IJCI v10 -01(2023) 36-53
International Journal of Computers and Information
(IJCI)
Available online at *https://ijci.journals.ekb.eg/*

Keystroke dynamics based user authentication using Histogram Gradient Boosting

Mina Ibrahim, Hussien AbdelRaouf [*], Khaled M. Amin, Noura Semary

Information Technology dept., Faculty of Computers and Information, Menoufia University, Shebin Elkom, 32511, Egypt.
Hussain.abdalraouf5689@ci.menofia.edu.eg

Abstract

User authentication is a vital part of securing digital services and preventing unauthorized users from gaining access to the system. Nowadays, organizations use Multi-Factor Authentication (MFA) to provide robust protection by utilizing two or more identity procedures instead of using Single Factor Authentication (SFA) which became less secure. Keystroke dynamics is a behavioral biometric that examines a user's typing rhythm to determine the subject's legitimacy using the system. Keystroke dynamics have a minimal implementation cost and do not need special hardware in the authentication process since the gathering of typing data is reasonably straightforward and does not involve additional effort from the user. In this work, we present an efficient approach that uses the quantile transformation that transforms data distribution into uniform distribution which significantly reduces the impact of outlier and extreme values. Histogram Gradient Boosting is employed as the primary classifier for the training and testing phase. Our proposed approach is evaluated on Carnegie Mellon University (CMU) keystroke benchmark dataset which has achieved 97.96% of average accuracy and 0.014 of average equal error rate (EER) across all subjects which outperforms all the previous advances in both machine and deep learning approaches.

Keywords: Keystroke Dynamics, Authentication, Histogram Gradient Boosting, CMU, Machine Learning.;

1. **Introduction**

One of the most crucial parts of digital systems is user authentication. Organizations and enterprises are constantly looking for suitable security solutions to handle identity fraud and information leakage, which are big concerns. Furthermore, the recent occurrences of the global pandemic have emphasized the value of securing and validating people while accessing enterprise and data via the internet. Not only it was more important than ever to be able to work from home, but it was also more important than ever to be able to complete daily duties remotely, such as finalizing insurance papers and requesting orders online. This is triggered by the fact that security breaches have increased after the recent occurrences [1].

Biometrics is a term that refers to something a person is or does. These kinds of properties could be nearly split into physiological and behavioral kinds [2]. The behavioral biometric paradigm of keystroke dynamics is based on a user's keyboard typing rhythm. The major benefits of keystroke dynamics are;

- **Uniqueness**: Software can measure keystroke inputs to millisecond precision [3]. As a result, replicating one's typing sequence at such a high resolution without exorbitant effort is impossible.

- **Low Cost**: In contrast to a typical physiological biometric system that depends on standalone devices and physical hardware, such as face recognition, iris, and fingerprint recognition, keystroke dynamics is fully implementable in the industry of software. [4], [5]. The advantage of getting little dependency on dedicated hardware lowers installation costs and creates the perfect setting for remote authentication. As a result, several service providers are seeking to expand their own solutions or hire a third party to validate their users' credentials [6].
- **Boost Password Lifespan and Strength**: Even though systems that entirely depend on a unique credential set are weak and vulnerable, passwords continue to be the most commonly used authentication mechanism. Keystroke dynamics have been investigated by scientists as a potential means for adding an extra layer of security and extending the life of passwords. [4].
- **Continuous Monitoring and Authentication**: Keystroke dynamics biometrics provide a mechanism to continuously verify a user's legal identity [7]. As long as users are continuing to interact with the system via input devices, keystroke patterns can be continuously examined and reevaluated.
- **Additional Security and Replication Prevention**: The combination of keystroke dynamics makes brute force password generation useless [7], and robbed credentials become completely irrelevant. Even if it is hacked, a new typing dynamics pattern can really be generated by selecting a new password.

Free text and fixed text are the two primary kinds of keystroke dynamics. The goal of free text keystroke dynamics is to authenticate user identity using unforeseen and impromptu text, which often necessitates large text instances and a lengthy training time. Fixed text, by contrast, concentrates on confirming user identity based on a specified repeated text that is generally short and needs a significantly shorter training time. Fixed text techniques are prominent among service suppliers for confirming a user's identity as he/she writes his/her login and password because of their obvious benefits. These techniques, consecutively, aid in the prevention of identity theft, fraud, and brute-force cyberattacks by adding a higher level of security without requiring the user to exert any more effort, as he/she is already typing his/her user name and password. In this paper, we consider two most important criteria to fulfill as follows:

- **Performance**: The recognition accuracy of the Keystroke dynamics should be improved and increased compared to the previous literature.
- **Robustness**: The most important criterion is to make sure that the proposed solution is robust against overfitting and underfitting problems. In addition, robust to noise and outliers.

It is very challenging to create a technique that satisfies the above-mentioned criteria. Therefore, in this work, we satisfy these two criteria by proposing an efficient technique that depends on efficient preprocessing and an excellent classifier in detecting whether this person is normal or an attacker. Numerous latest studies are proposed to develop and enhance keystroke dynamics authentication [Table I summaries these studies]. In this paper, we suggest a simple method that satisfies these previously mentioned criteria by using an efficient preprocessing and efficient classifier to develop and improve user authentication.

Our main contributions are:

- A novel approach for reducing the effect of extreme and outlier values using standard scaling and quantile transformation that transforms the data into uniform distribution which significantly increases accuracy and reduces the equal error rate.
- An efficient Histogram Gradient Boosting classifier which gives promising results in detecting imposter users.

The rest of our paper is arranged as follows. Section II presents the related work. The proposed approach with the preprocessing and learning algorithms is discussed in detail in section III. Section IV explains in detail the experimental results like dataset, evaluation metrics, and results. Finally, our work is summarized in Section V, which also gives a promising area for further research.

2. Related Work

In this section, we start with distance-based approaches and then we review more state-of-the-art works that use machine and deep learning approaches.

- **Distance Based Approaches**: Many algorithms based on distance measurements are an example of previous work in this domain. Euclidean distance was utilized in the beginning because of its straightforwardness [8], but it has limitations. It was quite susceptible to scale changes in the derived features, and it couldn't handle vector correlation. Mahalanobis distance uses data covariances to minimize heterogeneity in actual data. Mahalanobis distance is commonly utilized for evaluating features according to their covariance matrix [8]. One other approach, the Manhattan distance, has gained popularity due to its ease of calculation and breakdown into the contributions provided by every variable. According to [9], a distance based algorithm that utilizes a scaled Manhattan distance has the best performance of all distance-based approaches with an equal error rate of 9.6%.

- **Machine Learning Approaches**: Machine learning approaches have recently been heavily utilized in keystroke dynamics research such as nearest neighbor classifiers, K-means techniques, Bayesian classifiers, bagging classifiers and boosting classifiers. In [10] and [11], XGBoost with data augmentation and Random forest got an accuracy of 96.39% and 85% respectively on the CMU dataset. Ali. et al. [12] suggested partially observable hidden Markov model (POHMM). The POHMM, yields an ERR of 4.5%, which is a considerable enhancement over the HMM and additional effective techniques in user verification. A hybrid POHMM/SVM technique for user authentication is proposed in another work, which uses both generative and discriminative models. The suggested POHMM/SVM model has got 8.6% of average EER and 6.3% of standard deviation [13]. In [14], Generalised Fuzzy Model (GFM) is proposed, which is a set of Mamdani-Larsen and Takagi-Sugeno fuzzy models. For modelling, they employed the Gaussian Mixture Model (GMM) and they got 7.86% of equal error rate. In the context of behavioral biometrics, a modified differential evolution (MDE) based subspace anomaly detection mechanism is suggested for user authentication using keystroke dynamics features [15] which achieved an equal error rate of 3.48%. Alpar [16] suggested a new barcoding system that converts biometric data into storable barcodes in the form of very small barcode images. The primary algorithm for training and testing the barcodes is one-class support vector machines (SVM), and extremely encouraging outcomes are yielded provided the least EER of 1.83%. The K nearest neighbor (KNN) is utilized in many research studies. KNN is combined with dimensionality reduction and localization [17]. This approach is effective in dealing with outliers and scale variations with accuracy of 87.5%. Another research study used KNN with dependence clustering which achieved EER of 7.7% [18]. The clustering techniques are utilized in user authenticatoin which have proved efficient results. The X-means is used in [19] which is configured with two parameters. The two parameters are configured with 100 and 10 which refer to the max number of iterations and the max number of clusters respectively. This method got 94.2 of AUC and 11.2 of EER.

