HAWASSA UNIVERSITY

INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

FACULTY OF INFORMATICS

KEYSTROKE DYNAMICS BASED MULTI-FACTOR AUTHENTICATION USING MACHINE LEARNING

BY

MESERET DEGEFI HAILEGEORGIS

HAWASSA, ETHIOPIA

NOVEMBER, 2024

KEYSTROKE DYNAMICS BASED MULTI-FACTOR AUTHENTICATION USING MACHINE LEARNING

BY
MESERET DEGEFI

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF HAWASSA UNIVERSITY INSTITUTE OF TECHNOLOGY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

HAWASSA, ETHIOPIA

Program: M.Sc. Computer Science

Major Advisor: Asrat Mulatu (PhD)

Co-Advisor: Zemenu Haile (MSc)

APPROVAL SHEET-I

This is to certify that the thesis entitled, “Keystroke Dynamics based Multi-Factor Authentication using Machine Learning” submitted in partial fulfillment of the requirements for the degree of Master's with specialization in Computer Science, the Graduate Program of the Faculty of Informatics, and has been carried out by Meseret Degefi. Therefore, we recommend that the student has fulfilled the requirements and hence hereby can submit the thesis to the department.

Asrat Mulatu (Ph.D.)		05/12/2024
Name of major advisor	Signature	Date
Name of co-advisor	Signature	Date

EXAMINERS' APPROVAL SHEET

SCHOOL OF GRADUATE STUDIES

HAWASSA UNIVERSITY EXAMINERS' APPROVAL SHEET

(Submission Sheet-2)

We, the undersigned, members of the Board of Examiners of the final open defense by Meseret Degefi have read and evaluated his/her thesis entitled "Keystroke Dynamics based Multi-Factor Authentication using Machine Learning", and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirements for the degree.

Asrat Mulatu (Ph.D)		05/12/2024
Name of Major Advisor	Signature	Date
______________________	________________	____________
Name of Internal Examiner-I	Signature	Date
______________________	________________	____________
Name of Internal Examiner-II	Signature	Date
Tesfay Gidey (Ph.D)		Dec 03/2024
Name of External examiner	Signature	Date
______________________	________________	____________
SGS Approval	Signature	Date

Final approval and acceptance of the thesis is contingent upon the submission of the final copy of the thesis to the School of Graduate Studies (SGS) through the Department/School Graduate Committee (DGC/SGC) of the candidate's department.

Stamp of SGS Date: ____________

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisor Dr. Asrat Mulatu for his perpetual support, patience, and inspiration during my Master's program in Hawassa University. Secondly, I would like to express my sincere appreciation to my Co-advisor Zemenu Haile for guiding me throughout my thesis work. Meanwhile I would like to express my special thanks to my friends for helping me during the past two years. I humbly extend my thanks to Hawassa University, Institute of Technology and all concerned people who helped with me in this regard. My appreciation extends to all my instructors during the MSc program for providing the essential courses that were fundamental in shaping the mindset and knowledge necessary for this work.

Finally, I would like to express my greatest appreciation to my family for all unconditional love, support, patience, encouragement, and kindness they gave me during my whole life.

I am sincerely thankful to God for granting me the strength, wisdom, and time to accomplish this thesis.

DECLARATION

I affirm that this thesis is solely my creation and has not been submitted for any degree at any other academic institution. I have duly acknowledged and referenced all sources utilized in this thesis project.

Name: Meseret Degefi H/Georgis

Date: _________________

Signature: _______________

Name of Major Advisor: Asrat Mulatu (Ph.D)

Date: 05/12/2024

Signature: _______________

List of Acronyms and Abbreviations

AI	Artificial Inelegancy
CA	Continuous Authentication
CMU	Carnegie Mellon University
CNNs	Convolutional Neural Network
DBN	Deep Belief Network
EER	Equal Error Rate
EXRF	Extremely Random Forest
FAR	False Acceptance Rate
FMR	False Match Rate
FNMR	False Non-Match Rate
FRR	False Rejection Rate
GRU	Gated recurrent unit
ITAD	Instance-Based Tail Area Density
KSD	Keystroke Dynamics
MFA	Multi Factor Authentication
ML	Machine Learning
MLP	Multi-layer perceptions
PNN	Probabilistic Neural Network
RNNs	Recurrent Neural Networks
SA	Static Authentication
SVM	Support Vector Machine
XGBoost	Extreme gradient boosting

List of Figures

List of Tables

ABESTRACT

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 behavioural 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 research we used the CMU fixed benchmark data set of 20400 sizes which is used for keystroke dynamics. The data set collects 51 users' keystroke dynamics information where each user typed the same password. .tie5Roanl 400 times over 8 sessions and there are 50 repetitions in each session. We tested four different machine learning algorithms: Random Forest, Support Vector Machines, Multi-Layer Perceptron and Extra Trees, to determine which algorism is most effective on accuracy. We also tested these four algorithms with respect to Accuracy, Precision, Recall and F1 score evaluation matrix to compare the performance. The random forest classifier scores extremely high accuracy (99.19%) and with these final results, we can determine what method of machine learning is most effective at accurately authenticating users.

Keywords: *Behavioural Biometry, Keystroke Dynamics, Multi-Factor Authentication, Machine Learning*

Table of Contents

CHAPTER ONE
INTRODUCTION

1.1. Background

The most important components of digital systems are authentication methods. Businesses and organizations are always searching for appropriate security solutions to address identity theft and data leaks. In addition, the recent pandemic events have highlighted how important it is to secure and authenticate users when they access business and data online. Network security threats are more sophisticated and destructive than ever in the big data era. One of the most important lines of security against network threats is identity authentication. An increasing number of researchers are focusing on it. Traditional password-based authentication is still the most widely used method of identity authentication in the majority of network systems. Replay attacks, brute force cracking, and forward security are only a few of the security concerns and attacks that it will face. Biometric-based recognition techniques, such face and fingerprint recognition, have been widely proposed at the same time. Regretfully, by printing images and creating fingerprint stickers, these technologies will result in higher equipment prices and the emergence of numerous false users with phony physiological characteristics, deceiving users' access to resources. There were numerous problems with the security of the current system, which only used one-time authentication to determine a user's identity. Researchers suggested continuous authentication methods based on user behaviour to address this issue. Keystroke dynamics is a technology that is utilized extensively. This technology can greatly increase system security without the need for extra devices by detecting anomalous online visitor situations.

By eliminating their flaws and associated worries, behavioral biometrics builds upon the advantages of conventional biometrics systems. To achieve this, it stops depending on bodily parts or activity logs and instead analyzes distinct micro-patterns in movement, which are one-of-a-kind, hard to copy or fake, and only useful for verifying an individual's identity or biography. Users can be authenticated with this developing technology by showing patterns in their activity.

Iinstead of identifying bodily features (such as fingerprints or irises), possessions (such as phones or key fobs), or knowledge (such as passwords or biographical facts), it finds distinct, individual regularities in the way that people move and type. Behavioral biometrics solutions authenticate continuously by monitoring a user's ongoing interactions with their computer in real time, in contrast to traditional authentication methods that only authenticate when access is requested.

Forsen et al. [1]first investigated in 1977 whether users could be distinguished by the way they type their names, many different techniques and uses for keystroke dynamics have been proposed. A number of literatures also conducted an extensive survey of the keystroke-dynamics literature. Not all of that research is relevant to the use considered in this work, namely, analysing password-typing times. For instance, Gaines et al [2] experimented whether typists could be identified by analysing keystroke times as they transcribed long passages of text. Techniques for analysing passages are different from those for analysing passwords, and the same evaluation data cannot always be used to assess both. Further, even among studies of password-typing times, not all of the extant techniques can be evaluated using the same procedure. One class of techniques is anomaly detection. The typing samples of a single, genuine user are used to build (or train) a model of the user’s typing behaviour. When a new typing sample is presented, the detector tests the sample’s similarity to the model, and outputs an anomaly score. In contrast, another class of techniques is multi-class classification. The typing samples of multiple users are used to find decision boundaries that can be used to distinguish each user from the others. Since anomaly detectors train on a single user's data, while multi-class classifiers train on multiple users' data, these two techniques require different evaluation procedures. According to [6], ‘‘keystroke dynamics is not what you type, but how you type.’’ Most previous work on typing biometrics can be divided into either classification based on a fixed-text or authentication based on free-text. For fixed text, the text used to model the typing behaviour of a user and to authenticate the user is the same. This approach is usually applied to short text sequences, such as passwords. The classification can be made with the timing between each character in the same phrase along with multiple attempts. Moreover, with the methodology of combining password along with username, researchers can further strengthen the model. A comprehensive evaluation related to fixed-text data is mentioned. As for free text, the text used to model the typing behaviour of a user and to authenticate the user is not necessarily the same. This approach is usually applied to long text sequences, and can be viewed as a continuous form of authentication or IDS. In the early years, keystroke dynamics had been studied with many different distance-based methods such as the Euclidean distance, the Mahalanobis distance, and the Manhattan distance. Nowadays, lots of machine learning techniques had also been researched such as Support Vector Machine (SVM), RNN and so on. These techniques will be discussed in more details in the following subsections.

Recently Keystroke biometric authentication systems are a convenient and user-friendly authentication option. Though it shows great potential, its real-world application so far has been limited. Keystroke biometric authentication systems have several advantages[3].

1. They are cost effective as no additional hardware is required, only the regular keyboard is needed.

2. They are non-intrusive and does not create any extra hassle for the user.

3. The systems can be deployed remotely

Machine learning is a subfield of artificial intelligence that involves learning from data to make predictions. Traditionally, solving a problem consists of writing explicit instructions on how to process input to produce output. Using machine learning enables programs to learn the pattern of data to predict the output of a function, which is extremely useful for complex problems where explicit instructions are difficult or impossible to construct. A machine learning model is the program that learns from the given input data through training, and outputs predictions result. Machine learning Models are also known as classifiers, and the input values used to train and test models are called features. Figure 1.1 illustrates a high-level overview of this process.

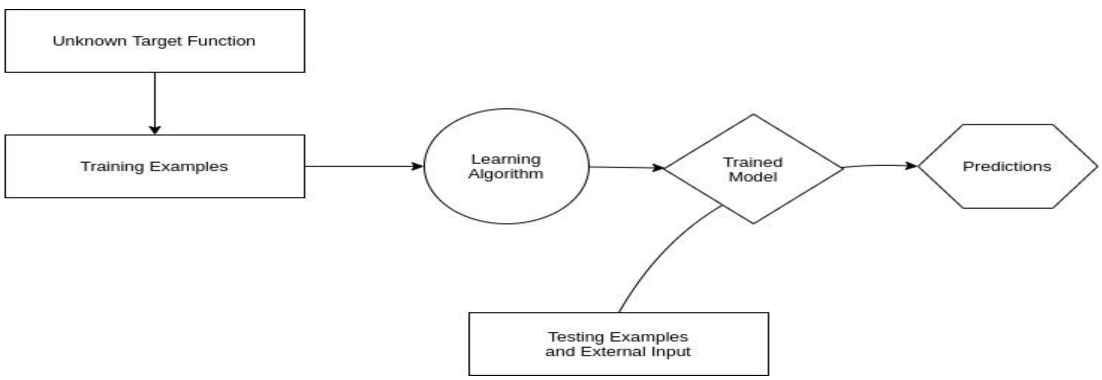


Figure 1. 1 Basic overview of creating a machine learning mode[1].

1.2 Behavioural Biometrics

Behavioural biometrics are devices that analyse particular behavioural characteristics or actions of an individual. It is often non-intrusive which means the information collected is not perceived by the users. They are unique to each individual on each device. It requires some interaction between the user and the system during the authentication process to reduce invasiveness. These systems are very effective in detecting a threat and can be improved in terms of accuracy over time. Some of the types of behavioural biometric devices are-

- Signature verification scanners

- Voice authentication scanners
- Keystroke and mouse movement scanners

These identifiers cannot be duplicated, and only the authorized person can gain access to the system [4]. If a security breach occurs, the information on who is responsible for it can be easily obtained increasing the accountability of the system.

These systems are very easy and safe to use without the need of end user training. By implementing a biometric system, there is no need for expert administrators as it does not require expensive password management. Comparison of different behavioural biometric authentication technique is illustrated in Table 1.1.

Table 1. 1 Comparison of biometric techniques [4]

Biometric Technology	Accuracy	Ease of Use	Cost	Devices Required	Acceptability
Iris	High	Medium	High	Camera	Low
Retinal	High	Medium	High	Camera	Low
Face	Low	Medium	Medium	Camera	Medium
Fingerprint	High	High	Low	Scanner	High
Voice	Low	Medium	Low	Microphone	Medium
Signature	Medium	High	Low	Optic pen, Touch panel	Medium
Hand geometry	High	High	Low	Scanner	Medium
Palm print	High	High	Low	Scanner	Medium
Thermogram	Medium	Low	High	Test equipment	Low
Keystrokes	High	High	Medium	Touch screen devices	High

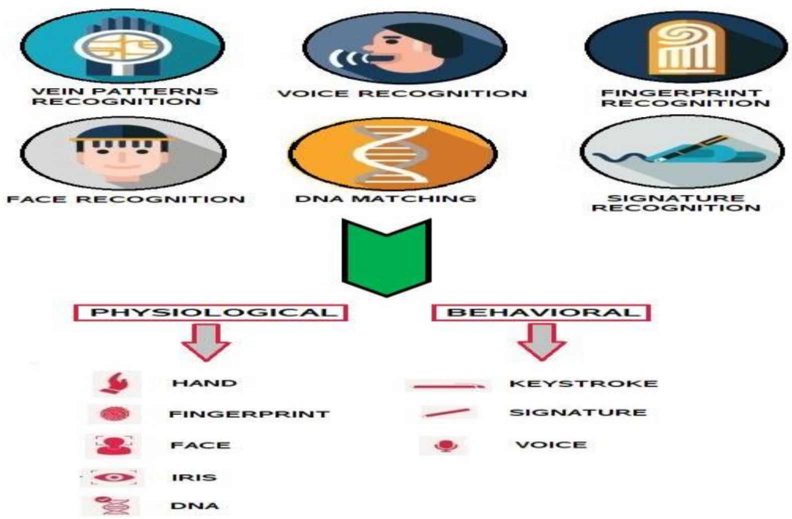


Figure 1. 2 Different type of Biometric used for Authentication [4]

1.2.1. Types of Behavioural Biometrics

Research on various biometrics, including dynamic and static authentication, has increased. With static authentication (SA), a user's identity is retained using immutable biometrics like fingerprints, veins, static passwords, and more. The password data needs to be flexible because it is kept in a physical database. The goal of continuous or dynamic authentication (CA) is to identify a user while they are logged in [4]. The user's behaviour profile, which combines patterns with usage characteristics, is the basis for the analysis and authentication of the real-time session data. Figure 1.3 illustrates types of authentication.

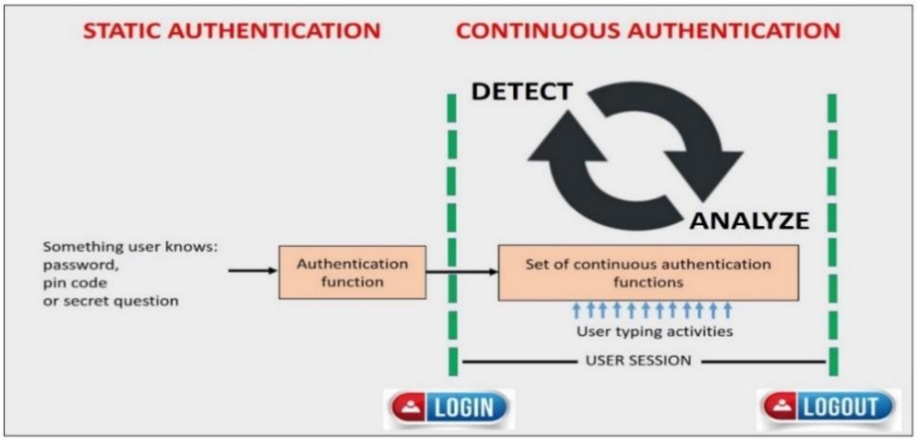


Figure 1.3: Types of behavioural biometrics showing static and continuous authentication [4]

1.2.2 Types of Behavioural Authentication

The matching algorithm's execution in a SA system is measured by false match rate (FMR) and false non-match rate (FNMR). Knowing the possibility that a system error has occurred, i.e., the a genuine user has not been permitted access or that a fake user has been granted access, is crucial for a SA system [5] Using behavioural biometrics for authentication, the user is permitted to begin a session and remains valid throughout. Static confirmation suggests that once the legitimate user has been given access, it is possible for a fraudster to hijack a session and take over a machine. CA is a different type of authentication that gets over the problem mentioned above. A CA biometric system verifies the user's identity throughout the entire session. The user of a traditional CA system shouldn't be aware that their personality is continuously checked. A true CA system makes use of all the user's actions throughout the procedure to determine the user's legitimacy. The user must go back to the static authentication access control mechanism in order to continue functioning when there are doubts about their identity.

