4-20-2018

Identification of User Behavioural Biometrics for Authentication using Keystroke Dynamics and Machine Learning

Sowndarya Krishnamoorthy
University of Windsor

Identification of User Behavioral Biometrics for Authentication using Keystroke Dynamics and Machine Learning

By

Sowndarya Krishnamoorthy

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2018

Identification of User Behavioral Biometrics for Authentication using Keystroke Dynamics and Machine Learning

by

Sowndarya Krishnamoorthy

APPROVED BY:

G. Bhandari
Odette School of Business

S. Saad
School of Computer Science

L. Rueda, Advisor
School of Computer Science

April 15, 2018

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

This thesis focuses on the effective classification of the behavior of users accessing computing devices to authenticate them. The authentication is based on keystroke dynamics, which captures the users behavioral biometric and applies machine learning concepts to classify them. The users type a strong passcode ".tie5Roanl" to record their typing pattern. In order to confirm identity, anonymous data from 94 users were collected to carry out the research. Given the raw data, features were extracted from the attributes based on the button pressed and action timestamp events. The support vector machine classifier uses multi-class classification with one vs. one decision shape function to classify different users. To reduce the classification error, it is essential to identify the important features from the raw data. In an effort to confront the generation of features from attributes an efficient feature extraction algorithm has been developed, obtaining high classification performance are now being sought. To handle the multi-class problem, the random forest classifier is used to identify the users effectively.

In addition, mRMR feature selection has been applied to increase the classification performance metrics and to confirm the identity of the users based on the way they access computing devices. From the results, we conclude that device information and touch pressure effectively contribute to identifying each user. Out of them, features that contain device information are responsible for increasing the performance metrics of the system by adding a token-based authentication layer. Based upon the results, random forest yields better classification results for this dataset. The research will contribute significantly to the field of cyber-security by forming a robust authentication system using machine learning algorithms.

DEDICATION

I would like to dedicate my thesis to my dear father Mr. Krishnamoorthy, my brother Mr. Ashwin, and especially my mother Mrs. Santhi Krishnamoorthy, who always persuaded me to take this program. Without their support and effort, I could not have achieved anything till now.

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my supervisor Dr. Luis Rueda for his perpetual support, patience, and inspiration during my Master's program in University of Windsor. It was such an honor for me to know you, and also be in your research team. Thank you so much for giving me an opportunity to learn from you.

Secondly, I would like to express my sincere appreciation to my internal committee member Dr. Sherif Saad for guiding me throughout my thesis work. I thank you for the unceasing encouragement and support.

Thirdly, I would like to thank Dr. Haytham Elmiligi from Thompson Rivers University for developing a Android application called iProfile and collaborating with us.

Also, I would also like to express my gratitude to my external committee member Dr. Gokul Bhandari for his beneficial advices and suggestions to my thesis.

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 the School of Computer Science and all concerned people who helped with me in this regard.

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.

This work has been partially supported by *NSERC*, the Natural Science and Engineering Council of Canada. This research has also been approved by the *Research Ethics Board* of University of Windsor, Canada (17-109 File No.34100). I would like to acknowledge the participants who actively took part in the research by inputting data.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Authentication

Authentication can be defined as verifying the validity of a user by using at least one form of the identification methods. To grant access to the system, the users identity should be verified by determining the following factors-

i **Knowledge-based factors:** It is defined as what the user knows. Some of them are any forms of a password, personal identification number, answer to the secret questions and many more [8].

ii **Possession-based factors:** It is defined as what the user has. Some of them are an identification card, security token, device token or any unique hardware identifier [42].

iii **Inherence-based factors:** It is defined based on what the user is or how he does. Some of the physiological factors are fingerprint, iris and DNA patterns and some of the behavioural factors are biometric identifiers, signatures, voice and face [36].

Authentication can be a combination of the above. The types of authentication categories include:-

- *Single-factor authentication:* It makes use of one factor to authenticate the user trying to login to the system. It is more prone to different cyber attacks.
- *Two-factor authentication:* It combines any two authentication factors to increase the level of security in the system. A practical example of this implementation is the real-time banking login where some banks generate a one time password (OTP) while the

user logins in by typing the correct password. Only if the password entered is valid the OTP gets generated, and if the user enters the generated number from his device correct, he gains access to the system.

- *Multi-factor authentication:-* It combines many authentication mechanisms to form a layered approach. The plethora of functionalities offered by multi-factor authentication includes protection from intrusion, enhancement of security, and reliable false proof system. My thesis focuses more on this multi-factor authentication to develop a robust system to identify the users via using machine learning algorithms. The idea is to add three factors of authentication by entering the correct password, verifying the device, and identifying the users typing pattern.

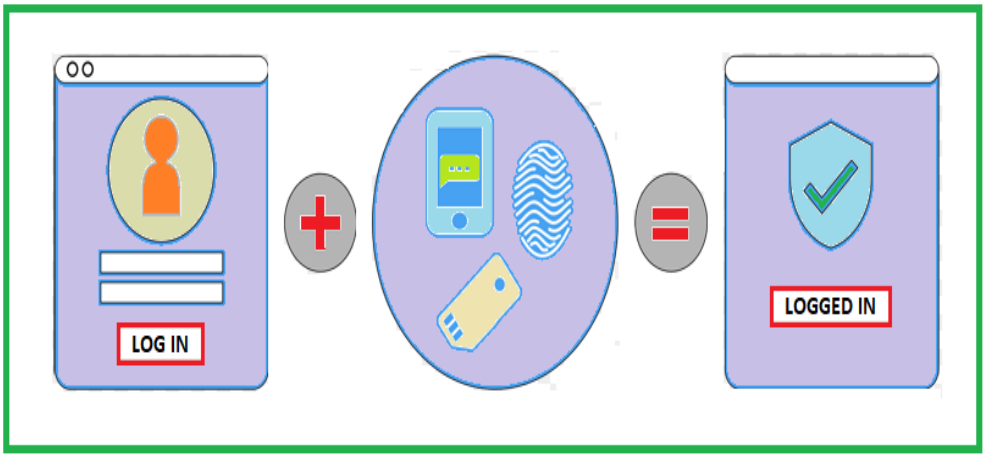

FIGURE 1.1.1: Multi-factor authentication using three forms of factors.

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 [22]. 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 [23]. 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 foolproof algorithm to be created.

1.2 Biometrics

Biometrics refer to measurable human characteristics to define and identify a user [37]. They are excellent and unique user characteristics to determine their identity. As the level of security decreases, the need for developing a highly secured identification and personal verification system increases. Physical biometrics are very useful to control the access to secure buildings. However, it has a major limitation that they can be easily compromised [40]. To verify users belonging to a large population, physiological characteristics such as fingerprints, iris, finger vein patterns, and face geometry play a vital role in user verification. On the other hand, technology is increasingly proposed to counter the cyber attacks on the transactions and Internet. Thus, another level of authentication based on the behaviour of the user is required. This scheme will reduce the performance issues and enhance the security of the existing techniques. Biometrics focuses on solid and vigorous distinguishing proof of users from their own attributes, for the most part for security and validation purposes, yet in addition for distinguishing and verifying the users of more astute applications. Every user has a unique pattern and signature to access the device; the system must identify the illegitimate user even if the correct login ID and password is typed from the users computing device.

1.2.1 Behavioural Biometrics

Behavioural biometrics are devices that analyze 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

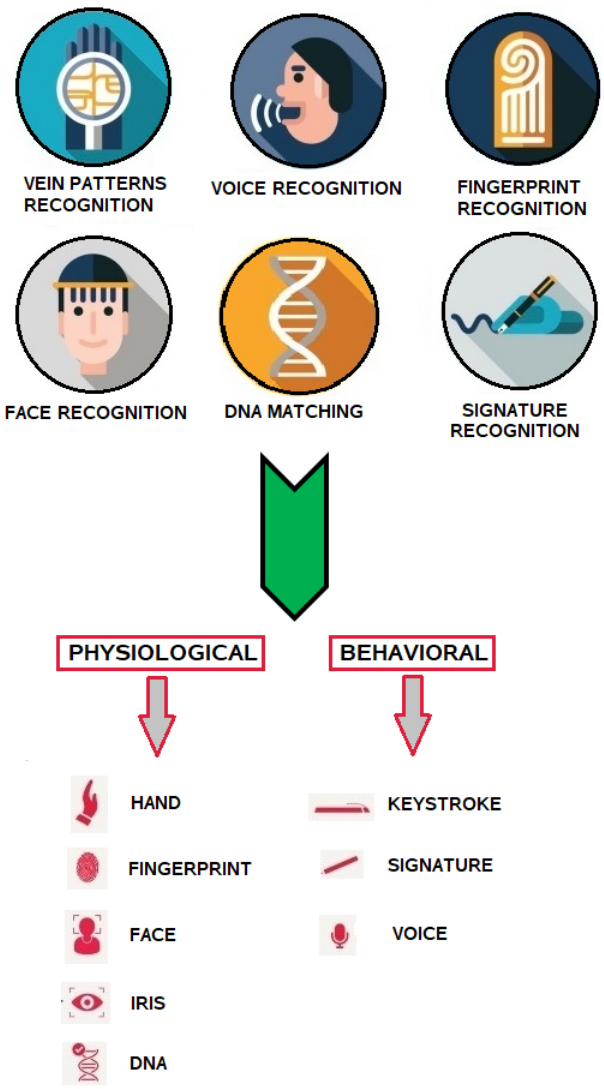

FIGURE 1.2.1: Different types of biometrics used for authentication.

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 [27]. If a security breach occurs, the information on who is responsible for it

TABLE 1.2.1: Comparison of biometric techniques.

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

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.

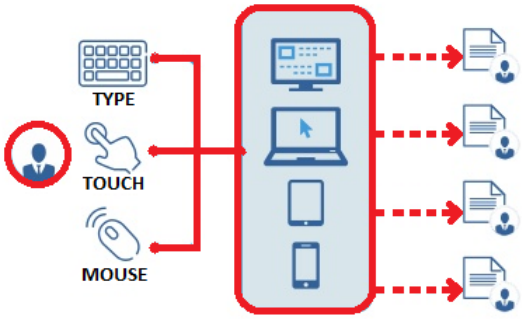

FIGURE 1.2.2: Behavioural biometrics as a form of authentication.

An extensive variety of behavioral biometrics have been proposed in view of human associations with machine amid the most recent couple of years [38]. Table 1.2.1 depicts the various biometric methods and its characteristics which can be used for multi-factor authentication [16].

1.2.2 Types of Behavioural Biometrics

There have been increasing research efforts on the types of biometrics such as static and dynamic authentication. Static authentication (SA) remembers a user based on unchangeable biometrics such as fingerprints, veins, static passwords and others. The password information is stored in a physical database and must be adaptable to change. Dynamic or continuous authentication (CA) focuses on identifying a user throughout the session while logged in [34]. The real-time information from a session is used to analyze and authenticate the user based on the behavioural profile which has patterns interwoven with the usage characteristics.

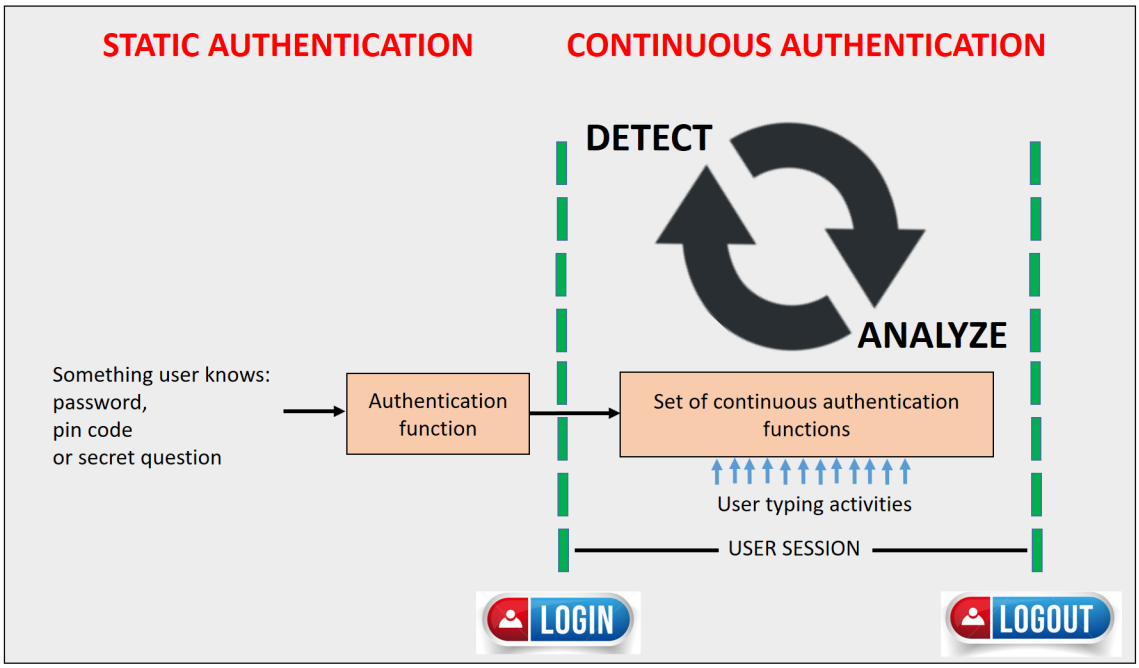

FIGURE 1.2.3: Types of behavioural biometrics showing static and continuous authentication.

1.2.3 Types of Behavioural Authentication

In an SA system, the execution of the matching algorithm is accounted for in false match rate (FMR) and false non-match rate (FNMR). For an SA system, it is imperative to know the likelihood that the system makes a mistake that is the likelihood that a legitimate user

has not been allowed access or that an impostor user has been granted access.

The user is authenticated using behavioral biometrics and is allowed towards the beginning of a session while legitimate for the full session. Static confirmation implies it is feasible for an impostor to seize a session and take control of a system after the veritable user has been granted access. An alternate kind of authentication, that defeats the issue portrayed above, is CA. A CA biometric framework checks the authenticity of the user amid the full session. In a conventional CA system, the user ought not to know that his or her personality is checked ceaselessly. A genuine CA system utilizes every different activity of the user in the process to decide his or her validity. When question emerges about the validity of the user, the system can lock, and the user needs to return to the static authentication access control mechanism to proceed with working. On the contrary, we find CA where the legitimate user is consistently verified in view of the action of the present user. It is essential to know whether an impostor user gets identified by the system and gets locked out, and it is significantly more critical to know how much activity can be done by the system to uncover an impostor. Subsequently, when looking at two CA systems that both distinguish all impostor users, the system that identifies the impostor faster is the best one.

1.3 Keystroke Dynamics

Keystroke dynamics, which is a behavioral biometry, refers to the unique patterns of rhythm and timing-based features that are created when a user types on a touchscreen in computing devices such as mobile devices [22]. The biometric system uses a pattern recognition system to classify users based on their physical and behavioral characteristics [5]. It is a method for identifying or verifying the users based on the way they type on either a physical or vrtual keyboard. This type of system uses artificial intelligence to differentiate legitimate users and illegitimate users. To protect a set of users from the illegitimate use of their accounts, the attributes of how they type and use the system are taken into account for authorizing the user.

The typing dynamics gives the detailed timing information of when exactly each key was pressed and when it was released while a person is typing on a touch screen [40]. This

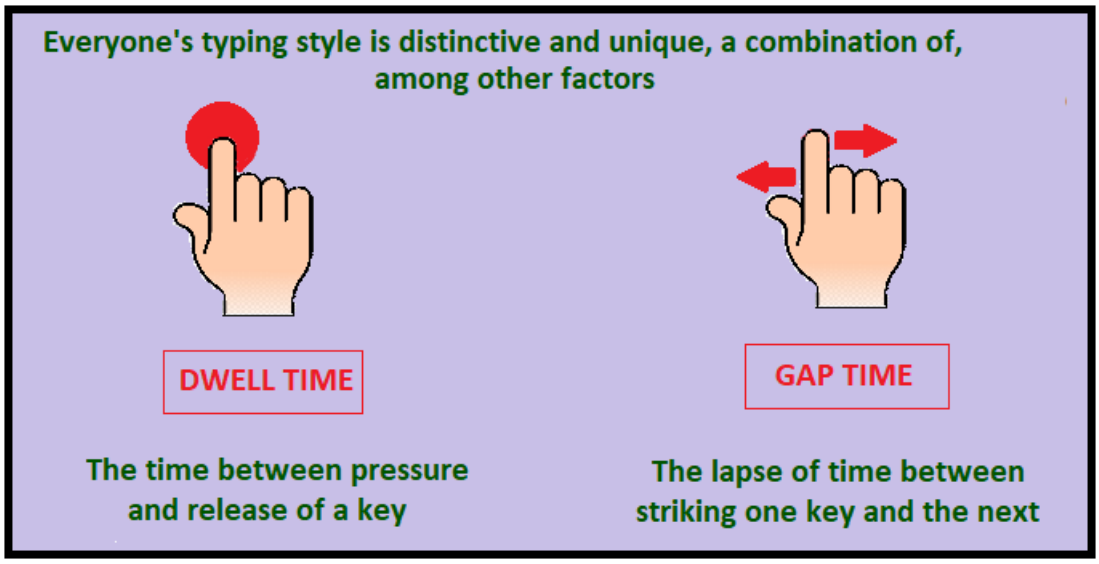

FIGURE 1.3.1: Keystroke dynamics is a combination of dwell time and flight time.

can be used as a primary pattern for future comparison.

1.3.1 Applications

Biometrics finds its application in various sectors such as police and prison services to access the closed-circuit television footages, hospitals to identify correct patients for treatment and procedures, facial recognition systems to prevent illegitimate entry, for security access of confidential rooms and servers, Web access, enterprise network access to incorporate encryption along with biometric-based authentication and in many other client and useful requirements [35]. The keystroke dynamics finds its application in these two kinds of systems:-

- *BioTracker* is a biometric authentication software that uses machine learning and keystroke dynamics to identity a user.
- *BioCheck* uses token based for ubiquitous Web-based login and workstation authentication, which is used to verify the user.

They can be combined with any computing device based on two modes such as identification and verification mode.

1.4 Machine Learning

Machine learning is the investigation of inspiring systems to learn without being customized by programmers automatically. It is a branch of Artificial Intelligence related to pattern recognition and computational theory [24]. In the previous decade, machine learning has given us self-driving automobiles, reasonable, viable Web look, and an immensely enhanced comprehension of the human genome. Machine learning is so unavoidable today that you most likely utilize it many times each day without knowing it. All the more imperatively, the hypothetical underpinnings of machine learning apply to the field of biometrics to distinguish the users based on their behaviour [32].

