Keystroke Dynamics Anomaly Detection

A. Khalifa
MEng. Student
School of Electrical Engineering and Computer Science
University of Ottawa, Ottawa, Canada
akhal126@uottawa.ca

S. Mohamed
MEng. Student
School of Electrical Engineering and Computer Science
University of Ottawa, Ottawa, Canada
smoha279@uottawa.ca

S. Ebrahim
MEng. Student
School of Electrical Engineering and Computer Science
University of Ottawa, Ottawa, Canada
sebra034@uottawa.ca

A. Ebraheem
MEng. Student
School of Electrical Engineering and Computer Science
University of Ottawa, Ottawa, Canada
aelba046@uottawa.ca

A. El Bassiouney
MEng. Student
School of Electrical Engineering and Computer Science
University of Ottawa, Ottawa, Canada
aelba046@uottawa.ca

***Abstract* — As cybersecurity attacks increase, the static methods for anomaly detection are no longer enough for protection. We proposed an algorithm for dynamic anomaly detection based on keystroke patterns that are unique for everyone, just like your fingerprint but digitally. Behavioral biometrics also overcome the most important limitation of physiological biometrics systems, as we can collect them without the knowledge of the user, allowing for continuous authentication. The user rhythm for 51 users with the fixed text password (.tie5Roanl) typed in 8 sessions for 50 repetitions per session. The proposed model Random Forrest achieve 94.09 % for F1 for this study. However, we tested the work by our own dataset which was collected and tested by typing the same password to check the user authentication.**

Keywords—Keystroke, anomaly detection, speed, genius, imposter.

I. INTRODUCTION

When you decide to connect to the internet, the virtual version of you is actually your username and password which is known as authentication, so if anyone succeed to know this critical information, he can easily pretend to be you and do electronic crimes or blackmail you. Identity monitoring is difficult to perform effectively in applications that rely on a collaborative architecture. Lately, information security required many powerful keys to be more effective. There are many authentication methods that provide the required security. However, it needs biometric techniques to improve security. One of the biometric techniques is the keystroke dynamics for users. Basically, it relies on the user's signature in typing and pressing. The keystrokes dynamic is an unexplored but promising form of behavioral features. Information extracted from typing, such as the time interval between key presses or releases, can distinguish different people.

II. STATE OF THE ART

1) Extensiveness of the related work

Previously, the methods used in keystrokes dynamics were distance-based methods like Euclidean and Manhattan distance. Now the state-of-the-art is going to the machine and deep learnings models, but before going to build the model, we should know more about the type of data set used to identify the user to different tasks like authentication. The dataset isn't the text itself. It's the actual key presses and the flight time among them. The row data represent a user behavior and it's captured multiple times. Therefore, the goal of classifying the user entering the text in such paper case is password [1] as a genuine or an imposter.

They divided keystrokes classification into:

• Classification-based fixed-text: In this type, the entered data to the model is a short text sequence, such as passwords, and sometimes, they include the username as another feature to strengthen the model.

• Classification-based free-text: The text used to mimic a user's typing activity and to authenticate the user isn't always the same.

verifying the user's identity based on their distinctive typing pattern is the proposed goal for their article [2]. by using the generalized Fuzzy Model (GFM), which is a blend of Mamdani-Larsen and Takagi-Sugano fuzzy models, to provide a unique method for keystroke dynamics-based authentication.

They employ the Gaussian Mixture Model (GMM) and GFM to model keystroke dynamics, using individual keystroke measurement types and combinations of data kinds. they tested the GFM on the CMU dataset to validate its performance on keystroke dynamics in a real-world context, and it outperforms GMM.

One of the most important objectives when dealing with keystroke dynamics is trying to reduce the detection time as much as possible, to let the allowed user fast access, and prevent otherwise. The two main types of keystroke dynamics are fixed and free text. The difference between them is the number of used characters. It's about ten characters in fixed-text while it's more than 676 in free-test, which makes it more difficult for free-text to show good results with test data. Free-test has two types: controlled (with some typing constraints) and uncontrolled (totally free typing).

In [3], They proposed an evaluation matrix called instance-based, tail area density (ITAD) which combines with scores after reducing the entire feature into five features with different weights according to the importance of each feature. they used two datasets:

I. Clarkson II: uncontrolled free-text.

II. Buffalo partially: controlled free-text.

Keystroke dynamics have a wide prominence as the second factor of authentication but it has a limitation that it is good at fitting the model to one user only which is not common in real-world applications that require two or more users to use the same credentials. So, to overcome this limitation in [4], they proposed a method that can leverage existing keystroke dynamics algorithms, to automatically determine the number of users sharing the same account using eight keystroke dynamics and three public datasets with up to five different users in one model.