Conversely, we identify CA where the current user's action continually verifies the legitimacy of the user. It is important to know if the system detects an impostor user and locks them out, but even more important is knowing how much work the system can put into identifying an impostor. Consequently, the optimal CA system is the one that can identify the impostor user more quickly.

1.2.3. Keystroke Dynamics

The distinctive patterns of rhythm and timing-based elements that are produced when a person types on a touchscreen in computer devices, such as mobile smartphones, are referred to as keystroke dynamics, which is a behavioural biometry [6]. Users are classified by the biometric system according to their physical and behavioural traits using a pattern recognition algorithm [7]. It is a technique for recognizing or authenticating people based on the characters they type on a virtual or real keyboard. Artificial intelligence is used by this kind of system to distinguish between authentic and fraudulent users. The characteristics of the way a user types and uses the system are taken into consideration when approving the user in order to safeguard a group of users from the unauthorized use of their accounts. When a person is typing on a touch screen, the typing dynamics provide precise timing information on when each key was hit and released [8].

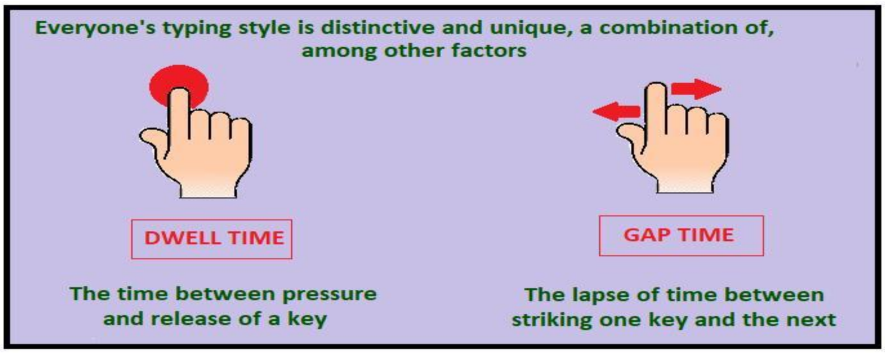


Figure 1. 4 Keystroke dynamics is a combination of dwell time and flight time [4]

Multi-Factor Authentication (MFA) is a security measure that requires users to provide two or more forms of identification in order to access a system or application. It is widely used to protect against unauthorized access, but it is not fool proof. One way to enhance the security of MFA is by using Artificial Intelligence (AI) and Machine Learning (ML). This can help to improve the accuracy and effectiveness of MFA, while also reducing the burden on users.

Authentication using Behavioural Biometrics: One way that AI and ML can be used to improve MFA is through the use of behavioural biometrics. Behavioural biometrics are unique characteristics of an individual's behaviour, such as typing rhythm, mouse movement, and browsing habits. AI and ML can be used to analyse these behavioural biometrics and create a unique profile for each user. This profile can then be used to authenticate users, in addition to traditional forms of authentication such as a password or fingerprint. A strong authentication procedure involves typing a password, which is resistant to cyber-attacks and the way of typing it. This double-layer protection offers more security from all Internet attacks such as brute-force, dictionary attack, and physical shoulder surfing. The brute-force attack involves a hacker to try out all the possible password combinations and is more easily guessable if the attacker knows what we know. The dictionary attack consists of trying the common passwords in the world which work in most of the cases [5] Shoulder surfing involves the attacker looking for the password while one is typing it. One of the best solutions for all these attacks is to combine the biometric pattern with the password. By doing so, even if the attacker looks at or knows our password he or she cannot type the same way as the legitimate user types. It is interesting to know that; every user has a unique way of typing and is subject to different conditions. Thus, by training the model to learn all the biometric movements of the user will lead to building an efficient authentication system. Building a good model requires the best fool proof algorithm to be created.

1.3. Motivation of the Study

In the new era, stronger authentication techniques are required to detect security breaches. The primary focus of this thesis is to develop a system using keystroke dynamic authentication system based on deep machine learning methods to authenticate the user correctly. The other Motivation of this research is that we need a flexible system that can authenticate user and must not be dependent on additional device. This research motivated by:

1. Keystroke dynamics require just a small amount of data to train a recognition system, resulting in a shorter processing time. It is impervious to external environmental conditions, making the authentication process easier.
2. The absence of suitable model that considers static authentication scenario identifying and understanding the characteristic and requirements of each type of static authentication scenario.
3. The need for new feature selection techniques that present user typing behaviour which guarantee that frequently typed features reflect user typing behaviour.
4. The relative absence of research in SA utilizing by biometric sources in general and specifically using keystroke dynamics.
5. The lack of an automated resource requirements and thus is suitable for real time detection.

1.4 Statement of the Problem

Control mechanisms established on the boundary of an information system are an important preliminary step to minimising losses from security breaches. The primary function of such controls is to restrict the use of information systems and resources to authorized users. Password-based systems remain the predominant method of user authentication despite many sophisticated and viable security alternatives that have emerged from research and development. However, literatures show that passwords are often compromised through the poor security and management practices of users. Although, a few organizations do exist that provide secured authentication for users when they login but on the other hand there are still most sites that skip the process of identification leading to major crimes across the globe.

At the forefront of user security is user authentication (Cheng et al., 2015; Kang et al., 2020). User authentication refers to the security methods used to verify a user before granting him/her access to secured content on a system. Several user authentication methods exist such as text (passwords and passphrases), images (clicking on specific areas of an image/images in the correct sequence), pattern (usually mobile devices where a user connects the dots in the right order), audio/voice recognition and physical scans (face, fingerprint and retinal recognition). Although biometrics seem to have proliferated, research conducted by Zimmermann and Gerber [9]explain that passwords are here to stay. To add to this, there is a high risk that if someone discovers how to crack a form of biometrics, it will become permanently unusable as a form of authentication.

The problem identified is that current user authentication methods either increase security and reduce usability or reduce security and increase usability (Kang et al., 2020). Strict system security protocols such as password policies have resulted in high login failures and time spent on password resets (Shay et al., 2014). Login failures/resets are created by users experiencing problems memorising the password. This has resulted in users creating common passwords which are easy to guess (Braunstein, 2015; Sonwalkar, 2020). Some users even write the password down on a piece of paper or capture it digitally in a notepad text file (Braunstein, 2015; Syahrial et al., 2019) users also experiencing problems during typing the password.

This becomes troublesome when special characters and numbers (which are required by most password policies) need to be used (Maoneke and Flowerday, 2019; Shay et al., 2014). To add to this, different keyboard layouts and key sizes also have an impact on typing (Roy et al., 2020; Shay et al., 2014). Also, when a password is mistyped, additional effort is expended as the user needs to repeat the action of inserting the password.

Current user authentication security measures tend to focus on system security with little consideration being given to the impact on usability. However, this has been tolerated by users as these measures protect them from security breaches such as having their passwords hacked. Alomari et al. (2019) Voris et al., (2019). and Choong et al. (2014) state that employees spend a total of 2.25 days within 60 days on password-related activities. This is caused by user memory failure and/or user typing failure.

Any application or data can be easily accessed through a simple password or PIN, if it is known. Methods like Key logger and social engineering can be used to steal the passwords with alphanumeric letters which are not even in the vocabulary. Moreover, the recent work suggests that an attacker needs only eleven attempts to guess most users’ passwords [10]. Thus, there is a vital need for robust security mechanisms that safeguard user’s data. One of the potential approaches is the use of physiological biometric features which are not based on the knowledge of user but based on the user itself. It includes interactive attributes such as retina, fingerprints, hand geometry etc [10].These biometric methods make use of one or more hardware devices in order to recognize a user. Moreover, such authentication methods could be easily compromised and thus security could be easily breached. Another variant to provide authentication is the behavioural biometrics. These methods include keystroke, voice, gait, etc. The behavioural characteristics largely remain unique to a person and do not change drastically.

Moreover, it does not require any additional hardware and hence such system can be cost effective.

This study focused on the user authentication method that will increase both security and accuracy. The solution entails the use of passphrases as the first tier of authentication coupled with a keystroke authentication algorithm as the second tier of authentication.

The keystroke authentication algorithm (also referred to as keystroke dynamics) based on Machin learning is a backend solution that records the keystroke patterns (using time as a metric) of a user when he/she inserts their text-based authentication into the system. This pattern is then used to validate the user's identity. As a result, in order to access the system, the attacker must be able to guess the password and comprehend how it was entered.

Hence the research work in this thesis aims to suggest better security systems that could be analysed using the typing behaviour of individuals using keystroke dynamics. Main emphasis is being put on to detect the typing behaviour of individuals using the concepts of Machine Learning Model such as Support Vector Machine (SVM), Random Forest (RF), MultiLayer Perceptron (MLP) and ExtraTrees (EXRF) to validate the features of users and give them a more secure and efficient system than ever before.

1.5 Research Questions

Biometrics has developed a space of natural choice for the purpose of identity verification, as it is based on detecting the unique characteristics of individuals such as physical and behavioural traits. It is seemed to be an excellent source of identification because all other forms of recognition methods might fail at some point of time either in the absence of physical damage or the data being lost or stolen. On the other hand, physiological characteristics of an individual such as fingerprints tend to become unique for every individual resulting in a precise authentication for access to systems. As a result, the primary goal of the thesis is to generate an environment for user credential safeguard and login security. To evaluate keystroke dynamics as a suitable methodology for user authentication, the following questions must be addressed:

RQ1: What attributes can be extracted from users based on their typing styles or dynamics?

RQ2: What authentication algorithm(s) can be employed to maximize the subtle differences between the typing styles of individuals?

RQ3: Which is the best model for identifying and understanding the characteristics and requirements of static authentication scenarios?

1.6 Objectives

1.6.1 General Objective

The main objective of this research work is to investigate how keystroke dynamics can improve multi factor continuous authentication mechanisms.

1.6.2 Specific Objectives

- Extract optimal attributes/features that are representative of user typing behaviour.
- Develop generic model for identifying and understanding the characteristics and requirements of static authentication scenarios.
- Develop flexible and simple authentication system for the user with minimum implementation cost without depending on special hardware in the authentication process.

1.7. Scope and Limitation of the study

This thesis focuses on Keystroke Dynamic Authentication, class of behavioural Biometrics, which are unique characteristics of an individual's behaviour such as typing rhythm of the user and improve using deep machine learning algorithm. Other biometric authentication such as mouse movement and browsing habits of the user are not considered in this study. The thesis also evaluates the classification model with fixed-Text dataset and not include Free-Text dataset as the highest accuracy achieved with the fixed-text dataset[11] and Fixed Text dataset is also constant during the authentication process [12].

Yet keyboard dynamics is not without drawbacks, the most notable of which the fact that typing patterns can be erratic and inconsistent. Emotion, Cramped muscles, sweaty hands and type of keyboard can potentially alter a person's typing pattern, which could affect verification.

The main limitation of random forest is that a large number of trees can make the algorithm too slow and ineffective for real-time predictions. In general, these algorithms are fast to train, but quite slow to create predictions once they are trained.

Another limitation of Keystroke dynamic also shows sensitivity to the position of the user which implies that if the system was trained when the user was standing up, logging in while in a different position such as sitting down or walking may affect the performance of the verification process, possibly leading to false rejection by the system.

1.8. Significance of the study

The implication of keystroke Dynamics method is that it can be extended beyond typist authentication. It is generic might be applied to any biometric sources such as mouse movements, gait or speech. Additionally, this method can distinguish unknown user sample and decide whether they are from the same user or not. This might help in forensics investigation application where we have two different typing samples and want to decide if they are related to one user or two different users.

1.9. Contributions

In this research work the following contribution are made:

- This thesis advances the knowledge in the area of behavioural biometric authentication schemes.

- The research helps to choose best feature selection and build classification model for the right accuracy measurement to secure the authentication system.
- It is helpful for the most important field of security for forensic investigation and medical field, a number of studies have used keystroke dynamics for disease detection.
- Enhanced keystroke dynamics authentication also used for Online education to authenticate students in its courses. In order to confirm their identity, students are required to type out a specific sentence at the start and end of the course which would then be matched against each other.
- It is also useful for other industries such as banking, financial services, retail, healthcare, education and defence, as they have started adopting keystroke dynamics for multimodal biometrics in a bid to be one step ahead of cybercriminals.

1.10. Research Methodology

This section of the research paper is devoted to materials and methods provides a detailed explanation of the materials used in conducting the study and the experimental procedures followed. This section is crucial because it enables other researchers to replicate and validate the study's findings. Additionally, the materials and methods section should include information about the data selection and analysis process, Literature review, specific measurements taken and any methods used for analysis, software programs used should also be described, including their version and relevant settings.

1.10.1. Dataset selection

Our proposed approach is evaluated on Carnegie Mellon University (CMU) keystroke benchmark fixed dataset and available for research (A Scientific Understanding of Keystroke Dynamics). The reason that we selected this dataset is that, it has higher accuracy compared to free text[11]. 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[13].

1.10.2. Literature Review

Through the thesis, we covered all recent developments of keystroke dynamics research and their applications. We collected Relevant literatures on machine learning techniques from 2020 to present on keystroke dynamics from Google Scholar, ACM Digital Library, IEEE Xplore and Science Direct. We also categorized them based on their contribution and uniqueness. Previous surveys on keystroke dynamics could not cover the recent scenarios. Others focused on specific areas like keystroke dynamics on touch screen and mobile devices, keystroke authentication algorithms, keystroke datasets etc.

1.10.3. Machine Learning Model

Baseline models: To the more commonly used models, will refer to them as "baseline" models as this will serve as a base comparison against more advanced approaches that we will test. There are dozens of models that can be considered to be "baseline" and therefore we will test the ones which are commonly occurring in other literature, such as, Random Forest, Support Vector Machine, Extra Trees and Multilayer Perceptron.

Ensemble learning: Ensemble learning consists of combining the use of multiple machine learning models at the same time, thus potentially minimizing the negatives of each different model. Ensemble models have been shown to often outperform single model approaches and while single algorithms such as random forest use some of these techniques, the use of ensemble models in keystroke Dynamics prediction remains an unexplored opportunity.

1.10.4. Keystroke Dynamics Performance Metrics

There are a number of performance metrics used in keystroke dynamics and each is used to present some specific information about the biometric system. This section describes four metrics and when they are preferably used.

Accuracy: It is the ratio of correctly classified points (prediction) to the total number of predictions. Its value ranges between 0 and 1.

$$Accuracy = (TP + FP)/(TP + FN + FP + TN)$$

Precision Recall: Both precision and recall are crucial for information retrieval, where positive class mattered the most as compared to negative. While searching something on the web, the model does not care about something irrelevant and not retrieved (this is the true F

Precision: Out of all the positive predicted, what percentage is truly positive.

$$Precision = TP/(TP + FP)$$

The precision value lies between 0 and 1.

Recall: Out of the total positive, what percentage are predicted positive. It is the same as TPR (true positive rate).

$$Recall = TP/(TP + FN)$$

F1 Score: It is the harmonic mean of precision and recall. It takes both false positive and false negatives into account. Therefore, it performs well on an imbalanced dataset.

$$F1\ Score = \frac{2}{\frac{1}{Precision}+\frac{1}{Recall}} = 2 * (Precision * Recall)/(Precision + Recall)$$

Precision refers to the confidence with which a positive class is predicted as positive, while recall measures how well the model identifies the number of positive class instances from the dataset. Note that the positive class is the class of interest. Empirically speaking, precision and recall are best understood with the help of a confusion matrix which consists of four key terms:

- True Positive (TP): Number of correctly identified positive class instances
- False Positive (FP): Number of negative class instances wrongly identified as positive class instances
- True Negative (TN): Number of correctly identified negative class instances
- False Negative (FN): Number of positive class instances wrongly identified as negative class instances.