- **Deep Learning Approaches**: Deep networks are known for taking a long time to train so Adam optimization and LeakyReLU activation functions are frequently utilized to accelerate the training [20]. Their architecture consists of 3 hidden layers which have 100, 400, and 100 units respectively. LeakyReLU is utilized in the hidden layers and the softmax in the output layer. This architecture got an EER of 3% and overall accuracy of 93.59%. In [21], the autoencoder model is employed to enhance keystroke authentication. They used two phases, the encoder, and the decoder. The encoder phase resembles the MLP architecture, in which they used multiple hidden layers to extract the features. The decoder phase receives the extracted features through multiple hidden layers to reconstruct features similar to the original one. Then, an error is computed between the original features and reproduced features in the training phase. The encoded features with the least error are fed into a Gaussian Mixture Model to compute the anomaly probability. This method obtains an EER of 6.51. Muliono. et. al. [22] used many separate learning layers that are added to be capable of learning separately. They emphasized the usage of Nadam optimizer which showed high accuracy of 92.60%. Andrean. et. al. [23] proposed a deep learning technique using Multilayer Perceptron (MLP). Their MLP has 1 input layer and 2 hidden layers and 1 output layer. The input layer contains 31 neurons according to the number of dataset features. The hidden layer contains 23 neurons. This number of neurons is chosen based on the highest accuracy achieved by the model. This model achieved an optimum EER of 4.45%. Another research study [24] used 3 different feed forward neural network models. The first model has 3 hidden layers with 20, 30, 20 neurons respectively. The second model has 4 hidden layer with 20, 40, 40, 20 neurons respectively. The third model has 5 hidden layers with 20, 30, 40, 30, 20 neurons respectively. The greatest results are achieved using the setting of the first model with 3 hidden layers. They adopted resilient backpropagation (Rprop) mechanism changing the weights using a momentum operator to reduce the variation in weight fluctuates over successive repetitions. They achieved ERR equal to 4.9% for authentication with average identification accuracy of 94.7%. Table I summaries the state-of-the-art literature.

TABLE I: Recent Advances in CMU Dataset.

Reference	Classifier	Date	Performance	
			Accuracy (%)	EER (%)
[10]	XGBoost-augment	2021	96.39	-
[11]	Random forest	2021	85	-
[12]	Partially Observable Hidden Markov Model (POHMM)	2017	-	4.5
[13]	POHMM+SVM	2018	-	8.6
[14]	Generalized Fuzzy Model (GFM)	2018	-	7.86
[15]	Modified Differential Evolution (MDE)	2019	-	3.48
[16]	One-class SVM	2021	-	1.83
[17]	Kernel PCA with KNN.	2020	87.5	-
[18]	Dependence Clustering + KNN	2017	-	7.7
[19]	X-means with QT	2021	AUC is 0.942	11.2
[20]	Deep Secure	2017	93.59	3
[21]	Autoencoder model	2019	-	6.51
[22]	Deep Learning using optimizer (Nadam)	2018	92.60	-
[23]	MLP	2020	-	4.45
[24]	Feed Forward Multilayer Neural Network	2020	94.7	-

According to the literature, it’s common and popular that most researchers used different techniques either machine learning or deep learning techniques trying to develop and improve user authentication. In our work, we worked on two dimensions. The first is to process the data efficiently to reduce the outliers which impede the improvement. The second is to use an excellent classifier which has a great impact on the performance and robustness. Also, it’s obvious from the literature that all the studies are contributing to the EER and the accuracy which still needs further enhancement and development.

3. Proposed Approach

As shown in Figure.1, our proposed technique starts with data collection that is based on the CMU dataset which is collected from 51 users entering their keystroke dynamics 400 times for each user. Then, the Random Forest classifier is used to highlight the best features and remove the least important features. After performing feature reduction, standard scaling is applied to the data which makes the mean of the distribution equal to 0. After that, the important phase of quantile transformation is applied which converts the data distribution into normal distribution which has a great impact on the results of our experiments. These transformed features are fed into the Histogram Gradient Boosting classifier which has proven superb results comparable to the previous and other techniques. To the best of our knowledge, we are the first research study to use the Histogram Gradient Boosting classifier in the keystroke dynamics field. In the rest of this section, the techniques used in our approach like preprocessing and learning techniques are described in depth.

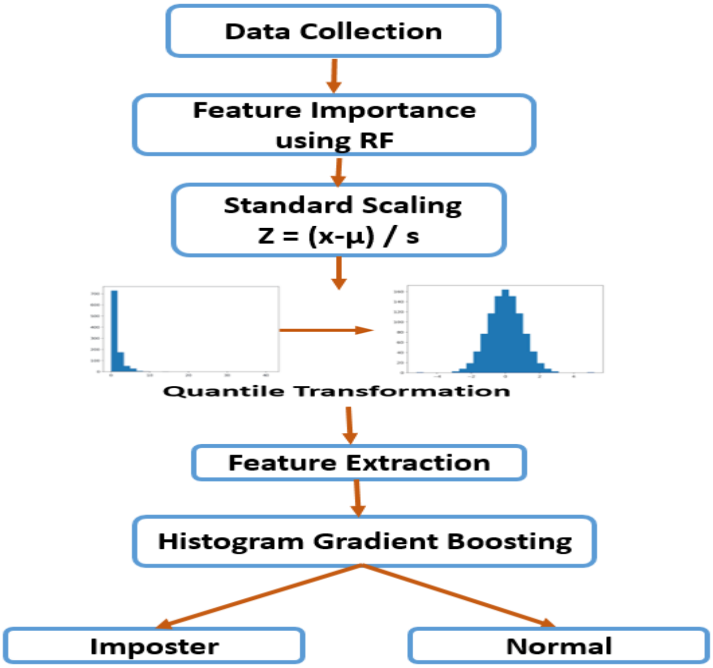


Fig. 1: Proposed Architecture

A. Preprocessing

- **Feature Importance**: It's very important to remove the redundant and irrelevant features that impede our model and have not any effect on the performance of our model. The feature importance shows which features are essential. By utilizing feature selection, this could help with a clear comprehension of the problem and, occasionally, result in model improvements. The random forest classifier is used as a feature importance technique which depends on measuring the impurity or uncertainty in the data [25] using the following entropy equation.

$$I_G = \sum_{j=0}^{c} Pj \quad (1)$$

where Pj: is the probability of the instances that fall to class c for a specific node. There are hundreds of decision trees in the random forest algorithm which work in parallel. Every decision tree computes the impurity of every feature using the entropy equation. As long as the impurity of the feature is less, the better feature's importance. According to the random forest, the impurity for every feature in every tree is accumulated to estimate the ultimate importance of the feature. The selection of features at the peak of the trees is typically more significant than that of features at the bottom nodes of the trees, as the upper divisions typically result in less entropy.

- **Standard Scaling**: The Standard Scaler assists in obtaining a standardized distribution with a zero mean and one standard deviation (unit variance) [26]. It standardizes features by subtracting the mean value from the feature and dividing the result by the standard deviation of the features as shown in eq.2. It helps in speeding up the calculations in our proposed method.

$$Z = (X - \mu)/\sigma \tag{2}$$

where μ: is the average of the training instances and σ: is the standard deviation of the training instances

- **Quantile Transformation**: is an efficient technique in which the features are changed to pursue a uniform or normal distribution. Every feature is exposed to this transformation on its own. The original features are mapped to new features that are subjected to a uniform distribution based on the cumulative distribution function [27]:

$$\mathrm{F(x)} = (x - a\ )/(b - a) \tag{3}$$

where a and b: are two constants such that a < x <b. This technique is very practical and effective in reducing the anomalies and outliers which impacts greatly on the performance.

B. Learning Algorithms

A wide range of learning approaches are investigated in our experiments. This subsection introduces these learning methods.

- **K-Nearest Neighbor (KNN):** is a supervised technique which is used for classification and regression problems. This technique allocates new data depending on how similar or close it is to the points in the training data [28]. The number of neighbors examined to classify the new data point is denoted by 'K'.

- **Support Vector Machine (SVM):** It's a supervised machine learning algorithm. It is based on the idea of maximizing the distance between different classes [29]. The data items/features are plotted as points and the SVM algorithm should separate the classes using the most suitable hyperplane.