2) Description of the solutions

In [1], they used The Carnegie-Mellon University (CMU) fixed-text dataset, which is commonly used in keystroke dynamics research. This dataset contains the keyboard dynamics of 51 individuals, each of which entered the password ". tie5Roanl" 400 times, with 50 repeats in each of eight sessions. Figure 1 shows the 31 time-based attributes that were gathered each time this password was input.

Notation	Number	Summary	Description
H	11	hold time	The length of time that a key is pressed
DD	10	down-down	The length of time from one key press to the next key press
UD	10	up-down	The length of time from one key being released until the next key is pressed
Total	31	—	—

Figure 1:table keystroke features in CMU

They applied a preprocessing step called "Data augmentation" and entered it into the fine-tuned XGBoost algorithm which achieved its performance in the data has outliers and misclassifications.
Data augmentation is creating new data from a dataset that already exists. This type of "fake" data can compensate for a problem's lack of data. Data enhancement is quite useful. Using a range of (-0.02, 0.02), they randomly disrupt each time feature.

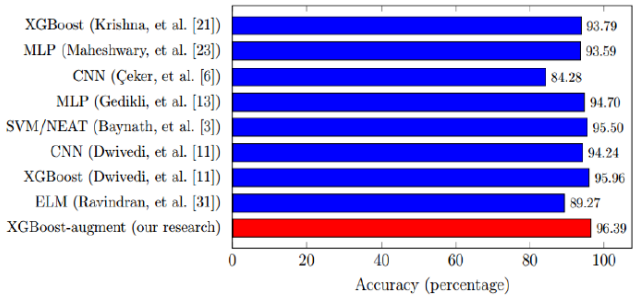

Figure 2: Accuracies of models trained in

In [2], they proposed learning GFM or GMM for each user. As a result, authentic & impostor samples were used while training these models. Genuine samples are the user's keystroke timing data that was used to train the model, whereas imposter samples are the remaining users' keystroke timing samples.

In [3], Instance-Based Tail Area Density Metric (ITAD): it basically doesn't follow any specific distribution. Instead, it uses only the area under the probability density function (PDF).

The matrix formula is:

$$Si = \begin{cases} CDFgi\ (xi) \text{ if } xi \le Mgi \\ 1 - CDFgi\ (xi) \text{ if } xi > Mgi\ , \end{cases}$$

Where: N in the of the shared graphs between the profile and test data (let's say N=5 which present the important five features), CDFgi is the cumulative distribution function, Mgi is the median, and xi is the single test duration for each ith graph. The range of the output of Si is from 0 to 0.5, going score close to 0.5 mains high similarity. Then calculate the summation of ith graph to get single score.

Results: ITAD shows better results with the Buffalo dataset because it's controlled free text. Moreover, these results are better than the previous work by 3%. In [4], Their proposed solution goes through the training phase and testing phase. In the training phase, they divide the training instance into subsets according to the user who produced this instance, but without knowing the instance-to-user association or the number of users. Interesting phase they associate the test instance with the relevant training subset, so the main steps are:
I. Training:
• Collecting data
They used 3 public datasets and get 3 different users from each dataset with 30 instances for each user.
• Feature extraction
The most important and commonly used feature in keystroke dynamics is a digraph, which has four types (press-to-release, press-to-press, release-to-press, and release-to-release). The output of the feature extraction step is a list of n feature vector instances.
• Quantile transformation
They perform quantile transformation on each feature separately, which is an efficient technique for dispersing close values, and for reducing the impact of extreme values and this facilitates the mission for clustering and getting higher scores.

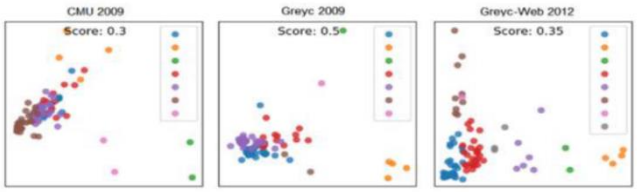

Figure 3: Projection of three users' instances for each dataset, with no preprocessing