It is to develop a system which automates the automation and allows the data to do the work instead of the programmers [20]. The major categories of machine learning tasks include the following:-

- **Supervised learning:** It comprises parametric/non-parametric algorithms, kernels, neural networks, support vector machines and many more classifiers.
- **Unsupervised learning:** It comprises clustering, recommender systems, deep learning, dimensionality reduction and many more algorithms.
- **Semi-supervised learning:** It includes the best practices such as predisposition and difference hypothesis in artificial intelligence.

It has its applications in various contextual investigations such as data mining, information retrieval, content comprehension, recognition and control, search engine optimization, autonomous networks and many more. It is a branch of artificial intelligence and has widespread into technologies such as deep learning, natural language processing, computer vision, robotics, speech recognition, and others mainly for commercial use. Another significant use of machine learning is optimization of the existing algorithms where the parameters can be altered to establish the hidden relationships. Machine learning problems are but not restricted to landscapes such as pattern generation, pattern recognition, anomaly detection, prediction, speech recognition, image processing, deep learning, fraud detection, diagnosis and many others [14].

1.5 Classification and feature selection

Classification is a machine learning approach used to differentiate and categorize the objects as they are recognized. According to machine learning terminology, classification falls under supervised learning where the training data is labeled to be correctly identified. On the other hand, clustering is a technique that falls under unsupervised learning, where the training data is unlabelled, and grouping is performed based on a similarity or dissimilarity measure. Classification is the process of getting to know records by the training data to identify the data points and determine to which set of categories it belongs. The variables involved in this classification process are quantifiable properties and are known as features. The feature types can be:-

i Categorical (It contains a String representation of features.)

ii Integer-valued

iii Real-valued

In machine learning, an algorithm that implements a statistical classification problem is called a classifier. The input data is mapped to a category based on the features and observations or instances.

The dataset usually contains some features that are redundant or irrelevant for the classification process. Thus, they can be filtered out or removed after pre-processing the data without loss of information during the classification [10]. This process is called attribute or feature selection where a subset of relevant features for the classification model is selected. After performing feature selection, the model can yield higher performance metrics, reduce the generalization error, reduce the training time, reduce over-fitting and can avoid the curse of dimensionality. The algorithm performs an exhaustive search in the space to find a new feature subset and scores them based on an evaluation measure. The goal is to reduce the error rate and increase the performance of the classification system. It is computationally useful if the dataset is large containing more number of features. The feature selection algorithms are of different categories [10]:-

- **Wrapper methods:** The subset search is performed using a predictive model and computes a score to select the features. Every new subset is used in the process of training the model.

- **Filter methods:** The scoring method for a feature subset is based on a proxy measure such as mutual information, correlation coefficient, inter or intra class distance. They are useful to find the relationship between features and rank the features based on cross-validation.

- **Embedded methods:** During the model generation process, the non-zero regression coefficients are selected as part of the feature selection process. Some of the commonly used algorithms are feature elimination, LASSO, ridge regression, elastic net regularization and many more.

However, feature selection is different from feature extraction process. Feature extraction is used to obtain the features from the raw data to form a dataset of class and features.

1.5.1 Classification algorithms

Based on the ability to make predictions the machine learning algorithms are classified into various learning methods such as supervised, unsupervised and semi-supervised.

TABLE 1.5.1: Different classification algorithms based on learning methods.

SUPERVISED LEARNING	UNSUPERVISED LEARNING
Support Vector Machines	k means clustering
k-Nearest Neighbour	Hierarchical clustering
Decision Trees	Hidden Markov models
Neural Networks	Apriori algorithm for association rule
Logistic Regression	Expectation-maximization algorithm (EM)
Naive Bayes	Principal Component Analysis
Random Forest	Generative Adversarial Networks
Linear/Polynomial Regression	Singular Value Decomposition

1.5.2 Evaluation method

The system is separated into two phases. In the training phase, the training information is utilized to develop the classifier models and store the models in a database for use amid the testing stage. Each veritable user has his/her own particular classifier models and preparing features. In the testing phase, system will utilize test information which was isolated from the training information for comparison.

1.6 Motivation of this Thesis

The primary focus of this work is to develop a system using machine learning methods to authenticate the users correctly. In the new era, stronger authentication techniques are required to detect security breaches. One such approach would be to introduce a multi-layer authentication mechanism. In the past [1, 3, 13, 29, 31], mRMR feature selection with SVM classification (One vs. One approach optimized by grid search) was not performed for keystroke dynamics authentication. The model proposed in this thesis is novel and concentrates on enhancing the classification performance by applying feature selection while the prior research included only a few features.

1.7 Contributions

In our experiments, the users were allowed to input data with any Android device using the iProfile app from anywhere. The app has a virtual keypad which has the same coordinate position, location of the keys and spacing of the keys. Thus, the built classification model can be more robust and can recognize users more accurately. In this thesis, we demonstrate how random forest yields high classification accuracy for this data set.

In Chapter 2 we review some of the related works for user authentication using keystroke dynamics and machine learning, and in Chapters 3, 4, 5, 6, 7 and 8 we discuss the proposed method, results, and conclusion, respectively.

CHAPTER 2

Literature review

In this chapter, we review some of the literature about keystroke dynamics for authentication that uses various feature selection and classification algorithms.

2.1 Existing approaches for keystroke dynamics authentication

Many works have been done in this area to identify and verify the users, and increase the performance metrics.

2.1.1 Analysis of Strong Password Using Keystroke Dynamics Authentication in Touch Screen Devices

Asma Salem and Dema Zaidan published a paper which examined the use of verification and identification system for touch screen mobile devices. They built a multi layer perceptron neural network model for classification using WEKA. This paper also combines the timing and non-timing features together and conclude that non-timing features increase the security level [33]. The experiment is carried out using five users and four features are extracted from the dataset. The authors put forth the problem of using different types of keyboards and developed a virtual keyboard for data collection.

TABLE 2.1.1: KSD selected features with timing and non-timing information.

CLASSIFIER	FEATURE	NOTATION	TYPE
KSD Neural Network Model Multilayer Perceptron	Duration	DU=$\{DU_1,..,\ DU_n\}$ where each value DU_i=U_i-D_i	Timing
	Pressure	P=$\{P_1,...,P_n\}$ For n successive characters	Non-timing
	Position	L=$\{X$=$X_1,..,X_n\}$,$\{Y$=$Y_1,...,Y_n\}$	Non-timing
	Size	S=$\{S_1,...,S_n\}$	Non-timing

2.1.2 Feasibility study on authentication based keystroke dynamics over touch-screen devices

Jeanjaitrong and Bhattarakosol presented a review of the literature carried out in the keystroke dynamics over touch dynamics so far. They also outlined the process of authenticating from biometric behavior in detail. They summarized how people use mobile devices as part of their daily life and the security when compromised causes the risk of data getting stolen high [13]. The authors extracted four features such as dwell time, interval time, interval timing ratio and the distance between buttons to classify the data. The data collection process involves ten users pressing four symbols out of 16 to serve as the granted password. They also built a Bayesian Network to find the relationship between feature factors and summarizes them in the classification phase.

2.1.3 Statistical Keystroke Dynamics System on Mobile Devices for Experimental Data Collection

In 2016, Al-Obaidi conducted experiments to extract features such as pressure, finger area and sensor readings for the mobile devices since other comparative studies have extracted features based on desktop keyboards [1]. Based on these features, pressure and finger area were selected as necessary features to build the statistical distance to-median anomaly detector. The experiment was carried out on Nexus smartphones to record 56 users and 71 feature elements. The classifier is a maximum mean discrepancy (MMD) model which classifies above a fixed pass mark specific to their dataset. The author draws the comparison among two different datasets and concludes with their Equal Error Rate (EER) values.

2.1.4 Evaluation of One-Class and Two-Class Classification Algorithms on Mobile Devices

There has also been some work done by Margit Antal and Laszlo Zsolt Szabo on mobile device keystroke authentication using one class and two-class classification algorithms. They applied Bayesian networks and random forest classifiers on the data set to obtain the EER comparisons for two-class classification [3]. The one-class classification is used for verifying the user by distinguishing them from outliers, and the two class classification is used for identifying the user. The authors conclude that the best EER value is obtained using Random Forest for a data set of 42 users and 71 features and all one class classifiers are better in classifying negative class than the positive class.

2.1.5 Keystroke dynamics for authentication in smartphones

One of the other research efforts is the work done by Roh and Lee that uses one class classification techniques; they applied feature selection and classification for each users posture. The users typing patterns recorded features such as time interval, strength, position, and usage angle using smartphone sensors. Along with these features, the users posture characteristics were also collected. The postures were walk, hand, and table [29]. A test population consisting of 15 users were used for building the model with five extracted features from smartphone sensors. The authors did some pre-processing, scaling and standardization over their data which yielded good EER values. They proposed a feature extraction algorithm which includes accelerometer and gyroscope sensor to find the users keystroke pattern.

2.1.6 Factors affecting keystroke dynamics for verification data collecting and analysis

The ideas presented in the work of Dema Zaidan is to verify the users by collecting data using HTML-Javascript-self-constructed Web pages [44]. The keystroke dynamics used here involves two techniques such as authentication and verification. Mobile systems now

a days have been designed such that the touch screens can record keystroke dynamics pattern. Their dataset consists of 71 users with five essential features which were collected based on machine-dependent characteristics. The author proposes that typing complex and hard passwords require more features into consideration as compared to typing a simple password.

2.1.7 Authenticating User Using Keystroke Dynamics and Finger Pressure

In [31], P. Bhattarakosol and H. Saevanee drew attention by obtaining 99% classification accuracy. The data collection was carried out on a notebook with six female and four male users as population. The authors extract three features such as inter-key, hold time and finger pressure to build the k-NN model. The authors conclude that if all three features are interacting the accuracy obtained is 91% and if inter-key and hold time features are alone present their accuracy drops to 71%. However, the author concludes that finger pressure along contributes to the obtained high accuracy scores. The major drawback of this paper is statistical insignificance as the experiment was carried out with very few users.

2.1.8 The MOBIKEY Keystroke Dynamics Password Database

Giot present a review of the literature carried out in the keystroke dynamics so far. He also outlines the process of authenticating from biometric behavior in detail [2]. He summarizes the different types of biometric systems used for authentication such as static and dynamic. He also explains more about continuous authentication where the system understands how the user interacts with it. There are different biometric modalities such as the face, iris, hand veins, fingerprint and keystroke which act as a biometric authentication. The author puts forth the problem of cross devices which is to use the same device to input the data. Since in real time, different users can possess various devices comprising different keyboards and screen coordinates. Thus, the model must be trained in such a way to recognize the users using various computing devices.

2.1.9 Two novel biometric features for touch screen devices

In 2013, Cheng Jung Tasia conducted experiments to extract features such as pressure and timing events and concluded that classification depends on these features. He considered only pressure, size, and button press timing events as features leaving it open for future researchers to discuss other features which can be responsible for classifying with more accuracy [39]. The methodology involved in that paper consists of three different phases such as enrolment phase, classifier building phase, and an authentication phase. During the classifier building phase, the illegitimate users typing patterns are not constructed by using a statistical classifier. The keystroke dynamics-based authentication proposed increases the security by verifying based on the alphanumeric-based and personal identification number-based schemes. The data collection process involves inputting a pin and not an efficient passcode and concludes his paper by finding only Equal Error Rate (EER).

2.1.10 Keystroke dynamics as a biometric for authentication

Similar experiments were carried out by Monrose who carried research over few participants from Bell Communications Lab [22]. In order to overcome the cyber threats such as network intrusion, malicious attack, and many others, dynamic biometric techniques were introduced based on the typing pattern of the user. The feature extraction process used is factor analysis which forms a lower dimensional representation among features based on correlation and dependence. The feature subset consists of class instances with similar and dissimilar user typing patterns. He depicts the covariance matrices for various features and performs classification using k-NN (Nearest Neighbor) classifier. He concludes with the applications of keystroke dynamics which can be combined with any system to form its security layer.

2.1.11 Authenticating mobile phone users using keystroke analysis

There has also been some work done by N. L. Clarke on mobile user authentication by using keystroke dynamics. He applied neural network classifiers on the data set to obtain the EER comparisons [7]. Mobile phones have intervened into our life so much and involves two

important handset interactions such as entering phone numbers and typing text messages. The author aims at maximizing the security while trying to authenticate the user during the handset interactions. The neural network layer utilizes a bac*k*-propagation perceptron algorithm to train the classifier and concludes that the performance of the classifier depends on usage characteristics of the mobile user.

2.1.12 Greyc keystroke

One of the many research efforts is the work done by El-Abed and Rosenberger using support vector machine techniques, although he applied some conditions during the enrolment of the users. The users are restricted to five captures during the data collection process, and he concludes that these operational conditions cause the classifier to outperform. The test population used the password greyc laboratory for a reason as it was lengthier. The experimental results are EER and gain values [9]. One shortcoming of this paper is that password can be easily guessed and does not meet the standards of the universal password policies.

2.1.13 User authentication through typing biometrics features

The ideas presented in the work of C. F. Araujo and H. R. Sucupira are to generate only timing latency features and reduce the false rejection and false acceptance rate. They proposed an adaptive mechanism to create a new template by adding the new samples and ignoring the old ones [4]. This leads to the modification of standard deviation and thresholds for each feature and contributes to the concept of two-trial authentication. The biometric system records key up, down, and ASCII codes as part of keystroke capturing when a user is typing on the screen. The existing password authentication mechanism is improved with the help of four major features and used when the password is not a secret.

2.2 Inspiration from the Previous Works

The main inspiration from previous works comes from the size of the datasets and the number of features that they used in their experiment. The previously mentioned works are primarily in light of user eccentric methods for connection with input devices and do not consider any other features, insight, or interest of the users. In any case, research uncovered that human practices that are firmly ruled by user's aptitude, knowledge, and interest show solid individual characteristics too. Thus, we decided to propose a method to deal with this problem, such that identification of users can be performed effectively by using large number of features obtained from much larger datasets.

In Chapter 3, we review the proposed method to form a robust authentication system using keystroke dynamics and machine learning.

CHAPTER 3

Proposed method

Machine learning is a field of artificial intelligence which focuses on training the algorithms to learn and make predictions from the data. The steps involved to solve machine learning problems are defined by:-

1. Define a problem
2. Prepare the data
3. Evaluate the algorithms
4. Improve the results
5. Present and Interpret the results

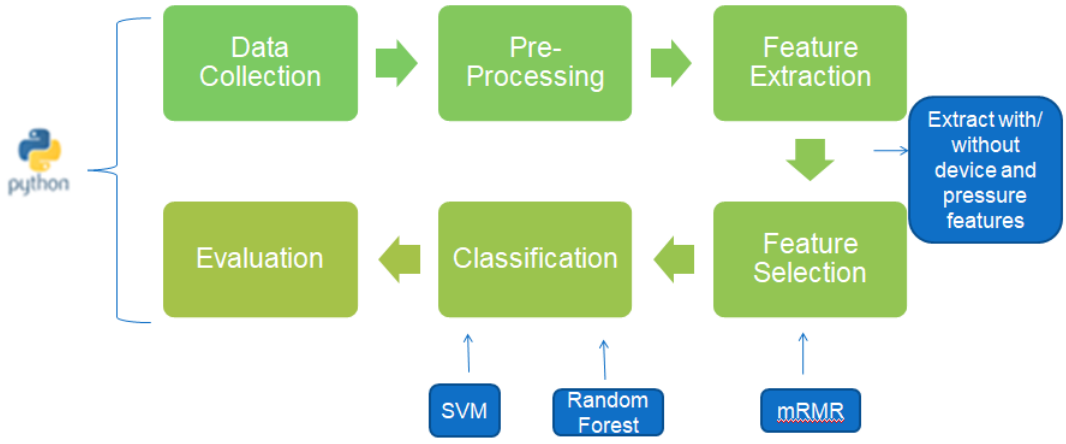

FIGURE 3.0.1: The proposed method for user behavioural biometric authentication.

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 3.0.1 shows the stages involved in the method are as follows:-

- Data Collection
- Pre-processing
- Feature extraction
- Feature selection
- Classification
- Evaluation

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, the feature selection algorithm that is used is mRMR and the classifiers used are SVM and random forest. 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 algorithm.

3.1 Data Collection

3.1.1 iProfile

iProfile is a Android application to collect keystroke events from Android devices. The app is available for all Android users freely and can be downloaded on any Android device. The application registers all up and down events when a user touches any key. We created our own keyboard and developed a local database that keeps a record of all events as long as the user is typing. The local database contains all the users typing and device information. In order to design the schema of the database table, a SQL script is written which contains

the attributes as column names. The table has all the attribute information as not null constraints and an auto-increment ID as the primary key. The application asks the user to type a specific passcode. Once the user is done, the application checks the passcode to confirm that it is correct. Once the confirmation is done, the application sends all the events information along with the unique user ID to a cloud database server. The application allows users to change their user name, the format of the JSON (Javascript Object Notation) object that is sent to the server and the server URI link. The JSON objects are easy to store and parse since they are lightweight data-interchangeable and their format is compatible with many languages. The JSON object which is stored in the cloud server is then sent to a PHP program which parses the object and stores the data in the local database. The PHP program contains the attribute information that is present in the JSON object and decodes them accordingly. After decoding the required data, the program inserts the records into the database table.