- **Random Forest (RF**): It consists of several numbers of decision trees; it is considered to be an ensemble method [30]. It makes the predictions by averaging the predictions from the various decision trees. It solves the limitation of the decision tree which reduces the overfitting, and it is more accurate than decision trees.

- **Decision Tree (DT):** It is commonly used for classification and regression tasks (binary and multiclassification) [31]. It works as follows: it learns some rules from the features of the data called decision rules, which force the decision tree to predict the target variable.

- **Naive Bayes (NB):** It is one of the most popular and efficient probabilistic models that is used for classification problems. The naive Bayes model dramatically improves learning by supposing that features are independent of a given class. In reality, naive Bayes frequently performs well comparable with many other complex classifiers [32].

- **Histogram Gradient Boosting (HGB)**: It is one of the boosting techniques that is very efficient and have proven recently its efficiency and improvement of the performance and the time [33]. Boosting techniques depend on integrating multiple weak learners to form a powerful and high effective strong learner as shown in Figure 2.

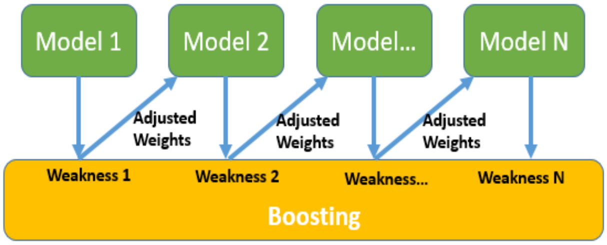


Fig. 2: The Principle of Boosting Technique

The first model is trained on the weighted data. Consequently, some of the features might participate frequently in the new sets of the other models. The first model identifies the misclassified data samples and then adjusts and increases their weights. These adjusted weights are fed into the second model so that, it can focus on these incorrect predictions and can fix them. This process is repeated until the n models are implemented as shown in this Figure 2. The boosting techniques use the same classifier for the n models like decision tree or any other classifier.

Gradient boosting is a boosting technique that uses a decision tree in its implementation. It's a newly developed algorithm for classification and regression problems that performs so well across a wide range of different datasets [34]. Gradient boosting has a big drawback in that it takes a long time to train the model. The problem is related to the nature of the decision tree itself and the huge number of features and samples. The construction of the decision tree depends on the splits or divisions for all the continuous values and all the continuous features. This takes a long time and heavy computation, especially when dealing with thousands of data samples and features. This also degrades the performance.

The HGB comes up with a solution to the problems of efficiency and a huge dataset. From its name, it uses the histogram to bin the continuous samples into a constant number of bins as shown in Figure 3.

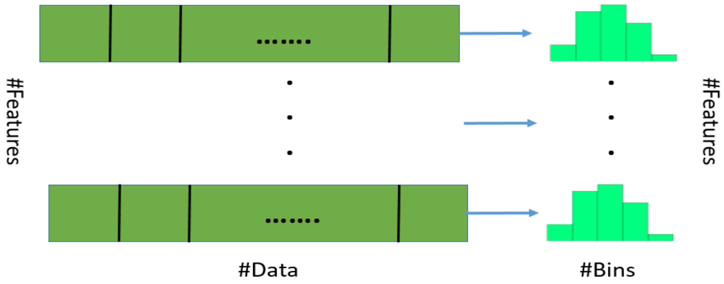


Fig. 3: Converting Data into Bins

This technique reduces the distinct values for all the features into a few and a small number of values. This enables the decision tree to work on a small number of features and then the splits or the division of the tree reduce. This improves and speeds up the implementation of the trees comparable to the gradient boosting. In our implementation, HistGradientBoostingClassifier class from the sklearn library is used to implement HGB.

4. Experiments and Results

In this section, we represent the CMU dataset, evaluation metrics, data exploration, feature importance, quantile transformation effect, and the results.

A. Dataset

The Carnegie-Mellon University (CMU) dataset contains information on 51 users' keyboard dynamics, in which the password ".tie5Roanl" was input 400 repetitions by every user, with 50 times in every of 8 sessions. A user has to delay a minimum of one day between sessions in order to take the day-to-day fluctuation of every user' s rhythm [9]. Figure 4 shows different timing features given to our model as input, with the down arrow indicating key press and the up arrow indicating key release [24]. In Figure 4, Hold Time refers to the time taken for a single key to be pressed and released. Down Down Time refers to the amount of time from one key press to the next key press. Up Down Time refers to the time taken from the key release to the next key press. The CMU dataset contains 31 timing features. They are classified as follows: 11 features as Hold Time, 10 features as Down Down Time and 10 features as Up Down Time.

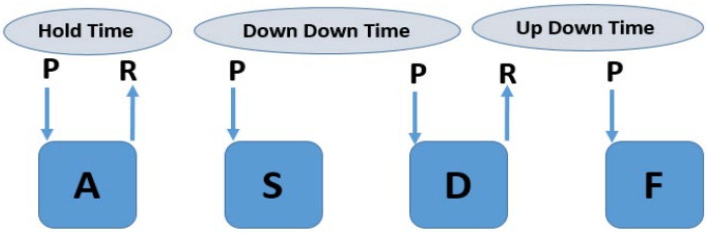


Fig. 4: Keystroke Dynamic Features

B. Evaluation Metrics

Most of the previous studies use either the accuracy or Equal Error Rate (EER) to measure model performance. In our study, the accuracy, EER, precision, recall, and F1 score are used to make sure that our model performance is robust and not stuck in overfitting or underfitting problems. The accuracy can be expressed as shown below in eq.4. False Acceptance Rate and False Rejection Rate are used for calculating Equal Error Rate [35] as shown below in eq.5 and eq.6. The EER is calculated from the ROC curve which is the point where the false acceptance and false rejection rates are equal. Generally, the smaller the equal error rate, the higher the biometric system's accuracy. In addition, the recall, precision, and f1 score are used as formulated in eq.7, 8, and 9 which TN and TP refer to true negative and true positive while FN and FP refer to false negative and false positive respectively.

$$Accuracy= (TP+TN) / (TP+TN+FN+FP) \tag{4}$$

$$False\ Acceptance\ Rate = FP/ (FP + TN) \tag{5}$$

$$Fasle\ Rejection\ Rate = FN\ /\ (\mathrm{FP} + \mathrm{TN}) \tag{6}$$

$$Recall = \ TP\ /\ (\mathrm{FN} + \mathrm{T}P) \tag{7}$$

$$Precision = \ TP\ /\ (\mathrm{FP} + \mathrm{T}P) \tag{8}$$

$$\mathrm{F1\ Score} = (2*Precision*Recall)\ /\ (\mathrm{Precision} + Recall) \tag{9}$$

C. Data Exploration

In the CMU dataset, there are 31 timing features that may be classified into three groups: Down Down (DD), Up Down (UD), and Hold Time (H). It is very important to do some statistics to see if there's a substantial difference between these three groups. Four users of the 51 users are chosen randomly for data discovery.
Every line graph in Figure 5 depicts the 400 input samples for a certain user. It is obvious from this diagram that the majority of the feature samples are quite constant, following a comparable pattern for each user. This shows that users are rather consistent when it comes to this specific feature group (Down Down). This observation can be interpreted as a sign that the users have the possibility to be correctly classified. When the average cases of the four subjects are compared in Figure 6, the results demonstrate that the users' typing patterns are pretty similar.