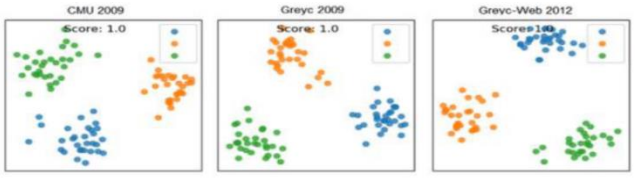

Figure 4: projection of three users' instances for each dataset, using quantile transformation; each color represents a cluster detected using X-means. The score above represents ARI. Instances for each dataset, with no preprocessing

• Data clustering

They preferred using X-means, which overcomes the limitation of k-means that require the number of clusters "users" in advance, which they do not know. X-means receives two parameters: τ, which is the maximal number of iterations; and γ, which is the maximal number of clusters. At the end of this step, they got several clusters hopefully equal to the number of users

• Training sub-models

And finally, they moved on to building the sub-models using the keystroke dynamics algorithm selected earlier to get a list of keystroke dynamics models KDM where each KDM corresponding to one cluster.

II. Testing:

The testing instance follows almost the same steps of training, and they want to associate it with one of the existing training clusters $\hat{C}$ i ∈ $\hat{C}$.

By applying their approach, they achieve an average improvement in the verification of 9.2% for the AUC and 8.6% for the EER in the multi-user cases, with just a negligible reduction of 0.2% for the AUC and 0.3% for the EER in the one-user cases.

c) Identified strengths/weaknesses of each solution

The strengths in this [1] research used the best-performed models and were fine-tuned and, in most cases; they achieved higher accuracy than the first publisher used the same model. They suggest in future work using a POPCORN technique which is promising with outside disturbances and makes the model more robust. However, their weaknesses they didn't explain the mathematics they used in the data augmentation and how it actually works. All they illustrate they create synthetic data by using a range they detect. Strengths In [2] include that they used the GFM, and, in most cases, it achieved higher performance than GMM for speaker identification using speech signals.

Because both speech signal and keyboard dynamics fall within the domain of behavioral biometrics, they studied the appropriateness of GFM and GMM on measurements of keystroke dynamics for user authentication in this study. Whereas one of their weaknesses is instead of using type-2 fuzzy sets, they used type-1 fuzzy sets, although the use of type-2 fuzzy sets gives higher accuracy. Strengths in [3], First, the ITAD matrix is better with non-Gaussian distrusted data because it doesn't follow any specific distribution. Second, gives good results with a wide range of keystrokes presses. However, their weakness is It doesn't work properly with Gaussian distribution data. Going through this [4], they illustrate very well the solution and methods. It is a very strong point to deal with multiple users in shared accounts, for example, to get better authentication, especially without knowing the number of users in advance. The results they achieved are excellent compared with the current methods. Whereas a weakness point, the ability to divide the data correctly into clusters reduces with the addition of more users. And the model's performance decrease with too few instances used.

III. DTATASET

The project use two dataset, first one is the benchmark dataset that was constructed from 51 different users, each one of them wrote the word (. tie5Roanl) as a password and saved to a CSV file, while each row represents one user, and the columns contain the keystroke-timing information which will be used to distinguish between the different users. This process was repeated 400 times for each user, to have enough data that can be used in the ML models.

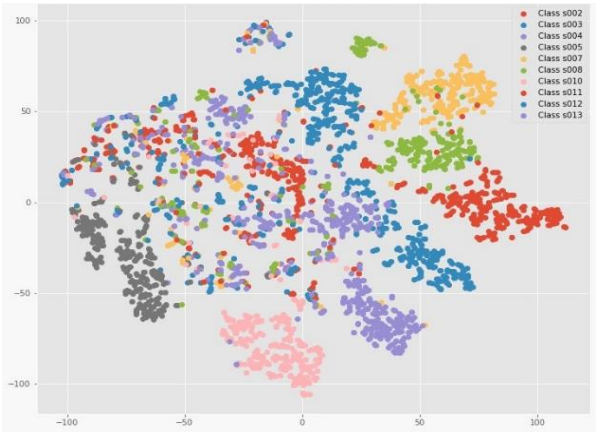

Figure 5:TSNE visualization for all features for the first 10 classes

The second one is collecting be running the train-collecting which allow user to enter his own data (name and password), then take this output and generate features from it which represent Hold, Up-Down and Down-Down timing information, then passed to producer that pass it to Kafka and finally the consumer consume it to be provided to the model

