Proficient User Authentication based on the Dynamic keystroke using Machine Learning

MSc Research Project
Cyber Security

Trecy Soumya
Charles
Student ID: x21143650

School of Computing
National College of Ireland

Supervisor: Michael Pentridge

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Trecy Soumya Charles		
Student ID:	x21143650		
Programme:	MSc in Cyber Security	**Year:**	2022-23
Module:	Research Project		
Supervisor:	Micheal Pantridge		
Submission Due Date:	15/12/2022		
Project Title:	Proficient user authentication based on dynamic keystroke using machine learning		
Word Count:	4975	**Page Count: 19**	

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Trecy Soumya Charles

Date: 15/12/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Proficient User Authentication Based on the Dynamic keystroke using Machine learning

Trecy Soumya Charles
x21143650

Abstract

The authentication conditions of today go beyond PINs and passwords. It needs a significant amount of Keystroke biometrics can offer security. even though they have the login. This study attempts to identify a legitimate user using his information. The paper's objective is to examine the various approaches and reach an agreement. By including a keystroke mechanism, The existing setup increases safety. In the industry of information technology, which is quickly growing User authentication is one of the time-consuming tasks of today. Everyone must employ swift as well as safe authentication. We developed and implemented the keystroke to address this problem technology. Keystroke data may be successfully used to imitate user input, which is our main assumption behavior. Utilizing a neighbor key pattern enhances the precision of user identification and assists in identifying legitimate users and pretenders. A few significant keystroke variables have been used to establish the User Type. The space between successive strokes of the user identifying characters is utilized in this method. using behavioral characteristics like the typing habits of multiple users. Executing a Combining parameters like recall, F1, accuracy, precision, and run time in comparison analysis is a important part of the research.

1 INTRODUCTION

In this fast-paced world, security is major worry when it comes to authentication systems electronic age. The experts employ several authentication techniques, like as comprising a person's voice, fingerprint, and retinal biometric information. They desire to create a recognition system using deep learning, artificial intelligence, and conventional methods acquiring skills. Each system has its own benefits and drawbacks, but the commercially available fingerprints show that the technology is currently advancing. Items offered for sale. However, to deploy such systems, more sensors are needed. Input requires those sensors, even though their cost varies on the market to the system's reader. Our goal is to implement a keystroke-based authentication system in proposed tactic. The unique method that each person types on the keyboard using each key is referred to as their keyboard habit. Since the system doesn't require any, we believe that because it doesn't require any additional sensors and is user-friendly, it may be used as a promising identification method that is less expensive than the biometric one by only identifying the traits from the typing habits of each user. Unusual biometric. A technique of authentication has been developed that authenticates and identifies users using their typing style Over the past ten years or so, it has been growing. A user's timing is used by Keystroke

Dynamics. data recorded during key press and key release, commonly referred to as keystroke latencies, to build a template for every user, which is then contrasted with the timing information across login process. Because user input patterns are so varied and dynamic, similarity measuring is a difficult problem. It may not be best to use the traditional Nearest Neighbor distance measurement. Machine learning is a better option to address this issue because learning techniques can typically create models that are more adaptive than Nearest Neighbor. Machine learning, often known as experience, is a method for automatically enhancing algorithms by extracting knowledge from existing data. It has been applied in a variety of fields, including speech recognition, time series prediction, and image recognition. Due to the intricacy of the data, these activities are typically challenging. Machine learning has successfully been applied in many fields, demonstrating its competence. This leads us to believe that keystroke authentication may also benefit from machine learning.

Research question: What makes the dynamic keystroke utilizing machine learning more accurate than the graphical password?

2 Literature Survey

Numerous studies have been conducted on dynamic keystroke. According to research by Bergadano et al. on dynamic keystroke for user authentication utilizing volunteer self-gathered datasets, just 0.01 percent of impostor pass the authentication. Yu and Cho 8 conducted an early experimental investigation on feature subset selection for keystroke dynamics identification verification in 2003. They found that GA-SVM had a good learning rate and accuracy. In 2007, Revett et al. conducted a study on user authentication, and research into dynamic keystroke authentication was also initiated. Although biometrics, in particular fingerprints, are trustworthy, the author said easily be manipulated.

i. Analysis of the current system:

- ✓ Keystroke dynamics is a system that verifies the system's legitimacy based on the typing of users.
- ✓ A growing area of focus in security is rhythm. Because of this's supremacy
- ✓ the following considerations favor authentication technology above other technological fields:
- ✓ Easy implementation because only data entry is required.
- ✓ No hardware is required.
- ✓ Reasonably priced computation
- ✓ Does not demand a special authorization from the user.

A few statistical models have been suggested for use in keystroke dynamics research. Classifiers like been developed using machine learning approaches. Hybrid models are also considered in the accurate construction of the model. These models have access to a sizable amount of data and are able to identify and establish a specific pattern of 4 typing frequencies. Keystroke dynamics are said to vary depending on a person's neurophysiological behavior in numerous studies. This behavior reflects the unique typing style of the person. To further categorize the concept of a keystroke, MLP and clustering algorithms were employed. As an alternative, a system was created to produce and arrange a user's typing pattern by fusing the traits of applied typing lag time difference. To authenticate certain users, they applied artificial neural networks as classifiers. The average training time was 0.9094 seconds, and the classification rate was 100%. Using the RBFN technique, the comparable pressure and time lag notion was established. 97 percent of the time, an authentication was successful using MLP and a Radial Basis Function. 97.5 percent of the contribution was accurate. used a 17-digit password in an experiment where each person typed the password multiple times over the course of each session, with the error rate being lowered by accounting for elements like finger size and timing frames. In a manner similar to this, use the space between keystrokes while using MLP and principal component analysis methodologies. From this experiment, accuracy rate of 80% was found. Data from Android devices was analyzed using SVM and Naive Bayes classification methods on keystroke datasets without touchscreen features. This strategy outperformed a standard authentication procedure by 10%.

ii. Keystrokes Dynamics:

A user's typing style is recorded by keystroke dynamics, a subtype of behavioral biometrics. The way you type takes into account various factors, such as how long it takes you to enter your login. ID/password, how long we hold down a key for while typing, and how long it takes us to type each key after that. Figure 1 illustrates the fundamental example of data that may be extracted by pressing two keys on a standard keyboard. By assembling all possible digraphs (two-letter combinations) from the login ID and password, one can, for example, build a model of how the user logs in. Along with this static information, One can examine how a person's typing style evolves over time. This learning curve, sometimes referred to as the practice effect, can be quantified and used as a statistic. All information obtained during the authentication process, including any qualities, must be updated over time. In addition to the static direct qualities previously mentioned, secondary or derived qualities should be acquired. Entropy, edit distance, and typing speed are a few of these. The range of qualities that can be used in the classification process is at least expanded by these traits. Furthermore, they potentially provide classification information that is useful but not available in the basic attributes. Next in this work, we go through how main and secondary attributes are used for authentication. procedure. There must be some objective metric by which the precision may be evaluated. in addition to the attributes one collects, of the authentication procedure.

Figure 1 illustrates the concept of a digraph and the various combinations that can be made. retrieved and applied to biometric identification. The word "no" in this situation serves as the digraph's supporting structure.