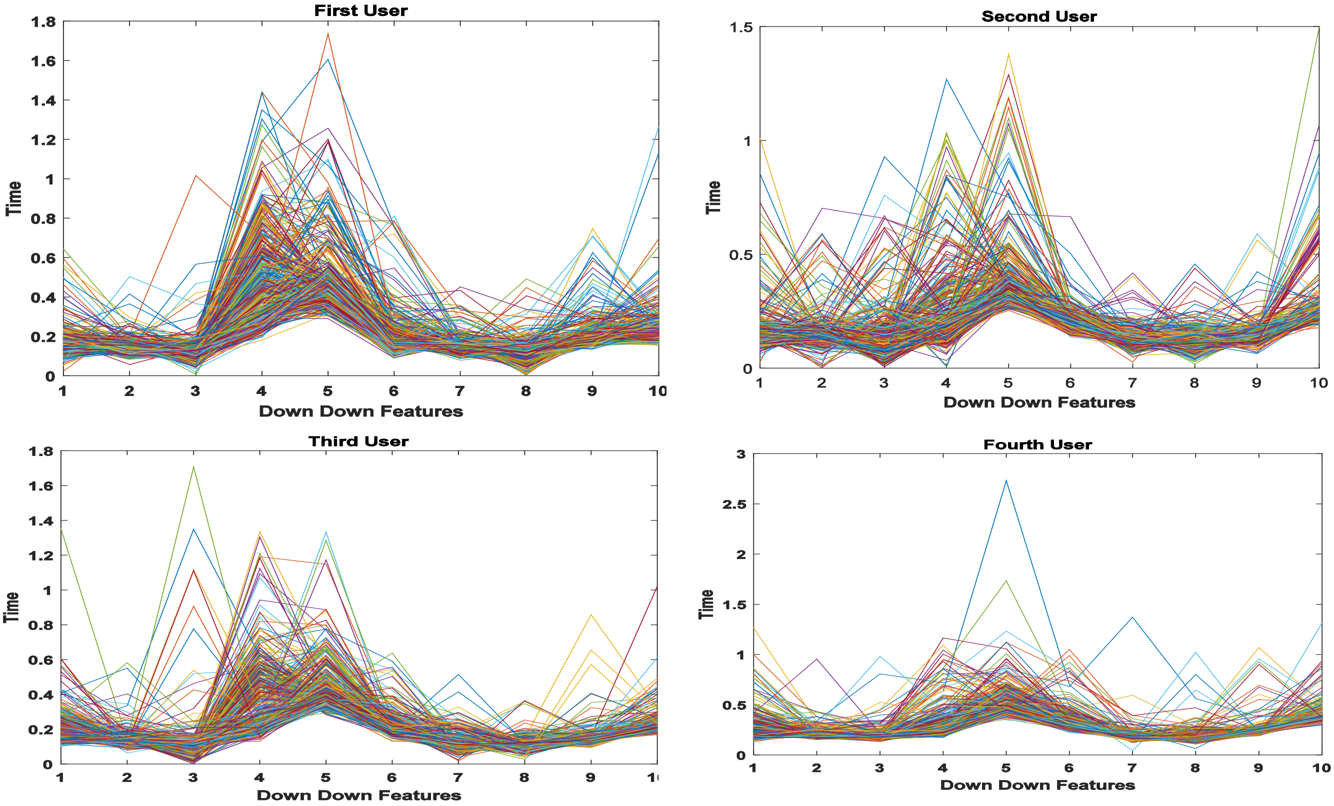


Fig.5: Down Down timing features for four subjects

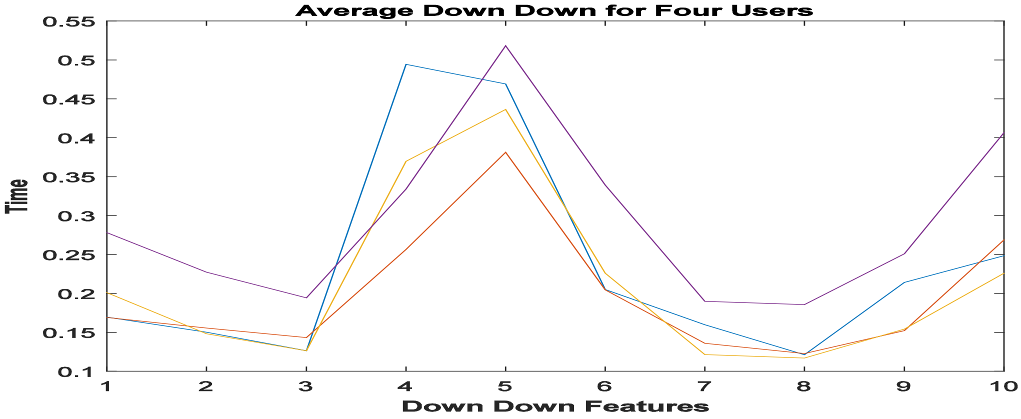


Fig.6: The average Down Down timing features for four subjects

Figure 7 and Figure 8 show the results for the Up Down timing features and their average. This data appears to be comparable to the Down Down timing features in Figure 5 and Figure 6.

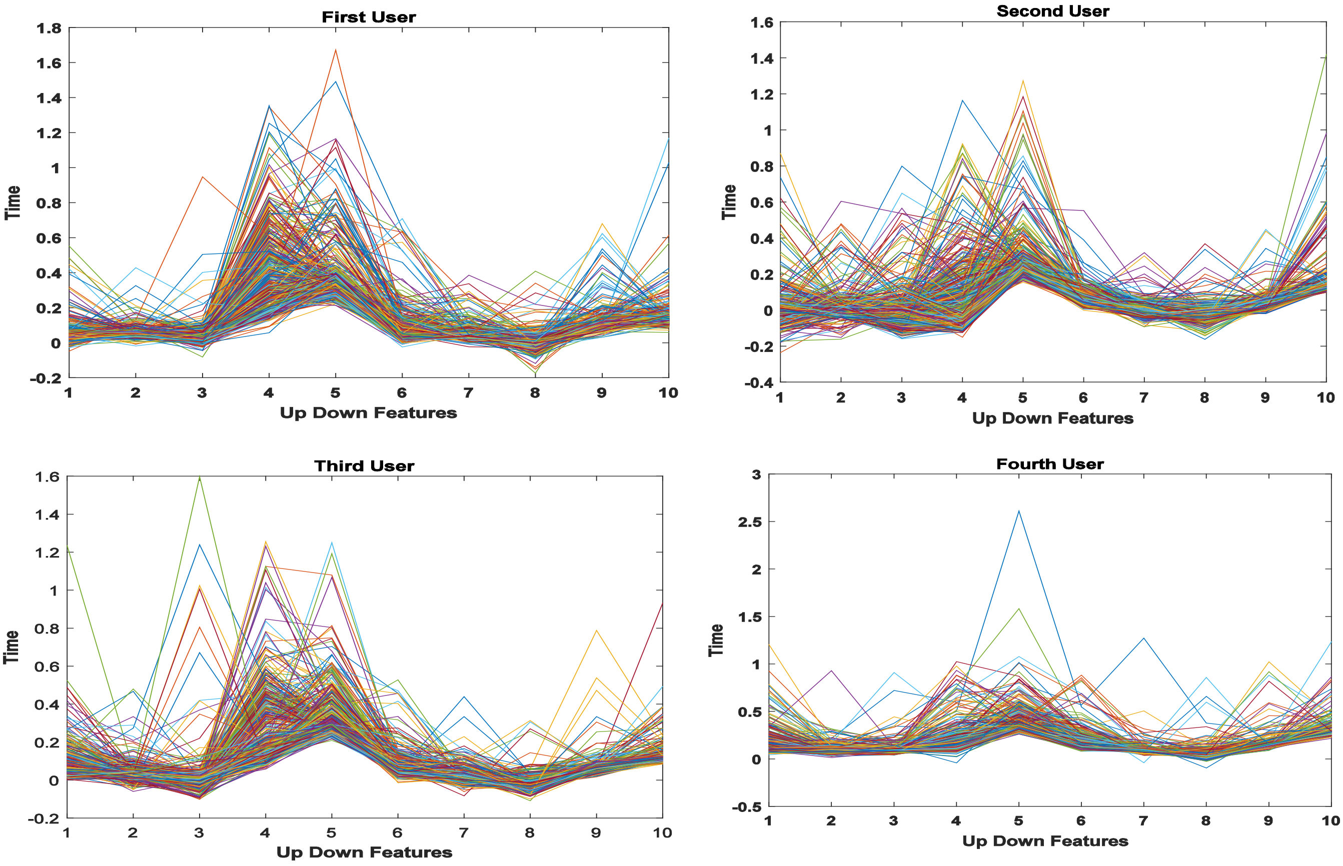


Fig. 7: Up Down timing features for four subjects

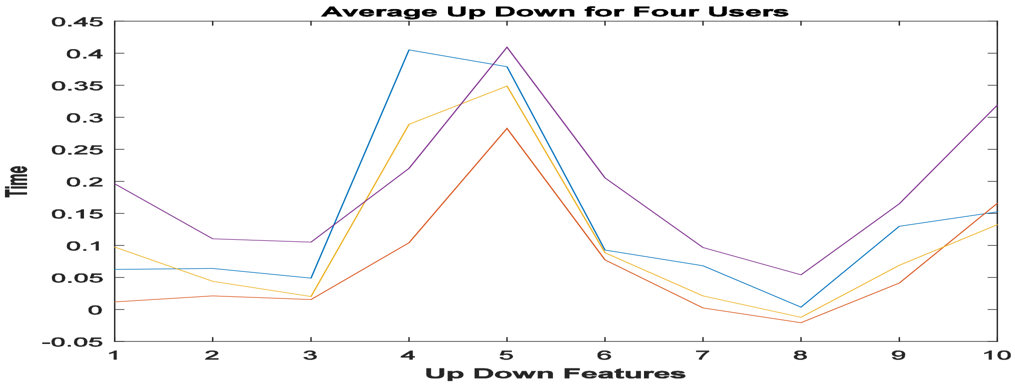


Fig. 8: The average Up Down timing features for four subjects

The four users according to the hold-time timing feature are compared in Figure 9, and the variations are more obvious here. The average examples in Figure 10 demonstrate even more significant disparities. The hold time should really be a major factor for differentiating users, according to these observations.