3.1.2 Process involved

The participants in this study were selected by sending invitation emails to different user groups to participate. The participants must be between 18 and 65 years old. The participants can take part in the data collection process only if they possess an Android device. The interested participants volunteered to download and install the application from Google Play store. While downloading the app from Play store, the app will ask the user's consent, and once he or she agrees, the app will be installed on their device. After installing the app and as soon as they provide their consent to participate in the input process, they will be redirected to the data collection screen. This process of data collection was carried out for five days. Each day, the user must type the passcode which appears on the iProfile app six times. Over five days, 30 passcode-entries will be collected by the application for each user. The data input process requires less than six minutes to enter the passcode six times per day. This version of the iProfile app uses the passcode ".tie5Roanl" because it combines capital, small letters and numbers. It also forces the user to navigate through different keyboard layouts by using the shift key or switching to the numbers-keyboard and vise versa. The data such as how the users access their screen is collected and sent to

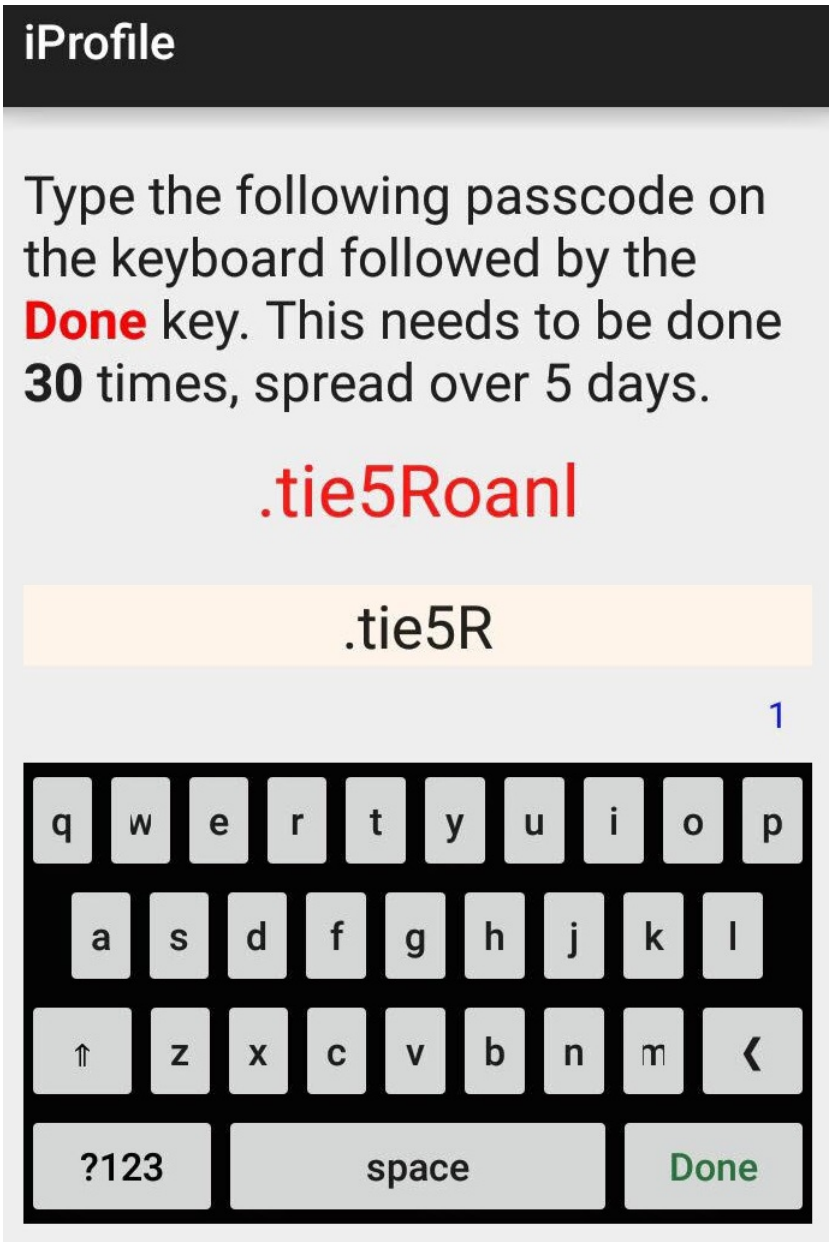

FIGURE 3.1.1: Screenshot of the iProfile app, which allows the users to type the passcode via the virtual keypad.

one or more servers; it involves *X* and *Y* coordinates of the touch location, touch pressure, touch speed, touch type (up or down), touch time stamp and other time or location data will be recorded. The location data collected will be dropped while storing it in the local database. The research is voluntary and if the participant refuses to take part of the study in the middle of the survey can exit or uninstall the app without any data being collected or stored. The entire data collection process took about four to six weeks. Even after the user completes entering the passcode for 30 times over five days, the user can continue inputting data as much as he or she likes. Participants are free to decline to answer any particular question if they do not wish to answer it for any reason and their responses are anonymous. The user's personal details and sensitive information are never collected from their device. User's responses helped us in the data collection process and to improve our research results, and hence enhanced the mobile authentication system with direct a benefit to the society.

Figure 3.1.1 shows the human-computer interface in the iProfile app which is used to collect data from the users. It indicates how the passcode is being typed; the number 1 in the figure denotes the number of completed strokes. It has a counter to keep track of the completed strokes for the user and a virtual keypad so that the users can enter the passcode with a common keypad. The password is .tie5Roanl which keeps repeating after submitting an individual stroke by pressing the Done key. As the user is typing a string, the key up, down and other attributes are captured to form a static authentication system.

3.2 The Dataset

We describe the dataset used in our experiments. The dataset lists values for all the attributes to represent the behaviour of a user. The data stored in a local database table is then exported as an Excel file. The database used is Microsoft SQL and has an option to export the entire table as an Excel sheet. The Excel file contains all the instances for every user who took part in the data collection process. It is essential to enroll the users and collect many samples from them to fit the classifier and predict it using the test data. The number of instances for every user is different since some users did not input data during

the entire data collection process. Even if the user presses a single button on the touch-screen, it creates an entry in the database, and he or she is recorded as a user. If the number of instances/samples are very few for a user, it will automatically be ignored during the pre-processing process.

TABLE 3.2.1: Dataset description that depicts the information of the collected data.

Dataset Description	Values
Number of users	94
Number of valid users	77
Number of attributes	24
Number of extracted features	155
Password	.tie5Roanl
Keys	. t i e [123] 5 [abc] [Shift] R [Shift] o a n l
Number of samples per user	40-700
Number of valid instances	30
Number of sessions	at least five per user
Devices	53

The dataset has 94 users who participated in the data collection process; out of which, only 77 users are valid after performing pre-processing. The users with instances less than thirty are dropped and not included in the experiment. The raw data has 24 attributes which are helpful to identify a user. After the feature extraction process, 155 features are obtained. The instances are recorded while typing the password that appears on the screen. The user has to open the app and enter the password every day. The sessions can be any number of times, but it is essential that the user logs in at least five times to enter the password spread over five days.

The Excel file contains the dataset for all users with attributes such as ID, Unique User Identification (UUID), language, hardware model, SDK version, manufacture, screen size, time zone, date time, country code, number of CPU cores, country location, location latitude, location longitude, button, touch pressure, touch size, X coordinate, Y coordinate, X precision, Y precision, action type, action time stamp and HR time stamp. The dataset is split into two images as illustrated in the Figures 3.2.1 and 3.2.2 to represent all of the attributes.

The user data collected involves the following:-

ID	UUID	Language	Hardware_Model	SDK_Version	Manufacture	Screen_Size	Time_Zone	Date_Time	Country_Code	Num_of_CPU_Cores	Location
145394	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145395	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145396	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145397	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145398	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145399	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145400	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145401	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145402	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145403	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145404	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145405	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145406	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145407	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145408	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145409	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145410	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145411	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145412	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145413	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145414	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145415	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145416	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145417	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145418	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145419	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145420	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145421	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145422	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN
145423	FFKQX1500469787961	English	Motorola XT1068	21	motorola	4.691011	America/Toronto	492274995	GB	4	HIDDEN

FIGURE 3.2.1: The dataset showing first half of the collected data for a single user.

- touch pressure is defined as how hard the user presses the button
- touch size is the size of the touching key
- X and Y coordinates describes the position with respect to X and Y axes
- X and Y precision describes the stroke corresponding to the screen size
- button pressed denotes which key a user is pressing while entering the password
- location data represents the latitude and longitude information collected
- device information contains hardware model, operating system version, manufacturer details, time zone of the user, screen size, language which the user selected, and the country code

Location_lat	Location_long	Button	Touch_Pressure	Touch_Size	X_Coordinate	Y_Coordinate	X_Precision	Y_Precision	Action_Type	Action_Timestamp	HR_Timestamp
0	0	NUMBERS	0.1843137	0.1843137	102	1063	1	1	Down	492274995	7/19/2017 9:09
0	0	LETTERS	0.1843137	0.1843137	102	1063	1	1	Up	492275091	7/19/2017 9:09
0	0	.	0.1843137	0.1843137	173	962	1	1	Down	492275892	7/19/2017 9:09
0	0	.	0.1843137	0.1843137	173	962	1	1	Up	492275991	7/19/2017 9:09
0	0	LETTERS	0.1843137	0.1843137	100	1060	1	1	Down	492276568	7/19/2017 9:09
0	0	NUMBERS	0.1843137	0.1843137	100	1060	1	1	Up	492276658	7/19/2017 9:09
0	0	t	0.1921569	0.1921569	342	768	1	1	Down	492277151	7/19/2017 9:09
0	0	t	0.1960784	0.1960784	339.6109	765.6108	1	1	Down	492277151	7/19/2017 9:09
0	0	t	0.1960784	0.1960784	337.0325	766	1	1	Down	492277151	7/19/2017 9:09
0	0	t	0.1843137	0.1843137	336	766	1	1	Down	492277151	7/19/2017 9:09
0	0	t	0.1843137	0.1843137	336	766	1	1	Up	492277234	7/19/2017 9:09
0	0	i	0.1882353	0.1882353	525	765	1	1	Down	492278278	7/19/2017 9:09
0	0	i	0.1882353	0.1882353	525	765	1	1	Up	492278358	7/19/2017 9:09
0	0	e	0.1803922	0.1803922	217	771	1	1	Down	492279001	7/19/2017 9:09
0	0	e	0.1803922	0.1803922	217	771	1	1	Up	492279084	7/19/2017 9:09
0	0	NUMBERS	0.1803922	0.1803922	90	1079	1	1	Down	492279986	7/19/2017 9:09
0	0	LETTERS	0.1803922	0.1803922	90	1079	1	1	Up	492280075	7/19/2017 9:09
0	0	5	0.1882353	0.1882353	339	767	1	1	Down	492280618	7/19/2017 9:09
0	0	5	0.1882353	0.1882353	339	767	1	1	Up	492280692	7/19/2017 9:09
0	0	LETTERS	0.1803922	0.1803922	95	1079	1	1	Down	492281911	7/19/2017 9:09
0	0	NUMBERS	0.1803922	0.1803922	95	1079	1	1	Up	492281984	7/19/2017 9:09
0	0	r	0.172549	0.172549	270	764	1	1	Down	492283870	7/19/2017 9:09
0	0	r	0.172549	0.172549	270	764	1	1	Up	492283892	7/19/2017 9:09
0	0	DELETE	0.1803922	0.1803922	642	959	1	1	Down	492284879	7/19/2017 9:09
0	0	DELETE	0.1843137	0.1843137	639.493	956.493	1	1	Down	492284879	7/19/2017 9:09
0	0	DELETE	0.1843137	0.1843137	635.3051	956	1	1	Down	492284879	7/19/2017 9:09
0	0	DELETE	0.1803922	0.1803922	634	956	1	1	Down	492284879	7/19/2017 9:09
0	0	DELETE	0.1803922	0.1803922	634	956	1	1	Up	492284967	7/19/2017 9:09
0	0	SHIFT	0.1882353	0.1882353	82	953	1	1	Down	492285509	7/19/2017 9:09
0	0	SHIFT	0.1843137	0.1843137	78.79242	953	1	1	Down	492285509	7/19/2017 9:09

FIGURE 3.2.2: The dataset showing the last half of the collected data for a single user.

- action type denotes where it is an up/down event, and records action and HR timestamps
- The data stored on the server has an encrypted user id to make the username anonymous.

The device attributes contain various device details since the users used their device to input the data during the data collection process. The device attributes contain 53 devices which were used by different users all over the world. Figure 3.2.3 shows some of the devices used in the experiments and the distinctive count for every user. We can also infer that three users used Lava V2s and Lenovo A7 devices to take in the data collection process.

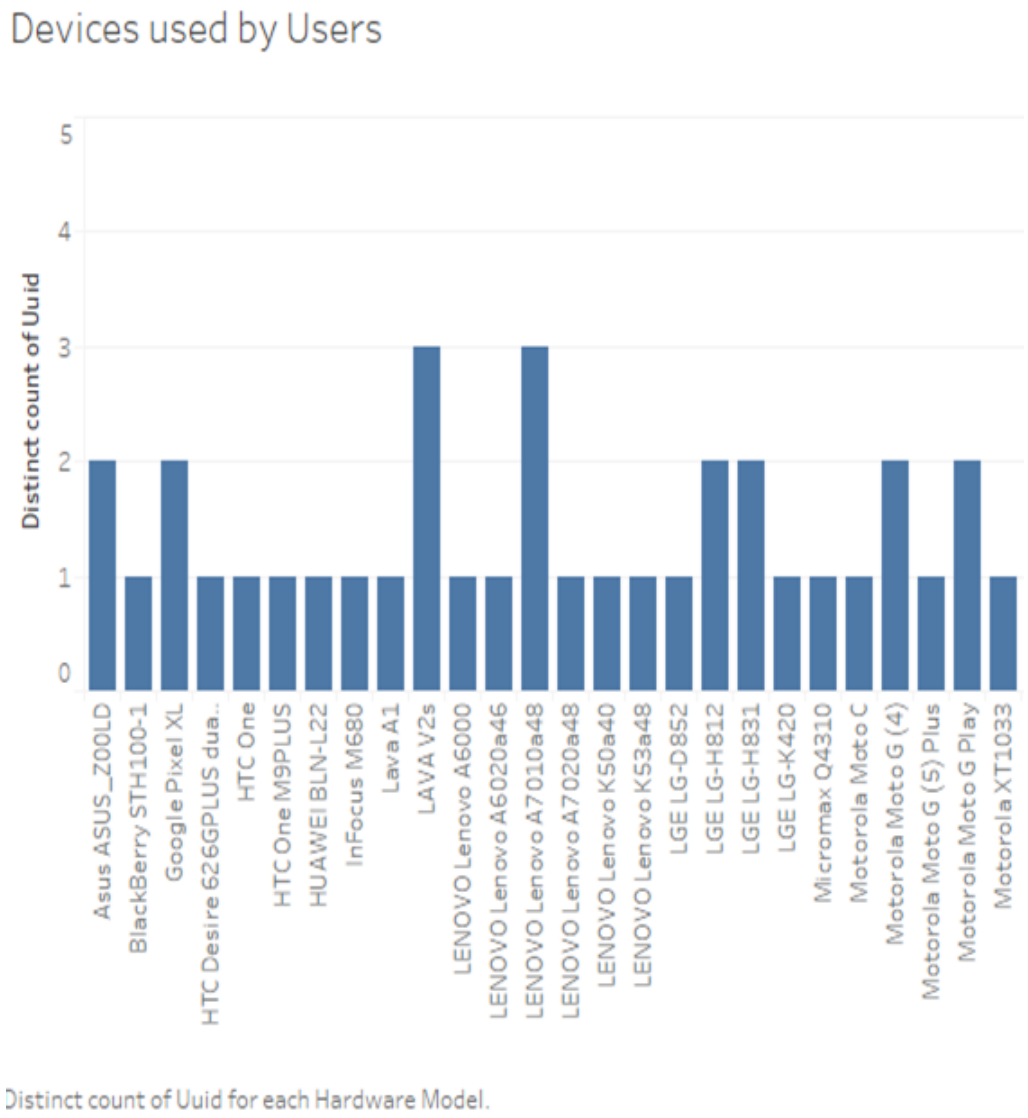

FIGURE 3.2.3: Various devices used in the data collection process.

3.3 Feature Extraction

To represent the data well, the data needs to be transformed to form the features. The raw data contains the attribute information out of which the features have to be formed to represent the distinctive properties of the input patterns. It derives new features to distinguish the differences among the users and to increase the classification metrics. The mode to determine the features depends on the dataset and problem.

3.3.1 Pre-processing

The raw data needs to be filtered, scaled and normalized before passing it into the classifier. Pre-processing makes the data more understandable. The pre-processing script is run on the entire raw data to extract the required features. The pre-processing steps are as follows:

i The entire data is loaded and sorted based on the unique User IDs (UUID) and action timestamps after removing the duplicate timestamps from the same user.

ii Exclude the instances below a certain threshold if the frequency count is less than 2 attempts (32 counts are 1 attempt).

iii Delete the location data, such as latitude and longitude information if present. The features are extracted based on the derivatives and averages of the raw data.

iv Find the records for touch events based on Button pressed and Action Type inclusive of Backspace touch events. While entering the passcode, the user tends to make mistakes. If the mistake is repeated mostly for a user, then it is the typing pattern for that user. The mistake is taken into consideration while calculating the correct touch event instances for a user. After all these filters, 77 users were obtained from a refined set of 94 users.

v Generate the features from the raw data attributes for every button pressed and action type events. The features for X and Y coordinates are formed from the distance formula 1 as it corresponds to the distance from the screen for all keystrokes of letters. The features from X and Y precisions are added to generate new features for every letter in the passcode. The timestamps for the letters are formed from the derivatives. The touch pressure and size of every letter corresponds to an individual feature.

Pre-processing the raw data generated 155 features for 77 users.

As illustrated in Figure 3.3.1, the first row contians the device-specific features. Attributes such as pressure and size are expanded to form 16 features each for various Down action type (press) events.

- The features starting with 'p' denote the touch pressure of the user while typing the passcode.

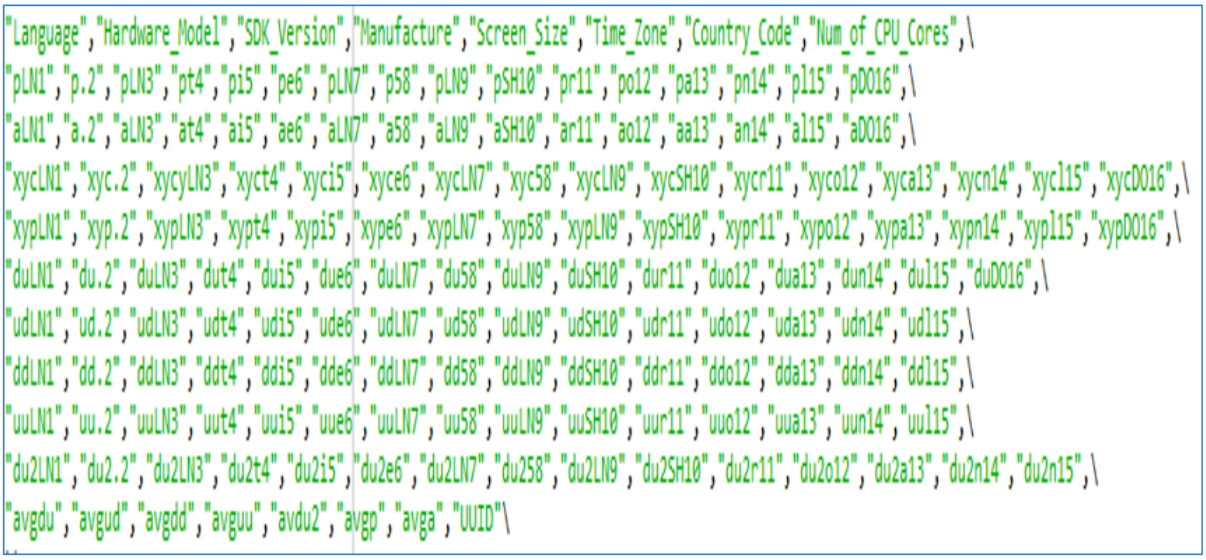

FIGURE 3.3.1: Extracted features from the raw data after pre-processing step.