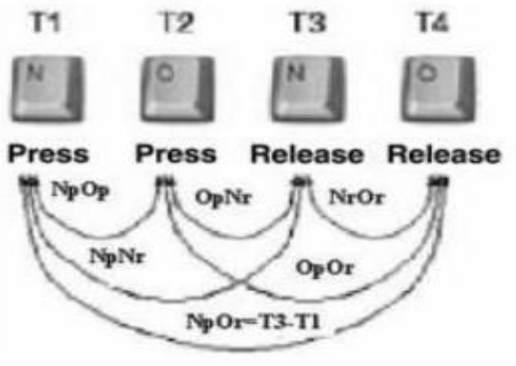


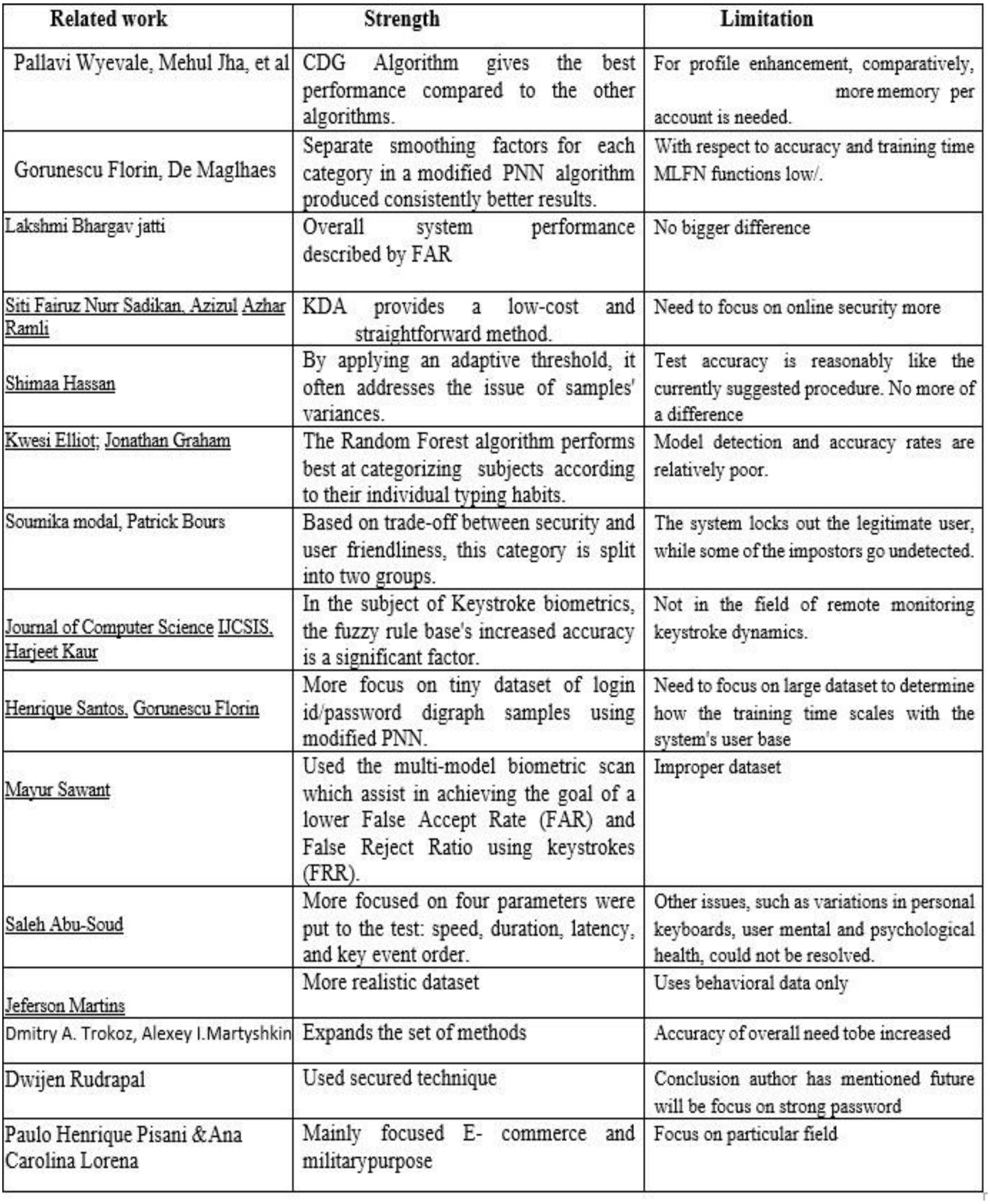
Fig 1: Diagraphs

In the biometrics literature, the two primary objective metrics used to assess the efficacy of the authentication process are the false rejection rate (FRR) and false acceptance rate (FAR) (FAR). False rejection, a type I error, is often measured by the former, and false acceptance, a type II error, by the latter. They are called as the Cross-over Error Rate or Equal Error Rate , respectively, and they provide a measure of how delicate the biometric is at balancing ease of use for the real user while at the same time limiting the imposter access rate. These two measures are traded off by every biometric technology now in use. typically have high FRRs and efficiently detect imposters. The last step entails developing the operational components of the authentication process utilizing the currently known attributes and suitable criteria for calculating the degree of process inaccuracy. The majority of behavioral biometrics employ a two-stage process that consists of an authentication phase and an enrollment step. During enrollment, users must repeatedly submit their login details so that the system can collect a statistically significant sample their typist practices the biometric system is undoubtedly collecting data in the background.

iii. Approach to machine learning:

Over the years, KDA research has extensively utilized the latest machine learning and classification methods. A machine learning approach is used to categorize, find a pattern, and arrive at the right conclusion based on the provided data. Many metrics have been studied based on, including the Manhattan distance, Euclidean distance, and Mahalanobis distance. Support vector machines (SVMs), Bayesian classifiers, KNearest Neighbor (KNN) classifiers, fuzzy logic, and both traditional and complex classifiers can be exemplified by neural networks. Artificial neural networks (ANNs) have been used recently for a number of applications due to their capacity to recognize complex noisy patterns, particularly in keystroke classification issues. Keystroke dynamics has previously used these networks extensively. Neural networks, however, cannot entirely replace well-established methods like statistical regression, time series analysis, and pattern recognition due to their own limitations. Due to this restriction, neural networks struggle to effectively train, which lowers the detection accuracy of their algorithms. While neural networks are amazing at being able to distinguish unseen data points, fuzzy clustering is fantastic at allowing the algorithm to generalize well. These two methods are used in conjunction with one another. In the past, various hybrid strategies have been investigated to address the drawbacks of each distinct approach.