1.10.5. Implementation Tools

Python Programming Language: Python is a versatile, simple-to-learn, and potent programming language. Python is a general-purpose, high-level, interpreter programming language. Initially published in 1991, and created by Guido van Rossum. It enables a more efficient, integrated, and rapid system. It offers straightforward but powerful object-oriented programming techniques along with effective high-level data structures. Python is a great language for scripting and quick application development across a wide range of platforms because of its beautiful syntax, dynamic typing, and interpreted nature.

Y can download the large standard library and the Python interpreter for free. New functions and data types written in C, C++, or other languages that can be called from C can be added to the Python interpreter with ease. Python is a good language to use as an extension for apps that may be customized.

Anaconda: Anaconda is a free, cross-platform, open-source distribution of Python and R programming languages tailored for scientific computing needs like data science, machine learning, large-scale data processing, and predictive analytics. Its primary goal is to streamline package management and deployment processes. The package management system Conda oversees the versions of packages, analyzing the existing environment along with any specified version constraints.

Sicikit-learn: Scikit-learn, a Python module, seamlessly integrates traditional machine learning algorithms within the scientific Python ecosystem, which includes numpy, scipy, and matplotlib. Its objective is to furnish straightforward and effective solutions to learning challenges that are accessible to all and can be applied across different scenarios.

Category Encoders: Through a variety of techniques, it transforms categorical variables into numerical values. The current version of scikit-learn provides similar functionalities for ordinal, one-hot, and hashing encoding. Additionally, it boasts compatibility with sklearn, data frames, and column configuration to handle diverse types of input seamlessly.

Matplotlib: Matplotlib, a Python library for 2D plotting, finds applications in various Python environments such as scripts, IPython shells, Jupyter Notebook, Spyder, and web application servers. It facilitates the creation of plots, histograms, power spectra, bar charts, scatterplots, and more.

Testing Environment: This thesis is implemented using the following hardware and software setup.

Hardware Specification:

System: 12th Gen Intel(R) Core(TM) i7-1255U 1.70 GHz

Hard disk: 500GB

RAM: 16GB

Software specification:

Operating system: Windows 10 Pro,64-bit operating system, x64-based processor

Coding Language: Python 3.11

Tools: Jupyter, Notebook, Anaconda, Spyder.

1.11. Organization of the Thesis

The rest of this Thesis is organized as follows:

Chapter 2: Review of Related Works this chapter surveys e related works for user authentication using keystroke dynamics and machine learning,

Chapter 3: Proposed Approaches This chapter proposed generic model for Keystroke Dynamic authentication based on Machine Learning, describe and analyses in depth the dataset that is used in the research.

Chapter 4: Result and Discussions This chapter analyses and compare different Machine Learning Models.

Chapter 5: Conclusion and future Work Conclusion and directions for future research are presented in this chapter.

CHAPTER TWO

REVIEW OF LITRETURE AND RELATED WORKS

2.1. Introduction

Digital security is critical in today's world because both businesses and users store sensitive information online. Everyone interacts with applications, services, and data that are stored on the internet using online accounts. A breach, or misuse, of this online information could have serious real-world consequences, such as financial theft, business disruption, and loss of privacy.

While passwords protect digital assets, they are simply not enough. Expert cybercriminals try to actively find passwords. By discovering one password, access can potentially be gained to multiple accounts for which you might have reused the password. Keystroke dynamic behavioural biometric authentication is a growing area of research today, especially with the use of machine learning algorithms. This area was first explored in 1977 to investigate whether users could be distinguished based on the way they typed their name (George Forsen, 1977). Since then, an extensive amount of research in the area has been carried out.

2.1. Systematic Literature Review

A systematic literature review, also known as a systematic review, is a type of secondary research that employs a precise methodology to locate, assess, and interpret all previously published research that is relevant to a particular question in a way that is objective and repeatable (Kitchenham 2004).

Reason for adopting Systematic Literature Review: Systematic literature reviews are conducted using a predetermined search strategy that must enable evaluation of the search's completeness. There many justifications for conducting a thorough literature study. Kitchenham (2004) the following are the most common reasons:

- To review the evidence that already exists regarding a specific phenomenon;
- To identify any gaps in the current research in order to suggest areas for further investigation;
- To offer a structure for planning upcoming studies on the subject.

According to the advantages of a systematic review described by Kitchenham [14], systematic reviews can be an efficient process to allow a researcher to get the information about the effects of a phenomenon across a wide range of settings and empirical methods, where it is less likely the results are biased.

The process this literature review follows are the guidelines which have been set out by Okoli & Schabram [15]. These guidelines recommend an eight-step literature review process that is academically thorough, complete and reproducible.

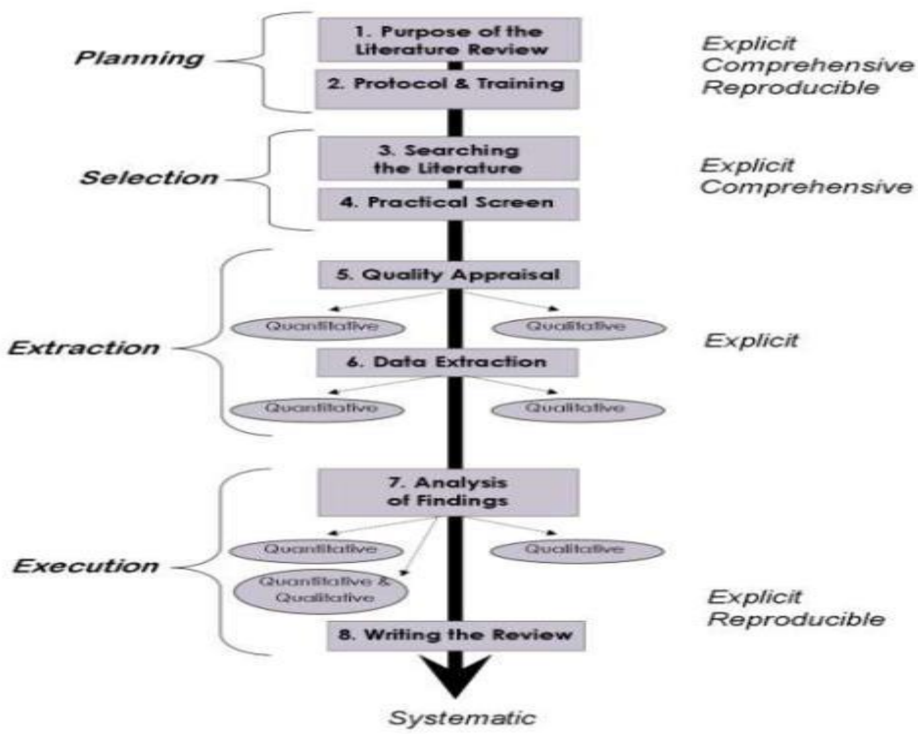


Figure 2. 1 A systematic guide to literature review development *[15]*

Table 2. 1 Steps for a systematic literature review *[15]*

Step	Title
1	Purpose of the literature review
2	Protocol and Training
3	Searching for the Literature
4	Practical Screen
5	Quality Appraisal
6	Data extraction
7	Synthesis of studies
8	Writing the review

The application of the systematic review has the goal of studying the state of the art in keystroke dynamics in order to identify:

- Advantages and disadvantages of using keystroke dynamics in Authentication;
- Extracted features;
- Classification algorithms applied;
- Performance measures commonly adopted;
- Benchmarking datasets, which are useful for conducting comparative experiments in the area.

There are three main stages to applying the systematic review:

- Planning,
- Conduction and
- Presentation of results.

In the first phase, a review protocol is defined, in which research questions are specified along with search strategies. After that, in the second phase, the review protocol is applied and the information is extracted from the returned references. References used for the extraction of information are called primary studies, while the review is a secondary study. Finally, the third phase defines the way to present the results and the final report is done[15]. The items comprehended in each of the three phases are:

Planning: Identification of the review need: a systematic review has the goal of summarizing all information regarding a specific topic. However, before starting a systematic review, the need of this review has to be checked. This checking, for instance, should verify the existence of previously published systematic reviews that deal with the topic under investigation and whether the protocol of these reviews meets the requirements of the research.

Table 2. 2 Number of returned Reference

Database	No of Reference
ACM Digital Library	14,100
IEEE Xplore	13,800
Google Scholar	22,400
Science Direct	33,400
Total	**83,700**

Database Number of references Total 83,700

Specification of the research questions*:* This is considered to be the most important part of the systematic review, as these questions will guide all the following steps, as the search for primary studies, extraction and analysis of information. In view of the need of the systematic review, we defined a research question and some respective sub-questions to meet the established goals:

Development of the review protocol: This step defines strategies to be used for the search, selection and evaluation of the references. In addition, the information to be extracted from each of the selected references is also defined. After defining the research question, we enumerated a list of terms related to papers that could answer it. The basic search expression is the conjunction of each term in the list using the logical connective [16].

(Keystroke) and (Dynamics) And (Authentication) And (Deep) And (Machine) And (Learning)

This search expression was applied in several data-bases that included references in the computing area. As each database has differences in its syntax for search expression, the basic search expression presented here was adapted to each database. The following databases were considered in this work:

- ACM Digital Library (http://dl.acm.org/)
- IEEE Xplore (http://ieeexplore.ieee.org/)
- Google Scholar (https://scholar.google.com/
- Science Direct (http://www.sciencedirect.com/)

Reference search: in order to prevent bias, look for the most references that can address the study topic. Compared to typical reviews, the systematic review search is conducted with greater rigor and pre-definition of search terms and databases.

In order to choose the literature for my study, we had to look for studies that would fit into the major body of my own research and weed out those that didn't fit the bill. When searching for any publications, conferences, or academic papers about Keystroke Dynamic Based Multi-Authentication using Machine Learning, a broad approach was used at first. A comprehensive exploration was conducted inside the library's database to identify pertinent publications, journals, and international conferences. A substantial number of publications and journals were compiled for more thorough review using key-word Keystroke Dynamic searches. Lastly, each paper was searched for citations in other works using Google Scholar, and these newly discovered articles were also examined to determine their applicability to this study. The search proceeded to the "practical screen" stage once it was evident that no fresh articles had been found.

Selection of primary studies: after reference search, the studies that are in fact relevant for the research must be selected, by the use of inclusion/exclusion criteria.

2.2. Inclusion and exclusion criteria

Inclusion criteria

1. Artickles, conference paper, journals and thesis written in English
2. Full-text of #1
3. Studies that focus on Key stroke Dynamic based Multi Factor Authentication using Deep Machine Learning.
4. Studies from the year 2020 up to 2023
5. Papers that are related to the domain of study.

Exclusion Criteria

1. Studies that are not written in English
2. Articles do not have full access
3. Duplicated articles
4. Short papers (E.g. Poster)
5. Old papers that don't satisfy the inclusion criteria set in #4

To make the results more relevant the timescale of the search was narrowed to 2020 –2023. The reason for this is that "Keystroke Dynamics based on Deep Machine Learning" is a relatively a key word and it is intended to filter out unnecessary results. Where the option was available, the results were limited to peer-review journals. The results of narrowing the search follow:

Table 2. 3 Number of selected references

Database	Number of Reference	Selected Literatures
ACM Digital Library	14,100	3
IEEE Xplore	13,800	2

Google Scholar	22,400	7
Science Direct	33,400	8
Total	**83,700**	**20**

Table 2. 4 A summary of keystroke authentication algorithms

Ref.	Published Year	Deep-Model Classifier	Dataset	Performance Metrics			
				Accuracy	FAR	FRR	EER
[3]	2023	-	-	-	-	-	2.94%
[11]	2022	RF (Random Forest)	-	98.8%	-	-	-
[17]	2022	CNN	GREYC-NISLAB	-	-	-	4.89%
[18]	2022	CNN & Bi-LSTM	Buffalo	-	3.03%	3.09%	4.23%
[19]	2023	CNN $ RNN	Aalto mobile	-	-	-	3.84
[20]	2022	ITAD & D-Vector	Bilingual	-	-	-	-
[21]	2022	CNN & GRU	SUNY Buffalo	high	-	-	0.096
[22]	2022	XGBoost & MLP	CMU	93.79%	-	-	-
[23]	2021	Novel	-	-	-	-	Reduced EER 5.32%
[13]	2022	Histogram Gradient Boosting	-	97.96%	-	-	0.01
[24]	2020	MLP	CMU	-	-	-	4.45%
[25]	2020	Feed forward Multilayer NN	CMU	94.7%	-	-	0.049
[26]	2020	CNN-Detect	CMU	-	-	-	0.009
[27]	2023	KNN and K-means++	CMU	87.67%	-	-	-
[28]	2023	DBN	-	95%	-	-	5%
[29]	2023	SVM	-	97.90%	-	-	-
[30]	2020	RF (Random Forest)	Greyc	95.65	-	-	-
[31]	2023	Neural network & MLP	CMU	-	-	-	1.26%
[32]	2023	XGBoost	-	-	0.3%	1.5%	0.9%
[33]	2022	NN	Buffalo	82%	-	-	-

During each of the searches, a number of articles were examined looking at the abstract, conclusion and keywords to see if they were relevant to the research. From the total 74 documents 20 were put forward to the screening stage

- ***Quality evaluation:*** each of the selected references undergo a quality evaluation. This evaluation may be used with diverse aims, like contributing for the inclusion/ exclusion criteria or supporting the summary results, by measuring the importance of each study
- ***Information extraction:*** the information extraction from the references must be done with the support of forms defined during the planning phase of the systematic

review. Of the 20 academic articles which has been found it was necessary then to explicitly examine these to see which ones should be put forward to the next stage. The criteria used to narrow down the search further were based on 3 questions:

TABEL 6: KSD screening criteria

No	Question
1	What attribute can be extracted from users based on their typing styles or dynamism?
2	What authentication algorithm(s) can be employed to maximize the subtle differences between the typing styles of individuals?
3	Which one is a best model for identifying and understanding the characteristics and requirements of continuous authentication scenarios?

Table 2. 5 Selected Primary Study Checklist

S.N.	Ref.	Title	RQ1	RQ2	RQ3
1	[3]	Keystroke Dynamics: Concepts, Techniques, and Applications	✓		
2	[11]	Fixed-Text vs. Free-Text Keystroke Dynamics for User Authentication	✓	✓	✓
3	[17]	Keystroke Dynamics based User Authentication using Deep Learning Neural Networks	✓		

4	[18]	ACBM: attention-based CNN and Bi-LSTM model for continuous identity authentication	✓		
5	[20]	Shared Multi-Keyboard and Bilingual Datasets to Support Keystroke Dynamics Research	✓		
6	[21]	Machine Learning-Based Analysis of Free-Text Keystroke Dynamics	✓		
7	[22]	Machine Learning and Deep Learning for Fixed-Text Keystroke Dynamics	✓		
8	[23]	Modified Distance Metric That Generates Better Performance for The Authentication Algorithm Based On Free-Text Keystroke Dynamics	✓		
9	[13]	Keystroke dynamics-based user authentication using Histogram Gradient Boosting	✓	✓	✓
10	[24]	Keystroke Dynamics Based User Authentication using Deep Multilayer Perceptron	✓		
11	[25]	A Simple Authentication Method with Multilayer Feedforward Neural Network Using Keystroke Dynamics	✓	✓	✓
12	[26]	Keystroke Dynamics Analysis for User Authentication Using a Deep Learning Approach	✓		
13	[27]	Evaluating Learning Algorithms for Keystroke Based User Authentication	✓		
14	[28]	Dynamic Keystroke Technique for a Secure Authentication System based on Deep Belief Nets	✓		
15	[29]	Extraction of Hidden Authentication Factors from Possessive Information	✓	✓	✓
16	[30]	A Comparison of Authentication Methods via Keystroke Dynamics	✓	✓	✓
17	[31]	User authentication through behavioral biometrics using multiclass classification algorithms	✓		
18	[32]	Enhancing User-Level Security: Performance Analysis of Machine Learning Algorithms for Dynamics Keystroke Analysis	✓		
19	[33]	Biometric Identification Based on Keystroke Dynamics	✓		
20	[34]	Continuous authentication using deep neural networks ensemble on keystroke dynamics	✓		

Data synthesis: this step corresponds to summarizing the results attained during the review. This summary may involve qualitative and quantitative aspects. For quantitative aspects, a meta-analysis may also be applied

Many methods used to perform a password authentication using user's biometrics such as fingerprint recognition, retina recognition, voice recognition, etc. However, additional sensors needed to perform most of biometric recognition methods and will be invasive to users caused by additional tools needed to perform a password authentication. Keyboard Dynamics is one of the solutions to perform password authentication without adding any tools which being disruptive to some users. The biometric keystroke dynamic system is relatively unexplored compared to another behavioral authentications discipline. Coupled with the limited number of studies that have been done compared with other biometric systems. Several machine learning researches has been conducted but few of them applying deep learning for solving this problem. This research will be focusing in deep learning using optimizer to beat the previous research which using another machine learning technique.