- The features starting with 'a' denote the touch area of the device.
- The features starting with 'xyc' denote the distance between X and Y coordinates for various button press and Down action type events (16):

$$\sqrt{X^2_{coordinate} + Y^2_{coordinate}}, \tag{1}$$

where *Y* is the target.

- The features starting with 'xyp' define the addition of *X* and *Y* precision values for various button press and Down action type events (16):

$$XPrecision_{Down} + YPrecision_{Down}, \tag{2}$$

where *Y* is the target.

- The timestamp differences for the button presses and various combinations of action type events form features starting with du(16), ud(15), dd(15), uu(15) and du2 (15):

$$ActionTimestamp_{Down} - ActionTimestamp_{Up}, \tag{3}$$

where Y is the target.

- The features starting with 'du' represent the dwell time for down up movements, starting with 'ud' denote the flight time for up down movements, starting with 'dd' denote the flight time for down down movements, starting with 'uu' denote the flight time for up up movements, and starting with 'du2' denote the flight time for di-graph down up movements by a user.

- Finally, the average of down/up, up/down, down/down, up/up movements, touch pressure, touch size, features forms a set of new features for the corresponding button events.

- The class is the UUID which is encrypted to make the username anonymous.

Table 3.3.1 shows the different feature types, definition, and their number present in the dataset after pre-processing. From the raw data, 155 features are formed through computation of the attributes. The keystroke measure includes digraph which denotes consecutive key types and the latency which denotes the time interval between two key types [15]. The action type denotes whether it is an Up or Down press [13, 2]. This leads to various latencies such as:

- Down-Up (du): time interval between press and release of a key

- Up-Down (ud): time interval between release and press of a key

- Down-Down (dd): time interval between presses of two consecutive keys

- Up-Up (uu): time interval between releases of two consecutive keys

The features are formed based on the passcode button presses and the action type associated with it for these various latencies, touch pressure, size, X and Y coordinates and action timestamps.

Thus, the features were formed to represent their behaviour with utmost detail.

TABLE 3.3.1: 155 features extracted from raw data based on touch events for 77 users.

N	Feature-Type	Definition	Number of features
1	Down-up	Dwell time	16
2	Up-down	Flight time	15
3	Down-Down	Flight time	15
4	Up-Up	Flight time	15
5	Down-up (2-graph)	Flight time	15
6	Pressure	Touch pressure	16
7	Size	Touch area	16
8	X-Y P	X-Y precision	16
9	X-Y C	X-Y coordinates	16
10	Averages for timing features	Average UD, DD, DU, UU, 2-graph	5
11	Average pressure	Average pressure for all keys	1
12	Average size	Average area for all keys	1
13	Device specific features	Screen size, hardware information	8
Total			**155**

3.4 Feature Selection

Feature selection or variable selection or attribute selection is used to select the most useful or relevant features which can be used to build the classification model. By ignoring irrelevant features, the data is reduced, thus, reducing the runtime to run the model. It also increases the performance metrics by running the classifier on the essential features. Feature selection may help boost the performance and may aim to reduce the classification errors. It selects a portion of the extracted features to apply the classification algorithm. There are three major types of feature selection algorithms:-

- Filter methods
- Wrapper methods
- Embedded methods

3.4.1 Minimum Redundancy Maximum Relevance approach

The Minimum Redundancy Maximum Relevance (mRMR) approach selects the features that correlate very strongly with the classification variable. It can use sequential forward, backward, and floating selections to select a subset features. The best features to classify

efficiently are selected based on mutual information with the criteria of min-redundancy, and max-relevance among the features by searching through the subspace of the extracted features. This is a wrapper-based feature selection algorithm to maximize the conditional likelihood of the iterations based on the testing/validation accuracy [26].

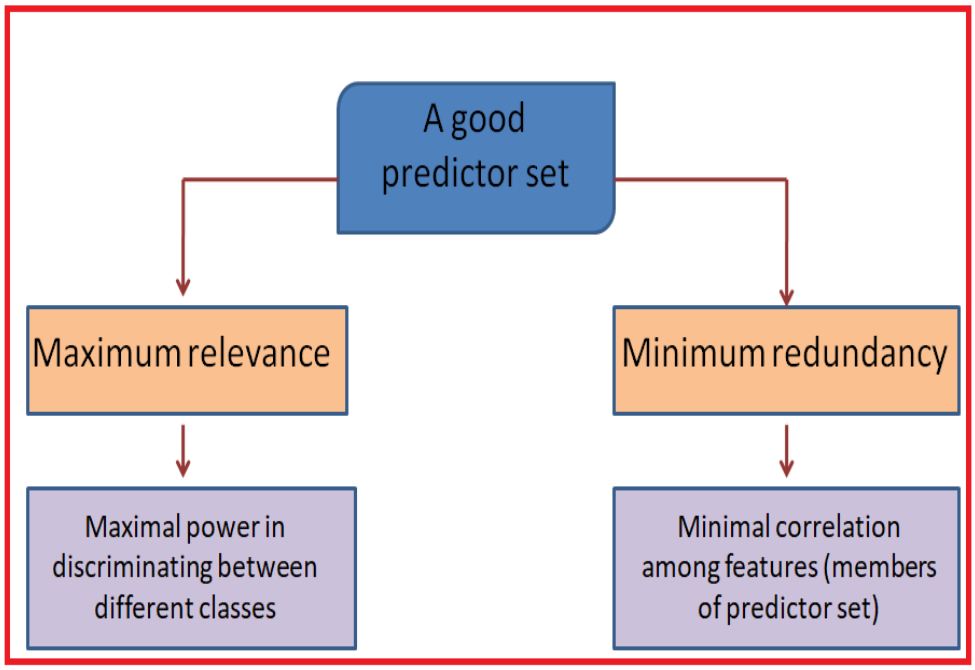

FIGURE 3.4.1: Description of the mRMR feature selection algorithm.

We also show that mRMR used as a wrapper approach produces a compact subset of features from the large number of features which mainly contributes to better classification at a lower computational expense. The feature set can be optimized by first picking the best features and then building the classifier to use them. In mRMR, the forward feature selection step, which is a sequential search method is used to consider the features one by one for addition or removal to the optimized feature set. The selection heuristic approach in this mRMR algorithm follows a greedy iterative maximization technique, where it does not add another feature if the mutual information is zero but adds features that produce large conditional likelihood. The mutual information for two variables of the data set is given by:

$$I(x,y) = \int\int p(x,y) log \frac{p(x,y)}{p(x)p(y)dxdy}, \qquad (4)$$

where x and y are discrete variables leading to the entropy and the mutual dependence

between them. The features that have the most significant relevance have the most substantial mutual information and denotes that this feature depends more on the target class; However, the features that are dependent on each other and are redundant are not the best features that contribute to the classification task.

Table 3.4.1 describes the data set in terms of instances and features after the feature extraction process. The total number of users after pre-processing are 77. These are the users who typed every passcode correctly and submitted it after completing a single passcode in the Android interface. Every user has 10 correct instances, which leads to 770 instances that are used for classification. The pre-processing script forms the features that contribute to 155 features. Among these, 146 are numerical features, and the rest 9 are categorical features. The categorical features contain string representations of the input data. These categorical features are converted into numerical features by using label encoding approach. All the features are then normalized and scaled before the wrapper based model selects a good number of features. Using mRMR, 36 features are selected to yield better classification results.

TABLE 3.4.1: Dataset description in terms of features and instances.

Total Instances	Number of Features	Numerical Features	Categorical Features	Selected Features
770	155	146	9	36

The 36 selected features include avga, avgp, al15,a58, aSH10, aDO16, aLN1, aLN9, xyc.2, xycLN1, xycr11, xycn14, xycyLN3, xycl15, xycLN9, xycDO16, xyca13, xypLN1, xypLN3, xypLN9, xypSH10, xyp58, xypl15, ud.2, uuLN1, udLN3, duLN1, du2LN1, ddSH10, ddLN1, Hardware model, manufacture, timezone, country code, number of CPU cores and screen size. These are the features that are responsible for classifying and authenticating the users. We noticed that features such as average pressure, average size, device specific features, latency features such as du, ud, dd, uu, du2, pressure, and size for particular button presses play a vital role to classify the users and to reduce the false acception or false rejection rate. These selected features give high classification performance when compared to classification on the entire data set.

3.5 Methods

In this chapter, the classification algorithms that were applied on different experiments are explained. The classifiers discussed in this chapter are SVM-linear, SVM-RBF (grid search optimization), and random forest. The experiments conducted on the dataset are also discussed.

3.5.1 Classification

Classification is the process of approximating a mapping function to map input variables to output variables. The mapping function is called the model which predicts the class for the given set of data. The classifier can classify the data points into one or two or more classes provided the input variables are real-valued or discrete. The different types of classification problem are-

- One class classification
- Binary class classification
- Multi-class classification
- Multi-label classification

Classification algorithms have many parameters that have to be set based on the problem. Python scikit library offers the classification algorithms which can be tailored according to our problem [25]. For our dataset, we use multi-class classification approach to classify the instances using various classifiers.

The block diagram in Figure 3.5.1 depicts how a user enrolls himself through inputting data via his or her Android device. This legitimate user types the password .tie5Roanl which is considered as a strong password by database administrators. These typing events are stored as instances of the user behavior leading to data acquisition phase. The raw data needs to be pre-processed by scaling and normalizing the attributes from which the features can be extracted. The features are generated by applying computational logic from the pre-processing script on the attributes. These features have to be chosen optimally to lead to

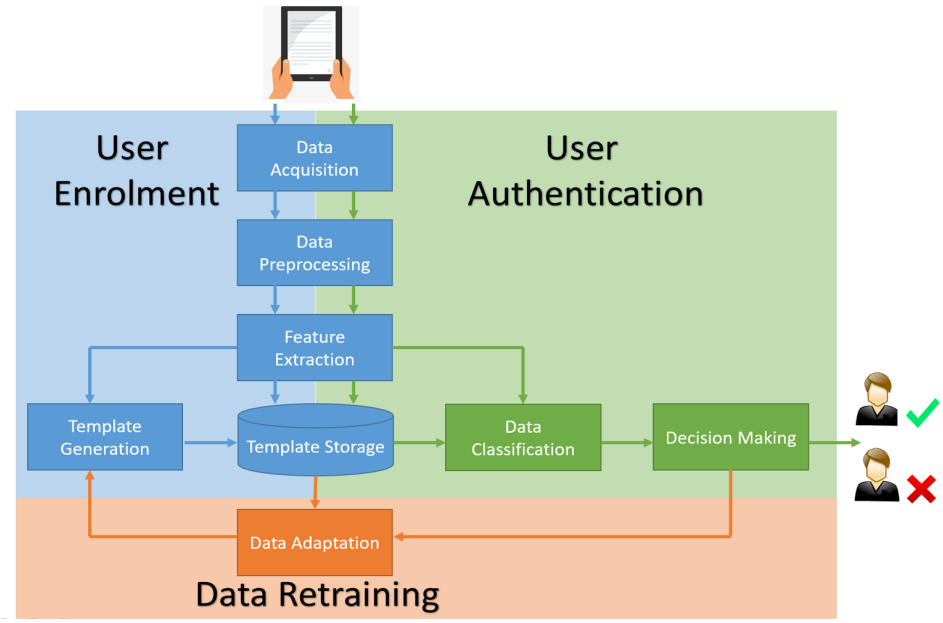

FIGURE 3.5.1: The overall classification process for user authentication based on continuous behavioral biometrics using machine learning. The block diagram explains the involved steps.

better classification; hence, the best number of features have to be selected. This is the template of data on which we need to perform classification using the various classifiers to compare the results. Classification is a decision-making process of recognizing if the user is a legitimate user or an illegitimate user.

3.5.2 Support Vector Machines

A SVM is a discriminative classifier formally defined by a separating hyperplane. The SVM maximizes the margin to separate the data serving as a maximal-margin classifier. The support vectors are the data points that lie in the classifier boundary area, and that the margin pushes up against. In other words, given labeled data (supervised learning), the algorithm outputs an optimal hyperplane that categorizes new samples.

Figure 3.5.2 shows how the hyperplane separates the datapoints of the two classes which are represented by circles and triangles. A kernel is a function to measure the similarity between the data points. The choice of the kernel depends on the problem and is data

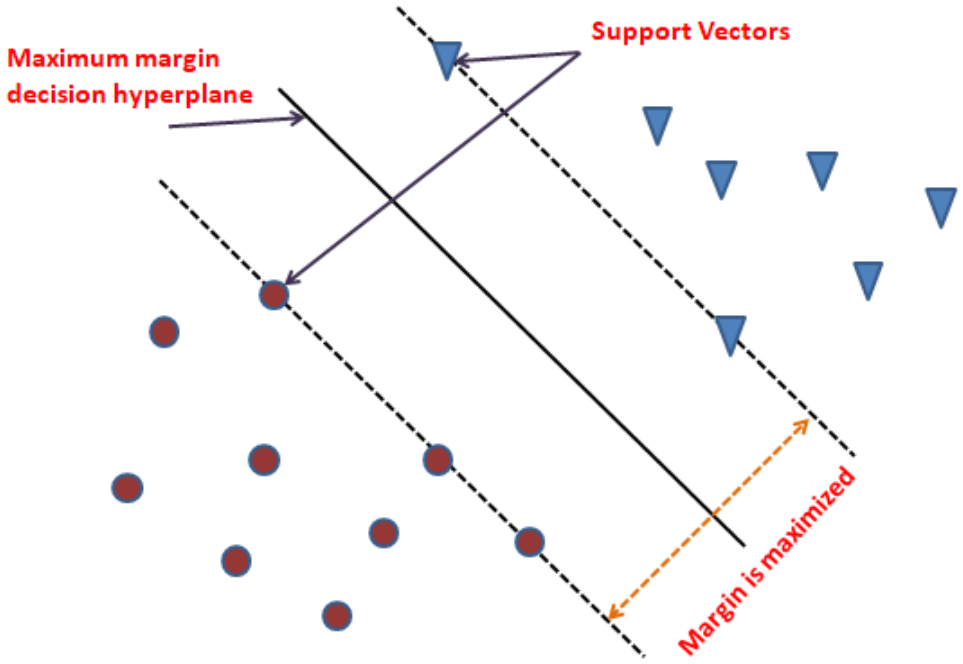

FIGURE 3.5.2: Support vector machine classifier showing the separating hyperplane.

dependent. If the data is not linearly separable, kernels can be used to map the samples to higher dimensional space.

In machine learning, SVMs are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training samples, each marked belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic linear classifier. When data is not labeled, supervised learning is not possible, and an unsupervised learning is required, which would find natural clustering of the data to groups, and map new data to these formed groups.

3.5.2.1 Multi-class One Versus One Approach

The multi-class classification using SVM with one vs. one decision function enlarges the feature space to make the separation between classes possible. Combining several binary classifiers generally form the multi-class classifier. The different users who are present after the pre-processing stage represent the various classes, thus leading to a multi-class problem.

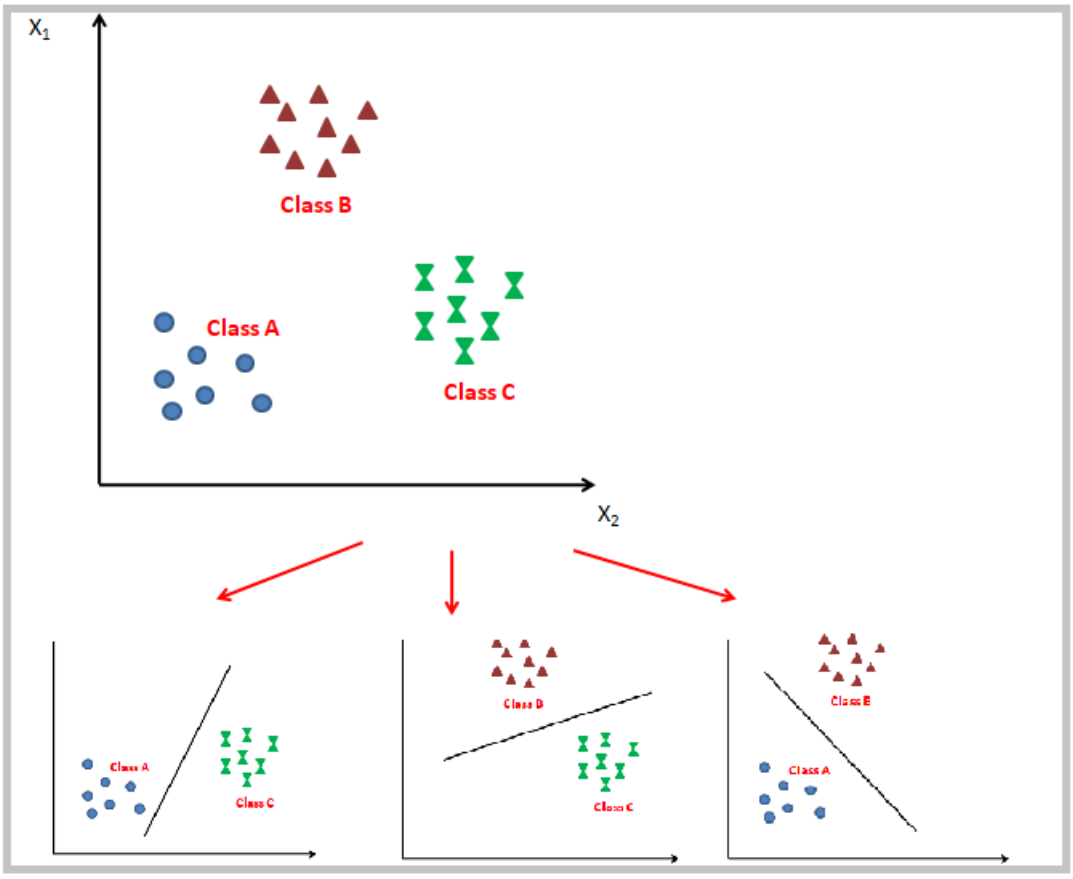

FIGURE 3.5.3: Multi-class classification using one vs one approach representation containing three classes

Figure 3.5.3, there are three classes which are shown in two dimensional space. It is treated like a binary class problem using one vs one approach to solve the multi-class problem.