i. Research Niche



Related work	Strength	Limitation
Pallavi Wyevale, Mehul Jha, et al	CDG Algorithm gives the best performance compared to the other algorithms.	For profile enhancement, comparatively, more memory per account is needed.
Gorunescu Florin, De Maglhaes	Separate smoothing factors for each category in a modified PNN algorithm produced consistently better results.	With respect to accuracy and training time MLFN functions low/.
Lakshmi Bhargav jatti	Overall system performance described by FAR	No bigger difference
Siti Fairuz Nurr Sadikan, Azizul Azhar Ramli	KDA provides a low-cost and straightforward method.	Need to focus on online security more
Shimaa Hassan	By applying an adaptive threshold, it often addresses the issue of samples' variances.	Test accuracy is reasonably like the currently suggested procedure. No more of a difference
Kwesi Elliot; Jonathan Graham	The Random Forest algorithm performs best at categorizing subjects according to their individual typing habits.	Model detection and accuracy rates are relatively poor.
Soumika modal, Patrick Bours	Based on trade-off between security and user friendliness, this category is split into two groups.	The system locks out the legitimate user, while some of the impostors go undetected.
Journal of Computer Science IJCSIS, Harjeet Kaur	In the subject of Keystroke biometrics, the fuzzy rule base's increased accuracy is a significant factor.	Not in the field of remote monitoring keystroke dynamics.
Henrique Santos, Gorunescu Florin	More focus on tiny dataset of login id/password digraph samples using modified PNN.	Need to focus on large dataset to determine how the training time scales with the system's user base
Mayur Sawant	Used the multi-model biometric scan which assist in achieving the goal of a lower False Accept Rate (FAR) and False Reject Ratio using keystrokes (FRR).	Improper dataset
Saleh Abu-Soud	More focused on four parameters were put to the test: speed, duration, latency, and key event order.	Other issues, such as variations in personal keyboards, user mental and psychological health, could not be resolved.
Jeferson Martins	More realistic dataset	Uses behavioral data only
Dmitry A. Trokoz, Alexey I.Martyshkin	Expands the set of methods	Accuracy of overall need tobe increased
Dwijen Rudrapal	Used secured technique	Conclusion author has mentioned future will be focus on strong password
Paulo Henrique Pisani &Ana Carolina Lorena	Mainly focused E- commerce and militarypurpose	Focus on particular field

Fig 2: Literature Review Table

3 Research Methodology

Collection of Data: In this section, the various phases of the study will be thoroughly explained, along with the computed keystroke timing data. Data gathering, training, testing, and algorithm comparison are the four primary components of the study. The techniques and types of user data that are collected, the machine learning algorithms that are used, the techniques by which these algorithms are trained and tested, and the metrics by which the algorithms are evaluated are all covered by these four processes..

Dynamic keystroke, first Benchmark Dataset:

The data were collected from 51 typists who entered the password (.tie5Roanl) a total of 400 times. A table with 34 columns has been created with the information.

Each row of data represents the time information for a single password repetition by a single participant. The first column, Subject, acts as the unique identifier for each subject (e.g., s002 or s057). Even though 51 people participated in the data collection, the identifiers do not start with s001 and end with s051 since each subject received a unique ID for a range of keystroke studies, and not all participants participated in every experiment.

The remaining 31 columns show the password's time-related information. The column name contains an encoding of the timing information type. The column names in the H.key table give the recognized key's hold time, or the interval between pressing and releasing the key. For the identified digraph, keydown-keydown times are indicated by column names of the format DD.key1.key2 (i.e., the interval between key1 and key2 pressing). The column names for the recognized digraph's keyup-keydown times have the format UD.key1.key2 (i.e., the interval between key1 being released and key2 being pushed). Remember that UD times can be negative and that H times and UD times combine to become DD times.

3.1 Training Phase:

Training Phase: After gathering the keyboard input from the test subjects, the computer needs to be taught. 80% of the data were used to train the computer algorithms, while 20% of the data were used to test their accuracy. Training datasets are the collections of information that each machine learning algorithm uses to learn how to classify a subject's keyboard rhythm. The team has developed four different machine learning algorithms that each follow the process in a different way.

3.2 Model Builder:

We have considered a wide range of learning strategies for our trials. In this part, we present some teaching strategies.

- ✓ Decision Tree
- ✓ SVC Model
- ✓ k-nearest neighbors
- ✓ Gaussian Model
- ✓ Random classifier model

4 Design Specification:

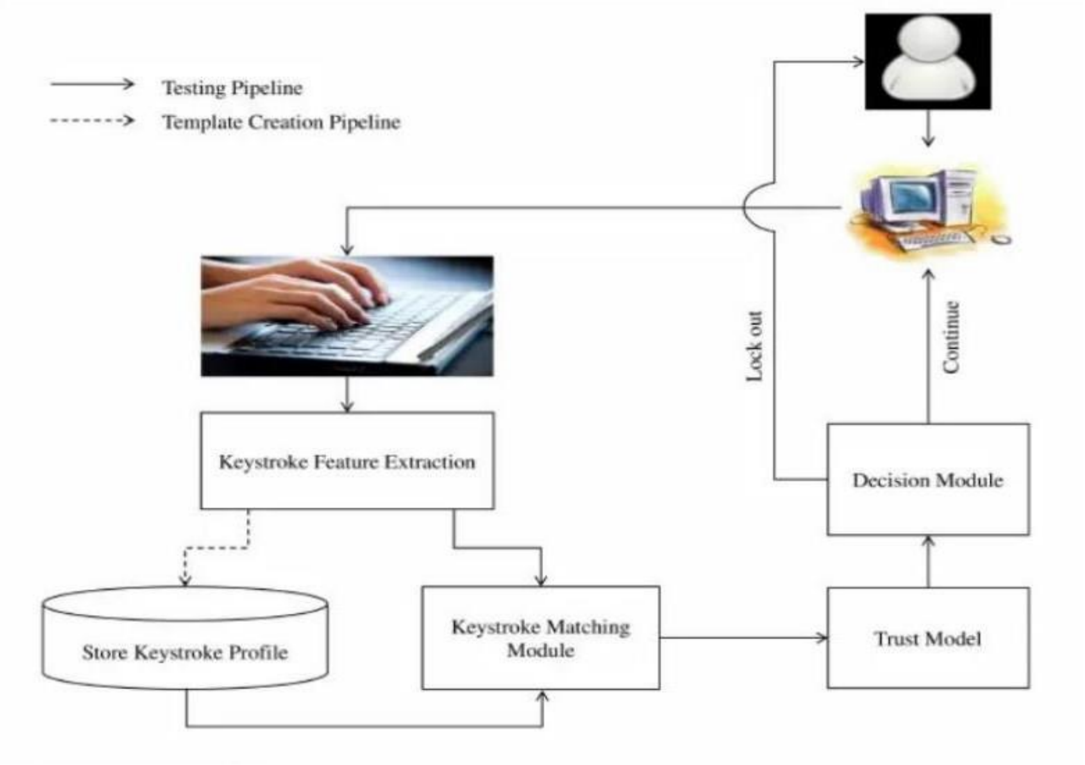


Fig 3: block diagram of Dynamic keystrokes

The block diagram above gives a description of the two phases that make up the system. The classifier models are created from the training data in the training phase and stored in a database for usage in the testing phase (showed by the dotted arrow). Each sincere user has received their own training.

models and classifier characteristics During the testing phase, test data that were separated from training data will be contrasted. We will compare the outcomes and calculate the classifier score (probability) for each sample of the test data based on the action taken using the models and training features that are kept in the database. This score will subsequently be used to modify the trust value in the trust model. Based on the trust value, the decision module decides whether to lock out the user or keep them logged in. This decision is made based on the lockout threshold and the existing trust value (T lockout). In order to calculate the change in trust as indicated by parameters A, B, C, and D, which are based on the type of action and were optimized using a genetic algorithm with the aim of maximizing the cost function, the system computes the score for each action made by the current user.