The Systematic literature research findings summarized based on the research questions as below**:**

Brice et al. [35] explored and proposed a new approach based on the conversion of behavioural biometrics data (time series) into a 3D image to address the problem of user authentication considering the keystroke dynamics modality and the transformation also allowed to train image based on convolutional neural networks. According to the authors the transformation process kept all the characteristics of the behavioural signal an image transformation is applied to keystroke dynamics considered as time series, before applying deep learning architectures. The time series did not receive any filtering operation with this transformation and the method was bijective. They evaluated the performance of the authentication system in terms of Equal Error Rate (EER) on a significant dataset and the authors showed the efficiency of the proposed approach on a multi-instance system. The authors used two important error rates to determine the performance of a biometric authentication system a False Match Rate (FMR) and False

NonMatch Rate (FNMR). The Equal Error Rate (EER) was found when the FMR is equal to the FNMR

FMR is the proportion of a specified set of completed non-mated comparison trials that result in a comparison decision of match.

FNMR is the proportion of completed mated comparison trials that result in a comparison decision of non-match.

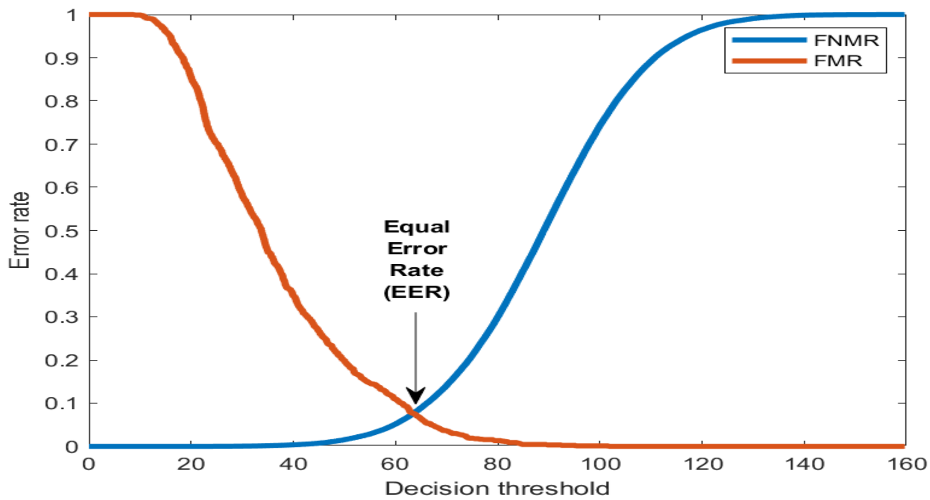


Figure 2. 2 Relationship between FMR, FNMR and EER[17]

The study provided insights into the potential of deep learning methods to improve passphrase user authentication by analysing keystroke dynamics data. It also suggests future research directions, including adding psychological features to enhance accuracy[17].

Other article written by Mao et al. [18] proposed a keystroke dynamic identity authentication model based on deep learning. This model combined convolutional neural network (CNN), bi-directional Long ShortTerm Memory (BI-LSTM), and the attention mechanism. According to the authors it was different from most existing models that only used keystroke time as the feature vector, this model also used keystroke content and keystroke time as the feature vector. First, CNN is used to process feature vectors. Then the normalized vector was input into the bi-LSTM network for training. The model in this paper was tested using Buffalo open data set. The results showed that FRR (False Reject Rate), FAR (False Accept Rate), and EER(Equal Error Rate) were 3.09%, 3.03%, and 4.23%, respectively. The validity and accuracy of the model in continuous identity authentication were proved. The paper suggested that the ACBM model shows promise for practical applications in continuous identity authentication and plans for future work include optimizing the model and testing with more public datasets.

Stragapede et al. [19] focused on improving keystroke biometric systems on the free-text scenario. The authors discussed the feature extraction process and the use of Gaussian range encoding to handle the variability in keystroke dynamics, aiming to minimize system input duration and improve authentication performance, the scenario was characterized as very challenging due to the uncontrolled text conditions, the influence of the user's emotional and physical state, and the in-use application. To overcome these drawbacks, methods based on deep learning such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been proposed in the literature, outperforming traditional machine learning methods. However, these architectures still have aspects that need to be reviewed and improved. To the best of the authors knowledge, this was the first study that proposed keystroke biometric systems based on Transformers. The proposed Transformer architecture had achieved Equal Error Rate (EER) values of 3.84% in the popular Aalto mobile keystroke database using only 5 enrolment sessions, outperforming by a large margin other state-of-the-art approaches in the literature. The authors Suggested future directions for enhancing the Transformer architecture, optimizing training approaches, and investigating the subject information embedded in the feature embeddings for privacy considerations.

Wahab et al.[20] explained that keystroke dynamics has proven to be a promising technique for user authentication including the prevention of identity theft, account takeovers, and transaction fraud. However, the paper discussed the use of keystroke dynamics for user authentication, focusing on the impact of different keyboards and languages on system accuracy setting due to the variable nature of keystroke dynamics, which could have an impact on the accuracy of the system. The authors used discussed two Models: Instance-Based Tail Area Density (ITAD) and D-Vectors. ITAD is an instance-based metric that evaluate keystroke Dynamic bay analysing the tail area under the probability density function of features. D-Vector is a deep learning approach that converts keystroke events in to images and uses a convolutional neural network for feature extraction.

The strength of ITAD is that it provides a context-aware assessment of strengths and weaknesses, allows candidates to highlight positive qualities relevant to the job and encourages measurable results to back up claims. In other hand the benefit of D-Vectors it is Non-traditional approach that captures non-intuive features, reasoning over arbitrary-length keystroke sequences and once it is trained it is computationally inexpensive. The weakness of ITAD is that it is limited by the quality and quantity of data available and may not capture all nuances of keystroke dynamics. D-Vectors weakness is that Data-hungry and requires substantial training data and the performance is highly dependent on the quality of training sample. This research does not explicitly mansion a gap, but it highlights the need for further research on multi-keyboard and multi-language effects on keystroke dynamics.

Chang et al. [21] The authors focused on user authentication and identification in cybersecurity using keystroke dynamics and presented a deep learning model combining a CNN and a GRU to analyse free-text keystroke dynamics. According to the authors explanation the Previous work has shown that user authentication and classification can be achieved based on keystroke dynamics. In this research, they consider the problem of user classification based on keystroke dynamics features collected from free-text. They implemented and analysed a novel a deep learning model that combines a convolutional neural network (CNN) and a gated recurrent unit (GRU). The authors optimized the resulting model and considered several relevant related problems. The model was competitive with the best results obtained in previous comparable research. The strength of this paper was Keystroke dynamics offer no need for special hardware and allow for passive data collection. The model also achieved high accuracy and competitive Equal Error Rate (EER) in biometric systems. Potential performance issues and with injured users or different hardware and the need for model optimization through attention layers and pre-trained models is the weakness of this paper. The gap of this paper is further research is suggested on robustness evaluation and data enhancement.

Chang et al. [22] explained that Keystroke dynamics can be used to analyse the way that users type by measuring various aspects of keyboard input. Previous work has demonstrated the feasibility of user authentication and identification utilizing keystroke dynamics. As the authors presented the research compared various ML and DL models, including k-nearest neighbours (k-NN), random forests, support vector machines (SVM), and neural networks, to optimize user identification based on keystroke pattern, they optimized the resulting models, and they compared their results to those obtained in related research. The authors found that models based on extreme gradient boosting (XGBoost) and multi-layer perceptions (MLP) performed well in their experiments. The authors concluded that their models best outperformed compared to the previous research.

The strength of this paper is that the study provided a comprehensive analysis of different ML and DL techniques, offering a broad comparison of their effectiveness and it also demonstrated the potential of extreme gradient boosting (XGBoost) and Multi-layer perceptions (MLP) in achieving high accuracy in use identification.

The reliance on fixed-text keystroke dynamics may limit the applicability in scenarios where users input free text and the models might require regular updates to accommodate variations in typing pattern due to factors like keyboard changes or user emotions are the weakness of this study.

When we see the gap, the paper does not discuss the implications of its findings on real-world application extensively and even if it acknowledges the limitation of keystroke dynamics as a standalone authentication method but does not explore potential solution in depth.

In 2021 A . Iapa and V. Cretu [23] discussed about an authentication algorithm based on free-text keystroke dynamics, focusing on improving the efficiency of the algorithm by reducing the Equal Error Rate (EER). The authors proposed a modification of the Manhattan distance calculation formula to improve the performance of an authentication algorithm based on free-text keystroke dynamics. The document provided a comprehensive analysis of keystroke dynamics and suggests a novel approach to enhance authentication algorithms in this domain. The strength of this paper is that The proposed model utilizes keystroke dynamics, a behavioural biometric feature that doesn't require additional devices and the modified distance metric shows a significant improvement in the algorithm's performance, reducing the EER value (5.32%). The paper does not explicitly mention weaknesses or gaps, but generally, such systems might face challenges in accuracy when users have varying typing patterns or when used across different devices. The gap identified is the need for a more efficient metric to lower the EER in keystroke dynamics authentication.

Nanglae et al. [13] proposed keystroke dynamics as a behavioural biometric for user authentication, which is cost-effective and doesn't require special hardware. The proposed approach employs Histogram Gradient Boosting as the primary classifier, which has shown to outperform other machine and deep learning methods in accuracy and error rates. The model achieved a high average accuracy of 97.96% and a low average equal error rate (EER) of 0.014, indicating strong performance and robustness against overfitting and noise. A novel pre-processing technique involving quantile transformation significantly improved the model's ability to handle outliers, contributing to its high accuracy. The strength of the model lies in its innovative use of quantile transformation and Histogram Gradient Boosting, leading to high accuracy and low EER. However, the document does not explicitly mention any gaps, but generally, such systems may face challenges in real-world scenarios due to variations in typing patterns over time or under stress. Future work could focus on addressing these potential gaps. Andrean et al. [24] the authors improved user authentication by analysing typing rhythms using keystroke dynamics, which is cost-effective and doesn't require additional hardware. In this document discussed keystroke Dynamics for user authentication, focused on a deep learning model using Multilayer Perception (MLP) for enhanced security. The MLP model was chosen for its ability to learn hierarchical non-linear features and dependencies, which traditional machine learning methods may struggle with. This study aims to propose deep learning model using Multilayer Perceptron (MLP) in keystroke dynamics for user authentication on CMU benchmark dataset. The user typing rhythm from 51 subjects collected based on the static password (.tie5Roanl) typed 400 times over 8 sessions and 50 repetitions per session The MLP model achieved an Equal Error Rate (EER) of 4.45%, outperforming benchmark classifiers. And t showed promise in providing reliable protection by requiring both the password and the rhythm in which it's typed.

The study is limited to one dataset and does not consider external factors that might affect performance. Future research could explore more complex models and datasets, and extend to mobile platforms for broader applicability. The document emphasized the potential of keystroke dynamics as a low-cost biometric authentication method, with the deep learning approach showing promising results in distinguishing between genuine users and impostors. However, it also acknowledges the need for further research to enhance accuracy and address limitations.

Ahmet Melih Gedikli and Mehmet Önder Efe [25] introduced a simple, accurate, and lightweight method for user authentication using keystroke dynamics and a multilayer feedforward neural network with resilient backpropagation model. According to the authors explanation, the study showed the effectiveness of their approach through comparisons with existing methods, which have also used the CMU keystroke dynamics benchmark dataset and achieved an Equal Error Rate (EER) equal to 0.049 for authentication with overall identification accuracy of 94.7% which is quite impressive compared to other techniques. The Strengths of this approach it is software-based, making it easy, cheap, and online. It also benefits from being non-intrusive and does not require additional hardware. The Weaknesses and Gaps of this model might be affected by external factors like different keyboard layouts and internal factors such as emotion or stress, potentially leading to genuine users being incorrectly rejected.

Najwa Altwaijry [26] presented a convolutional neural network, which is called CNN-Detect, to detect unauthorized users that attempt to access resources by their typing patterns. they tested their model on the publicly available CMU keystroke dynamics dataset, after suitable feature engineering. According to the authors the proposed model showed significant improvement over other models in the literature, achieving an average equal error rate (EER) of 0.009, and a zero-miss false acceptance rate (ZM-FAR) of 0.027. The strengths of CNN-Detect include its ability to detect unauthorized users by their typing patterns with high accuracy and its suitability as a component in a user authentication system, The document suggests that while CNN-Detect performs well, there is a potential for misclassification due to the physical and emotional state of an individual affecting the biometric. Additionally, the document acknowledges the need for periodic retraining to capture variations in users' typing behaviour over time.

Xi et al. [27] investigated the comparative performance of supervised and unsupervised learning two algorithms, KNN and K-means++, and explored the impact to the keystroke-based user authentication. This study also used the CMU Keystroke Dynamics Benchmark Dataset, which includes keystroke timing information acquired from 51 typists. Each typist has a subject (id) and 8 sessions, with each session consisting of typing a fixed strong password (.tie5Roanl) 400 times. The authors were selected three keystroke features: dwell time, flight time and press-to-press latency. FAR, FRR and accuracy are used as the performance metrics of KNN while purity and silhouette coefficient are selected as the performance metrics of K-means++. The experimental results show that KNN is more suitable for the analysed scenarios and had a slightly higher accuracy of 74.58% than K-means++. they further proposed a method of reprocessing the dataset based on modifying the outliers when unsupervised algorithm. K-means++: Initially had lower performance, but after reprocessing the dataset to modify outliers, it reached a purity of 0.8767.

The Strengths of KNN it was More suitable for the analysed scenarios with higher stability against outliers. Whereas K-means++ After dataset reprocessing, it showed very good performance, indicating its potential when data is pre-processed effectively. Weaknesses KNN Higher false rejection rate (FRR), meaning legitimate users might be denied access. K-means++: Required significant pre-processing to achieve high purity, indicating sensitivity to noisy data. The paper proposed a novel method for pre-processing noisy datasets to improve the performance of unsupervised learning algorithms like K-means++. It also highlighted the importance of choosing the right model based on the nature of the dataset and the specific requirements of the authentication system.

Aversano et al. [34] presented a study on continuous authentication using deep neural networks ensemble on keystroke dynamics. The study proposed an ensemble learning approach using basic deep neural network classifiers to identify users based on their typing style. The ensemble super-classifier compares different voting techniques and training allocation strategies. The proposed approach achieves an accuracy of up to 0.997 under certain evaluation conditions, indicating a high level of precision in user identification through keystroke dynamics. The use of a large integrated dataset composed of 169,678 different users enhanced the robustness and generalizability of the model. The ensemble method, which combines multiple classifiers, tends to be more accurate than single classifiers. The study acknowledged the need for large and balanced datasets to train deep learning models effectively, which could be a limitation in real-world scenarios. It also mentions that the advantage of using deep learning in keystroke analysis is clear but highlighted the challenge of managing large intra-class variations and noisy biometric data.

The research contributes to the field by designing a unique feature set characterizing user typing events and integrating different keystroke datasets into a large dataset, which could be beneficial for future studies in continuous authentication.

S. Zeid et al [11] The authors studied two distinct datasets to test the prospect of classifying individuals based on attributes derived from their keystroke dynamics: (1) a fixed-text dataset with varying degrees of difficulty, and (2) a free-text dataset that did not impose any constraints on what a user may type on the keyboard. Several classification algorithms were used in the investigation, including K-Nearest Neighbors (KNN), RandomForest (RF), Support Vector Machines (SVM), and BayesNet (BN). When utilizing RF for classification, the fixed-text dataset yielded the best accuracy of 98.8%, whereas the free text dataset yielded the maximum accuracy of 87.58% when using an RF classifier. The document suggested that while fixed-text systems are more accurate, free-text systems are necessary for continuous authentication during a session, not just at login. The gap lies in improving the accuracy of free-text systems to match that of fixed-text systems. Fixed-text systems showed higher accuracy but are limited to initial login verification. Free-text systems offer continuous authentication but with lower accuracy. The study highlights the need for better free-text system accuracy and suggests combining classifiers to improve performance.