The one versus one approach fits all classes against one and another through pairwise classification, to classify the class that wins the most pairwise competitions. The one vs. one decision shape is more suitable for practical use. This approach creates *k(k-1)/2* classifiers which take data from i^{th} and j^{th} class and is trained on them. For every split, if the data point *a* is in i^{th} class then it obtains a vote otherwise j^{th} is incremented [11]. The data point *a* is then predicted based on the number of votes. In case the data point ends up with the same number of votes, a tie-breaking strategy can be incorporated.

3.5.2.2 SVM Linear

The Linear SVM is a simple kernel with the maximum margin linear classifier to define the maximum boundary before hitting the data point. The linear SVM can be represented as follows:

$$K(a, b) = a^t b, \tag{5}$$

which is the inner product between a and b vectors. The linear kernel finds a plane that passes through the origin and separates the classes in the feature space. It predicts the data point based on the classifier boundary; if the data point falls on the plus-plane side of the decision boundary, then it classifies it as a positive class or vice versa. In some cases, to increase the accuracy and to obtain higher classification results, the linear kernel may not work regardless of the cost parameter. Therefore, Radial Basis Kernel(RBF) can be used with best cost and gamma values to classify better, where the parameters can be obtained using Grid Search.

3.5.2.3 SVM RBF

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick. SVM has many kernels, and one such is polynomial. However, RBF outperforms all of them since it is a squared exponential kernel, which defines the function space a lot larger. RBF maps the feature space implicitly to a very high dimension by controlling the variance [11]. The RBF SVM classifier is given as follows:

$$K(a, b) = exp(-\frac{\|a-b\|^2}{2\sigma^2}), \tag{6}$$

where the numerator is the diameter of the smallest sphere which encloses the high-dimensional feature vectors, and the denominator refers to the margin the SVM chooses. For choosing the σ parameter, Structural Risk Maximization (SRM) can be used. However, in our experiments, cross-validation and grid search optimization are used. Radial transformation results in far more appropriate decision boundary and rule and solves the convex quadratic

optimization problem. The SVM RBF is much better in unpredictable situations and relies on the *gamma* value while tuning for optimization.

3.5.2.4 Grid Search

Grid search is used for tuning and optimizing the parameters of the SVM kernel by finding the parameters including Cost(C), gamma(σ) and degree. The search is performed on three or higher dimensional space after the mapping and entirely depends on the data set. The range for gamma and cost values are given as follows:

$$\sigma \in 0.01 to 2^8 \qquad Cost \in 1 to 10^8 \qquad (7)$$

For the linear kernel, we evaluate Cost to find the optimized metrics, However for RBF kernel, we evaluate Cost and gamma to find the optimized metrics.

Algorithm 1 Classification of users using SVM kernels with grid search after feature selection

Input A labeled set D_i, of *m* number of features:
D_i =(x_i,y_i), *i* = 1, 2,...,*m*
Output classification metrics of linear and RBF SVM

1: S_i = Encode the categorical values and scale the data
2: **for** $i = 3$ to m **do**
3: *classify*:
4: **for** each Stratified split obtain train and test , *j*=1 To 5 **do**
5: $N \leftarrow$ MRMR(S_{train},S_{test},*i*) ▷ Selecting features
6: Train the **SVM linear** classifier with *one Vs one* decision function shape
7: Validate on test and calculate performance metrics
8: Select *N* features with highest classification performance metrics
9: Perform **Grid Search** to obtain best *Cost* and *gamma* parameters for RBF SVM
10: **goto** *classify*.
11: Repeat the inside for loop and classify S_N using **RBF SVM** classifier *(ovo)* with *grid search* parameters and *N* features

Algorithm 1 depicts how the users are classified using SVM kernels such as linear and RBF. The extracted features contains UUID and it serves as the class attribute. The

extracted features consist of categorical values which need to be encoded to replace all the string values to numerical values. On the other hand, the numerical data needs to be normalized and scaled to [0,1], to make the convergence faster and to use a distance calculation function between points uniformly. The features are split into train and test samples. The training model may over-fit the data and hence five-fold stratified cross-validation is applied to obtain decision values before minimizing the negative log likelihood. The averages from the cross-validation are taken to obtain the classification scores and choose the best number of features based on the highest score. After performing grid search, we obtain the parameters for the RBF kernel and perform the classification with the selected number of features.

3.5.3 Random Forest

Random forests is a notion of the general technique of random decision forests that are an ensemble learning method for classification, regression and other tasks, which operate by constructing a multitude of decision trees at training time and outputting the class that is the model of the classes (classification) or mean prediction (regression) of the individual trees. It can also be used in unsupervised mode for assessing proximities among data points. They make the predictions by combining the prediction of the individual trees which have been constructed during the classification process [41].

Random forest is one of the bagging approaches that builds the decision trees based on bootstrap samples. The original and sub-sample space will always be the same. If the bootstrap parameter is set, then the samples are chosen at random with replacement. There is a class weight parameter that can be altered; in the experiments conducted, it is set to balanced mode since it adjusts the weights automatically. The parameters need to be controlled to reduce the size, memory, and complexity of the growing trees. If the default parameters are set, it can lead to unpruned fully grown trees especially if the dataset is large.

The random forest classifier pseudocode is explained in Algorithm 2 as follows [28]:

Algorithm 2 Pseudocode for classification of users using random forest classifier and feature selection

Training Phase
Given

- X: the objects in the training data set
- Y: the labels of the training set
- L: the number of features
- K: the number of subsets
- ω_1,ω_2,....,ω_c: the set of class labels

For i=1...L

- Randomly select k features from L features
- Split $\boldsymbol{F}$ (the feature set) into K subsets: $\boldsymbol{F}_{i,j}$ (for j=1...K) using best split point
- Let $X_{i,j}$ be the dataset X for the features in $\boldsymbol{F}_{i,j}$
- Eliminate from $X_{i,j}$ a random subset of classes
- Build classifier D_i using (X_i,Y) as the training set

Classification Phase

- Test features are used to predict the outcome of randomly created decision tree.
- Assign X to the class with the largest confidence.

3.5.4 Experiments

In this part, we curated our datasets by refining the features and performing various experiments to observe the importance of each feature in the classification process. Table 3.5.1 describes the various experiments conducted by altering the dataset instances and features, and the total number of features present in the dataset. The following experiments were conducted on the dataset.

TABLE 3.5.1: Different experiments carried out on the dataset.

Experiments	Number of features
With device-specific features with ten samples	155
With device-specific features with thirty samples	155
Without device-specific features from experiment 2	147
Without pressure-related features from experiment 3	130

3.5.4.1 Experiment 1: With device-specific features with ten samples

The dataset is extracted from the database table and processed by the pre-processing script. The Python pre-processing script filters out the instances for the users. Initially, the script sets the threshold to ten, so that ten samples are selected for every user. The experiment is carried out using SVM-linear, SVM-RBF, and random forest classifiers.

3.5.4.2 Experiment 2: With device-specific features with thirty samples

Experiment 1 is repeated by increasing the number of samples per user from ten to thirty. The pre-processing threshold parameter is set to thirty to filter out the instances which fall short of thirty. The maximum number of correct instances for each user is thirty since the users entered the passcode only thirty times during the data collection process. This experiment takes all the valid data and considers them as a valid user in the experiment. Thus, increasing the number of samples leads to better classification performance metrics and helps form a more robust secured authentication system. The dataset contains all the features with thirty instances per user. If the classifier is trained and tested with more data (instances), it performs better in real-time if new scenarios are encountered. The algorithm becomes more reliable to classify/identify the users better.

3.5.4.3 Experiment 3: Without device-specific features from experiment 2

From experiment 2, remove all the device-specific features from the dataset and carry out this experiment. To understand the importance of device information, it is essential to remove them and perform the classification process. The device features in the dataset are hardware model, SDK, number of CPU cores, manufacture, screen size, language, and country code. After removing these features from the dataset, the dataset features drops

from 155 to 147. The dataset now contains the UUID as the class and all other features except device-specific features. Train and test the classification models with this dataset. Device information is essential to add another layer of security as part of the authentication mechanism since the devices serve as tokens for each user.

3.5.4.4 Experiment 4: Without pressure-related features from experiment 3

From the experiment 3, remove all the pressure-related features for all the key up/down and button events from the dataset and carry out this experiment. It is essential to perform this experiment to know the importance of touch pressure in the identification process of the users. The dataset now contains the UUID as the class and all other features except pressure-related and device-specific features. Train and test the classification models with this dataset. Touch pressure is also one of the important features in the dataset since it is beneficial to identify the users operating the touchscreen devices.

CHAPTER 4

Results

For classification purposes, different classifiers such as Random Forest and SVM with two different kernels, Linear and RBF have been used. As performance metrics, classification accuracy and F1 score were used. The SVM classifier uses multi-class classification one vs. one approach to classify the users. Each user is compared to another user to determine if they are legitimate. The results are also obtained applying feature selection to remove noise by using mRMR as the ranking method and selecting "*WrapperSubsetEval*" as "*Attribute evaluator*" and "*RerankingSearch*" as its search method.

The results of classifying the original and filtered dataset using the mentioned classifiers, as well as application of feature selection are listed and discussed in this chapter. With these features as input, all algorithms are executed to analyze the accuracy and F1 score.

4.1 Ten samples with device specific features

For every user, ten samples are present to form the dataset for this experiment. The preprocessing script has a threshold parameter, which can be set to alter the number of instances. To classify the users using a multi-class classifier, the class has to be balanced with an equal number of samples. The number of features were selected to be 36 to yield better classification metrics.

4.1.1 Classification results on the original datasets using SVM linear

The original dataset containing the device specific information was classified using SVM linear classifier. The SVM uses multi-class one versus one approach to treat it as a multi-

class problem. From Figure 4.1.1, it can beinferred that the classification accuracy obtained before applying any feature selection algorithm is 96.31% by selecting 58 features.

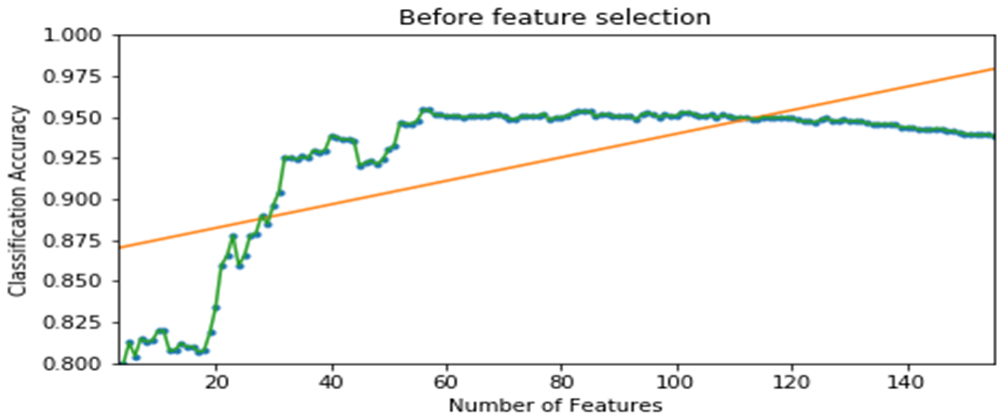

FIGURE 4.1.1: Classification accuracy obtained before feature selection on linear SVM.

From Figure 4.1.2, it can be inferred that the F1 score obtained before applying any feature selection algorithm is 95.78% by selecting 57 features.

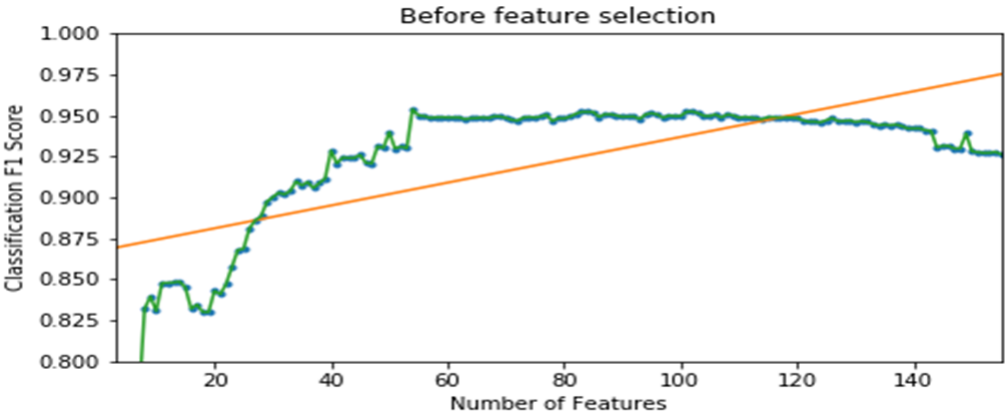

FIGURE 4.1.2: F1 score obtained before feature selection on linear SVM.

4.1.2 Classification results on datasets after mRMR feature selection using SVM linear

The dataset containing ten samples per user were classified first using SVM linear classifier. Since the mRMR feature selection algorithm is a wrapper-based approach, it is used along with the classification model for every cross-validation iteration. From Figure 4.1.3, it

can be inferred that the classification accuracy obtained after applying the mRMR feature selection algorithm is 97.27% by selecting 36 features. After applying the feature selection algorithm, the classification accuracy is increased from 96.31% to 97.27%.

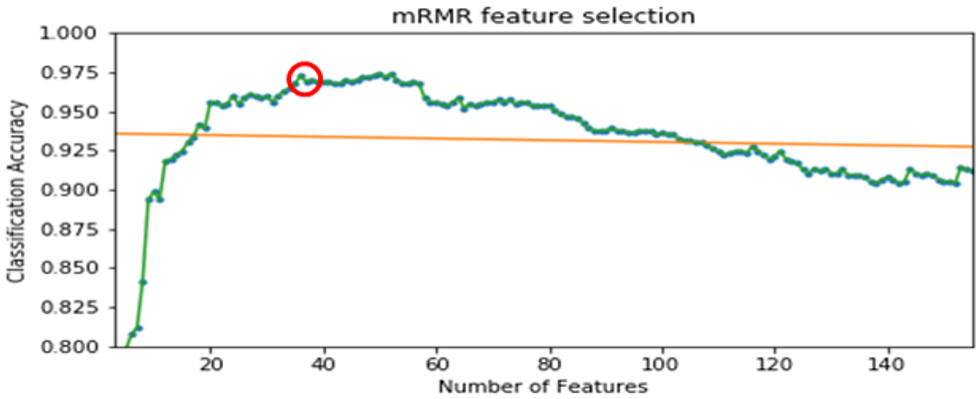

FIGURE 4.1.3: Classification accuracy using SVM linear and feature selection.

From Figure 4.1.4, it can be inferred that the F1 score obtained after applying mRMR feature selection algorithm is 96.99% by selecting 36 features. After applying feature selection, the F1 score is increased from 95.78% to 96.99%.

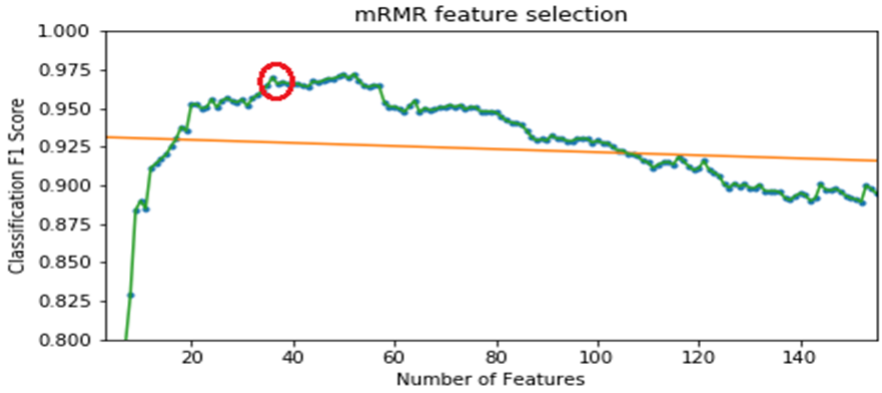

FIGURE 4.1.4: F1 score using SVM linear and feature selection.

4.1.3 Classification results on datasets using SVM RBF at N=36

After selecting a good number of features by applying mRMR feature selection algorithm, they can be selected to run the SVM RBF classifier. By selecting 36 features, the multi-

class SVM RBF classification is performed on the dataset using stratified cross-validation. A grid search must be performed on this filtered dataset with correct values of cost and gamma. To optimize the RBF kernel, the parameters need to be chosen correctly to yield the best classification results. After performing the grid search, the best classification accuracy was obtained for cost at 10^5 and gamma at 0.01. By setting these values as the SVM RBF parameters, the model was formed to perform classification. SVM RBF increases the performance metrics when compared to the SVM linear classifier. The classification accuracy obtained using RBF classifier is 97.40% and F1 score obtained is 97.01%.

4.1.4 Classification results on datasets using Random Forest at N=36

With the selected number of features by applying mRMR feature selection, the random forest classifier is run on the dataset. By selecting 33 features, the random forest classification is performed on the dataset using stratified cross-validation. Random forest increases the performance metrics when compared to the SVM RBF classifier. The classification accuracy obtained using RBF classifier is 98.44% and the F1 score obtained is 98.33%.

4.1.5 Comparison of SVM Linear, RBF and Random Forest

The comparison between the classifiers is essential to show the distribution of performance metrics as a result of carrying out the experiments. The box plot of Figures 4.1.5 and 4.1.6 displays the whiskers and quartiles of the classification metrics using this dataset. The classification metrics from different classifiers represented as a Numpy array is passed to the *x*, *y* or hue parameters of the box plot. Based on the maximum and minimum values in the 2D array, the box plot is plotted.

From Figures 4.1.5 and 4.1.6, the best classification accuracy and F1 score are obtained by using random forest classifier with 98.44% and 98.33% respectively, whereas the classification accuracy and F1 score obtained using SVM RBF classifier are 97.40% and 97.01%, and obtained using SVM linear are 97.27% and 96.99% respectively.