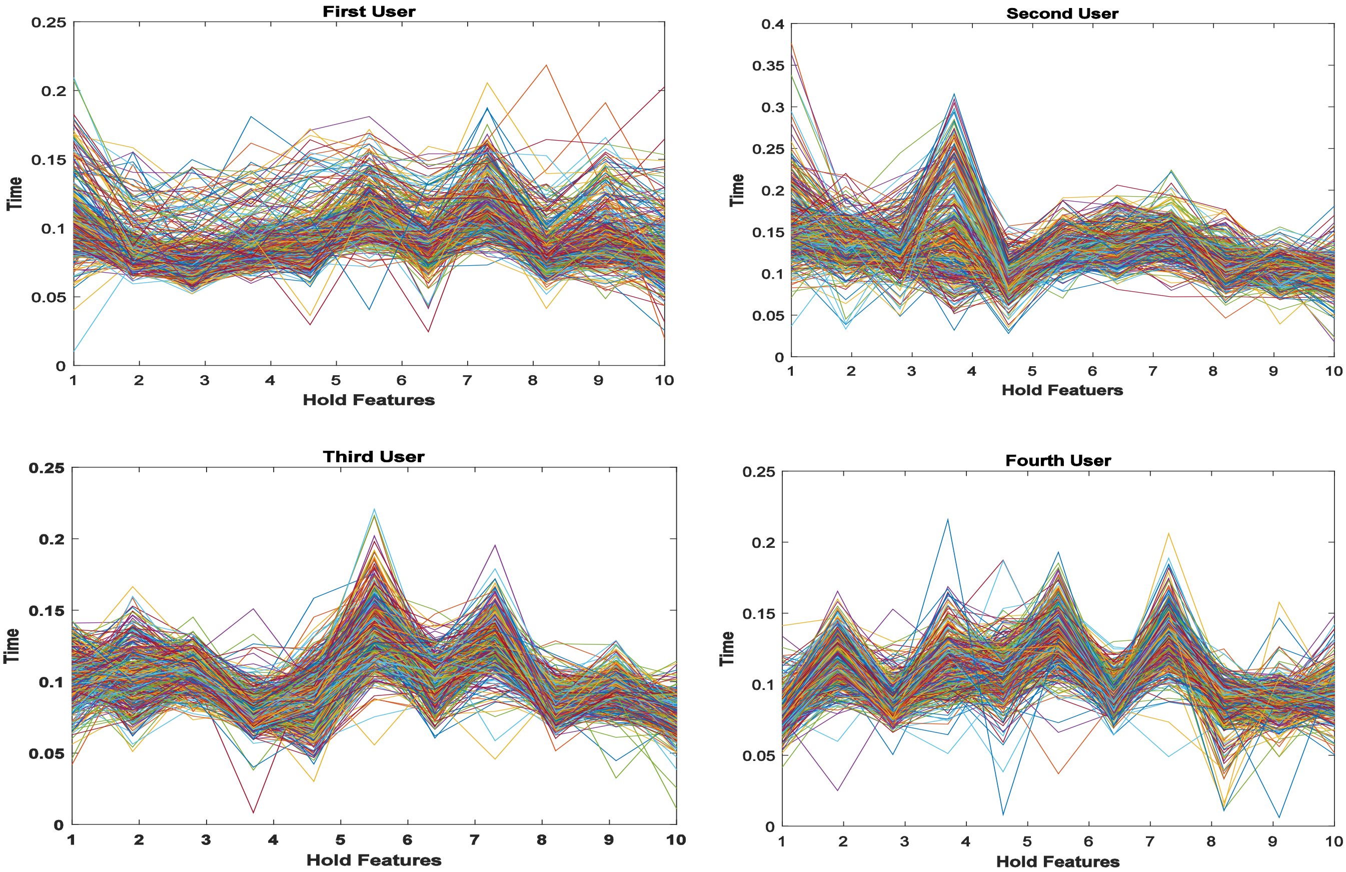


Fig. 9: Hold Time timing features for four subjects

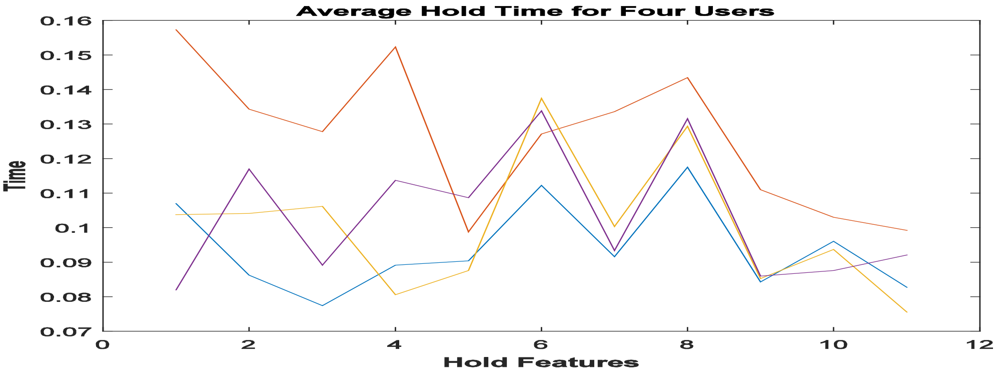


Fig. 10: The average Hold Time timing features for four subjects

D. Feature Importance

The importance scores for each feature of our dataset are shown in Figure 11 which the “rep and “sessionIndex” features are the least important compared to all other features. This is very reasonable because they don’t capture any information about user behavior. So it’s important to remove these two features so that our model can perform better. The selected features are 31 features which are (H.period, DD.period.t, UD.period.t, H.t, DD.t.i, UD.t.i, H.i, DD.i.e, UD.i.e, H.e, DD.e.five, UD.e.five, H.five, DD.five.shift.r, UD.five.shift.r, H.shift.r, DD.shift.r.o, UD.shift.r.o, H.o, DD.o.a, UD.o.a, H.a, DD.a.n, UD.a.n, H.n, DD.n.l, UD.n.l, H.l, DD.l.return,

UD.l.return, H.return) and the removed features are (Rep, sessionIndex). The selection of important features and removal of trivial features give us the best possible results.