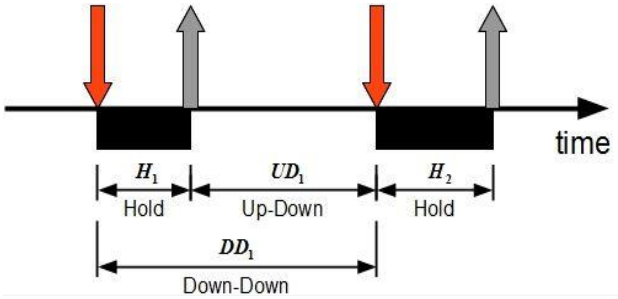

Figure 6: representation of timing information for pressing and releasing keys

Foe some sort of preprocessing figure 7 shows the correlation between features and figure 8 illustrate the behavior analysis for user pattern

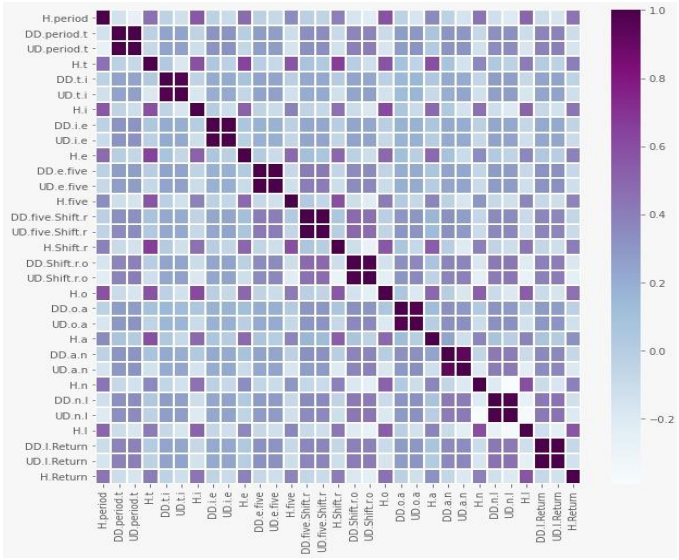

Figure 7: correlation between features

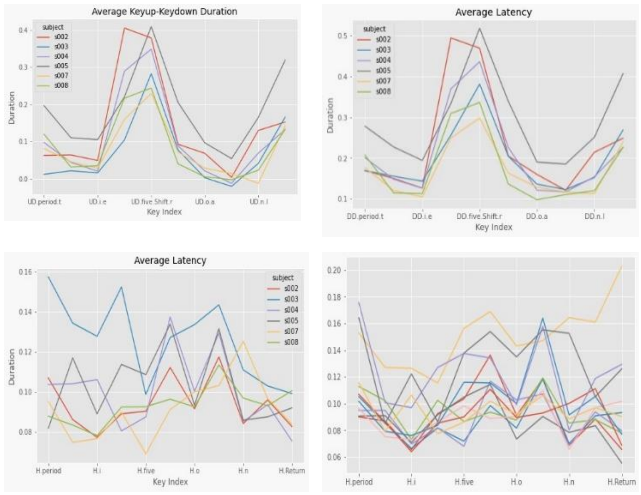

Figure 8:behavior user analysis

IV. METHODOLOGY

To analyze patterns of user, some approaches were applied. first apply clustering "figure 9" by computing similarity between user behaviors through distance metrics to group similar behaviors together this can be calculated through:

$$\sum_{i=1}^{p} \frac{|xi - yi|}{\alpha i}$$

Where xi and yi are the features of the testing and training data, where the training data represented by the mean vector for the 300 time by the user. The αi is the features of mean absolute deviation. We are calculating the city-block distance but with scaling each feature by the mean absolute deviation.

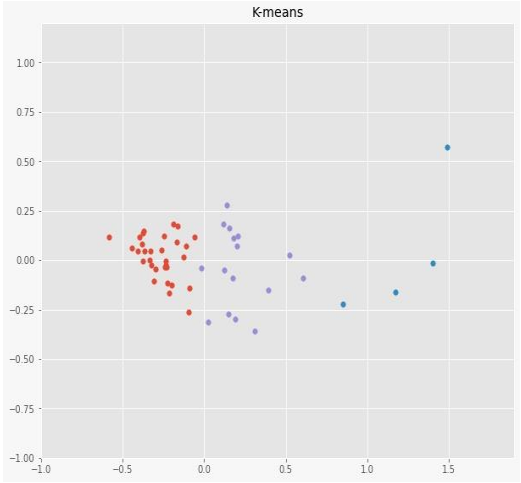

Figure 9: k-means clustering

Second approach is training classification models by taking subject "user" ID as a label and let model learn the features of each user to apply authentication criteria, to get the best performance different classification models to differentiate between genuine and impostor users based on the rhythm of the typing. SVM, those models include Logistic regression with OVR technique, k-nearest neighbors, Gaussian Naive Bayes classifier, Decision tree classifier, and Random Forrest are applied on the dataset and getting the best performance between them based on the accuracy and the F1 score. Choosing the F1 score because the data are imbalanced and we want to get the average between the recall and precision.