5 Implementation

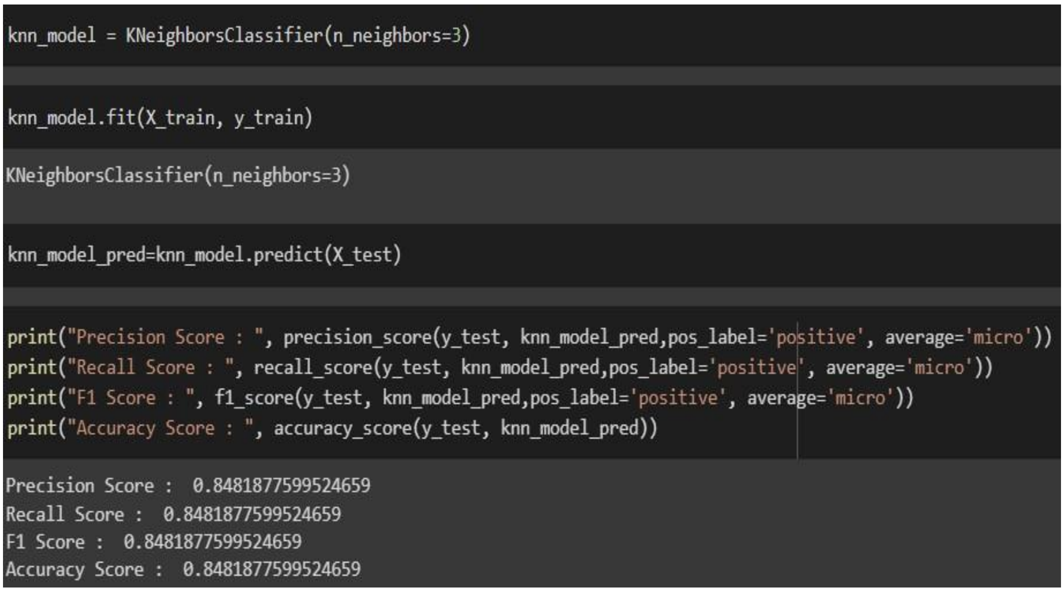
The fixed-text keystroke dynamics dataset that was used in this work is first described in this section. Then, we quickly go over the different learning methods we used on this dataset.

5.1 K-Nearest Neighbors:

An apparently clear method known as the k-nearest neighbors (k-NN) algorithm allows us to categorize a sample based on the k nearby samples in the training set. Despite being easy, k-NN frequently execute well, while overfitting is an issue, particularly for low values of k. Both k-NN and random forest are neighborhood-based algorithms, however they produce significantly distinct neighborhood structures. By calculating the distance between a query point and all of its neighbors in the training set, the basic idea behind it is to identify the k closest neighbors. During the training phase, the KNN classifier specifically builds two models using the training vectors. During the testing phase, the separations between the query location and each of its neighbors are calculated. Then, using a majority voting method, a new point is given to the most prevalent class among its K nearest neighbors. In other words, KNN awards categories new points based on a similarity criteria (e.g., distance functions). Any metric unit can be used to express the distance. The Euclidean distance, however, is the most popular choice. The Euclidean distance is given by the following equation..

$$\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

The number of nearest neighbors is a key factor in KNN accuracy (k). Large numbers for K would produce some noisy data, making it harder to differentiate across class boundaries. As a result, the accumulation of all these noisy data would have a negative effect on the classifier accuracy. However, picking small values for K can potentially have a negative impact on the classifier's accuracy because the prediction would then be based on insufficient data, which might result in over-fitting. For K values in this study, we choose the simplest approach, which is to choose K so that it is smaller than the square root of the number of training examples.



```
knn_model = KNeighborsClassifier(n_neighbors=3)
```

```
knn_model.fit(X_train, y_train)
```

```
KNeighborsClassifier(n_neighbors=3)
```

```
knn_model_pred=knn_model.predict(X_test)
```

```
print("Precision Score : ", precision_score(y_test, knn_model_pred,pos_label='positive', average='micro'))
print("Recall Score : ", recall_score(y_test, knn_model_pred,pos_label='positive', average='micro'))
print("F1 Score : ", f1_score(y_test, knn_model_pred,pos_label='positive', average='micro'))
print("Accuracy Score : ", accuracy_score(y_test, knn_model_pred))
```

```
Precision Score :  0.8481877599524659
Recall Score :  0.8481877599524659
F1 Score :  0.8481877599524659
Accuracy Score :  0.8481877599524659
```

Fig 4: Code snippet of K-Neighbors

5.2 Random Forest classifier

Decision tree-based machine learning technique called a random forest is frequently very successful for classification and regression problems. This method uses a lot of different decision trees, each of which is based on a small subset of the available features and a small subset of the training samples. Each decision tree's subgroups are chosen using replacement. The random forest categorization is chosen by a majority vote or by averaging the component decision trees.

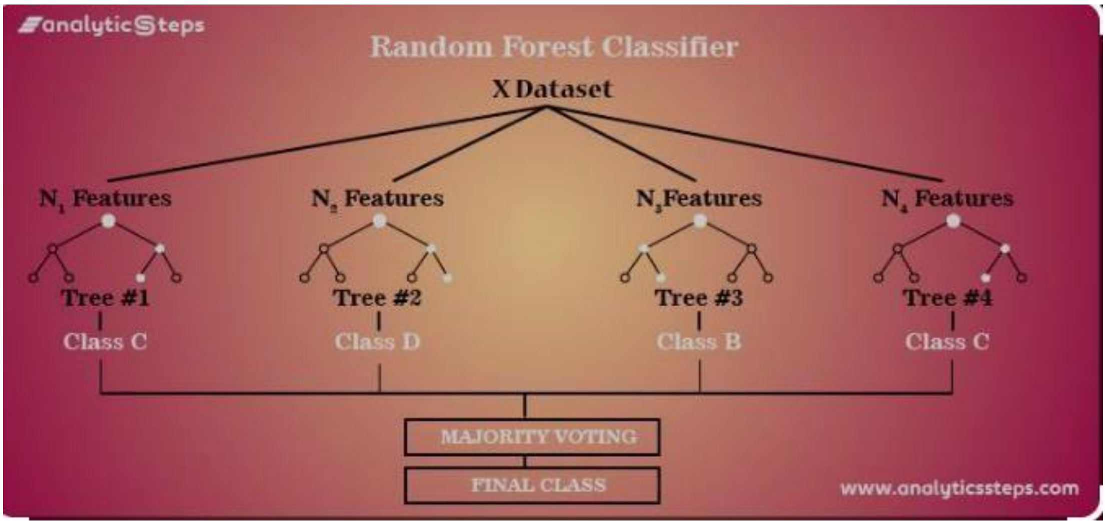
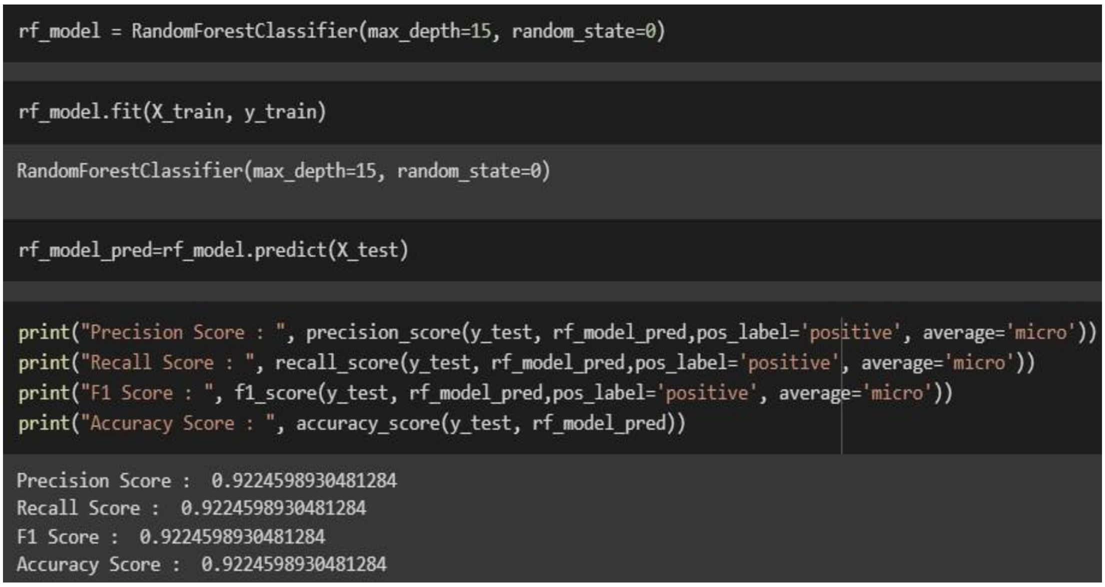