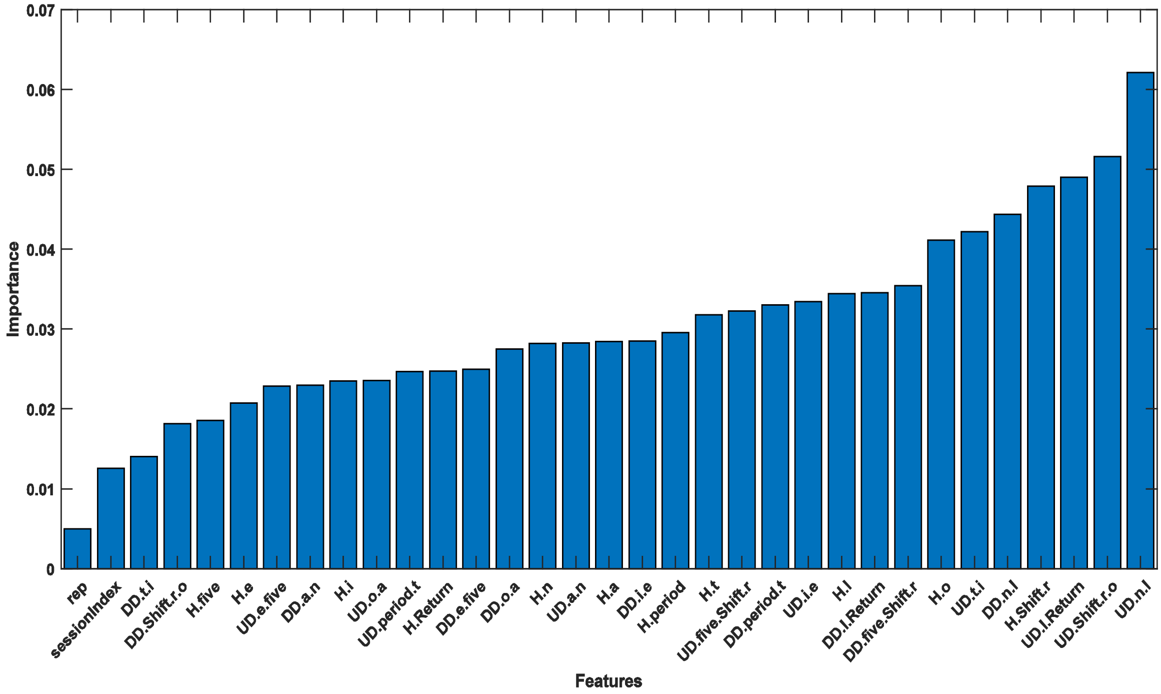
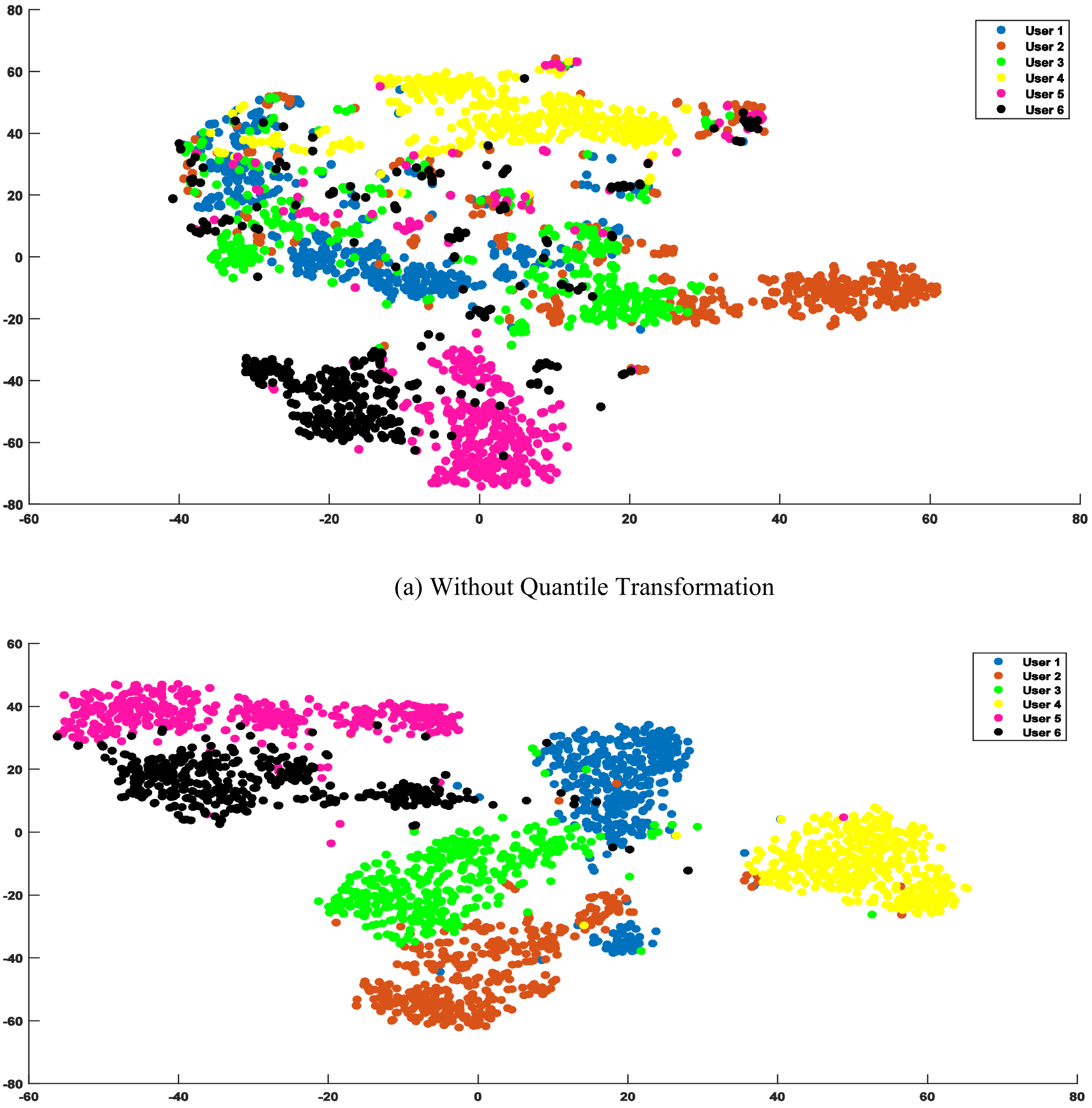


Fig. 11: Feature Importance

E. Quantile Transformation Effect

The data of six users from the dataset are represented in Figure 9 to highlight the significance of quantile transformation. For every user, 400 instances are considered. For each instance, the features are extracted and visualized using the two-component t-SNE technique [36]. The same data is shown in two ways: in Figure 12 (a), it is shown without any preprocessing, and in Figure 12 (b), it is shown with quantile transformation. The use of quantile transformation, as shown in Figure 12 (b), assists to bring users' anomalies and non-anomalies closer to one another, which makes it much easier for our model to identify users. Even though the samples of the six users are combined together, the correct clusters can be easily spotted once the quantile transformation has been applied.



(a) Without Quantile Transformation

(b) With Quantile Transformation

Fig. 12: Projection of 6 users' features without (a) and with quantile transformation (b)

F. Results

In our experiments, the proposed methodology is tested on the CMU Benchmark Dataset. The evaluation procedure begins with a user as a genuine user and the other 50 users as imposter users. 400 timing features for that user are taken as genuine and 400 timing features for the other 50 users as an imposter, random 8 timing features for each 50 users, yielding a total of 800 timing features, then total 800 timing features are split into 70% for training and 30% for testing. This evaluation procedure has been carried out for every user which treats that particular user as genuine and the others as impostors. This evaluation process for CMU Dataset is mentioned in [9].Our approach is evaluated with Histogram Gradient Descent and many other state-of-the-art algorithms like Decision Tree, Random Forest, Support Vector Machine (SVM), K Nearest Neighbor (KNN), and Naive Bayes. The performance and robustness are satisfied as followed.

- **Performance**: Many state-of-the-art algorithms are used to compare and prove that our methodology with the HGB classifier is outperforming them. The algorithms are trained on the CMU dataset and tested to evaluate the models' performance. In Table II, the results for five machine learning techniques are compared with Histogram Gradient Descent by calculating accuracy, precision, recall, and f1 score. The result for our approach is marked by the red color as shown in Table II.

TABLE II: Authentication Results on CMU dataset

Algorithm/ Metric	Accuracy	Precision	Recall	F1 Score
KNN	0.969	0.960	0.980	0.969
SVM	0.966	0.973	0.959	0.966
RF	0.970	0.985	0.956	0.970
NN	0.946	0.962	0.929	0.945
DT	0.839	0.856	0.838	0.839
HGB	0.979	0.985	0.973	0.979

The accuracy and EER for each of the six algorithms are represented in Figure. 13 in which Histogram Gradient Boosting comes first with the highest accuracy of 97.96% and with the lowest EER of 0.01.

