*, vol. 14, no. 14, pp. 200–230, 2019.

[19] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in *2015 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2015, pp. 4520–4524.

[20] M. V. Valueva, N. Nagornov, P. A. Lyakhov, G. V. Valuev, and N. I. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," *Mathematics and computers in simulation*, vol. 177, pp. 232–243, 2020.

[21] M. M. Taye, "Theoretical understanding of convolutional neural network: concepts, architectures, applications, future directions," *Computation*, vol. 11, no. 3, p. 52, 2023.

[22] "Characteristics of biometric systems," https://web.archive.org/web/20081017165633/http://www.ccert.edu.cn/education/cissp/hism/039-041.html, accessed: 2023-05-17.

[23] "Far and frr: security level versus user convenience," https://www.recogtech.com/en/knowledge-base/security-level-versus-user-convenience, accessed: 2023-05-17.

[24] A. K. Jain, S. C. Dass, K. Nandakumar *et al.*, "Soft biometric traits for personal recognition systems," in *ICBA*, vol. 3072. Springer, 2004, pp. 731–738.

[25] K. S. Killourhy and R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," in *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*. IEEE, 2009, pp. 125–134.

[26] E. D. Measure and W. P. Measure, "Keystroke dynamics."

[27] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation computer systems*, vol. 16, no. 4, pp. 351–359, 2000.