Aljahdali et al. [28] the researchers in this study created a dynamic keystroke technique for secure e-assessment using the Deep Belief Networks (DBNs) paradigm. The DBN algorithm is applied to a dataset collected from participants who typed free text using a standard QWERTY keyboard in a neutral state without inducing specific emotions. This allows the proposed system to classify the user's identity based on the features that were extracted from the pressure-time measurements, digraphs (dwell time and flight time), trigraphs, and n-graphs. The writers assessed the model's performance in relation to a certain task. computed parameters such as accuracy, recall, and F1 score by comparing its predictions to a ground truth. The Deep Belief Networks (DBN) model achieved an accuracy of 95% and an error rate of 5% in identifying users and detecting cheating attempts in e-assessment. Finding the things that the model accomplishes effectively is one of this research's strengths. This may encompass its capacity to extrapolate from the training set, its velocity of operation, or its resilience to a range of data inputs. The Weaknesses: Seek out instances when the model is inadequate. Overfitting, underfitting, or subpar performance with unbalanced datasets could be common problems. The authors suggested suggesting topics for improvement based on the gaps and limitations that have been found. This could entail expanding the data set, experimenting with other model structures, or using the model for novel applications. The DBN model is effective in providing secure e-assessment by leveraging multiple features for robust user authentication. The document mentions challenges in implementing the DBN model on free text data, requiring a large amount of data and significant training time.

Nanglae et al. [30]studied three major approaches on keystroke dynamics: distance-based, statistical-based and machine learning based approaches, which are often used to solve the problem. Nevertheless, according to the authors, in the literature there were several works which results were obtained from different comparison methodologies; this represents a great problem for future researchers who seek to improve or advance with prior works. Furthermore, by using proprietary databases, researchers do not provide a good overview of the overall performance of their methods, but rather an overview in a specific case: that represented by their database. In this investigation, they proposed to evaluate the performance of the most representative classifiers in two of the three most common approaches used in keystroke dynamics using the public Greyc dataset. The experimental results, reveal that machine-learning based approaches outperformed the distance-based techniques. Moreover, the Random Forest classifier, provided encouraging results of 95.65%. The paper suggested further research with more techniques and datasets to improve the real-world applicability of these models for user authentication.

Shadman et al. [3] discussed keystroke dynamics as a behavioural biometric for user authentication, highlighting its non-intrusiveness and cost-effectiveness since it requires no additional hardware beyond a keyboard authentication operates by creating a template for each user based on their typing pattern during an enrolment period. Once enrolled, the test sample from the user is contrasted with the representation of that same user and a matching score will be calculated. Matching scores are calculated based on the timing features of keystrokes. The document details performance metrics used in keystroke dynamics, like False Accept Rate (FAR), False Reject Rate (FRR), and Equal Error Rate (EER a 2.94%), which are crucial for evaluating the effectiveness of biometric systems. It identified gaps in previous research, such as outdated surveys and a lack of comprehensive coverage across different devices and languages, providing insights for future research directions in keystroke dynamics.

Lantz et al. [31] in this study keyboard and mouse behaviour have been considered. Previous research promised for this authentication method. The research however is scarce, old and often not comprehensive. This study focused on two available data sets, the CMU keystroke dynamics dataset and the ReMouse data set. The data was used together with a comprehensive set of multi-class supervised classification machine learning algorithms from the scikit-learn library for Python. By performing hyperparameter optimization, two optimal algorithms with modified hyperparameters were found that improved results compared with previous research. For keystroke dynamics a classifier based on a neural network, multi-layer perceptron, achieved an Equal Error Rate (EER) of 1.26%. For mouse dynamics, a decision tree classifier achieved an EER of 0.43%. The findings indicate that the produced biometric classifiers can be used in an authentication model and importantly to strengthen existing authentication models such as password-based login as a safe alternative to traditional Multi-Factor Authentication (MFA). The research provided insights into the viability of using machine learning algorithms for behavioural biometrics in cybersecurity, particularly for strengthening existing authentication models. It also suggests combining keystroke and mouse dynamics for enhanced security.

Srkrec, Jntuk [32] This paper presented a comprehensive performance analysis of machine learning algorithms for dynamic keystroke analysis, aimed at enhancing user-level security. The focus was on developing a robust method that ensures tight security while maintaining high accuracy. Previous research efforts have proposed various security mechanisms, techniques, and algorithms; however, their efficiency had been found lacking. The proposed approach leverages machine learning algorithms, specifically the KNN and XGB models, to analyse dynamic keystroke patterns. By extracted and analysed important parameters related to keystroke dynamics, the algorithms identified fraudulent login attempts. The performance of both algorithms is evaluated and compared, with a particular emphasis on accuracy and efficiency. The results of the analysis demonstrate that the XGB model (False acceptance rate (FAR) of 0.3%, False rejection rate (FRR) of 1.5%, and equal error rate (EER) of 0.9%,) outperforms the KNN algorithm in terms of security enhancement and user satisfaction. The XGB model leverages the dataset effectively, utilizing key parameters to make accurate predictions and classify keystroke patterns. In contrast, the KNN algorithm falls short in achieving comparable levels of accuracy and efficiency. By employing the superior XGB model, organizations can enhance user-level security, prevent password theft, and safeguard sensitive data. The XGB model emerged as a powerful tool, surpassing the limitations of the KNN algorithm and provided a more robust and accurate solution. The XGB model's strength lies in its superior performance in capturing essential parameters for keystroke dynamics, while the KNN algorithm's weakness is its lower accuracy in prediction. The paper suggests further research to address these gaps and provide deeper insights into keystroke dynamics as a biometric technique.

Kasprowski et al. [33] studied how neural network architecture and its hyperparameters affect the results of biometric identification based on keystroke dynamics. The publicly available dataset of keystrokes was used, and the models with different parameters were trained using this data. Various neural network layers—convolutional, recurrent, and dense—in different configurations were employed together with pooling and dropout layers. The results were compared with the state-of-the-art model using the same dataset. The results varied, with the best-achieved accuracy equal to 82% for the identification. The model showed promising results, with Rank-5 accuracies of 98.7% for 20 classes and 90.7% for 60 classes, indicating. a high potential for correct classification. Its performance decreased with a larger number of classes, and highlighted a need for further optimization and research.

Nanglae et al. [29], [36] suggested that the use of distinctive and consistent elements over an individual's lifespan may be employed to develop an authentication classification model. This model would be based on prevalent personal behavioural biometrics and could be readily implemented in security authentication systems. The biological biometrics acquired from an individual's typing abilities during data entry include their name, surname, email, and phone number. The study focused on extracting hidden authentication factors from possessive information, emphasizing the importance of using consistent elements over an individual's lifespan for authentication. The document mentioned several biometric methods and their accuracy rates, such as keystroke dynamics with 97.90% accuracy with the proposed SVM (Support Vector Machine) model. The paper analysed the strengths and weaknesses of different biometric methods, noting that while they are effective at detecting imposters, they have limitations such as being affected by emotions or physical changes. It also suggests the need for further research to optimize authentication factors for individual users.1

2.3. Summary of Related Works and Research Gaps

Thirteen studies evaluated the performance of the keystroke Dynamics authentication system in terms of Equal Error Rate from these the lowest EER is 0.009 with CNN-Detect classifier.

Eight studies evaluated the performance of the Keystroke Dynamics authentication system in terms of accuracy with the highest accuracy of 98.8%, with Random Forest classifier.

Most of the reviewed articles of the keystroke dynamics based authentication explained that if a user has an injured hand or is simply distracted or overly emotional, used different devices and languages their typing patterns might not be consistent with the patterns used for training and proposed the need for periodic retraining to capture variations in users' typing behaviour over time.

Some researchers also provided insights into the viability of using machine learning algorithms for behavioural biometrics in cybersecurity, particularly for strengthening existing authentication models and suggested combining keystroke and mouse dynamics for enhanced security.

Most of the researches in review paper also proposed that the performance of keystroke Dynamics can be more improved by using different Classifiers and feature extraction methods. From the identified gap, this research aimed to improve the accuracy of keystroke Dynamic authentication system using different Machine learning algorithms and feature extractions as a result security of the authentication system will be increased.

CHAPTER THREE

THE PROPOSED SOLUTION

3.1 Introduction

In this section, following phases of the research will be discussed. The four main phases of the research are the dataset selection, data pre-processing, training, testing, and comparing algorithms phase. These four phases will go through how dataset will be selected and pre-processed, what machine learning algorithms will be used, how these algorithms will be trained and tested, and what criteria will the algorithms compared. Figure 3.1 illustrate Framework for behavioural biometric authentication.

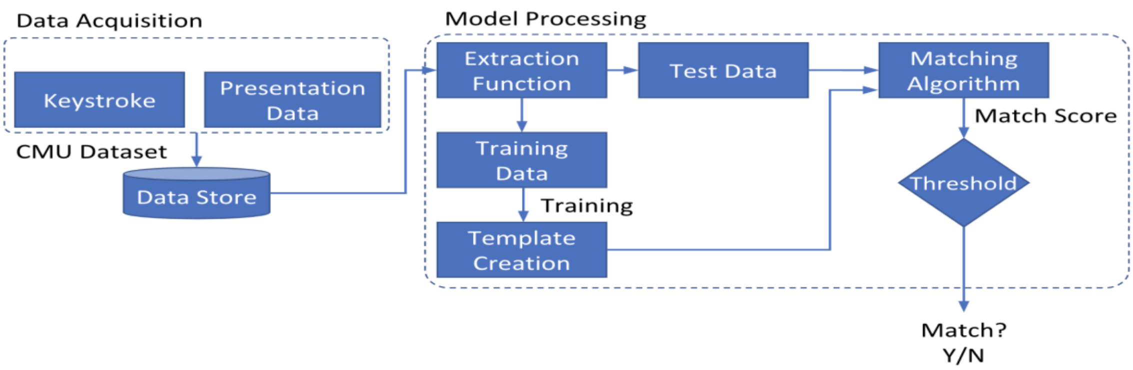


Figure 3. 1 The framework of behavioral biometric authentication

3.2 Datasets Selection

The model evaluated using existing keystroke dataset CMU (Carnegie Mellon University) which is benchmark dataset for keystroke dynamics [32] and available for research (A Scientific Understanding of Keystroke Dynamics). The data in keystroke dynamics is a fixed-text dataset. The reason that we selected the Fixed-Dataset is that it is more accurate than Free-Text[8]. The "DSL-StrongPasswordData" is downloaded from Kaggle website and saved as a .csv format in the computer. After the "DSL-StrongPasswordData.csv" dataset saved to the computer, Python 3.11.8 version notebook server with required packages like Anaconda is properly installed.

3.3. Data Pre-processing

The first step of data pre-processing is data cleansing it is a crucial step in the data preparation process, playing an important role in ensuring the accuracy, reliability, and overall quality of a dataset. In the field of data science, where insights and predictions are drawn from vast and complex datasets, the quality of the input data significantly influences the validity of analytical results. Data cleaning involves the systematic identification and correction of errors, inconsistencies, and inaccuracies within a dataset, encompassing tasks such as handling missing values, removing duplicates, addressing outliers and encoding data, splitting dataset, scaling features, feature importance.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model. The data pre-processing involves below steps:

- Getting the dataset
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data
- Splitting dataset into training and test set
- Feature scaling

Get the Dataset

To create a machine learning model, the first thing we required is a dataset as a machine learning model completely works on data. The selected data for a particular problem in a proper format is known as the dataset. To use the dataset in our code, we usually put it into a csv file. The following Python code is used during data pre-processing: Upload the Dataset "df=pd.read_csv(r'C:\Users\Mdegefi\Documents\UntitledFolder/DSL-StrongPasswordDataN.csv')".

Import Libraries

A library is a collection of functions that an algorithm can call and utilize. we can streamline data pre-processing procedures using tools and frameworks that make the process easier to organize and execute. Without certain libraries, one-liner solutions might take hours to code and optimize. In order to perform data pre-processing using Python, we need to import some predefined Python libraries. These libraries are used to perform some specific jobs. Some of the common libraries are NumPy, Scikit-learn, Pandas, Seaborn, Matplotlib, Keras for our machine learning.

Import Datasets

The next key step is loading the dataset that will be utilized in the machine learning algorithm. This is the most critical machine learning pre-processing step. Now we need to import the CMU fixed datasets which we are selected for our machine learning. But before importing a dataset, we need to set the current directory as a working directory.

- To import the dataset used,

“df=pd.read_csv(r'C:\Users\Mdegefi\Documents\UntitledFolder/DSL-StrongPasswordDataN.csv')”.

- To Displaying the head and tail of the dataset from the Python, we use “df.head()”, df.tail() and to get the size of the dataset in python, we use “df.shape” result (20400, 34).

```
df = pd.read_csv(r'C:\Users\Mdegefi\Documents\Untitled Folder/DSL-StrongPasswordDataN.csv')
df.head()
```

	subject	sessionIndex	rep	H.period	DD.period.t	UD.period.t	H.t	DD.t.i	UD.t.i	H.i	...	H.a	DD.a.n	UD.a.n	H.n	DD.n.l	UD.n.l	H.l	D
0	s002	1	1	0.1491	0.3979	0.2488	0.1069	0.1674	0.0605	0.1169	...	0.1349	0.1484	0.0135	0.0932	0.3515	0.2583	0.1338	
1	s002	1	2	0.1111	0.3451	0.2340	0.0694	0.1283	0.0589	0.0908	...	0.1412	0.2558	0.1146	0.1146	0.2642	0.1496	0.0839	
2	s002	1	3	0.1328	0.2072	0.0744	0.0731	0.1291	0.0560	0.0821	...	0.1621	0.2332	0.0711	0.1172	0.2705	0.1533	0.1085	
3	s002	1	4	0.1291	0.2515	0.1224	0.1059	0.2495	0.1436	0.1040	...	0.1457	0.1629	0.0172	0.0866	0.2341	0.1475	0.0845	
4	s002	1	5	0.1249	0.2317	0.1068	0.0895	0.1676	0.0781	0.0903	...	0.1312	0.1582	0.0270	0.0884	0.2517	0.1633	0.0903	

```
df = pd.read_csv(r'C:\Users\Mdegefi\Documents\Untitled Folder/DSL-StrongPasswordDataN.csv')
df.tail()
```

	subject	sessionIndex	rep	H.period	DD.period.t	UD.period.t	H.t	DD.t.i	UD.t.i	H.i	...	H.a	DD.a.n	UD.a.n	H.n	DD.n.l	UD.n.l	H.
20395	s057	8	46	0.0884	0.0685	-0.0199	0.1095	0.1290	0.0195	0.0945	...	0.1219	0.1383	0.0164	0.0820	0.1329	0.0509	0.1005
20396	s057	8	47	0.0655	0.0630	-0.0025	0.0910	0.1148	0.0238	0.0916	...	0.1008	0.0512	-0.0496	0.1037	0.0868	-0.0169	0.1445
20397	s057	8	48	0.0939	0.1189	0.0250	0.1008	0.1122	0.0114	0.0721	...	0.0913	0.1169	0.0256	0.0689	0.1311	0.0622	0.1034
20398	s057	8	49	0.0923	0.1294	0.0371	0.0913	0.0990	0.0077	0.0992	...	0.0882	0.0821	-0.0061	0.0576	0.0697	0.0121	0.0979
20399	s057	8	50	0.0596	0.1310	0.0714	0.0992	0.1103	0.0111	0.0998	...	0.0969	0.0784	-0.0185	0.0790	0.1133	0.0343	0.0807

5 rows × 34 columns

```
In [195]: df.shape
Out[195]: (20400, 34)
```

The CMU fixed dataset consists of a subject identifier (ID) variable, session number, repetition number, and 31 keystroke timing features (H, DD, and UD) collected from 51 users. The users were asked to type a secure password (.tie5Roanl) for each eight sessions with 50 typing repetitions, which lead to a total of 34 variables and 20400 observations. 31 timing features are recorded in the measurement unit of second. The dataset description is given in Table 3.1. Each row of data corresponds to the timing information for a single repetition of the password by a single subject. The first column, subject, is a unique identifier for each subject, for example s002 or s053. Note that even though the data set contains 51 subjects, the identifiers do not range from s001 to s051 since not every subject participated in every experiment. The second column ’sessionIndex’ is the session where the password was typed which ranges from 1 to 8. The third column ’rep’ which ranges from 1 to 50 is the repetition of the password within the session [38].