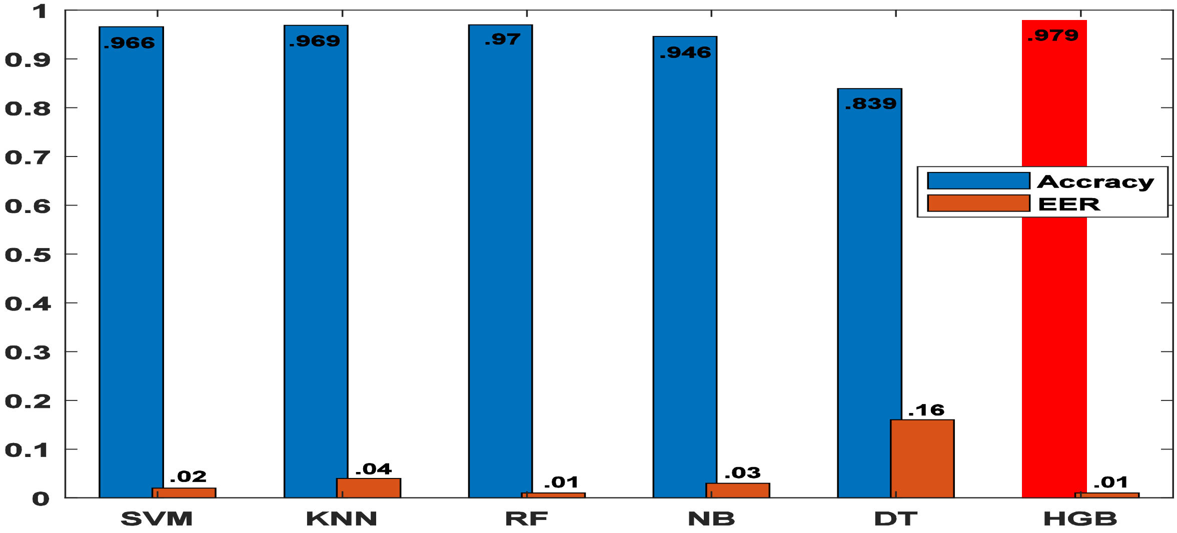


Fig. 13: Accuracy and EER for each model in our Experiments

- **Robustness**: To show how robust and efficient our technique is. The Roc Curve is used as an excellent metric for proving that. The Roc Curves for all 51 users are visualized together in Fig. 14 which all curves are approaching the top. The AUC average for 51 ROC Curves is 99.7% which represents the success and efficiency of our approach.

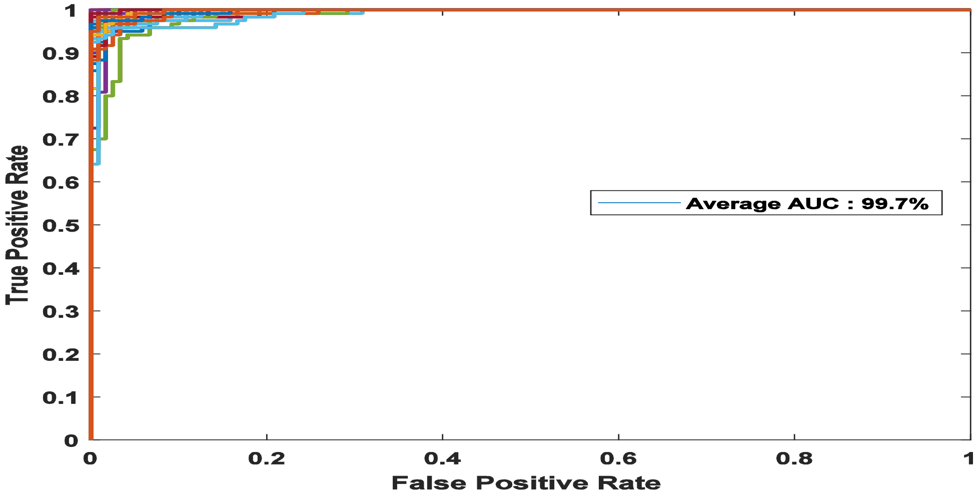


Fig. 14: ROC Curves for 51 Subjects

To compare our results with the literature, we find that the literature uses either accuracy or EER in their studies, so our results are compared according to their evaluation metric. In Fig. 15, the accuracy of our model is compared with the accuracy of previous studies, in which our model outperforms all the previous algorithms.

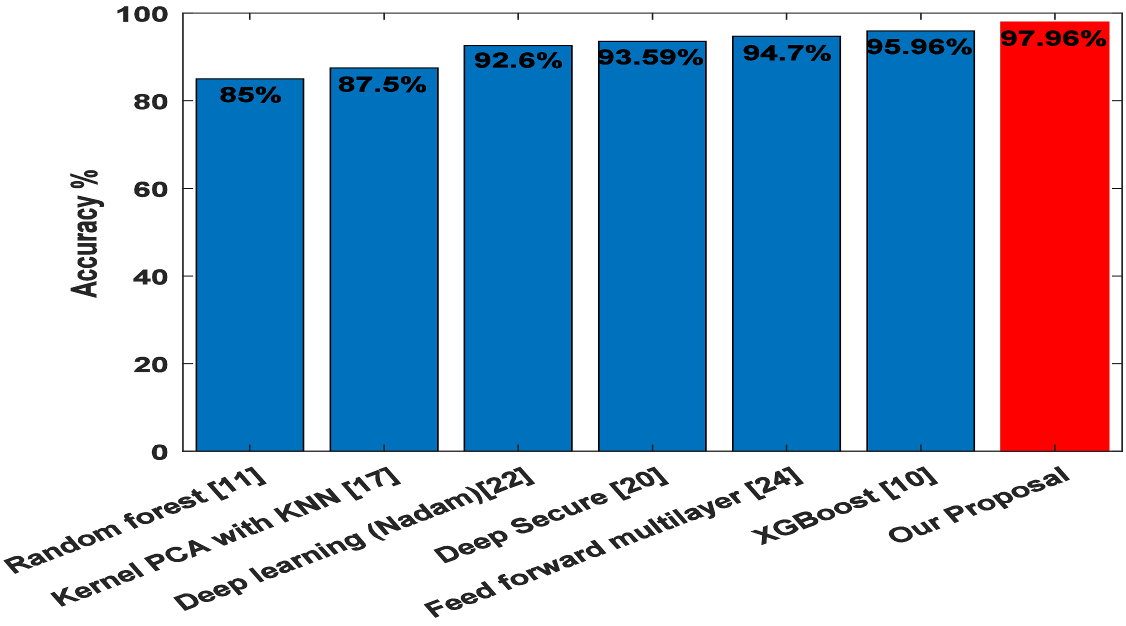


Fig. 15: Comparison with Literature using Accuracy

In Fig. 16, the EER of our model is compared with the EER of the previous studies, and our model gets the lowest EER.

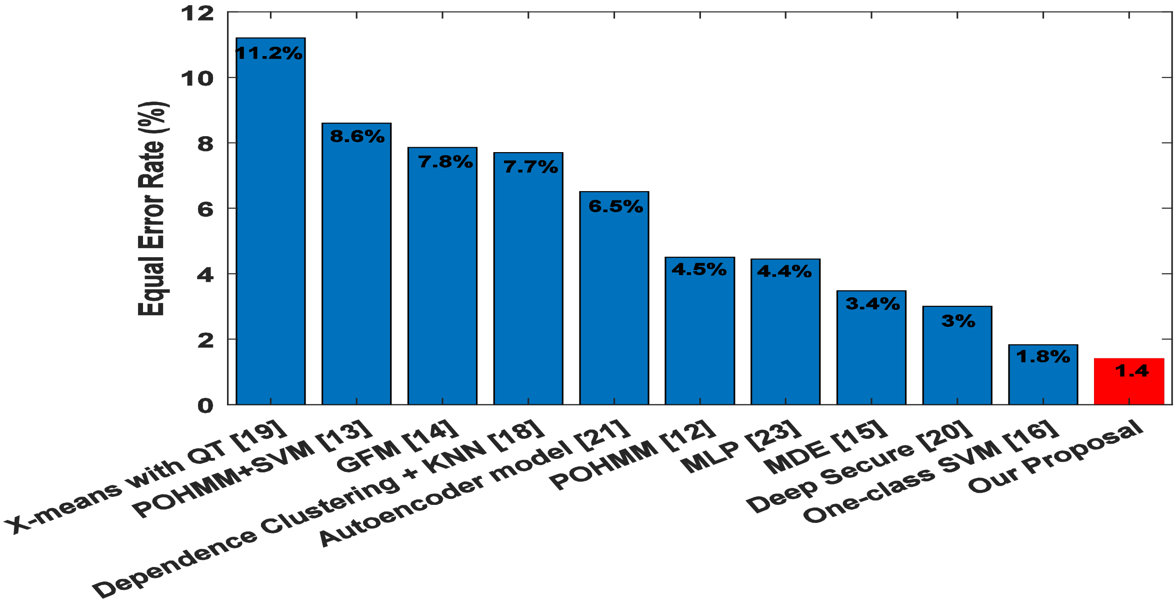


Fig. 16: Comparison with Literature using EER

It's obvious from our results that our methodology has proven great results and efficiency compared to the previous work. Quantile transformation played an important role in our work, which significantly succeeded in the reduction of outliers. This is a very important point to make sure that the data is free from the outliers that impede and degrade the performance. The novelty of the Histogram Gradient Boosting with the free outlier features had a great achievement in terms of performance and robustness.