Fig 5: Random Forest Classifier



```
rf_model = RandomForestClassifier(max_depth=15, random_state=0)
```

```
rf_model.fit(X_train, y_train)
```

```
RandomForestClassifier(max_depth=15, random_state=0)
```

```
rf_model_pred=rf_model.predict(X_test)
```

```
print("Precision Score : ", precision_score(y_test, rf_model_pred,pos_label='positive', average='micro'))
print("Recall Score : ", recall_score(y_test, rf_model_pred,pos_label='positive', average='micro'))
print("F1 Score : ", f1_score(y_test, rf_model_pred,pos_label='positive', average='micro'))
print("Accuracy Score : ", accuracy_score(y_test, rf_model_pred))
```

```
Precision Score :  0.9224598930481284
Recall Score :  0.9224598930481284
F1 Score :  0.9224598930481284
Accuracy Score :  0.9224598930481284
```

Fig 6: code snippet of Random Forest Classifier

5.3 Decision tree

A set of options is represented graphically in the shape of a tree by decision trees, which are decision assistance tools. The various options are found at the branches' ends (the "leaves" of the tree), and they are arrived at based on the choices made at each stage. They have the benefit of being quick and simple to read.

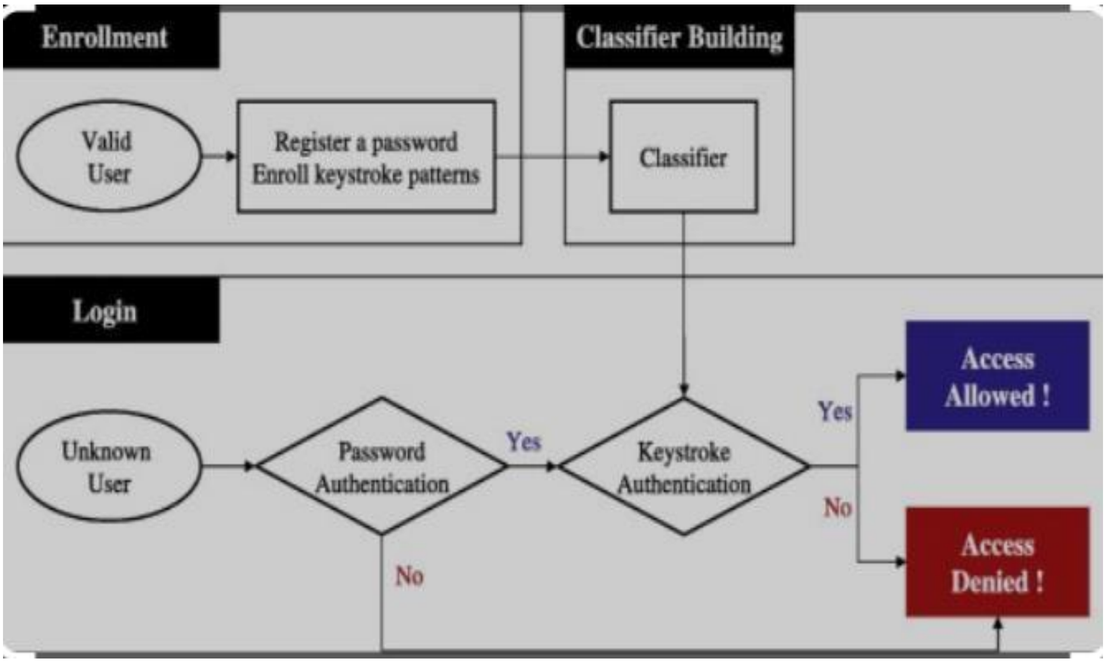
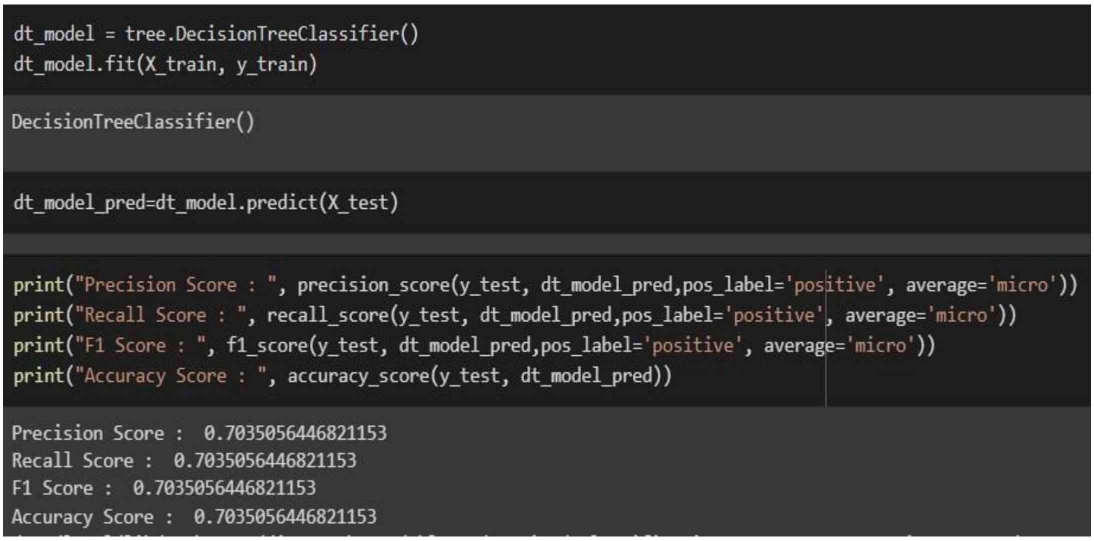