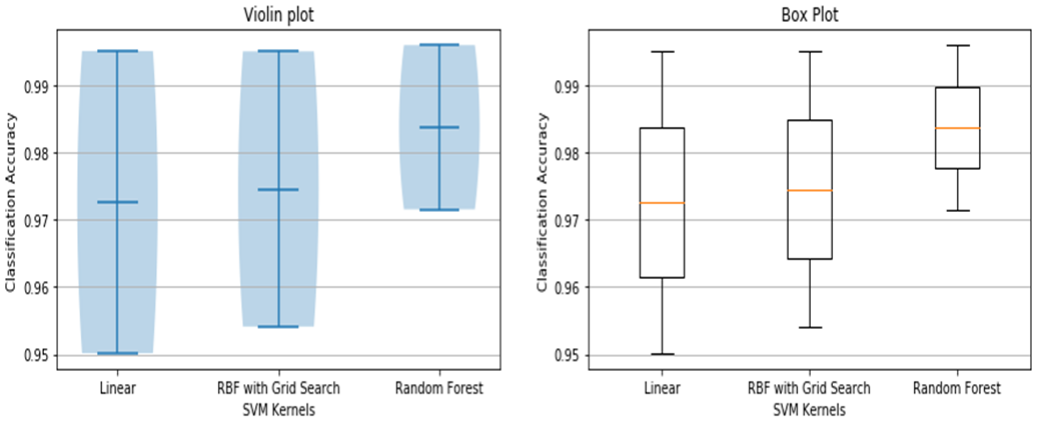

FIGURE 4.1.5: Comparison of classification accuracies for SVM linear, RBF, Random Forest.

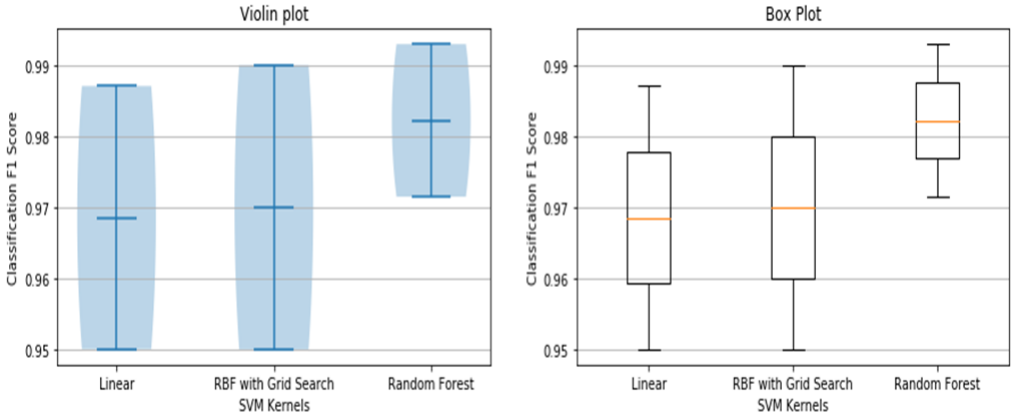

FIGURE 4.1.6: Comparison of F1 scores for SVM linear, RBF, Random Forest.

4.2 Thirty samples with device specific features

In this experiment, the number of samples was increased from ten to thirty per user. The experiment was carried for five days where the users had to input at least thirty strokes. Thus, the minimum value to be set in the threshold variable is thirty while performing pre-processing. This threshold selects all the users having at least thirty valid samples. The optimal number of features was selected to be 36 to yield better classification metrics.

4.2.1 Classification results on the original datasets using SVM linear

From Figure 4.2.1, it can be inferred that the classification accuracy obtained before applying any feature selection algorithm is 79.99% by selecting 85 features.

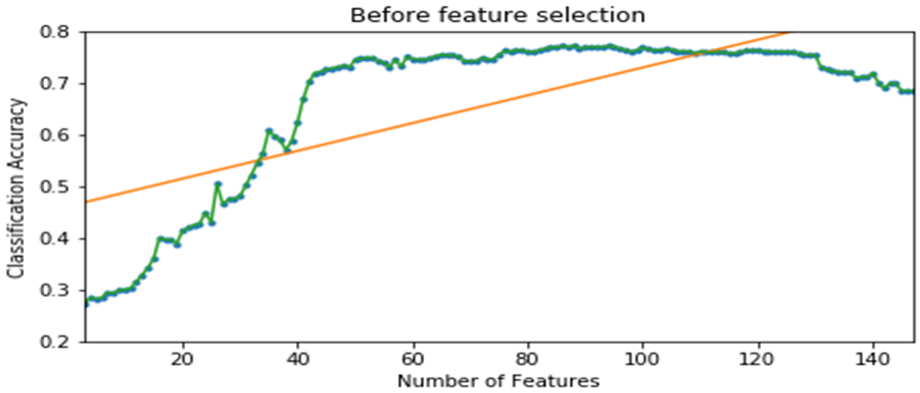

FIGURE 4.2.1: Classification accuracy obtained before feature selection on linear SVM.

From Figure 4.2.2, it can be inferred that the F1 score obtained before applying any feature selection algorithm is 79.96% by selecting 83 features.

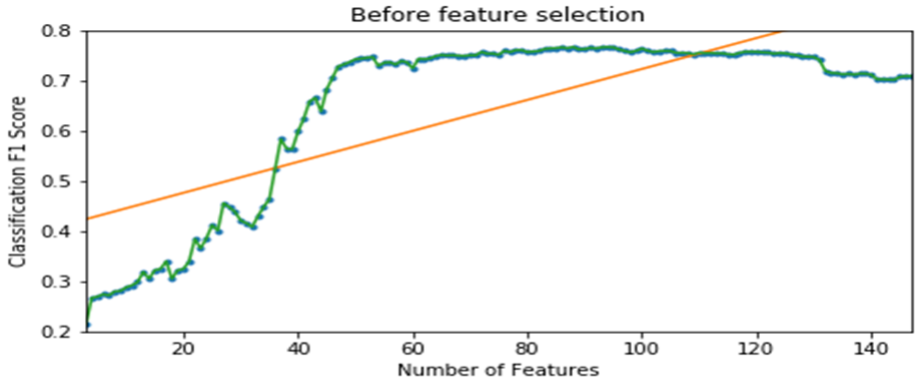

FIGURE 4.2.2: F1 score obtained before feature selection on linear SVM.

4.2.2 Classification results on datasets after mRMR feature selection using SVM linear

From Figure 4.2.3, it can be inferred that the classification accuracy obtained after applying mRMR feature selection is 95.23% by selecting 36 features. After applying feature selection algorithm, the classification accuracy is increased from 79.99% to 95.23%.

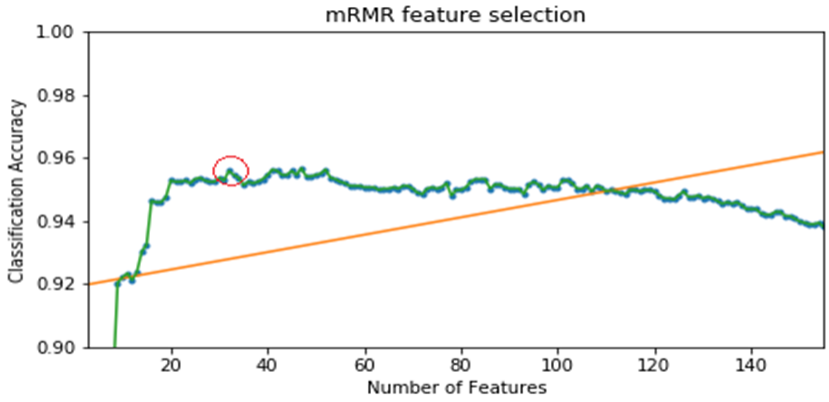

FIGURE 4.2.3: Classification accuracy using SVM linear and feature selection.

From Figure 4.2.4, it can be inferred that the F1 score obtained after applying mRMR feature selection algorithm is 94.92% by selecting 36 features. After applying feature selection, the F1 score is increased from 79.96% to 94.92%.

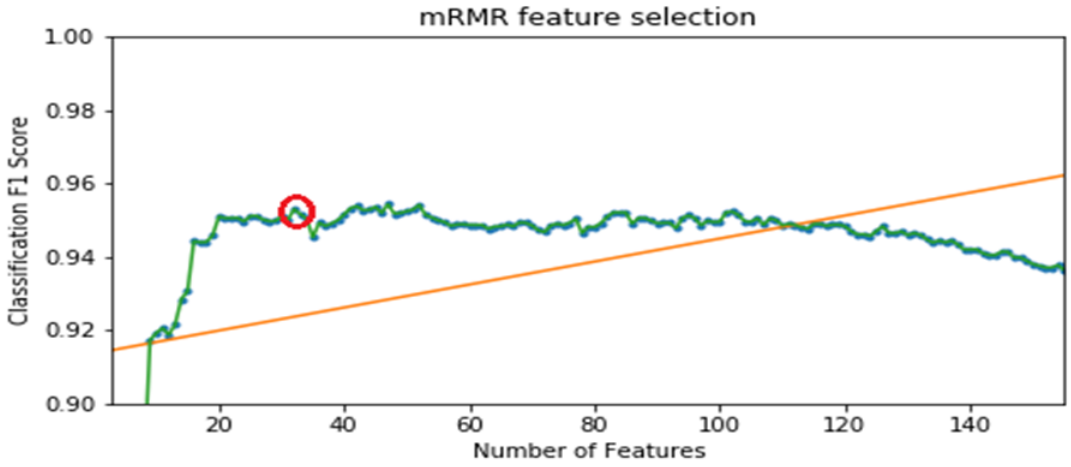

FIGURE 4.2.4: F1 score using SVM linear and feature selection.

4.2.3 Classification results on datasets using SVM RBF at N=36

After selecting the optimal number of features by applying the mRMR feature selection algorithm, they can be selected to run the SVM RBF classifier. By selecting 36 features, the multi-class SVM RBF classification is performed on the dataset using stratified cross-validation. After performing the grid search, the best classification accuracy was obtained for cost at 10^5 and gamma at 0.01. SVM RBF increases the performance metrics when compared to the SVM linear classifier. The classification accuracy obtained using RBF classifier is 96.27% and the F1 score obtained is 96.14%.

4.2.4 Classification results on datasets using Random Forest at N=36

With the selected number of features by by applying mRMR feature selection algorithm, the random forest classifier is run on the dataset. By selecting 36 features, the random forest classification is performed on the dataset using stratified cross-validation. Random forest increases the performance metrics when compared to the SVM RBF classifier. The classification accuracy obtained using the RBF classifier is 99.00% and F1 score obtained is 98.99%.

4.2.5 Comparison of SVM Linear, RBF and Random Forest

From Figures 4.2.5 and 4.2.6, the best classification accuracy and F1 score are obtained by using random forest with 99.00% and 98.99% respectively, whereas the classification accuracy and F1 score obtained using SVM RBF classifier are 96.27% and 96.14%, and obtained using SVM linear are 95.23% and 94.99% respectively.

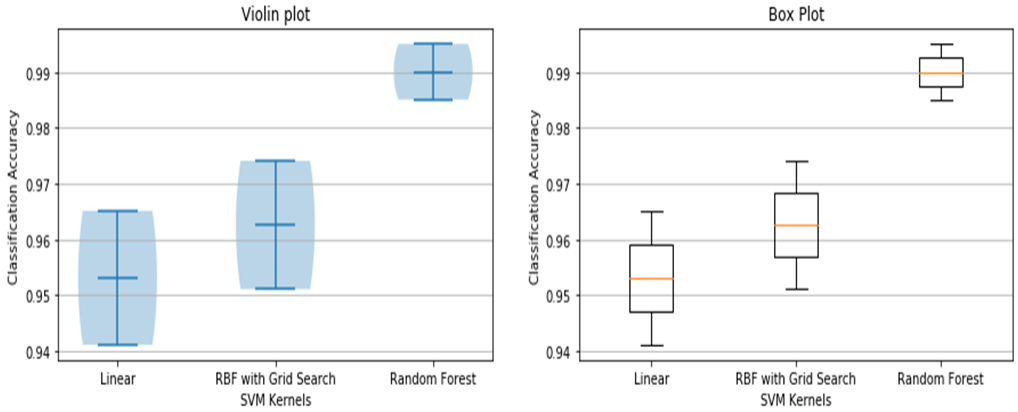

FIGURE 4.2.5: Comparison of classification accuracies for SVM linear, RBF, Random Forest.

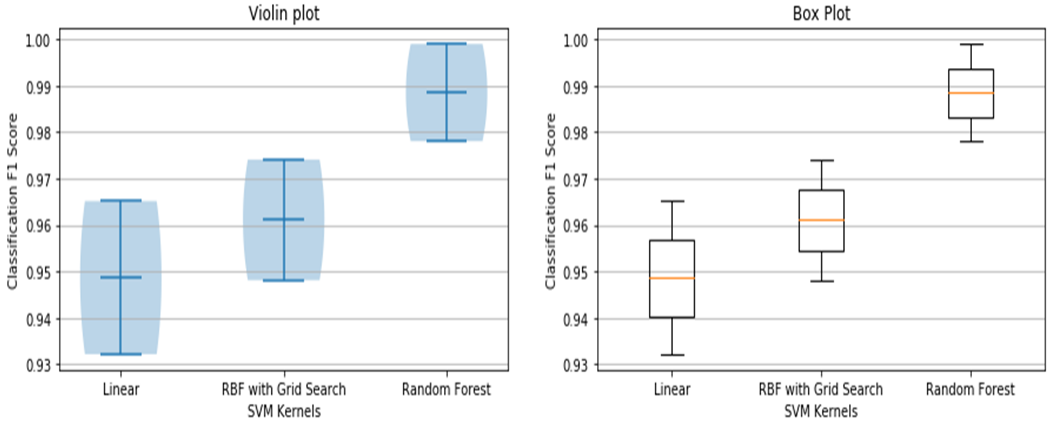

FIGURE 4.2.6: Comparison of F1 scores for SVM linear, RBF, Random Forest.

4.3 Thirty samples without device specific features

The previous dataset contains users with thirty samples for each of them. The dataset contains the maximum valid data which can be obtained from the data collection process.

To know the effect of different features in the dataset, this experiment was carried out; where the device-specific features such as hardware, manufacturer, SDK version, country code, language, and the number of CPU cores were removed. The results obtained are without device data on the entire dataset with thirty instances per user. The optimal number of features were selected to be 88 to yield better classification performance.

4.3.1 Classification results on the original datasets using SVM linear

From Figure 4.3.1, it can be inferred that the classification accuracy obtained before applying any feature selection algorithm is 75.31% by selecting 57 features.

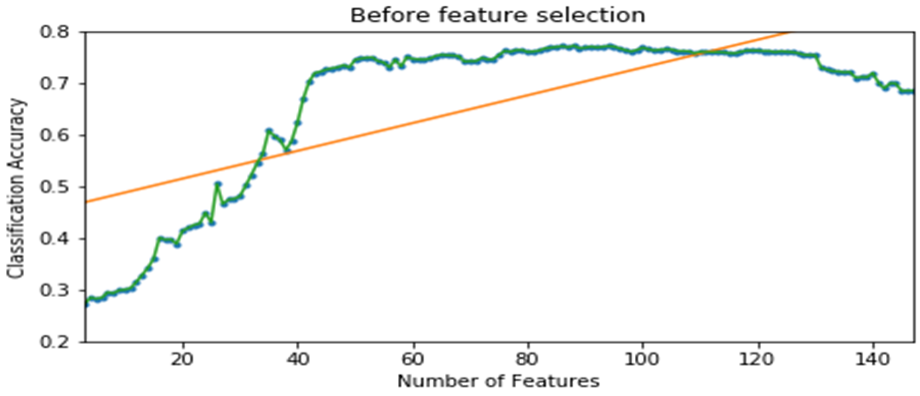

FIGURE 4.3.1: Classification accuracy obtained before feature selection on linear SVM.

From Figure 4.3.2, it can be inferred that the F1 score obtained before applying any feature selection is 74.94% by selecting 57 features.

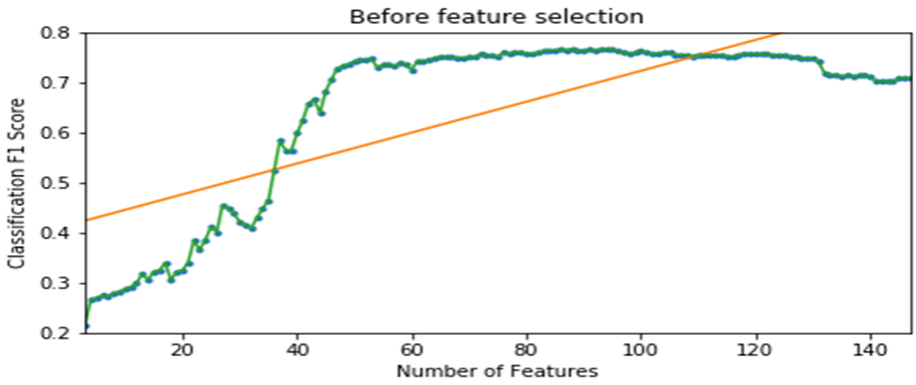

FIGURE 4.3.2: F1 score obtained before feature selection on linear SVM.

4.3.2 Classification results on datasets after mRMR feature selection using SVM linear

From Figure 4.3.3, it can be inferred that the classification accuracy obtained after applying mRMR feature selection algorithm is 77.27% by selecting 88 features. After applying feature selection, the classification accuracy is increased from 75.31% to 77.27%.

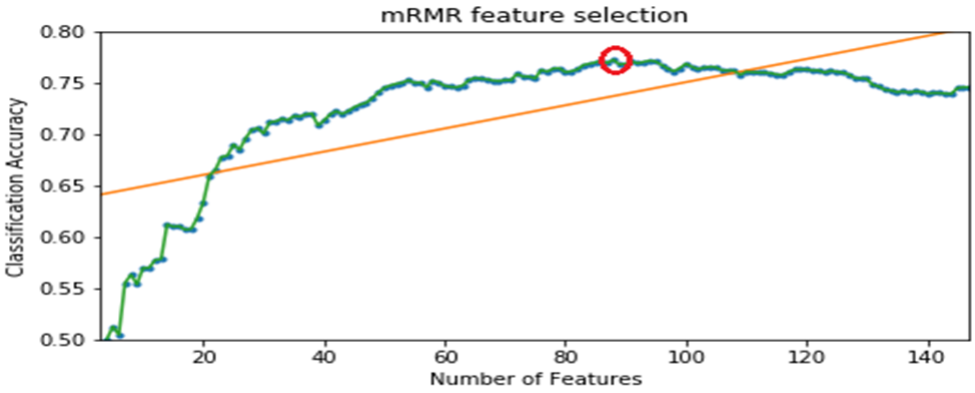

FIGURE 4.3.3: Classification accuracy using SVM linear and feature selection.

From Figure 4.3.4, it can be inferred that the F1 score obtained after applying mRMR feature selection is 76.77% by selecting 88 features. After applying feature selection, the F1 score is increased from 74.94% to 76.77%.