Table 3. 1. Summary of Dataset

No.	Variables	Details
1	Subject	Subject ID or class label for 51 users involved in typing task.
2	sessionIndex	A number of the session in the typing task; consists of 8 sessions in total.
3	Rep	A number of repetitions in the typing task; consists of 50 repetitions for each session.
4	H.period	The duration between pressing and releasing ‘.’ key.
5	DD.period.t	The duration between pressing ‘.’ key and pressing ‘t’ key.
6	UD.period.t	The duration between releasing ‘.’ key and pressing ‘t’ key.
7	H.t	The duration between pressing and releasing ‘t’ key.
8	DD.t.i	The duration between pressing ‘t’ key and pressing ‘i’ key.
9	UD.t.i	The duration between releasing ‘t’ key and pressing ‘i’ key.
10	H.i	The duration between pressing and releasing ‘i’ key.
11	DD.i.e	The duration between pressing ‘i’ key and pressing ‘e’ key.
12	UD.i.e	The duration between releasing ‘i’ key and pressing ‘e’ key.
13	H.e	The duration between pressing and releasing ‘e’ key.
14	DD.e.five	The duration between pressing ‘e’ key and pressing ‘five’ key.
15	UD.e.five	The duration between releasing ‘e’ key and pressing ‘five’ key.
16	H.five	The duration between pressing and releasing ‘.’ key.
17	DD.five.shift.r	The duration between pressing ‘five’ key and pressing ‘shift.r’ key.
18	UD.five.shift.r	The duration between releasing ‘five’ key and pressing ‘shift.r’ key.
19	H.shift.r	The duration between pressing and releasing ‘r’ key.
20	DD.shift.r.o	The duration between pressing ‘shift.r’ key and pressing ‘o’ key.
21	UD.shift.r.o	The duration between releasing ‘shift.r’ key and pressing ‘o’ key.
22	H.o	The duration between pressing and releasing ‘o’ key.
23	DD.o.a	The duration between pressing ‘o’ key and pressing ‘a’ key.
24	UD.o.a	The duration between releasing ‘o’ key and pressing ‘a’ key.
25	H.a	The duration between pressing and releasing ‘a’ key.
26	DD.a.n	The duration between pressing ‘a’ key and pressing ‘n’ key.
27	UD.a.n	The duration between releasing ‘a’ key and pressing ‘n’ key.
28	H.n	The duration between pressing and releasing ‘n’ key.
29	DD.n.l	The duration between pressing ‘n’ key and pressing ‘l’ key.
30	UD.n.l	The duration between releasing ‘n’ key and pressing ‘l’ key.
31	H.l	The duration between pressing and releasing ‘l’ key.
32	DD.l.return	The duration between pressing ‘l’ key and pressing ‘return’ key.
33	UD.l.return	The duration between releasing ‘l’ key and pressing ‘return’ key.
34	H.return	The duration between pressing and releasing ‘return’ key.

The remaining 31 columns represent the information of timing in the password and the title of the column indicates the type of timing information. For example, a column named 'H.period' means a hold time for the period key. The hold time is the time when the key was pressed until it was released. Another example is the column named 'DD.period.t' means the time interval between period key-down and t key-down. In other words, 'DD.period.t' can also represent the time where the period key was pressed to where the t key was pressed. The last example is the column named 'UD.period.t', it means the time interval between period key-up and t key-down, also known as the time where period was released to the time where t was pressed

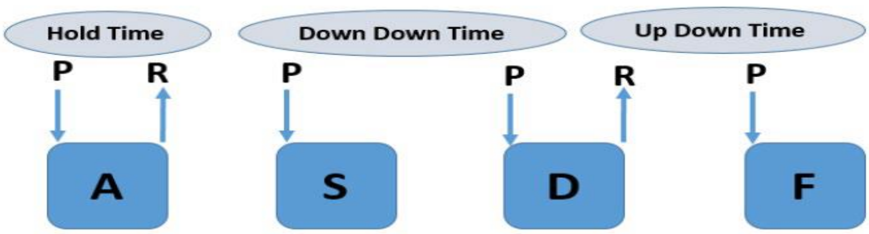


Figure 3. 2 *Keystroke Dynamics Features[39]*

The next step of data pre-processing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset. To check if there is a missing value in the dataset, we used "df.isnull().sum()" result "0" which means there is no missing value in our dataset.

```
:]:  df.isnull().sum()
:]:  subject             0
     sessionIndex        0
     rep                 0
     H.period            0
     DD.period.t         0
     UD.period.t         0
```

Finding Duplicated Data

The other step that we have done is checking duplicated records. As duplicated value does not add any value or information while using them in a model and would rather slow down the processing. So, it is better to check and remove duplicates before feeding the data to the model. To check for duplicate values in the dataset, we use “df.duplicated().sum()” result “0” and as the result, there is no duplicated value in our dataset.

```
In [196]: df.duplicated().sum()
Out[196]: 0
```

Finding Outliers

An Outlier is a data item/object that deviates significantly from the rest of the (so-called normal) objects. Using tactical Analysis of the data (df.describe), we can compare the min and Max value of each features to identify there is outliers and if there is extreme value b\n min & Max further investigation has to be done.

```
In [51]: df.describe()
Out[51]:
```

	sessionIndex	rep	H.period	DD.period.t	UD.period.t	H.t	DD.t.i	UD.t.i	H.i	DD.i.e	...
count	20400.000000	20400.000000	20400.000000	20400.000000	20400.000000	20400.000000	20400.000000	20400.000000	20400.000000	20400.000000	...
mean	4.500000	25.500000	0.093379	0.264148	0.170769	0.085727	0.169085	0.083358	0.081565	0.159372	...
std	2.291344	14.431223	0.029626	0.220534	0.226836	0.027424	0.123546	0.125755	0.026887	0.226928	...
min	1.000000	1.000000	0.001400	0.018700	-0.235800	0.009300	0.001100	-0.162100	0.003200	0.001400	...
25%	2.750000	13.000000	0.074400	0.146900	0.049800	0.066000	0.113600	0.027200	0.062000	0.089300	...
50%	4.500000	25.500000	0.089500	0.205950	0.108700	0.081000	0.140400	0.057800	0.077100	0.120900	...
75%	6.250000	38.000000	0.107900	0.306450	0.212400	0.099800	0.183900	0.096400	0.096900	0.173100	...
max	8.000000	50.000000	0.376100	12.506100	12.451700	0.241100	4.919700	4.799900	0.331200	25.987300	...

Encoding Categorical Data

Categorical data is data which has some categories such as, in our dataset; there is one categorical variable, Subject. Since machine learning model completely works on mathematics and numbers, but as our dataset would have a categorical variable “subject”, then it may create trouble while building the model. So it is necessary to encode these categorical variables into numbers.

Firstly, we converted the subject variables into categorical data. So to do this, we used LabelEncoder() class from preprocessing library.

'label_encode_x=LabelEncoder()' and

'x[:,0]=label_encode_x.fit_transform(x[:,0])'.

In above code, we have imported LabelEncoder class of sklearn library. This class has successfully encoded the variables into digits. In our dataset for Dummy Encoding, we have used OneHotEncoder class of preprocessing library.

'from sklearn.preprocessing import OneHotEncoder', 'onehotencoder=OneHotEncoder()' and 'onehotencoder.fit_transform(df.subject.values.reshape(-1,1)).toarray()' and the output values are "0" or "1"

Splitting dataset into training and test set

In this step we divide our dataset into a training set 80% and 70% and test set 20% and 30%. In machine learning splitting dataset is one of the crucial steps of data pre-processing as by doing this, we can enhance the performance of our machine learning model.

Figure 3. 3 Splitting data set in Training and test set

- **Training Set:** A subset of dataset to train the machine learning model (80%/70%).

- **Test set:** A subset of dataset to test the machine learning model (20%/30%).

For splitting the dataset, we have used the below lines of code:

"from sklearn.model_selection import train_test_split"

"x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2, random_state=0)"

```
train, test = train_test_split(df, test_size = 0.2, random_state=42)
```

The splitting a dataset into training and testing sets, such as 80/20 or 70/30. From a total of 20,400 final data sets collected for 80/20 dataset split, 16,320 for training and 4,080 for testing and for 70/30 dataset split, 14,280 for training and 6,120 for testing using Python 3.11 ensemble techniques and using RandomForestClassifier.

Splitting the Data:

- train_test_split(X, y, random_state=42, test_size=0.2): This line actually performs the split. It takes the feature matrix (X) and target variable (y), and it randomly shuffles and splits the data into training and testing sets.
- random_state=42: This parameter sets the random seed for reproducibility. If it uses the same seed, will get the same split each time you run the code.
- test_size=0.2: This parameter specifies that 20% of the data will be used for testing, and the remaining 80% will be used for training.
- train_test_split(X, y, random_state=42, test_size=0.3): This line actually performs the split. It takes the feature matrix (X) and target variable (y), and it randomly shuffles and splits the data into training and testing sets.
- random_state=42: This parameter sets the random seed for reproducibility. If it uses the same seed, will get the same split each time you run the code.

- test_size=0.3: This parameter specifies that 30% of the data will be used for testing, and the remaining 70% will be used for training.

Feature scaling

In the final step of date pre-processing, we used a technique to standardize the independent variables of the dataset in a specific range. In feature scaling, we put our variables in the same range and in the same scale so that no any variable dominates the other variable. For feature scaling,we have import StandardScaler class of sklearn.preprocessing library as: 'from sklearn.preprocessing import StandardScaler',
'st_x= StandardScaler()', 'x_train= st_x.fit_transform(x_train)' and we have get output from the python code 'x_test= st_x.transform(x_test)'. By executing the above lines of code, we have got the scaled values for x_train and x_test.

```
]: scaler = StandardScaler()
   scaler.fit(X)

   scaler.transform(X)
   scaler.transform(X_test)
```

```
]: array([[ 0.65629302,  0.52267595,  1.75794114, ..., -0.21079328,
           -0.50536119,  1.8930323 ],
          [-0.21580746,  0.87016701, -0.09628472, ..., -0.36634048,
           -0.3541073 , -0.3925109 ],
          [ 0.65629302, -0.3808008 , -0.88272055, ..., -0.60100994,
           -0.56104016, -0.58632496],
```

Feature Importance and Feature Selection

Random forest feature selection is an ensemble learning method, widely used for feature selection due to its inherent ability to rank features based on their importance. Feature selection aims to reduce the number of input variables to those that are most important to the model. This can enhance the model's performance by reducing overfitting, improving accuracy, and decreasing computation time.

Random Forest is particularly suited for feature selection for several reasons:

- **Intrinsic Feature Ranking**: Random Forest provides a built-in method to evaluate the importance of features.
- **Handles High Dimensionality**: Effective even when the number of features is much larger than the number of samples.
- **Non-Linearity**: Can capture complex interactions between features without requiring explicit specification of interactions.

From 31 timing features, we used Random Forest feature selection to select the most important 20 features and we have trained with Random Forest classifier using the selected features and evaluated its accuracy on the test set and increased model accuracy.

Benefits of Using Random Forest for Feature Selection

- **Improved Model Performance:** By selecting the most relevant features, the model can achieve higher accuracy and generalize better to new data.
- **Reduced Overfitting**: Fewer features can reduce the risk of overfitting, especially in models prone to this issue.
- **Enhanced Interpretability:** With fewer features, it becomes easier to interpret the model and understand the relationship between the features and the target variable.
- **Efficiency:** Reducing the number of features can lead to faster training and prediction times.

For Selecting Features, we used the below python code

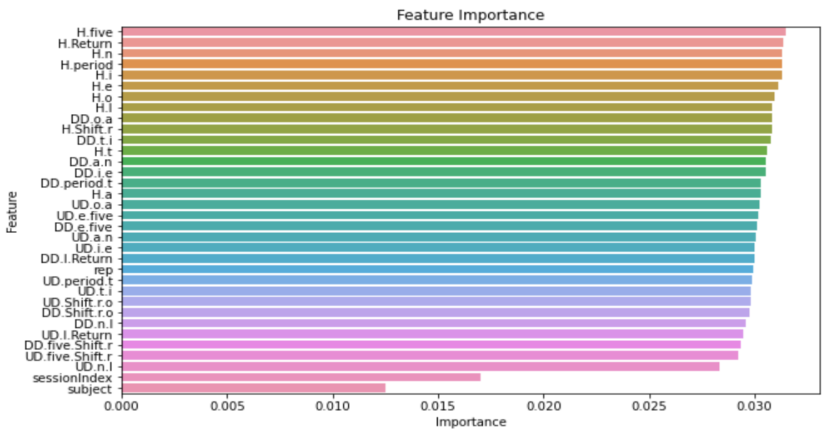
```
In [481]:  #Get feature importances
feature_importances = rf.feature_importances_

# Create a DataFrame for visualization
feature_importance_df = pd.DataFrame({
    'Feature': X_train.columns,
    'Importance': feature_importances
})

# Sort the DataFrame by importance
feature_importance_df = feature_importance_df.sort_values(by='Importance', ascending=False)
# Rank features by importance
feature_importance_df = feature_importance_df.sort_values(by='Importance', ascending=False)
print(feature_importance_df)

# Plot feature importances
plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Feature', data=feature_importance_df)
plt.title('Feature Importance')
plt.show()
```

```
          Feature  Importance
15         H.five    0.031509
33       H.Return    0.031387
27            H.n    0.031312
3        H.period    0.031307
9             H.i    0.031305
12            H.e    0.031134
21            H.o    0.030978
30            H.l    0.030839
22         DD.o.a    0.030832
18      H.Shift.r    0.030810
7          DD.t.i    0.030779
6             H.t    0.030598
25         DD.a.n    0.030530
10         DD.i.e    0.030529
4     DD.period.t    0.030317
4     DD.period.t    0.030317
24            H.a    0.030297
23         UD.o.a    0.030260
14      UD.e.five    0.030206
13      DD.e.five    0.030146
26         UD.a.n    0.030058
11         UD.i.e    0.030000
31      DD.l.Return    0.029998
2             rep    0.029955
5     UD.period.t    0.029889
8          UD.t.i    0.029845
20     UD.Shift.r.o    0.029797
19     DD.Shift.r.o    0.029780
28         DD.n.l    0.029590
32      UD.l.Return    0.029470
16  DD.five.Shift.r    0.029364
17  UD.five.Shift.r    0.029253
29         UD.n.l    0.028340
1     sessionIndex    0.017044
0         subject    0.012542
```



3.4 Training Phase

After selecting the most important feature from the dataset, next phase is to train the machine. We used 80% and 70% of the dataset to train the machine algorithms and 20% and 30% for testing the accuracy, precision, Recall and F1 score of the algorithm. The training datasets are the data that will train each machine learning algorithm to classify the subject's keystroke rhythm accurately. We have implemented four types of machine learning algorithms that follow the above procedure in different ways.