Models	Logistic Regression	RandomForest	KNeighborsClassifier	SVM	GaussianNB	Decision Tree Classifier
Accuracy	75%	94%	75%	76%	67%	88%
F1 Score	0.739	0.9409	0.747	0.758	0.664	0.875

Figure 10: conclusion of applied algorithms

As shown from the figure 10, Random Forrest achieves the highest accuracy and f1 score so it was the champion model.

The data was applied to the model and the predicted probability was obtained for each class, how much is the probability of the new test data close to one of the 51 users, Threshold of .04 was chosen for this probability by try and error which give better results, if it is larger than 0.4 it will get the argmax for the predicted class and predict it perfectly. If it below this threshold it will be considered as an Imposter.

We applied the RF_model to the dataset and get the probability of belonging to each class of the 51 users. We set a threshold equal to 0.4 by try and error, which gives us most appropriate performance. If the probability exceeds the threshold it, then we get the predicted user which is so close to the user. However, if the probability below the threshold it will be considered as an Imposter.

V. TEST AND EVALUATION

The models were evaluated using the existing keystroke dataset and the data that have collected. This study will use both datasets, testing was done with the collected dataset in real-time which illustrated in figure 11 below

```
Enter your text:
□□.tie5□roanl
ouput
[('Fares', 0, 'Down', 920635218), ('Fares', 0, 'Down', 920635343), ('Fares', 0, 'Up', 920635484), ('Fares',
 0, 'Up', 920635531), ('Fares', 46, 'Down', 920637875), ('Fares', 46, 'Up', 920638046), ('Fares', 116,
'Down', 920639250), ('Fares', 116, 'Up', 920639390), ('Fares', 105, 'Down', 920641406), ('Fares', 105,
'Up', 920641593), ('Fares', 101, 'Down', 920644390), ('Fares', 101, 'Up', 920644562), ('Fares', 53, 'Down',
 920649484), ('Fares', 53, 'Up', 920649640), ('Fares', 0, 'Down', 920651984), ('Fares', 114, 'Down',
920652609), ('Fares', 0, 'Up', 920652828), ('Fares', 114, 'Up', 920652828), ('Fares', 111, 'Down',
920655687), ('Fares', 111, 'Up', 920655828), ('Fares', 97, 'Down', 920659546), ('Fares', 97, 'Up',
```

Figure 11: testing online dataset

Figure 12 shows the confusion matrix for the prediction for each user after applying random forest, the false negative and false negative values are acceptable

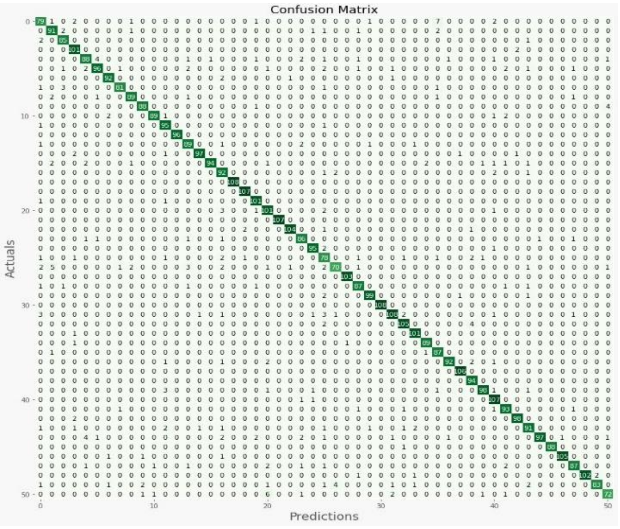

Figure 12: confusion matrix for random forest

VI. BASELINE TESTING METHODS

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

VII. RESULTS

For the online prediction, we injected the both datasets to Kafka topic, the received the data from the consumer, converting these data into logs or Jeson in particular, then sent to elastic search to be visualized in Kibana. the dashboard provides the results of testing. figure 13 illustrate the number of test records and the percentage of Genuine and imposter from this total number.