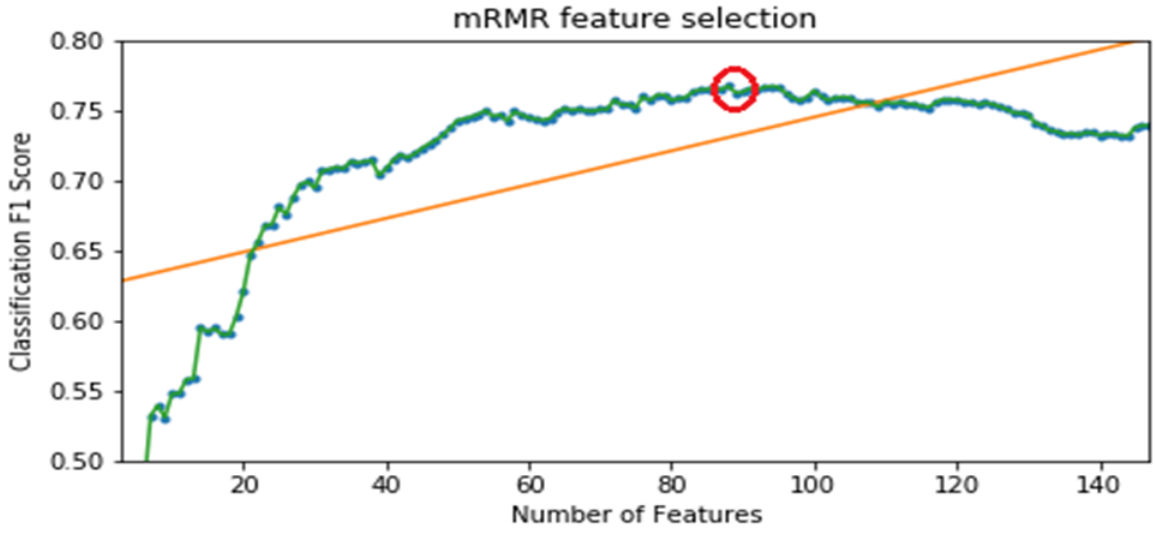

FIGURE 4.3.4: F1 score using SVM linear and feature selection.

4.3.3 Classification results on datasets using SVM RBF at N=88

After selecting the optimal number of features by applying mRMR feature selection, they can be selected to run the SVM RBF classifier. By selecting 88 features, the multi-class SVM RBF classification is performed on the dataset using stratified cross-validation. After performing the grid search, the best classification accuracy was obtained for cost at 10^5 gamma at 0.001. SVM RBF increases the performance metrics when compared to the SVM linear classifier. The classification accuracy obtained using SVM RBF classifier is 78.13% and the F1 score obtained is 77.43%.

4.3.4 Classification results on datasets using Random Forest at N=88

With the selected number of features by by applying mRMR feature selection algorithm, the random forest classifier is run on the dataset. By selecting 88 features, the random forest classification is performed on the dataset using stratified cross-validation. Random forest increases the performance metrics when compared to the SVM RBF classifier. The classification accuracy obtained using the SVM RBF classifier is 86.66% and F1 score obtained is 86.28%.

4.3.5 Comparison of SVM Linear, RBF and Random Forest

From Figures 4.3.5 and 4.3.6, the best classification accuracy and F1 score are obtained by using random forest with 86.66% and 86.28% respectively, whereas the classification accuracy and F1 score obtained using SVM RBF are 78.13% and 77.43%, and obtained using SVM linear are 77.27% and 76.77% respectively.

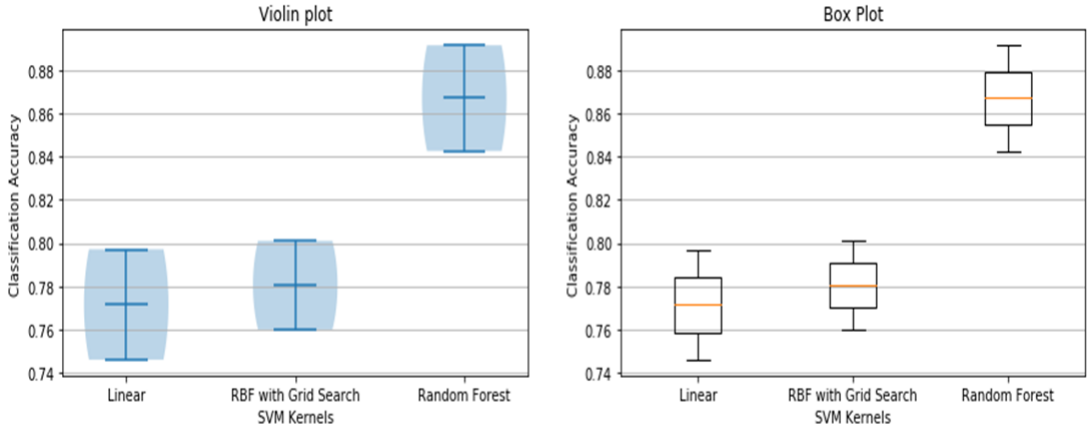

FIGURE 4.3.5: Comparison of classification accuracies for SVM linear, RBF, Random Forest.

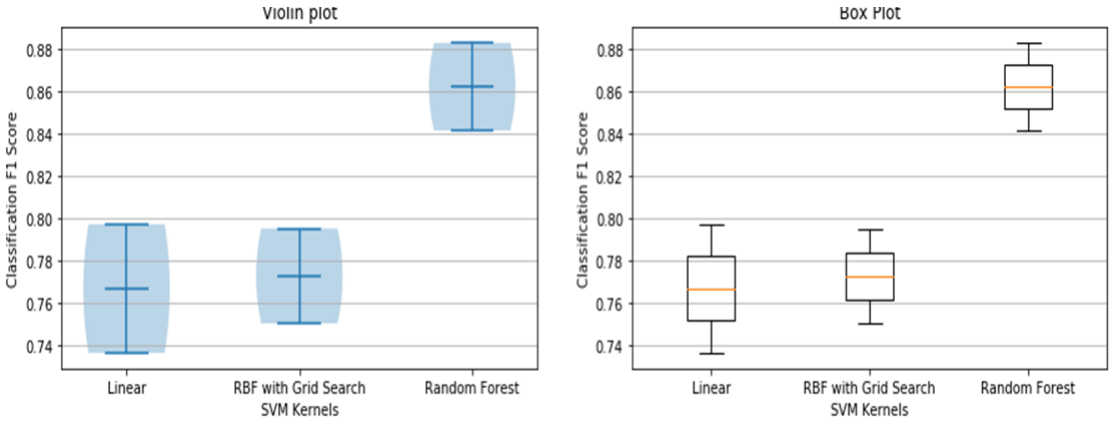

FIGURE 4.3.6: Comparison of F1 scores for SVM linear, RBF, Random Forest.

4.4 Thirty samples without pressure related features

The previous dataset contains the entire dataset without device-specific features. To know the effect of pressure-related features, this experiment was carried out; where the touch pressure features for various buttons and action type events were removed. The results obtained below are without pressure and device features data on the entire dataset with thirty instances per user. The optimal number of features were selected to be 33 to yield better classification metrics.

4.4.1 Classification results on the original datasets using SVM linear

From Figure 4.4.1, it can be inferred that the classification accuracy obtained before applying any feature selection algorithm is 69.31% by selecting 79 features.

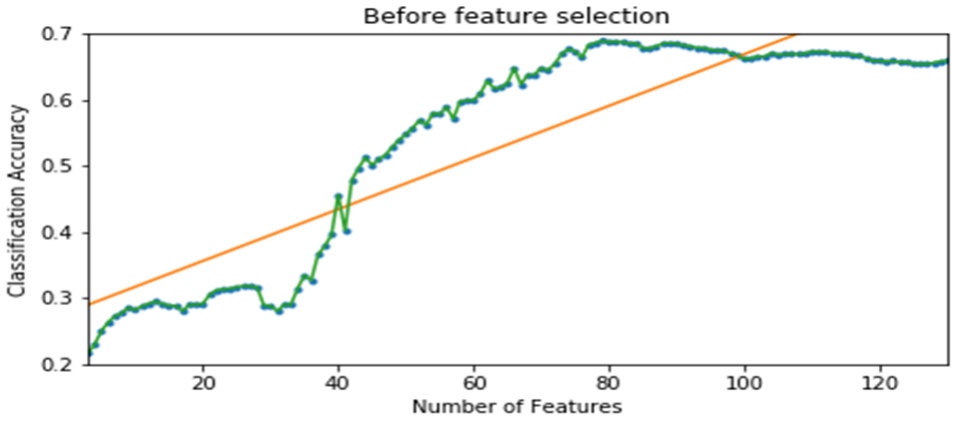

FIGURE 4.4.1: Classification accuracy obtained before feature selection on linear SVM.

From Figure 4.4.2, it can be inferred that the F1 score obtained before applying any feature selection algorithm is 68.27% by selecting 78 features.

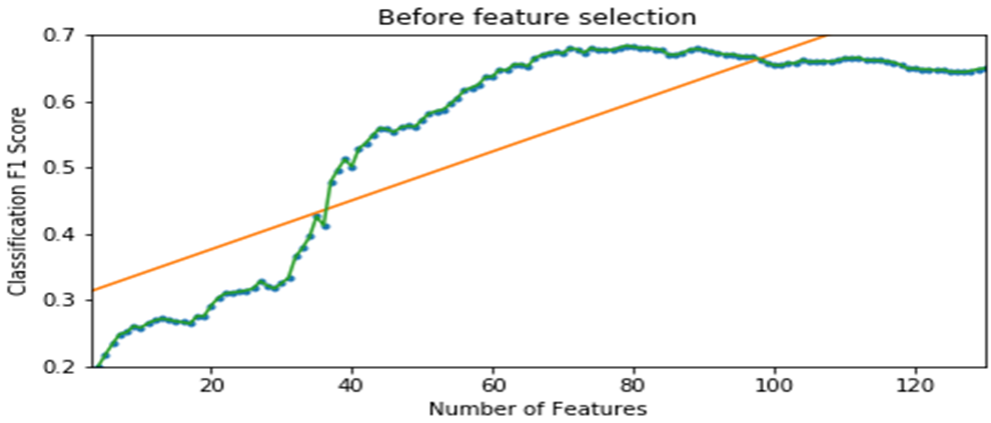

FIGURE 4.4.2: F1 score obtained before feature selection on linear SVM.

4.4.2 Classification results on datasets after mRMR feature selection using SVM linear

From Figure 4.4.3, it can be inferred that the classification accuracy obtained after applying the mRMR feature selection is 69.91% by selecting 33 features. After applying feature selection algorithm, the classification accuracy is increased from 69.31% to 69.91%.

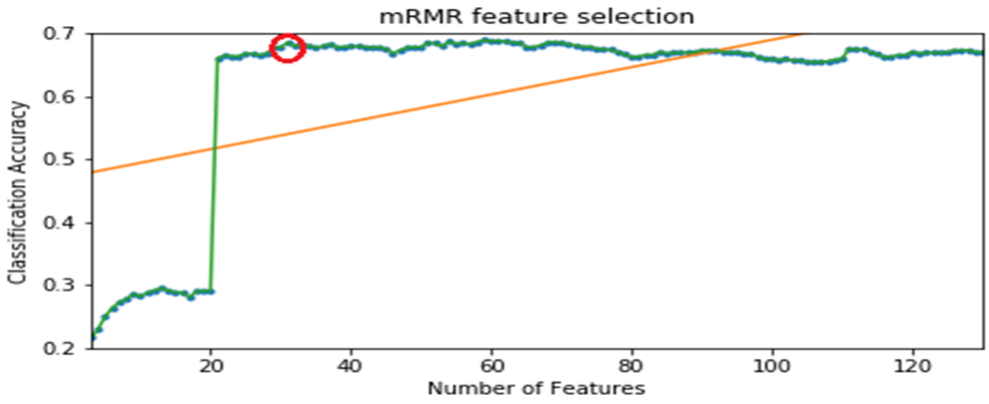

FIGURE 4.4.3: Classification accuracy using SVM linear and feature selection.

From Figure 4.4.4, it can be inferred that the F1 score obtained after applying mRMR feature selection algorithm is 69.83% by selecting 33 features. After applying feature selection, the F1 score is increased from 68.27% to 69.83%.

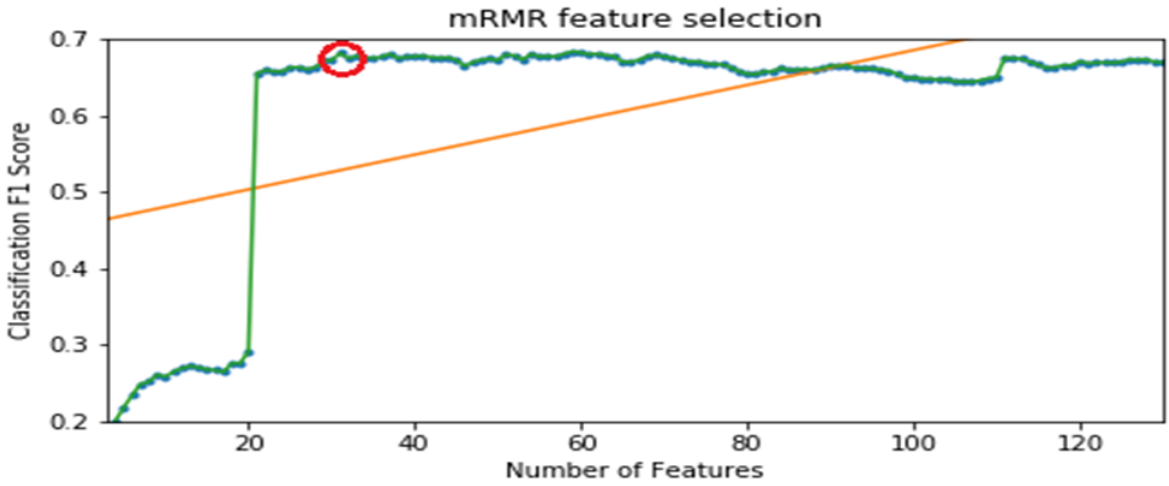

FIGURE 4.4.4: F1 score using SVM linear and feature selection.

4.4.3 Classification results on datasets using SVM RBF at N=33

After selecting the optimal number of features by applying mRMR feature selection algorithm, they can be selected to run the SVM RBF classifier. By selecting 33 features, the multi-class SVM RBF classification is performed on the dataset using stratified cross-validation. After performing the grid search, the best classification accuracy was obtained for cost at 10^6 and gamma at 0.001. SVM RBF increases the performance metrics when compared to the SVM linear classifier. The classification accuracy obtained using RBF classifier is 73.11% and F1 score obtained is 72.49%.

4.4.4 Classification results on datasets using Random Forest at N=33

With the selected number of features by applying the mRMR feature selection algorithm, the random forest classifier is run on the dataset. By selecting 33 features, random forest classification is performed on the dataset using stratified cross-validation. Random forest increases the performance when compared to the SVM RBF classifier. The classification accuracy obtained using RBF classifier is 79.09% and F1 score obtained is 78.61%.

4.4.5 Comparison of SVM Linear, RBF and Random Forest

From Figures 4.5.5 and 4.5.6, the best classification accuracy and F1 score are obtained by using random forest with 79.09% and 78.61% respectively, whereas the classification accuracy and F1 score obtained using SVM RBF classifier are 73.11% and 72.11%, and obtained using SVM linear are 69.91% and 69.83% respectively.

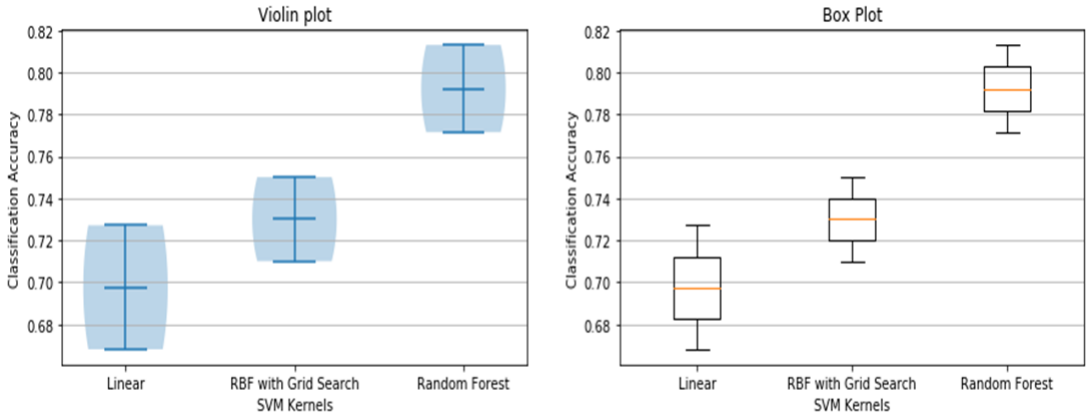

FIGURE 4.4.5: Comparison of classification accuracies for SVM linear, RBF, Random Forest.

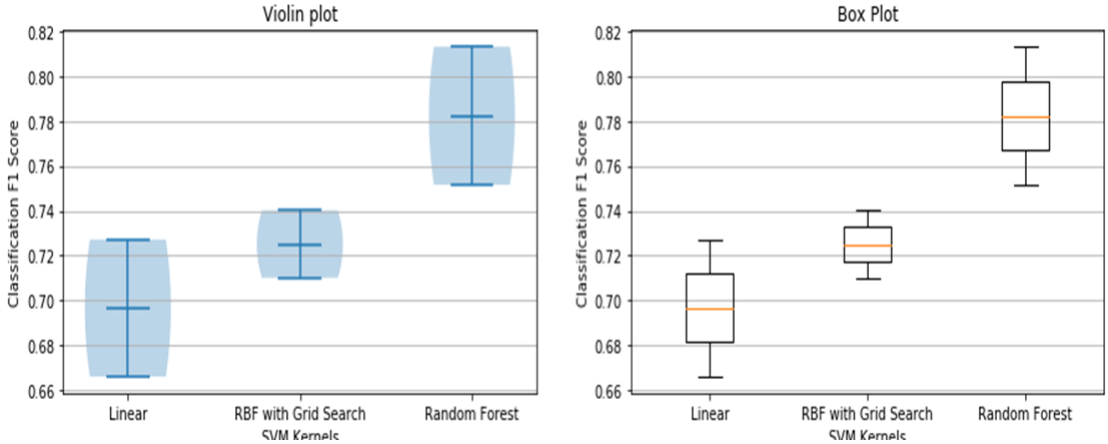

FIGURE 4.4.6: Comparison of F1 scores for SVM linear, RBF, Random Forest.