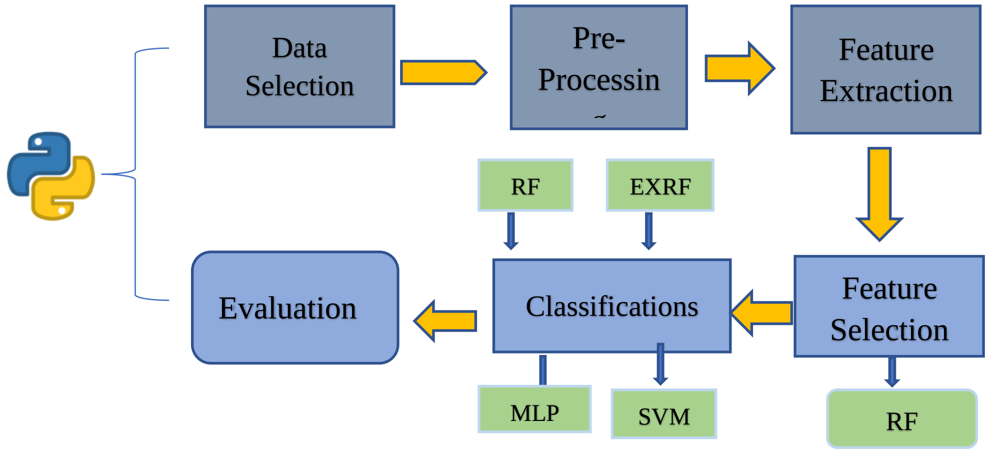


Figure 3. 4 The proposed ML method for user behavioral biometric authentication

The proposed method for constructing the behavioural biometric authentication system makes use of keystroke dynamics and machine learning. The method helps to automate the decision-making process while identifying the users as part of authentication. Figure 3.4 shows the stages involved in the method are as follows: - To solve a machine learning problem, the data has to be collected in a suitable format. Preparing the data is a main task in machine learning. After this, pre-processing has to be done to normalize and scale the data. The dataset contains the raw data from which the features have to be extracted. Once the dataset contains the class, the important features can be selected using feature selection algorithms to run the classifier. In this thesis, Random Forest, an ensemble learning method, is widely used for feature selection due to its inherent ability to rank features based on their importance. We tested the proposed model with four classifiers which are random forest (RF), Extra Trees (EXRF), support Vector Machine (SVM) and Multi-layer perceptron (MLP). The classification stage is to classify and identify the users by generating a model. Finally, evaluation has to be done to know the effectiveness of the algorithms.

The first machine learning algorithm is based on decision trees which is known as the random forest algorithm. The goal for this algorithm is the same as the decision tree, but it will complete the goal more than once per dataset to build a set of outcomes. These outcomes will then be compared together to create the final output. The purpose of random forest compared to decision trees is that this process will mitigate the effects of overfitting. Overfitting is a modelling error where the training data does not generalize well to the test data. This type of error will result in the data believing a certain outcome is the best answer in most cases, creating a bias in the algorithm.

The second machine learning algorithm is called Support Vector Machine (SVM). In this algorithm, data is plotted onto a graph and an optimal hyperplane is created that has the greatest margin between sections of data. This creates a classification where the groups on either side of the hyperplane make up the classes and allow a prediction to be made based on which class a new piece of data falls into. The larger the margin distance with the hyperplane, the more confidence there is in the prediction.

The third machine learning is ExtraTrees Classifier it is an ensemble ML approach that trains numerous decision trees and aggregates the results from the group of decision trees to output a prediction. However, there are few differences between Extra Trees and Random Forest. Random Forest uses bagging to select different variations of the training data to ensure decision trees are sufficiently different. However, Extra Trees uses the entire dataset to train decision trees. As such, to ensure sufficient differences between individual decision trees, it RANDOMLY SELECTS the values at which to split a feature and create child nodes. In contrast, in a Random Forest, we use an algorithm to greedy search and select the value at which to split a feature. Apart from these two differences, Random Forest and Extra Trees are largely the same.

The final machine learning is a Multi-layer perceptron (MLP) is a type of artificial neural network consisting of multiple layers of neurons. The neurons in the MLP typically use nonlinear activation functions, allowing the network to learn complex patterns in data. MLPs are significant in machine learning because they can learn nonlinear relationships in data, making them powerful models for tasks such as classification, regression, and pattern recognition.

A neural network consists of interconnected nodes, called neurons, organized into layers. Each neuron receives input signals, performs a computation on them using an activation function, and produces an output signal that may be passed to other neurons in the network. An activation function determines the output of a neuron given its input. These functions introduce nonlinearity into the network, enabling it to learn complex patterns in data.

The network is typically organized into layers, starting with the input layer, where data is introduced. Followed by hidden layers where computations are performed and finally, the output layer where predictions or decisions are made.

Neurons in adjacent layers are connected by weighted connections, which transmit signals from one layer to the next. The strength of these connections, represented by weights, determines how much influence one neuron's output has on another neuron's input. During the training process, the network learns to adjust its weights based on examples provided in a training dataset. Additionally, each neuron typically has an associated bias, which allows the neuron to adjust its output threshold.

Neural networks are trained using techniques called feedforward propagation and backpropagation. During feedforward propagation, input data is passed through the network layer by layer, with each layer performing a computation based on the inputs it receives and passing the result to the next layer.

Backpropagation is an algorithm used to train neural networks by iteratively adjusting the network's weights and biases in order to minimize the loss function. A loss function (also known as a cost function or objective function) is a measure of how well the model's predictions match the true target values in the training data. The loss function quantifies the difference between the predicted output of the model and the actual output, providing a signal that guides the optimization process during training.

3.5 Testing Phase

After the algorithm completes a session of training with 80% and 70% of the dataset, it has been evaluated with the testing data, with the remaining 20% and 30% of the data from the dataset and the testing has to be done before and after data pre-processing. The test datasets are used to predict which data connects to each user. Once the testing concludes, the given results from the algorithm are then compared to the expected outcomes using accuracy, Precision, Recall and F1 Score.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Experiment

This chapter goes over our experimental design and implementation, followed by the results and a discussion. All our experiments were written in Python (3.11.8). We used the machine learning library Scikit-learn (0.23.1) for machine learning implementation and evaluation. We also used the Python libraries numpy, matplotlib, pandas and seaborn for additional data processing and visualization functionality. Our programs were executed on a system running windows 10 with 12th Gen Intel(R) Core(TM) i7-1255U, 1700 Mhz, 10 Core(s), 12 Logical Processor(s) and 16GB of RAM.

4.2 Machine Learning Classifier

For classification purposes, different classifiers such as Random Forest, Multi-Layer perceptron, Extra Trees and SVM are used.

Random forest is an ensemble that consists of many decisions' trees. It uses bagging and feature randomness when building each individual tree. While predicting, for the purpose of maximizing the prediction accuracy, it considers the prediction which has been generated by the maximum trees.

A support vector machine (SVM) is a supervised machine learning model that can be used for both the tasks of classification and regression. After giving SVM model sets of labelled training data they're able to predict the target. The SVM models use kernel functions to avoid complex computations which make them suitable for handling the large data.

Multi-Layer Perceptron (MLP) is one of the most common neural network models used in machine learning. A multi-layered perceptron consists of interconnected neurons transferring information to each other.

The MLP is a feedforward neural network, which means that the data is transmitted from the input layer to the output layer in the forward direction. The connections between the layers are assigned weights. The weight of a connection specifies its importance. The technique of 'Backpropagation' is used to optimize the weights of an MLP till the weights converge to predict the correct values.

Extra trees (short for extremely randomized trees) is an ensemble supervised machine learning method that uses decision trees and like the random forest algorithm, it creates many decision trees, but the sampling for each tree is random, without replacement. This creates a dataset for each tree with unique samples. A specific number of features, from the total set of features, are also selected randomly for each tree. The most important and unique characteristic of extra trees is the random selection of a splitting value for a feature. Instead of calculating a locally optimal value using Gini or entropy to split the data, the algorithm randomly selects a split value. This makes the trees diversified and uncorrelated. Summarized result ML classifier with respect to evaluation matrix is illustrated in Figure 4.1.

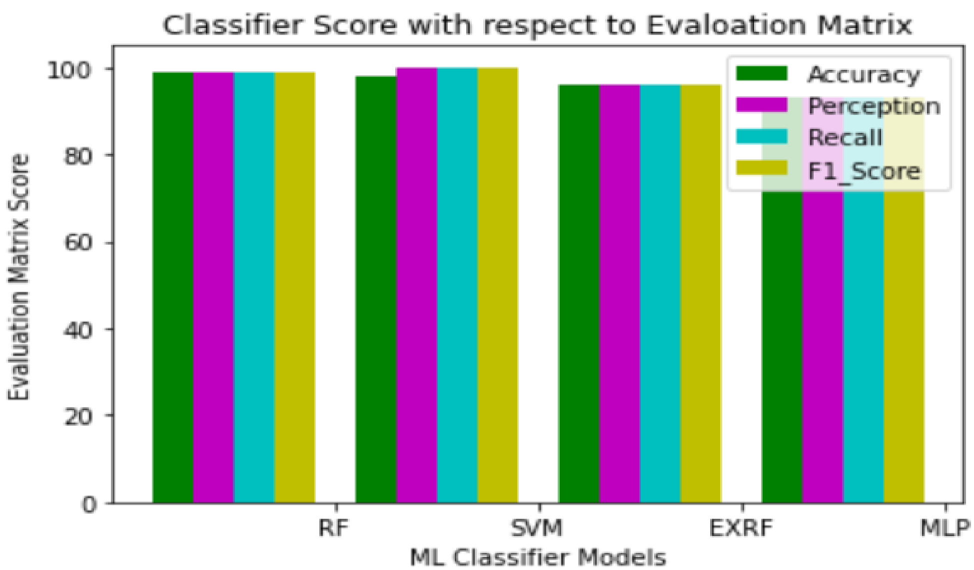


Figure 4. 1 ML Classifier Vs Evaluation Matrix Score

4.3 Evaluation Matrix

To test the effectiveness and access the validities of the proposed techniques for the analysis of user authentication using keystroke dynamics, there is a need to identify some performance measurements. The performance of our system is determined by computing accuracy, perception, recall, F1 score.

4.4 Evaluation of Algorithm

Eight different experiments were conducted in the experiment. The first four experiments are performed on a 20% test data (1 * 4 algorithms), and the second experiments are performed on 30% test data (1 * 4 algorithms). The four classifiers that were used in this study are a Support Vector Machine (SVM), Multilayer perceptron (MLP) and Random Forest (RF) and extra tree classifiers. Table 4.1 illustrate comparison of 20% and 30% test data accuracy result.

Table 4. 1 Comparing accuracy result of Classifiers with 0.2% and 0.3% test dataset

Test Data	**Classification Model with 0.2 and 0.3 Testing Data**			
	Random Forest (RF)	Support Vector Machine (SVM)	ExtraTrees The accuracy result is shown in Table . (XRF)	Multi-Layer Perceptron
0.2%	99.19	0.98	96.00	92.62
0.3%	99.10	0.98	95.29	91.65

In the experiment the result for both 20% and 30% test data, Random forest classifier score decreased from the highest 99.19 to 99.10. Support vector Machin score the same for both test data. The accuracy decreased from 99% to 95.29 for Extra Tress and from 92.62 to 91.65 for Multi-Layer perceptron.

From the experiment we observed that the accuracy of some models increases in larger training data. The reason for the accuracy increases for large training data is that More information being provided to the model and it is a key reason why bigger training datasets might lead to improved accuracy. Machine learning models are built to gain knowledge from the information in the data they are fed, have missed with a smaller sample. Furthermore, the training model accuracy improves as the amount of available data increases. Larger training datasets may improve accuracy in machine learning for another reason: they lessen the likelihood of overfitting the data.

4.5 Evaluation of Algorithm Before and after pre-processing

In this section the algorithm evaluation has been done before pre-processing and after pre-processing of data. Eight different experiments were conducted in this experiment. The first four experiments are performed with the data before pre-processing using 20% test data (1 * 4 algorithms), and the second experiments are performed with the data after pre-processing using 20% test data (1 * 4 algorithms). The four classifiers that were used in this study are a Support Vector Machine (SVM), Multilayer perceptron (MLP) and Random Forest (RF) and extra tree. The accuracy result is shown in Table 4.2.

Table 4. 2 Comparing accuracy result of Classifiers before and After Pre-processing

Results 50 User (20400 Dataset)				
Machine Learning Algorithms	**Random Forest**	**Extra Tree**	**Multi-layer perceptron**	**Support Vector Machine**
Accuracy before Pre-Processing	94.24%	96.00%	92..62%	100%
Accuracy After Pre-Processing	99.19%	96.00%	92.62%	98%

From the above experiment the accuracy result of Random forest after processing is higher than before processing. The reason for this that pre-processing helps to improve the accuracy of the model. By cleaning and formatting the data, we can ensure that the algorithm is only considering relevant information and that it is not being influenced by any irrelevant or incorrect data. It is widely accepted in the machine learning community that pre-processing data is an important step in the ML workflow and it can improve the performance of the model. In the case of support vector machine, the experiment accuracy result decreased from 100% to 98% after pre-processing due to overfitting in SVM model and it is removed during the pre-processing of data. SVM is a powerful, supervised machine learning algorithm used for both classification and regression challenges. However, like any model, it can suffer from over-fitting, where the model performs well on training data but poorly on unseen data.

Overfitting occurs when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is more pronounced in SVMs when the decision boundary is overly complex and tries to accommodate all training points.

4.5 Comparing ML Classifiers Results

In the comparison of the accuracy of the result in the literature review with the current our experiment result, we found that the accuracy score of Random forest and support vector machine accuracy is higher than the previous. The reason for is that in the current experiment we used standard scaling and on-hot encoding during pre-processing of dataset. Standard scaling and one-hot encoding are both common pre-processing techniques used in machine learning, but they serve different purposes and are applied to different types of data. Comparison of accuracy in the reviewed paper with the current result is illustrated in Table 4.3.

Table 4. 3 Comparing Accuracy in the Literature review paper and the current with the easy password ***"tie5Roanl"***

S/N	Classification Model	Previous Accuracy Score	Current Accuracy Score	Current Accuracy Score
1	Support Vector Machine (SVM)	97.9	Support Vector Machine (SVM)	98.00
2	Random Forest (RF)	98.8	Random Forest (RF)	99.12
3	XGBoost & MLP	93.79	Multi-Layer Perceptron	92.62
4	ExtraTree (XRF)		ExtraTree (XRF)	96.00

Standard scaling is a technique used to transform numerical data so that it has zero mean and unit variance. It is used to normalize the scale of the features in a dataset, which can be useful for certain machine learning algorithms that assume that the input data is centred around zero and has a similar scale across features and enhances the accuracy and efficiency of machine learning models. One-hot encoding, on the other hand, is a technique used to represent categorical data as a set of binary features. It is used to convert categorical features into a format that can be used as input to machine learning algorithms. It enhances the accuracy and efficiency of machine learning models by avoiding the pitfalls of ordinality and facilitating the use of categorical data.

4.5 Comparing ML Classifiers Results with respect to different evaluation Metrix

The final experiment is done by comparing the algorithm result with respect to four evaluation Metrix: Accuracy, Precision, Recall and F1 core. Sixteen experiments are performed after pre-processing the dataset with 20% test data (4 algorithms * 4 evaluation matrix). The reason for evaluating algorithms with Accuracy, precision, and recall and F1 score help evaluate the quality of classification models in machine learning. Each metric reflects a different aspect of the model quality, and depending on the use case, we might prefer one or another. Comparison of four ML classifier with respect to evaluation matrix is illustrated in Table 4.4.

*Table 4. 4 Results of using different classifiers with respect to different Performance Metrix the easy password "**tie5Roanl**"*

S/N	Performance Metrix	Classification Model			
		Random Forest (RF)	Support Vector Machine (SVM)	ExtraTrees (EXRF)	Multi-Layer Perceptron
1	**Accuracy**	99.19	98.00	96	93
2	**Precision**	99	100	96	93
3	**Recall**	99	100	96	93
4	**F1 Score**	99	100	96	93

From these results, we concluded that Random Forest classifier is overall the most effective algorithm when in conjunction with keystroke authentication. This makes sense in comparison to Decision Trees because Decision Trees consists of one tree and it is limiting the number of outcomes it finds throughout the datasets. The other algorithm, Support Vector Machine accuracy shows the lower result compared to the Random Forest results. This is because the Support Vector Machine relies on classes that are separated on a hyper-plane and outliers could have affected that.

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORKS

5.1. Conclusions

In this paper, the keystroke dynamics based authentication was tested using ".tie5Roanl" password. In the implementation described four classifiers are considered. The first classifier is Random forest, it is a supervised learning algorithm. The "forest" it builds is an ensemble of decision trees, usually trained with the bagging method. The general idea of the bagging method is that a combination of learning models increases the overall result and the best classification performance metrics obtained were from the Random Forest classifier. Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

Another great quality of the random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction. Sklearn provides a great tool for this that measures a feature's importance by looking at how much the tree nodes that use that feature reduce impurity across all trees in the forest. It computes this score automatically for each feature after training and scales the results so the sum of all importance is equal to one.

The random forest classifier with its extremely high accuracy of 99.19% would be most effective if applied to systems with highly sensitive data.

