Behavioral Bio-metrics: 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 bio-metrics, keystroke dynamics, machine learning, user authentication, cybersecurity.**

***Abstract*—**

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 key topics of great importance in this field is user authentication. Different methods have been used over the years: passwords, iris recognition, fingerprint matching, etc. However, most of these methods seek to guarantee user authentication only at the beginning of the session of interaction between a human and a device, but not through it. This is a problem, since if a user leaves their session open, anyone can access their information. For this reason, the use of behavioral bio-metrics has been proposed, which allows the user to be authenticated not only at the begining of the session but also throughout it [2]. One of the behavioral bio-metrics applications is called Keystroke Dynamics, which consists of the authentication a user at the begining abd throughout a session by analyzing the patterns and rhythm that they present when writing using a keyboard.

The main problem with the Keystroke Dynamics technique as a method for user authentication is the different classification algorithms used to cluster users. Different comparison techniques have been used though the years, distance metrics [3], multiple machine learning techniques [4] and Neural Networks [5], [12]. All these techniques have very different results, some of the best achieve an average false rate 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 that if a person is writing code the algorithms can reach a precision of 98.6% [8].

The main objective of this work is to implement two algorithms for user authentication through Keystroke Dynamics that have already been proposed. It seeks 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, it seeks to determine the behavior of each of the algorithms with respect to the type of data that is entered, that is, if the algorithm is more accurate with free text or with text that is transcribed.

II. Theoretical Foundations

A. Data extraction

There are two types of text that are going to be used for feature extraction are the following:

- 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 is used to log in to 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 keystroke dynamics a digraph is a set of two consecutive characters that represents two keystrokes, these can be letters, numbers or symbols and other keys that do not necessarily represent a character like the space bar, shift key or the enter key. Logically, a trigraph is a set of three consecutive characters that represents 3 keystrokes. Some of the most common features that can be extracted from a single keystroke, a digraphs or a trigraphs are the following:

- 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.

- Edit time: Time that a takes to correct a mistake.
- 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 from the keystroke dynamics, the next step is to use a technique to identifyi the user. Most of these techniques can be classified into two groups.

The first group is based on the use of distance metrics. These set of techniques are based on 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. The features are displayed as a vector of multiple dimensions and the distance between the vectors is calculated using a distance metric. Some metrics 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. And can be described by 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 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 data.

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.

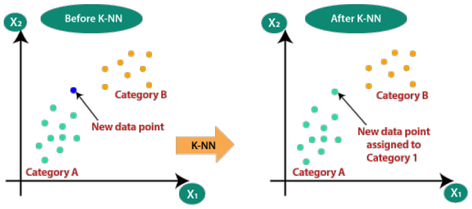

Fig. 1. K-nearest neighbors algorithm [15]

The second group is based on the use of deep learning techniques. VECTORIZATION CNN RNN LSTM GRU

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 Probabilistic Neural Networks [5].

Previos work has shown that in some cases simple distance based clustering algorithms can acomplish good performance [10], [11] this have been explained by the fact that the there are some distances that handle effectively the correlations and good interaction between the features like Mahalanobis which takes into account the covariance matrix of the features and others like Manhattan distance which 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 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 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.

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.

$$x' = \Phi x \tag{3}$$

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.

$$d(x', y') = \sum_{i=1}^{n} |x'_i - y'_i| \tag{4}$$

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 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.

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 2.

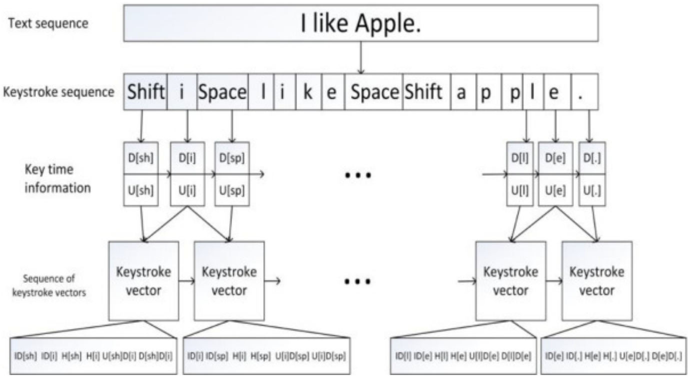

Fig. 2. Vector data format transform by the authors of [12]

Then, the authors proposed multiple configurations of the neural network architecture. However, the more effecive can be see in the image 3.

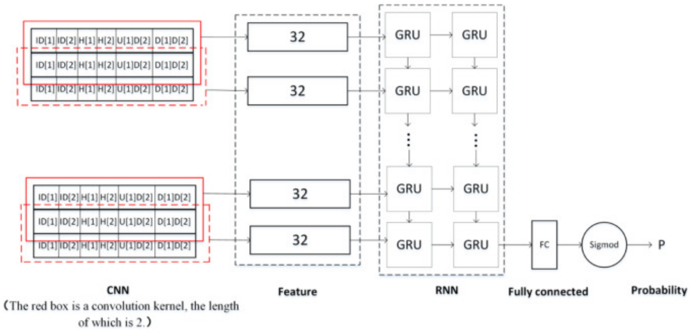
ID[1] ID[2] H[1] H[2] U[1]D[2] D[1]D[2]
ID[1] ID[2] H[1] H[2] U[1]D[2] D[1]D[2]
ID[1] ID[2] H[1] H[2] U[1]D[2] D[1]D[2]
ID[1] ID[2] H[1] H[2] U[1]D[2] D[1]D[2]
ID[1] ID[2] H[1] H[2] U[1]D[2] D[1]D[2]
ID[1] ID[2] H[1] H[2] U[1]D[2] D[1]D[2]
32
32
32
32
GRU
GRU
GRU
GRU
GRU
GRU
GRU
GRU
FC
Sigmod
P
CNN
(The red box is a convolution kernel, the length of which is 2.)
Feature
RNN
Fully connected
Probability

Fig. 3. Neural network architecture proposed by the authors of [12]

As it can be seen, all the vectors extracted from a given text with specific 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.

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.