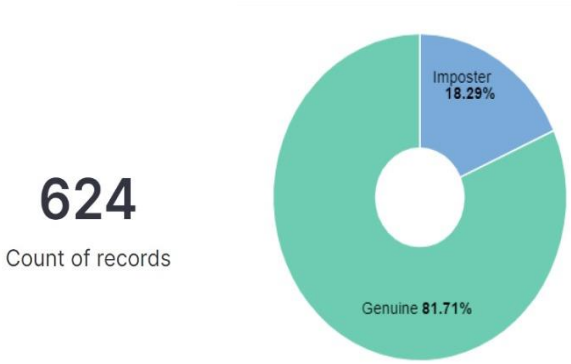

Figure 13:results of test records

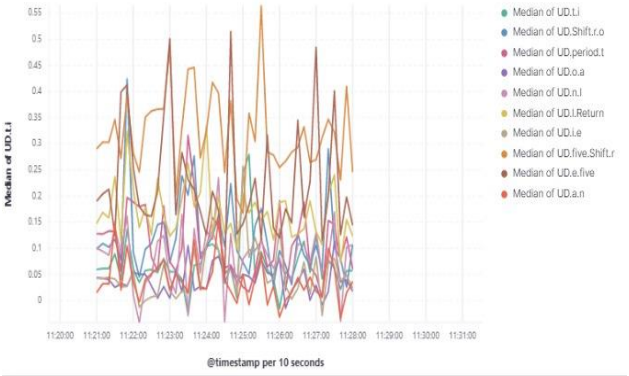

Figure 14: Median of up-down features in real-time

Figure 15:Median of down-down features in real-time

VIII. FUTURE WORK

Although it has less studies than other biometric modalities, keystroke dynamics is an exciting fields to explore as a sort of biometric authentication measure. Despite the fact that the field of study is still open to challenges and improvements, it has the potential to become a reliable, active, and low-cost biometric user authentication system.

The study on classification models development for keystroke dynamics has achieved a promising result as random forest result. However, the study's implementation did not address a number of problems. For starters, the study's classification implementation is based solely on a single model (random forest). Second, the study's model training is

limited to a single dataset. Although these limitations did not prevent the study's goal and goals from being achieved, it could have improved performance. As a result, future research may examine more complicated deep learning and unsupervised models for keystroke dynamics, such as autoencoders, recurrent neural networks, Local outliers factor and others. Building a deep learning& unsupervised models for the mobile platform is another future research project in the realm of keystroke dynamics.

IX. CONCLUSION

Multiple classifiers, such as statistical and machine learning, have been developed to classification between genuine users and imposter users as a result of advances in keystroke dynamics based on the rhythm of the typing.

This study aims to propose a model in keystroke dynamics using random forest method .Then for each class, How much is the probability of the new test data close to one of the 51 users, We set a threshold for this probability: if it's more than 0.4, it'll get the argmax for the predicted class; if it's less than 0.4, it'll be considered an imposter. Based on trial and error, we chose 0.4 since it produced exceptional results.

The scope of this research is focused to the implementation of a Random Forest model with a single dataset, and it excludes external factors that influence keystroke dynamics performance.

The evaluation metrics such as confusion matrix is selected and prioritized in this study to evaluate the performance of the random forest classifier. Based on the training result, the classifier has achieved the accuracy of 94% in classifying genuine user and impostor based on Fifty one users data as we have 400 record by user. So, we selected 300 records for training and the other 100 for testing.

X. REFERENCE

[1] Chang, H.-C., Li, J., Wu, C.-S., & Stamp, M. (2021). Machine Learning and Deep Learning for Fixed-Text Keystroke Dynamics. CoRR, abs/2107.00507. https://arxiv.org/abs/2107.00507

[2] Bhatia, A., Hanmandlu, M., Vasikarla, S., & Panigrahi, B. K. (2018). Keystroke Dynamics Based Authentication Using GFM. 2018 IEEE International Symposium on Technologies for Homeland Security (HST), 1–5. https://doi.org/10.1109/THS.2018.8574195

[3] Ayotte, B., Banavar, M., Hou, D., & Schuckers, S. (2020). Fast Free-Text Authentication via Instance-Based Keystroke Dynamics. IEEE Transactions on Biometrics, Behavior, and Identity Science, 2(4), 377–387. https://doi.org/10.1109/TBIOM.2020.3003988

[4] Hazan, I., Margalit, O., & Rokach, L. (2021). Supporting unknown number of users in keystroke dynamics models. Knowledge-Based Systems,221,106982. https://doi.org/https://doi.org/10.1016/j.knosys.2021.10698