Fig 7: Decision tree



```
dt_model = tree.DecisionTreeClassifier()
dt_model.fit(X_train, y_train)
```

```
DecisionTreeClassifier()
```

```
dt_model_pred=dt_model.predict(X_test)
```

```
print("Precision Score : ", precision_score(y_test, dt_model_pred,pos_label='positive', average='micro'))
print("Recall Score : ", recall_score(y_test, dt_model_pred,pos_label='positive', average='micro'))
print("F1 Score : ", f1_score(y_test, dt_model_pred,pos_label='positive', average='micro'))
print("Accuracy Score : ", accuracy_score(y_test, dt_model_pred))
```

```
Precision Score :  0.7035056446821153
Recall Score :  0.7035056446821153
F1 Score :  0.7035056446821153
Accuracy Score :  0.7035056446821153
```

Fig 8: code snippet of decision tree classifier

5.4 Gaussian Model

A parametric density function called the Gaussian Mixture Model (GMM) is created by adding the weighted sum of Gaussian components. The GMM produces a matrix of covariance containing the variances of the components as well as the covariances between them, as well as a vector of mean values corresponding to each component. While GMM may represent the data in additional dimensions by utilizing a discrete number of Gaussian functions, each with its own mean and covariance matrix, to enable better modeling, pure Gaussian can only fit the data by a single peak (mean) and an elliptic shape (variance).

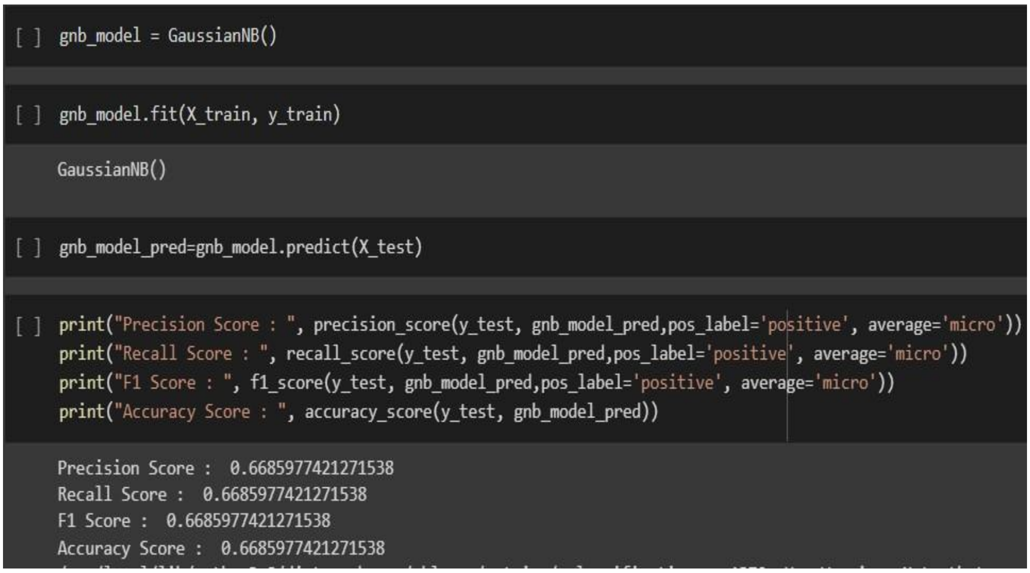
The parameter set for the GMM is made up of the covariance matrix I mean vector I and component weights (wi).

$$\lambda = \{w_i,\ \mu_i^{\rightarrow},\ \Sigma_i\}, i = 1, \ldots, M$$

Using the iterative expectation-maximization (EM) approach, the parameters are computed. Every iteration updates the parameter lambda (), increasing the possibility that the parameters will be fine-tuned and the distribution of the training dataset will be fit.

The weighted linear combination of M pure Gaussian distributions for the x vector is the mixing density:

$$p(\vec{x}|\lambda) = \sum_{\mathrm{i}=1}^{M} w_i p_i(\tilde{x})$$



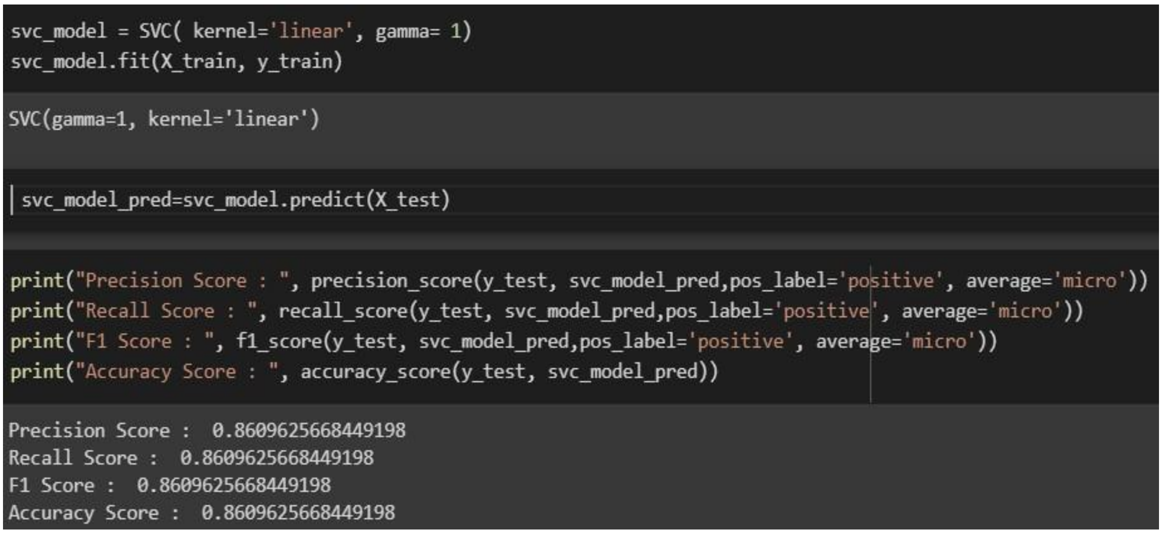
```
[ ] gnb_model = GaussianNB()

[ ] gnb_model.fit(X_train, y_train)

    GaussianNB()

[ ] gnb_model_pred=gnb_model.predict(X_test)

[ ] print("Precision Score : ", precision_score(y_test, gnb_model_pred,pos_label='positive', average='micro'))
    print("Recall Score : ", recall_score(y_test, gnb_model_pred,pos_label='positive', average='micro'))
    print("F1 Score : ", f1_score(y_test, gnb_model_pred,pos_label='positive', average='micro'))
    print("Accuracy Score : ", accuracy_score(y_test, gnb_model_pred))

    Precision Score :  0.6685977421271538
    Recall Score :  0.6685977421271538
    F1 Score :  0.6685977421271538
    Accuracy Score :  0.6685977421271538
```

Fig 9: Code snippet of Gaussian Model

5.5 SVC Model

SVMs have proven to be a very effective method for classifying data. The idea of a separating hyper-plane serves as the foundation for the discriminative classifier known as SVM. In other words, the goal of SVM is to discover an ideal separating hyper-plane that separates the training data set by a maximum margin given a set of labeled training data. When given two classes of training data, (xi, yi) I = 1,, n, x belongs to Rd, y belongs to +1, 1), the SVM generates an ideal hyper-plane that divides fresh data observations into (+ 1) and (1) categories. SVM classification is essentially a two-step procedure. The SVM classifier creates two models using the training vectors in the first phase, also known as the training phase. The technique searches for a linear separator between two classes after projecting two classes of training data into a higher dimensional space. As a result, all vectors with the label "1" are situated on one side of the hyperplane, while those with the label "1" are situated on the opposite side. The testing phase involves mapping the testing vectors onto the same high-dimensional space and predicting their category based on which side of the hyper-plane they fall in the second phase.