The second classifier is Support Vector Machine (SVM) with accuracy of 98% in the experiment. It is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well it's best suited for classification. The main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space. The third Classifier is ExtraTrees with 96% accuracy. It is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a “forest” to output it’s classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest. Each Decision tree in the Extra Trees Forest is constructed from the original training sample. Then, at each test node, each tree is provided with a random sample of k features from the feature-set from which each decision tree must select the best feature to split the data based on some mathematical criteria

The last classifier is A Multi-Layer Perceptron (MLP) with 93% accuracy. It is a type of artificial neural network that consists of multiple layers of neurons, or nodes, arranged in a hierarchical structure. It is one of the simplest and most widely used types of neural networks, particularly for supervised learning tasks such as classification and regression.

5.2. Future Works

Due to time constraints, the research focus was only on the general aspects of keystroke authentication in conjunction with machine learning. If given more time, there are many ways the experiment could be expanded for more results. Some ways include expanding the research into mobile devices such as; smartphones and tablets, increasing the number of criteria to test for each machine learning algorithm, and obtaining keystroke patterns around the entire day.

Another potential area of future research is that a study should be done in detecting emotions keystroke dynamics-based authentication explained that if a user has an injured hand or is simply distracted.

The user typing rhythm patterns might not be consistent through time, with the patterns used for training and proposed the need for periodic retraining to capture variations in users' typing behaviour over time

Yet keyboard dynamics is not without drawbacks, the most notable of which the fact that typing patterns can be erratic and inconsistent. Cramped muscles and sweaty hands can potentially alter a person's typing pattern. The type of keyboard also governs the typing patterns, which could affect verification. As concerning as these issues are, in the future sophisticated algorithms can easily remedy such effects.

It is also strongly recommended that IT professionals and hardware manufacturers adopt and integrate the use of enhanced multi-factor authentication in their newer products for enhanced security and mitigation of identity theft, pharming, sniffing, snooping, phishing and outright theft of possession factors of authentication.

In summary, we have achieved highest results for the challenging problem of user authentication based on keystroke dynamics, using Random Forest machine learning models. The results indicate that practical, Enhancing Multi-Factor authentication based on keystroke dynamics with Random Forest classifier is feasible.

Appendix

- **Random Forest Classifier Accuracy Precision, Recall and F1 Score**

```
In [697]: from sklearn.ensemble import StackingClassifier

          baseline = []
          for iteration in range(10):
              X_train, X_test, y_train, y_test = train_test_split(
                  df.iloc[:, 1:],
                  df.subject,
                  test_size=0.2,
                  random_state=iteration,
                  stratify=df.subject
              )

              c1 = RandomForestClassifier(random_state=0, n_estimators=500)
              c1.fit(X_train, y_train)


              y_pred = c1.predict(X_test)


          print(accuracy_score(y_test, y_pred))

          0.9919117647058824
```

```
In [648]: print(classification_report(y_test, y_pred))

              precision    recall  f1-score   support

           0       1.00      1.00      1.00        80
           1       1.00      1.00      1.00        80
           2       0.99      0.99      0.99        80
           3       0.99      1.00      0.99        80
           4       1.00      0.99      0.99        80
           5       0.98      1.00      0.99        80
           6       1.00      1.00      1.00        80
           7       1.00      0.93      0.96        80
           8       0.99      1.00      0.99        80
           9       1.00      1.00      1.00        80

          49       1.00      0.99      0.99        80
          50       0.98      1.00      0.99        80

    accuracy                           0.99      4080
   macro avg       0.99      0.99      0.99      4080
weighted avg       0.99      0.99      0.99      4080
```

- **Support Vector Machine Classifier Accuracy Precision, Recall and F1 Score**

```
In [365]: scores = cross_val_score(svm_model, X,Y, cv = loo, scoring='accuracy')

          print( scores.mean() )

          0.98

In [273]: from sklearn.metrics import classification_report

In [274]: scores = cross_val_score(svm_model, X,Y, cv = loo, scoring='precision')
          print(classification_report(y_test, y_pred))
```

```
In [274]: scores = cross_val_score(svm_model, X,Y, cv = loo, scoring='precision')
          print(classification_report(y_test, y_pred))

              precision    recall  f1-score   support

           1       1.00      1.00      1.00       503
           2       1.00      1.00      1.00       481
           3       1.00      1.00      1.00       490
           4       1.00      1.00      1.00       480
           5       1.00      1.00      1.00       545
           6       1.00      1.00      1.00       534
           7       1.00      1.00      1.00       512
           8       1.00      1.00      1.00       535

    accuracy                           1.00      4080
   macro avg       1.00      1.00      1.00      4080
weighted avg       1.00      1.00      1.00      4080
```

- **Extra Trees (XRF) Classifier Accuracy Precision, Recall and F1 Score**

```
[101]: from sklearn.ensemble import StackingClassifier

       baseline = []
       for iteration in range(10):
           X_train, X_test, y_train, y_test = train_test_split(
               data.iloc[:, 1:],
               data.subject,
               test_size=0.2,
               random_state=iteration,
               stratify=data.subject
           )

           c1 = ExtraTreesClassifier(random_state=0, n_estimators=500)
           c1.fit(X_train, y_train)


           y_pred = c1.predict(X_test)

       print(accuracy_score(y_test, y_pred))
       baseline.append(accuracy_score(y_test, y_pred))

       0.9600490196078432
```

```
In [102]: print(classification_report(y_test, y_pred))

              precision    recall  f1-score   support

        s002       0.91      0.94      0.93        80
        s003       0.99      0.96      0.97        80
        s004       0.93      0.96      0.94        80
        s005       0.95      0.95      0.95        80
        s007       0.96      0.91      0.94        80
        s008       0.95      0.93      0.94        80

        s056       0.97      0.91      0.94        80
        s057       0.97      0.93      0.95        80

    accuracy                           0.96      4080
   macro avg       0.96      0.96      0.96      4080
weighted avg       0.96      0.96      0.96      4080
```

- **Multi-Layer Perceptron Classifier Accuracy Precision, Recall and F1 Score**

```
In [165]: model_accuracy = metrics.accuracy_score(y_test, predicted_output)
          print('Accuracy of Model:',model_accuracy)

          Accuracy of Model: 0.9262254901960785
```

```
In [102]: print(classification_report(y_test, predicted_output, target_names=unisub))
```

```
              precision    recall  f1-score   support

        s002       0.89      0.86      0.87        72
        s003       0.81      0.93      0.86        69
        s004       0.92      0.92      0.92        73
        s005       0.97      0.95      0.96        73
        s007       0.92      0.83      0.87        54

        s054       0.85      0.83      0.84        83
        s055       0.97      0.98      0.98        65
        s056       0.94      0.94      0.94        84
        s057       0.95      0.86      0.90        93

    accuracy                           0.93      4080
   macro avg       0.93      0.93      0.93      4080
weighted avg       0.93      0.93      0.93      4080
```

References

[1] R. J. Forsen, G. E., Nelson, M. R., & Staron Jr, “Personal Attributes Authentication Techniques,” *Pattern Analysis and Recognition Corp Rome Ny.*, 1997.

[2] “Authentication by Keystroke Timing.pdf.”

[3] R. Shadman, A. A. Wahab, M. Manno, M. Lukaszewski, D. Hou, and F. Hussain, “Keystroke Dynamics: Concepts, Techniques, and Applications,” no. March, 2023.

[4] S. Krishnamoorthy, L. Rueda, S. Saad, and H. Elmiligi, “Identification of user behavioral biometrics for authentication using keystroke dynamics and machine learning,” *ACM International Conference Proceeding Series*, pp. 50–57, 2018, doi: 10.1145/3230820.3230829.

[5] S. Krishnamoorthy, L. Rueda, S. Saad, and H. Elmiligi, “Identification of user behavioral biometrics for authentication using keystroke dynamics and machine learning,” *ACM International Conference Proceeding Series*, pp. 50–57, 2018, doi: 10.1145/3230820.3230829.

[6] F. Monrose and A. Rubin, “Keystroke Dynamics as a Biometric,” *Future Generation Computer Systems*, vol. 16, pp. 351–359, 2000.

[7] F. Bergadano, D. Gunetti, and C. Picardi, “User Authentication through Keystroke Dynamics,” *ACM Transactions on Information and System Security*, vol. 5, no. 4, pp. 367–397, 2002, doi: 10.1145/581271.581272.

[8] P. S. Teh, A. B. J. Teoh, and S. Yue, “A survey of keystroke dynamics biometrics,” *The Scientific World Journal*, vol. 2013, 2013, doi: 10.1155/2013/408280.

[9] V. Zimmermann, “From the Quest to Replace Passwords towards Supporting Secure and Usable Password Creation,” 2021.

[10] Akshat Shah, Parth Shah, Hitarth Shah, and Chetashri Bhadane, “Strengthening user Authentication using Keystroke Dynamics,” *International Journal of Engineering Research and*, vol. V4, no. 11, pp. 436–440, 2015, doi: 10.17577/ijertv4is110437.

[11] S. S. Zeid, R. A. ElKamar, and S. I. Hassan, “Fixed-Text vs. Free-Text Keystroke Dynamics for User Authentication,” *Engineering Research Journal - Faculty of Engineering (Shoubra)*, vol. 51, no. 1, pp. 95–104, 2022, doi: 10.21608/erjsh.2022.224312.

[12] R. Shadman, A. A. Wahab, M. Manno, M. Lukaszewski, D. Hou, and F. Hussain, “Keystroke Dynamics: Concepts, Techniques, and Applications,” pp. 1–42, 2023.

[13] mina ibrahim, H. AbdelRaouf, K. Amin, and N. Semary, “Keystroke dynamics based user authentication using Histogram Gradient Boosting,” *IJCI. International Journal of Computers and Information*, vol. 0, no. 0, pp. 0–0, 2022, doi: 10.21608/ijci.2022.155605.1081.

[14] Barbara Kitchenham, “Procedures for Performing Systematic Reviews,” *Keele University Technical Report*, vol. 33, no. 2004, pp. 1–26, 2014.

[15] C. Okoli, “A guide to conducting a standalone systematic literature review,” *Communications of the Association for Information Systems*, vol. 37, no. 1, pp. 879–910, 2015, doi: 10.17705/1cais.03743.

[16] P. H. Pisani and A. C. Lorena, “A systematic review on keystroke dynamics,” *Journal of the Brazilian Computer Society*, vol. 19, no. 4, pp. 573–587, 2013, doi: 10.1007/s13173-013-0117-7.

[17] Y. Brice *et al.*, “Deep Learning Neural Networks To cite this version : HAL Id : hal-03716818 Keystroke Dynamics based User Authentication using Deep Learning Neural Networks,” 2022.

[18] R. Mao, X. Wang, and H. Ji, “ACBM: attention-based CNN and Bi-LSTM model for continuous identity authentication,” *J Phys Conf Ser*, vol. 2352, no. 1, 2022, doi: 10.1088/1742-6596/2352/1/012005.
[19] G. Stragapede, P. Delgado-Santos, R. Tolosana, R. Vera-Rodriguez, R. Guest, and A. Morales, “Mobile Keystroke Biometrics Using Transformers,” *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition, FG 2023*, 2023, doi: 10.1109/FG57933.2023.10042710.
[20] A. A. Wahab *et al.*, “Shared Multi-Keyboard and Bilingual Datasets to Support Keystroke Dynamics Research,” *CODASPY 2022 - Proceedings of the 12th ACM Conference on Data and Application Security and Privacy*, pp. 236–241, 2022, doi: 10.1145/3508398.3511516.
[21] H. C. Chang, J. Li, and M. Stamp, “Machine Learning-Based Analysis of Free-Text Keystroke Dynamics,” *Advances in Information Security*, vol. 54, pp. 331–356, 2022, doi: 10.1007/978-3-030-97087-1_14.
[22] H. C. Chang, J. Li, C. S. Wu, and M. Stamp, “Machine Learning and Deep Learning for Fixed-Text Keystroke Dynamics,” *Advances in Information Security*, vol. 54, pp. 309–329, 2022, doi: 10.1007/978-3-030-97087-1_13.
[23] A. C. Iapa and V. I. Cretu, “Modified Distance Metric That Generates Better Performance for the Authentication Algorithm Based on Free-Text Keystroke Dynamics,” *SACI 2021 - IEEE 15th International Symposium on Applied Computational Intelligence and Informatics, Proceedings*, no. March, pp. 455–460, 2021, doi: 10.1109/SACI51354.2021.9465601.
[24] A. Andrean, M. Jayabalan, and V. Thiruchelvam, “Keystroke Dynamics Based User Authentication using Deep Multilayer Perceptron,” *Int J Mach Learn Comput*, vol. 10, no. 1, pp. 134–139, 2020, doi: 10.18178/ijmlc.2020.10.1.910.
[25] A. M. Gedikli and M. Ö. Efe, “A simple authentication method with multilayer feedforward neural network using keystroke dynamics,” *Communications in Computer and Information Science*, vol. 1144 CCIS, pp. 9–23, 2020, doi: 10.1007/978-3-030-37548-5_2.
[26] N. Altwaijry, “Keystroke Dynamics Analysis for User Authentication Using a Deep Learning Approach,” *International Journal of Computer Science and Network Security*, vol. 20, no. 12, pp. 209–216, 2020, doi: 10.22937/IJCSNS.2020.20.12.23.
[27] T. Xi, I. Kuzminykh, B. Ghita, and T. Bakhshi, “Evaluating Learning Algorithms for Keystroke Based User Authentication,” *2023 IEEE International Black Sea Conference on Communications and Networking, BlackSeaCom 2023*, no. July, pp. 283–288, 2023, doi: 10.1109/BlackSeaCom58138.2023.10299695.
[28] A. O. Aljahdali, F. Thabit, H. Aldissi, and W. Nagro, “Dynamic Keystroke Technique for a Secure Authentication System based on Deep Belief Nets,” *Engineering, Technology and Applied Science Research*, vol. 13, no. 3, pp. 10906–10915, 2023, doi: 10.48084/etasr.5841.
[29] N. Nanglae, B. M. Yakubu, and P. Bhattarakosol, “Extraction of Hidden Authentication Factors from Possessive Information,” *Journal of Sensor and Actuator Networks*, vol. 12, no. 4, 2023, doi: 10.3390/jsan12040062.
[30] A. Lo, V. H. Ayma, and J. Gutierrez-Cardenas, “A Comparison of Authentication Methods via Keystroke Dynamics,” *Proceedings of the 2020 IEEE Engineering International Research Conference, EIRCON 2020*, 2020, doi: 10.1109/EIRCON51178.2020.9253751.
[31] E. Lantz, “User authentication through behavioral biometrics using multi- class classification algorithms User authentication through behavioral biometrics using multi-class classification algorithms,” 2023.
[32] J. Srkrec, “ENHANCING USER-LEVEL SECURITY : PERFORMANCE ANALYSIS OF MACHINE LEARNING ALGORITHMS FOR,” vol. 101, no. 13, pp. 5313–5324, 2023.

[33] P. Kasprowski, Z. Borowska, and K. Harezlak, “Biometric Identification Based on Keystroke Dynamics,” *Sensors*, vol. 22, no. 9, 2022, doi: 10.3390/s22093158.
[34] L. Aversano, M. L. Bernardi, M. Cimitile, and R. Pecori, “Continuous Authentication using Deep Neural Networks Ensemble on Keystroke Dynamics,” *PeerJ Comput Sci*, vol. 7, pp. 1–27, 2021, doi: 10.7717/PEERJ-CS.525.
[35] Y. Brice *et al.*, “Deep Learning Neural Networks To cite this version : HAL Id : hal-03716818 Keystroke Dynamics based User Authentication using Deep Learning Neural Networks,” 2022.
[36] H. Chang, “Keystroke Dynamics Based on Machine Learning,” 2021.
[37] K. S. Killourhy, “A Scientific Understanding of Keystroke Dynamics,” *Cmu-Cs-12-100*, vol. 12, no. January, p. 216, 2012.
[38] H. Chang, “Keystroke Dynamics Based on Machine Learning,” 2021.
[39] mina ibrahim, H. AbdelRaouf, K. Amin, and N. Semary, “Keystroke dynamics based user authentication using Histogram Gradient Boosting,” *IJCI. International Journal of Computers and Information*, vol. 0, no. 0, pp. 0–0, 2022, doi: 10.21608/ijci.2022.155605.1081.