5. Conclusion and Future Work

In this work, an intensive analysis of the CMU dataset is represented which has a promising ability to distinguish between normal and imposter users. A successful approach is introduced which employs the most recent advancements in prepossessing techniques and machine learning to automatically discover and learn keystroke data feature interactions. An efficient prepossessing technique that depends on quantile transformation with Histogram Gradient Boosting as the main classifier is utilized for this methodology which has given magnificent results. Experiments on the CMU dataset show that our suggested method performs significantly better than all other methods in both evaluation criteria. Our future work is to fine-tune the champion model and extend our research to use deep learning techniques like transfer learning and generative adversarial network (GAN).

References

[1] https://www.interpol.int/en/News-and-Events/News/2020/Preventing-crime-and-protecting-police-INTERPOL-s-COVID-19-global-threat-assessment. "Preventing crime and protecting police: INTERPOL's COVID-19 global threat assessment". In: Interpol Website (2022).

[2] Lawrence O'Gorman. "Comparing passwords, tokens, and biometrics for user authentication". In: Proceedings of the IEEE 91.12 (2003), pp. 2021–2040.

[3] Christian Senk and Florian Dotzler. "Biometric authentication as a service for enterprise identity management deployment: a data protection perspective". In: Sixth International Conference on Availability, Reliability and Security. 2011, pp. 43–50.

[4] Pin Shen Teh, Andrew Beng Jin Teoh, and Shigang Yue."A survey of keystroke dynamics biometrics". In: The Scientific World Journal 2013.

[5] Eesa Al Solami et al. "Continuous biometric authentication: Can it be more practical?" In: 12th International Conference on High Performance Computing and Communications (HPCC). 2010, pp. 647–652.

[6] Robert Moskovitch et al. "Identity theft, computers and behavioral biometrics". In: International Conference on Intelligence and Security Informatics.IEEE. 2009, pp. 155–160.

[7] Willem G De Ru and Jan HP Eloff. "Enhanced password authentication through fuzzy logic". In: IEEE Expert 12.6 (1997), pp. 38–45.

[8] Yu Zhong, Yunbin Deng, and Anil K Jain. "Keystroke dynamics for user authentication". In: computer society conference on computer vision and pattern recognition workshops.2012, pp. 117–123.

[9] Kevin S Killourhy and Roy A Maxion. "Comparing anomaly-detection algorithms for keystroke dynamics". In: International Conference on Dependable Systems & Networks.2009, pp. 125–134.

[10] Han-Chih Chang et al. "Machine Learning and Deep Learning for Fixed-Text Keystroke Dynamics". In: Cybersecurity for Artificial Intelligence. Springer, 2022, pp. 309–329.

[11] Adesh Thakare et al. "A Machine Learning-Based Approach to Password Authentication Using Keystroke Biometrics". In: Machine Learning, Deep Learning and Computational Intelligence for Wireless Communication. Springer, 2021, pp. 395–406.

[12] Md Liakat Ali, John V Monaco, and Charles C Tappert. "Biometric studies with hidden Markov model and its extension on short fixed-text input". In: 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON). 2017, pp. 258–264.

[13] Md L Ali, Kutub Thakur, and Muath A Obaidat. "A Hybrid Method for Keystroke Biometric User Identification". In: Electronics 11.17 (2022), p. 2782.

[14] Aparna Bhatia et al. "Keystroke dynamics based authentication using gfm". In: International Symposium on Technologies for Homeland Security (HST). 2018, pp. 1–5.

[15] Gutha Jaya Krishna and Vadlamani Ravi. "Keystroke based user authentication using modified differential evolution". In: TENCON Region 10 Conference (TENCON).2019, pp. 739–744.

[16] Orcan Alpar. "Biometric keystroke barcoding: A next-gen authentication framework". In: Expert Systems with Applications 177 (2021), p. 114980.

[17] Chinmay Sahu, Mahesh Banavar, and Stephanie Schuckers. "A novel distance-based algorithm for multiuser classification in keystroke dynamics". In: 2020 54th Asilomar Conference on Signals, Systems, and Computers.2020, pp. 63–67.

[18] Elena Ivannikova, Gil David, and Timo Hamalainen."Anomaly detection approach to keystroke dynamics based user authentication". In: Symposium on Computers and Communications (ISCC).2017, pp. 885–889.

[19] Itay Hazan, Oded Margalit, and Lior Rokach. "Supporting unknown number of users in keystroke dynamics models". In: Knowledge-Based Systems 221 (2021), p. 106982.

[20] Saket Maheshwary, Soumyajit Ganguly, and Vikram Pudi. "Deep secure: A fast and simple neural network based approach for user authentication and identification via keystroke dynamics". In: IWAISe: First International Workshop on Artificial Intelligence in Security. Vol. 59. 2017.

[21] Yogesh Patel et al. "Keystroke dynamics using auto encoders". In: International Conference on Cyber Security and Protection of Digital Services (Cyber Security). 2019, pp. 1–8.

[22] Yohan Muliono, Hanry Ham, and Dion Darmawan "Keystroke dynamic classification using machine learning for password authorization". In: Procedia Computer Science 135 (2018), pp. 564–569.

[23] Alvin Andrean, Manoj Jayabalan, and Vinesh Thiruchelvam. "Keystroke dynamics based user authentication using deep multilayer perceptron". In: International Journal of Machine Learning and Computing 10.1 (2020), pp. 134–139.

[24] Ahmet Melih Gedikli and Mehmet O¨ nder Efe. "A simple authentication method with multilayer feedforward neural network using keystroke dynamics". In: Mediterranean conference on pattern recognition and artificial intelligence. Springer. 2019, pp. 9–23.

[25] T Daniya, M Geetha, and K Suresh Kumar. "Classification and regression trees with gini index". In: Advances in Mathematics Scientific Journal 9.10 (2020), pp. 1857–8438.

[26] Maro Choi et al. “Keystroke dynamics-based authentication using unique keypad”. In: Sensors 21.6 (2021), p. 2242.

[27] https://machinelearningmastery.com/quantile-transforms-for-machine-learning/. “How to Use Quantile Transforms for Machine Learning”. In:machinelearningmastery.com (2022).

[28] Jiankun Hu, Don Gingrich, and Andy Sentosa. “A k-nearest neighbor approach for user authentication through biometric keystroke dynamics”. In: International Conference on Communications.2008, pp. 1556–1560.

[29] Hayreddin C¸ eker and Shambhu Upadhyaya. “User authentication with keystroke dynamics in long-text data”. In: 8th International Conference on Biometrics Theory, Applications and Systems (BTAS).2016, pp. 1–6.

[30] Paulo Angelo Alves Resende and Andr´e Costa Drummond. “A survey of random forest based methods for intrusion detection systems”. In: ACM Computing Surveys (CSUR) 51.3 (2018), pp. 1–36.

[31] Bahzad Charbuty and Adnan Abdulazeez. “Classification based on decision tree algorithm for machine learning”. In: Journal of Applied Science and Technology Trends 2.01 (2021), pp. 20–28.

[32] Irina Rish et al. “An empirical study of the naïve Bayes classifier”. In: IJCAI workshop on empirical methods in artificial intelligence. Vol. 3. 22. 2001, pp. 41–46.

[33] Alexey Natekin and Alois Knoll. “Gradient boosting machines, a tutorial”. In: Frontiers in neurorobotics 7 (2013), p. 21.

[34] Aleksei Guryanov. “Histogram-based algorithm for building gradient boosting ensembles of piecewise linear decision trees”. In: International Conference on Analysis of Images, Social Networks and Texts. Springer. 2019, pp. 39–50.

[35] John Swets. Evaluation of diagnostic systems. Elsevier, 2012.

[36] Laurens Van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.” In: Journal of machine learning research 9.11 (2008).