SVMs can successfully carry out a non-linear classification in addition to the conventional linear classification by making use of what are referred to as "kernel" functions.



```
svc_model = SVC( kernel='linear', gamma= 1)
svc_model.fit(X_train, y_train)
```

```
SVC(gamma=1, kernel='linear')
```

```
svc_model_pred=svc_model.predict(X_test)
```

```
print("Precision Score : ", precision_score(y_test, svc_model_pred,pos_label='positive', average='micro'))
print("Recall Score : ", recall_score(y_test, svc_model_pred,pos_label='positive', average='micro'))
print("F1 Score : ", f1_score(y_test, svc_model_pred,pos_label='positive', average='micro'))
print("Accuracy Score : ", accuracy_score(y_test, svc_model_pred))
```

```
Precision Score :  0.8609625668449198
Recall Score :  0.8609625668449198
F1 Score :  0.8609625668449198
Accuracy Score :  0.8609625668449198
```

Fig 10: Code snippet of SVM model

6 Evaluation

The outcomes of research on the dataset are presented in this section.
We examine our findings and offer some analysis.
As previously indicated, the data in the dataset is set up as a table with 31 columns, indicating the data gathered for one password period. For instance, H.period, the hold time for the "." key, is listed in one column. The amount of time the key was depressed is known as the hold time. Another illustration is the column DD. period.t, which represents the amount of time between pressing the "." and "t" keys. The total number of rows in the table is 31, and each row represents the time data for a only single recursive of the password by a participant.

6.1 Performance Assessment

Accuracy Precision, and Recall are the three performance evaluation indicators we employed to assess our outcomes.
Accuracy: is calculated as the total number of classifications divided by the proportion of samples that were correctly categorized. Accuracy is calculated more formally as

$$\textbf{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

Precision: is the ratio of correctly and erroneously categorized characteristics in a class to the number of correctly and wrongly identified positive samples in that class. The equation is provided by

$$\textbf{Precision} = \frac{TP}{TP + FP}$$

Recall: is the proportion of accurately anticipated positive observations to all observations that were in fact positive. The equation is provided by

$$\textbf{Recall} = \frac{TP}{TP + FN}$$

6.2 Experiment 1: *K*-Nearest Neighbors

This presumption must be true enough for the KNN algorithm to be effective. KNN illustrates the concept of similarity. As the score for accuracy is 84%.

Model	**Precision score**	**Recall score**	**F1 Score**	**Accuracy Score**
K- Nearest Neighbors	84%	84%	84%	84%

6.3 Experiment 2: Random Forest Classifier

The training subset is used to produce a random decision tree, known as the Random Forest. This method combines the votes from randomly constructed decision trees to determine the object class. Below table shows the scores of the random forest. The accuracy score of the random forest is 92% so this model classifies between authorized user and not authorized.
This model is more fit for the analysis.

Model	**Precision score**	**Recall score**	**F1 Score**	**Accuracy Score**
Random Forest	92%	92%	92%	92%

6.4 Experiment 3: SVM Model

To make sure the classification system can generalize effectively to new data, we divided the data into train and test (70–30 split). algorithm defined earlier provides a 86% accuracy. The model is functioning good.

Model	**Precision score**	**Recall score**	**F1 Score**	**Accuracy Score**
SVM Model	86%	86%	86%	86%

6.5 Experiment 4: Gaussian Model

MMs have a reputation for being excellent at clustering a feature space.
The score is 66% which is not as expected, the model which is not suitable for the analysis.

Model	Precision score	Recall score	F1 Score	Accuracy Score
Gaussian	66%	68%	66%	67%

6.6 Experiment 5: Decision tree

A decision tree chooses the most dominant and predictive variable from a set of independent variables, then subcategorizes it into branches.as the accuracy score of this algorithm is 70% which is good for the analysis.

Model	Precision score	Recall score	F1 Score	Accuracy Score
Decision tree	70%	70%	71%	70%

6.7 Discussion

In Figure 10, we list our experimental findings for the dataset.
The outcome demonstrates that random forest with data augmentation, also known as random forest-augment, obtains the highest accuracy at 92% among the models we have taken into consideration.
MLP performs almost as well as Random Forest with data augmentation, which yields the greatest results.

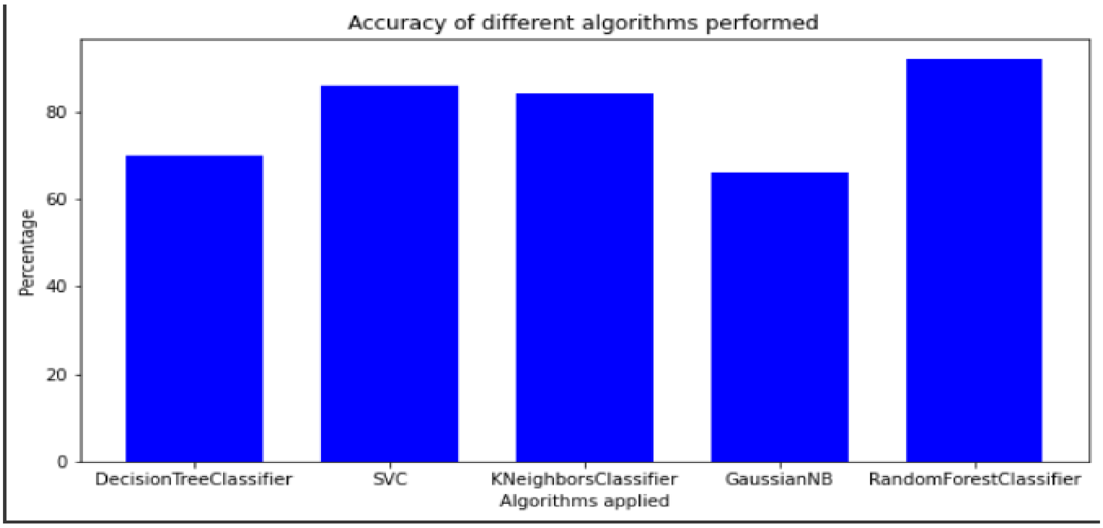


Fig 11: Accuracy of different algorithms performed

The Random Forest Classifier with data augmentation may be a superior option if high training efficiency is desired. Above graph shows a comparison between our best result and earlier work. Our greatest accuracy of 92% is a slight improvement over earlier research in this area.

7 Conclusion and future work

In this research, we test and assess a wide range of machine learning methods for fixed-text typing-based biometric authentication. We discovered that MLP performed nearly as well as Random Forest with data augmentation. Our findings were better than those of earlier studies using the same dataset.

There are numerous options for potential future employment. For instance, model fusion and optimization would be intriguing. To test if these methods may enhance our model, we could think about self-supervised and contrastive learning strategies for model optimization.

Another example of potential future study is the evaluation of the resilience of various approaches using the POPQORN algorithm. With Popcorn, you can watch how external disruptions affect the model and use that information to gauge its resilience.

Limitations of this Research:

A single dataset is utilized for all the models, and only one model's visualization is shown. Due to time and resource constraints, we were unable to create our own IoT dataset by simulating a network. Moreover, open-source simulators did not generally to enable. But since it was impossible to extract network information, this study employed the Dynamic keystroke, First Benchmark dataset.

Future work:

The user's emotions when inputting passwords for the training and testing procedure will be added as psychological characteristics in future study to increase accuracy because the user's emotional states can be deduced from their input behavioral style. Additionally, we want to enhance (and increase) the keyboard dynamics datasets that are necessary to advance this authentication problem significantly.

8 Acknowledge

I want to thank my supervisor, Michael Pantridge, I sincerely appreciate all the motivation and inspiration he has provided me during my projects.

9 References

- ✓ Kevin S. Killourhy; Roy A. Maxion (no date) *Analytical Hierarchy and neural network based landslide risk assessment ...* Available at: https://ieeexplore.ieee.org/abstract/document/9865763 (Accessed: December 1, 2022)
- ✓ Alaa Darabseh; Akbar Siami Namin (2015) *IEEE Xplore*, *Effective User Authentications Using Keystroke Dynamics Based on Feature Selections*. Available at: https://ieeexplore.ieee.org/Xplore/home.jsp (Accessed: December 13, 2022).
- ✓ panelYohanMulionoaEnvelopeHanryHambDionDarmawanb, A.links open overlay *et al.* (2018) *Keystroke Dynamic Classification Using Machine Learning for password authorization*, *Procedia Computer Science*. Elsevier. Available at: https://www.sciencedirect.com/science/article/pii/S1877050918314996 (Accessed: December 13, 2022).

- ✓ Kenneth Revett (no date) *Download a file - computer, Google Chrome Help*. Google. Available at: https://support.google.com/chrome/answer/95759?hl=en&co=GENIE.Platform (Accessed: December 11, 2022).
- ✓ Mohamad El-Abed and Baptiste Hemery (2011) *Unconstrained keystroke dynamics authentication with shared secret, Unconstrained Keystroke Dynamics Authentication with Shared Secret*. Available at: https://hal.archives-ouvertes.fr/hal-00628554/document (Accessed: December 13, 2022).
- ✓ Nick Bartlow (2005) *Username and password verification through keystroke dynamics - core, Username and password Username and password verification through keystroke dynamics* . Available at: https://core.ac.uk/download/pdf/230452244.pdf (Accessed: December 11, 2022).
- ✓ Tawab Attaie (2019) *Dynamic keystroke for authentication with machine learning algorithms, Dynamic Keystroke for Authentication with Machine Learning Algorithms*. Available at: https://www.researchgate.net/publication/336683999_Dynamic_Keystroke_for_Authentication_with_Machine_Learning_Algorithms (Accessed: December 13, 2022).
- ✓ Benoît Martin Azanguezet Quimatio (2022) *Home - archive Ouverte Hal, HAL Open Science*. Available at: https://hal.archives-ouvertes.fr/?lang=en (Accessed: December 13, 2022).
- ✓ Shambhu Upadhyaya (2015) *IEEE Xplore, User Authentification through Keystroke dynamics based on ensemble learning approach*. Available at: https://ieeexplore.ieee.org/Xplore/home.jsp (Accessed: December 13, 2022).
- ✓ Lakshmi Bhargav Jetti (2021) *(PDF) Keystroke Dynamics based user authentication using deep , User Authentication Based on the Keystroke Dynamics using Multi-Layer Perceptron*. Available at: https://www.researchgate.net/publication/327402801_Keystroke_Dynamics_Based_User_Authentication_using_Deep_Multilayer_Perceptron (Accessed: December 13, 2022).
- ✓ Han-Chih Chang (2022) ***https://link.springer.com/chapter/10.1007/978-3-030-97087-1_13**, Machine Learning and Deep Learning for Fixed-Text Keystroke Dynamics*. Available at: https://ieeexplore.ieee.org/abstract/document/8267667 (Accessed: December 13, 2022).
- ✓ Kołakowska, A. and Landowska, A. (2021) *Keystroke dynamics patterns while writing positive and negative opinions*, *Sensors (Basel, Switzerland)*. U.S. National Library of Medicine. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8434638/ (Accessed: December 13, 2022).
- ✓ Piugie, Y.B.W. *et al.* (2022) *Keystroke dynamics-based user authentication using Deep Learning Neural Networks*, *HAL Open Science*. Available at: https://hal.archives-ouvertes.fr/hal-03716818 (Accessed: December 13, 2022).
- ✓ Martins, J. (2018) *Keystroke dynamics authentication: A survey of free-text methods*, *Academia.edu*. Available at: https://www.academia.edu/36250614/Keystroke_Dynamics_Authentication_A_Survey_of_Free_text_Methods?email_work_card=view-paper (Accessed: December 13, 2022).
- ✓ Journal, I.J.S.R.D. (2016) *Keystroke Dynamics Authentication with Project Management System*, *Academia.edu*. Available at: https://www.academia.edu/20053762/Keystroke_Dynamics_Authentication_with_Project_Management_System?email_work_card=view-paper (Accessed: December 13, 2022).
- ✓ Asdkile, A. (2017) *Keystroke Dynamics as a biometric for authentication*, *Future Generation Computer Systems*. Available at:

https://www.academia.edu/32037526/Keystroke_dynamics_as_a_biometric_for_authentication?email_work_card=view-paper (Accessed: December 14, 2022).

- ✓ Organization, S.D.I.W.C. (2014) *A hybrid keystroke authentication dynamics, Academia.edu*. Available at: https://www.academia.edu/5538955/A_Hybrid_Keystroke_Authentication_Dynamics?email_work_card=view-paper (Accessed: December 14, 2022).
- ✓ Santos, H. and Florin, G. (2016) *Authenticating computer access based on keystroke dynamics using a probabilistic neural network*, *Academia.edu*. Available at: https://www.academia.edu/7068915/AUTHENTICATING_COMPUTER_ACCESS_BASED_ON_KEYSTROKE_DYNAMICS_USING_A_PROBABILISTIC_NEURAL_NETWORK?email_work_card=view-paper (Accessed: December 14, 2022).
- ✓ Shinde, P. and IJCSIS, J.of C.S. (2016) *Survey of keystroke dynamics as a biometric for static authentication*, *Academia.edu*. Available at: https://www.academia.edu/25107338/Survey_of_Keystroke_Dynamics_as_a_Biometric_for_Static_Authentication?email_work_card=view-paper (Accessed: December 11, 2022).