4.5 Overall Comparison

4.5.1 Comparison of classification accuracies for all experiments

As illustrated in Figure 4.5.1, the comparison between classification accuracies is shown for various classifiers and different experiments. Besides, the best results from the three classifiers (Random forest, SVM-RBF, and SVM-Linear) are achieved using random forest in our experiment which contains thirty instances per user with the device-specific information. The highest classification accuracy achieved using the random forest classifier is 99.00%. Thus, it can be concluded that the experiment containing device-specific features with more number of instances gives higher accuracy values.

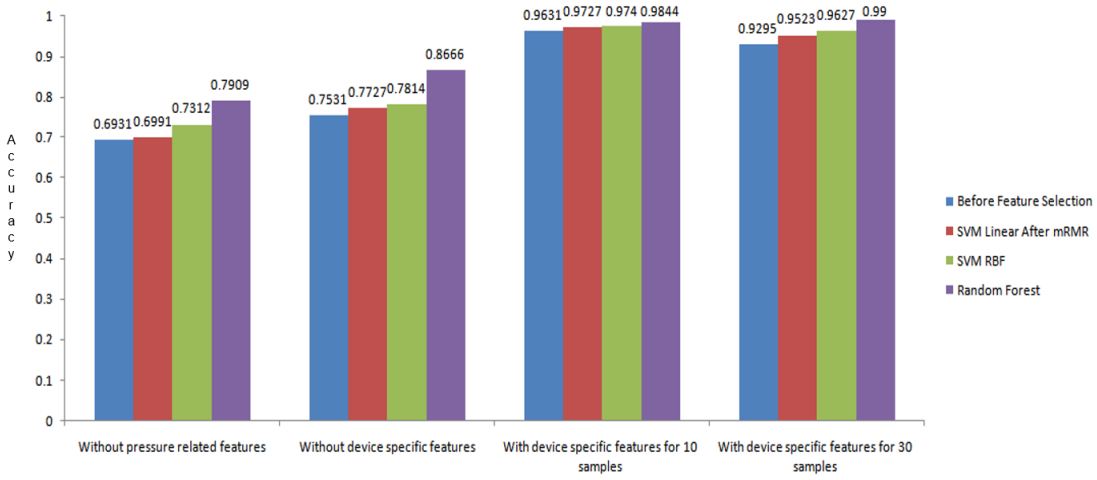

FIGURE 4.5.1: Comparing classification accuracy performance of each classifier for all the experiments.

4.5.2 Comparison of F1 scores for all experiments

As illustrated in Figure 4.5.2, the comparison between F1 scores is shown for various classifiers and different experiments. Besides, the best results from the three classifiers (Random forest, SVM-RBF, and SVM-Linear) are achieved using random forest in our experiment which contains thirty instances per user with the device-specific information. The highest classification accuracy achieved using the random forest is 98.99%. Thus, it can be concluded that experiment containing device-specific features with more number of instances

gives higher F1 score values.

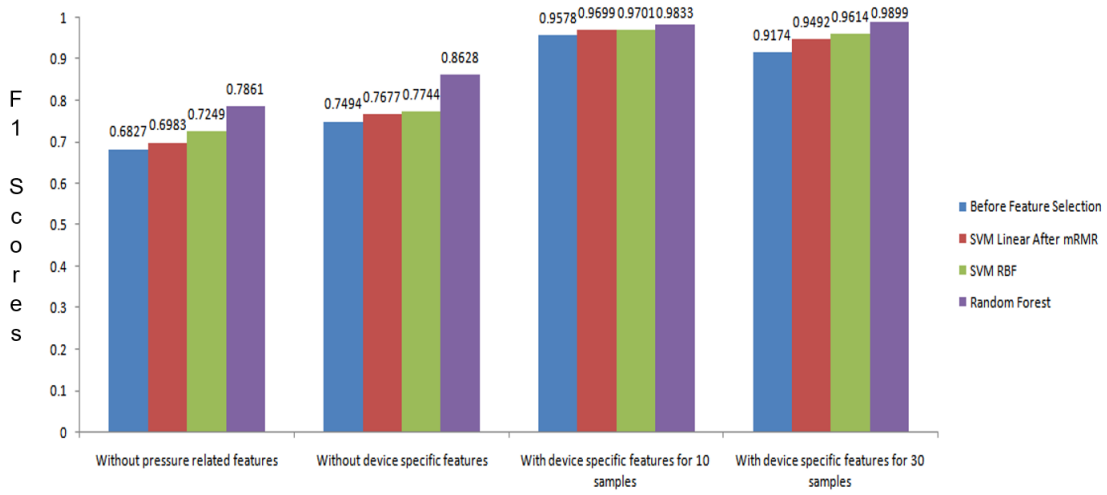

FIGURE 4.5.2: Comparing F1 score performance of each classifier for all experiments.

4.5.3 Compilation of Results

As shown in Table 4.5.1, different experiments were carried out by altering the dataset samples and features. As observed in the table, the best result for classifying the user was obtained by using random forest classifier for the experiment containing the device-specific features.

TABLE 4.5.1: Classifier results for all experiments.

Experiments	Classifiers	mRMR feature selection	Accuracy	F1 Score
With device specific features for 10 samples	SVM Linear	36	0.9727	0.9699
	SVM RBF		0.9740	0.9701
	Random Forest		0.9844	0.9833
With device specific features for 30 samples	SVM Linear	36	0.9523	0.9492
	SVM RBF		0.9627	0.9614
	Random Forest		0.9900	0.9899
Without device specific features from exp2	SVM Linear	88	0.7727	0.7677
	SVM RBF		0.7814	0.7744
	Random Forest		0.8666	0.8628
Without pressure related features from exp 3	SVM Linear	33	0.6991	0.6983
	SVM RBF		0.7311	0.7249
	Random Forest		0.7909	0.7861

The best accuracy and F1 score values for all the datasets after being classified by different classifiers have been shown in the table. Finally, among all the classifiers we used in our experiment, SVM-Linear was the weakest and SVM-RBF, and Random Forest

TABLE 4.6.1: Evaluation of our experiment with previous related works.

Reference Paper	No. of users	No. of features	Classifier	Classification Metrics
[29]	15	16	One Class	EER ranges from 11.65% to 25.16%
[31]	10	3	*k*-NN	Accuracy ranges from 71% to 99%
[33]	5	4	Neural Net	Accuracy is 85.4545%
[3]	42	71	Bayes Net	EER ranges from 4.3% to 9.8%
[13]	10	14	Bayes Net	Accuracy is 82.18%
[1]	56	71	MMD	FAR is 5% FRR is 5.6%
Our Paper	**94**	**155**	**Multi-class ovo SVM**	**Accuracy is 97.40% F1 Score is 97.01%**

all performed very well. Random forest performs well for this problem and can be used for all multi-class problems. From another point of view, among all the experiments, the datasets with device-specific features are almost equally the best datasets. As the features were removed, the classification performance metrics were affected and reduced. After carrying out the experiments, device-specific features contribute more to the metrics and thus, classify the users better.

4.6 Experimental Evaluation

It is evident that previous papers perform either identification or verification of users on less users and features. Different classifiers have been used by them to find the EER values. Our method optimizes the accuracy and F1 score metrics by performing wrapper-based feature selection using mRMR algorithm as it finds the most relevant features for classification.

From Table 4.6.1, we observe that all the experiments carried before contain less users and features when compared to our study. To record the biometric pattern of the user, all possible features must be recorded to obtain high classification results. This enhances to uniquely identify each user; feature selection after this will help identify the most important features to perform the identification of users accurately.

As concluded in [3], their main limitations are addressed in this thesis, the data is collected over all ranges of users between aged 18 to 65 with varying levels of touchscreen experience. Also, most of the previous works did not carry out the research extensively on a larger population. In an earlier work, P. Bhattarakosol and H. Saevanee [31] used *k*-NN classifier with three features to identify ten users with 99% accuracy. However, these high

scores are obtained from fewer features and small population. The feature extraction results show that 155 significant features are extracted which identify the users accurately. The data collection process lasted for more than a month to record 94 user patterns successfully. Thus, classification is performed effectively by properly combining feature selection and complex SVM and random forest classification algorithms on a large dataset.

CHAPTER 5

Conclusion and Future Work

5.1 Contributions

In this thesis, keystroke dynamics based authentication was tested using .tie5Roanl password and the best classification performance metrics obtained were from the random forest classifier. Grid search optimization was also used to determine the best parameters for the SVM RBF kernel. The anonymous data were collected from a relatively large number of users (94) who inputted their biometric pattern. Feature extraction and selection were performed on the raw data to increase the classification scores. Using the wrapper-based feature selection mRMR method, an optimized number of features were selected. From our study, we conclude that touch pressure, touch size and coordinates are mainly responsible for authenticating the user as a legitimate user. The experiments were carried using a virtual keypad of an app which contributes to a homogeneous environment for all users to obtain a foolproof algorithm. The classification metrics for the random forest classifier are found to be higher than the SVM kernels because of its generalization power, thus, forming a robust algorithm. The results prove that:

- Users accessing computing devices are effectively classified based on their behaviour.
- The research will contribute significantly to the field of cyber-security by forming a robust authentication system using machine learning algorithms.
- For building a highly secured system, multi factor authentication system is required. Password (knowledge-based) + device (token-based) + behaviour (pattern-based).

- The importance of features were studied by dividing the dataset and running various experiments.
- Random forest improves the performance metrics of the classification system.
- It is evident that the device specific features play a vital role to improve the performance of the biometric authentication system.

5.2 Future Work

The identification of users is based on the real-time data, hence different feature selection, and classification approaches can be used to minimize the classification error. Moreover, the error rates can be focused to be minimal since no legitimate user should be ignored as a faux user, and no illegitimate user should be classified as an authorized user. The data collection process might involve any other sensor information which can be used as an additional feature to record the behavioral biometry of the user. Future researchers can also combine different biometric modalities together such as combining the hard password with keystroke with fingerprint dynamics. This experiment is carried out using static keystroke dynamics, whereas more comprehensive research is required for dynamic or continuous keystroke dynamics on mobile devices. The users can also type dynamic passcodes which keep changing during the data collection process, leading to dynamic keystroke authentication algorithms. For the perspectives of this work, user verification can be implemented instead of identification or a combination of both can be experimented. The data collection process can also involve experimental users to observe and mimic the typing patterns of other users. Future work could also be done to increase the number of classes in the dataset by involving more participants. For the real user, it is essential that he/she does not get disturbed in his/her everyday business under the circumstances. Again here it is critical to know what number of activities a legitimate user can perform when locked out from the system. One of the ways to extend this work is to make the system learn from the user activities and action sequences about the user by incorporating machine learning and artificial intelligence when deciding about the user and to identify the intruders accurately. These

participants can participate in the research under a controlled environment to increase the number of samples for each user and to protect them from all kinds of cyber-attacks. Therefore, all options for extending this work can be summarized as follows:

- The data collection process might involve any other additional information supporting to record the behavioural biometry of the user.
- The users can replicate the typing patterns of other users to test the efficiency of the developed system.
- This experiment can be carried out using dynamic or continuous keystroke dynamics for mobile devices.
- One-class classifier can be used to build the model for the verification method.
- Participants can be monitored and guided to take part in the data collection process by using a single device to input the data.

REFERENCES

[1] Al-Obaidi, N. M. and Al-Jarrah, M. M. (2016). Statistical keystroke dynamics system on mobile devices for experimental data collection and user authentication. In *2016 9th International Conference on Developments in eSystems Engineering (DeSE)*, pages 123–129.

[2] Antal, M. and Nemes, L. (2016). *The MOBIKEY Keystroke Dynamics Password Database: Benchmark Results*, pages 35–46. Springer International Publishing, Cham.

[3] Antal, M. and Szabo, L. Z. (2015). An evaluation of one-class and two-class classification algorithms for keystroke dynamics authentication on mobile devices. In *2015 20th International Conference on Control Systems and Computer Science*, pages 343–350.

[4] Araujo, L. C., Sucupira, L. H., Lizarraga, M. G., Ling, L. L., and Yabu-Uti, J. B. T. (2005). User authentication through typing biometrics features. *IEEE transactions on signal processing*, 53(2):851–855.

[5] Bergadano, F., Gunetti, D., and Picardi, C. (2002). User authentication through keystroke dynamics. *ACM Transaction Information System Security*, 5:367–397.

[6] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[7] Clarke, N. L. and Furnell, S. M. (2007). Authenticating mobile phone users using keystroke analysis. *International Journal of Information Security*, 6(1):1–14.

[8] Erlich, Z. and Zviran, M. (2015). Authentication practices from passwords to biometrics. In *Encyclopedia of Information Science and Technology, Third Edition*, pages 4248–4257. IGI Global.

[9] Giot, R., El-Abed, M., and Rosenberger, C. (2009). Greyc keystroke: a benchmark for keystroke dynamics biometric systems. In *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*, pages 1–6. IEEE.

[10] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

[11] Hsu, C.-W. and Lin, C.-J. (2002a). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425.

[12] Hsu, C.-W. and Lin, C.-J. (2002b). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425.

[13] Jeanjaitrong, N. and Bhattarakosol, P. (2013). Feasibility study on authentication based keystroke dynamic over touch-screen devices. In *2013 13th International Symposium on Communications and Information Technologies (ISCIT)*, pages 238–242.

[14] Jordan, M. I. and Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260.

[15] Joyce, R. and Gupta, G. (1990). Identity authentication based on keystroke latencies. *Communications of the ACM*, 33(2):168–176.

[16] Kundu, S. and Sarker, G. (2016). An efficient integrator based on template matching technique for person authentication using different biometrics. *Indian Journal of Science and Technology*, 9(42).

[17] Liaw, A., Wiener, M., et al. (2002). Classification and regression by random forest. *R news*, 2(3):18–22.

[18] Liu, H., Dougherty, E. R., Dy, J. G., Torkkola, K., Tuv, E., Peng, H., Ding, C., Long, F., Berens, M., Liu, H., Parsons, L., Zhao, Z., Yu, L., and Forman, G. (2005). Evolving feature selection. *IEEE Intelligent Systems*, 20(6):64–76.

[19] Madzarov, G., Gjorgjevikj, D., and Chorbev, I. (2009). A multi-class svm classifier utilizing binary decision tree. *Informatica*, 33(2).

[20] Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (2013). *Machine learning: An artificial intelligence approach*. Springer Science and Business Media.

[21] Monaco, J. V., Perez, G., Tappert, C. C., Bours, P., Mondal, S., Rajkumar, S., Morales, A., Fierrez, J., and Ortega-Garcia, J. (2015). One-handed keystroke biometric identification competition. In *2015 International Conference on Biometrics (ICB)*, pages 58–64.

[22] Monrose, F. and Rubin, A. D. (2000). Keystroke dynamics as a biometric for authentication. *Future Generation Computer Systems*, 16(4):351 – 359.

[23] Montalvao, J., Freire, E. O., Jr., M. A. B., and Garcia, R. (2015). Contributions to empirical analysis of keystroke dynamics in passwords. *Pattern Recognition Letters*, 52(Supplement C):80 – 86.

[24] Nasrabadi, N. M. (2007). Pattern recognition and machine learning. *Journal of Electronic Imaging*, 16.

[25] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

[26] Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238.

[27] Prakash, S. and Gupta, P. (2015). Introduction. In *Ear Biometrics in 2D and 3D*, pages 1–20. Springer.

[28] Rodriguez, J. J., Kuncheva, L. I., and Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1619–1630.

[29] Roh, J. H., Lee, S. H., and Kim, S. (2016). Keystroke dynamics for authentication in smartphone. In *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1155–1159.

[30] Roy, U., Sinha, D., and Roy, S. (2017). User authentication: Keystroke dynamics with soft biometric features. In *Internet of Things (IoT)*, pages 123–142. CRC Press.

[31] Saevanee, H. and Bhattarakosol, P. (2009). Authenticating user using keystroke dynamics and finger pressure. In *2009 6th IEEE Consumer Communications and Networking Conference*, pages 1–2.

[32] Salah, A. A. (2012). Machine learning for biometrics. In *Machine Learning: Concepts, Methodologies, Tools and Applications*, pages 704–723. IGI Global.

[33] Salem, A., Zaidan, D., Swidan, A., and Saifan, R. (2016). Analysis of strong password using keystroke dynamics authentication in touch screen devices. In *2016 Cybersecurity and Cyberforensics Conference (CCC)*, pages 15–21.

[34] ShenTeh, P. and NingZhang (2016). A survey on touch dynamics authentication in mobile devices. *ScienceDirect*, 59(2):210–235.

[35] Shoniregun, C. A. and Crosier, S. (2008). *Applications of Biometrics*, pages 71–112. Springer US, Boston, MA.

[36] Singha, A. K., Singla, A., and Pandey, R. K. (2016). Study and analysis on biometrics and face recognition methods. *EPH-International Journal of Science And Engineering (ISSN: 2454-2016)*, 2(6):37–41.

[37] Smith, M., Mann, M., and Urbas, G. (2018). *Biometrics, Crime and Security*. Routledge.

[38] Sultana, M., Paul, P. P., and Gavrilova, M. (2014). A concept of social behavioral biometrics: motivation, current developments, and future trends. In *Cyberworlds (CW), 2014 International Conference on*, pages 271–278. IEEE.

[39] Tasia, C.-J., Chang, T.-Y., Cheng, P.-C., and Lin, J.-H. (2014). Two novel biometric features in keystroke dynamics authentication systems for touch screen devices. 7(4):750–758.

[40] Teh, P. S., Teoh, A. B. J., and Yue, S. (2013). A survey of keystroke dynamics biometrics. *The Scientific World Journal*, 2013:1–24.

[41] Vens, C. and Costa, F. (2011). Random forest based feature induction. In *IEEE 11th International Conference on Data Mining*, pages 744–753, Washington, DC, USA.

[42] Wayman, J., Jain, A., Maltoni, D., and Maio, D. (2005). An introduction to biometric authentication systems. In *Biometric Systems*, pages 1–20. Springer.

[43] Xu, H., Zhou, Y., and Lyu, M. R. (2014). Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones. In *Symposium On Usable Privacy and Security, SOUPS*, volume 14, pages 187–198.

[44] Zaidan, D., Salem, A., Swidan, A., and Saifan, R. (2017). Factors affecting keystroke dynamics for verification data collecting and analysis. In *2017 8th International Conference on Information Technology (ICIT)*, pages 392–398.

[45] Zhang, C. and Ma, Y. (2012). *Ensemble machine learning: methods and applications*. Springer.

VITA AUCTORIS

NAME:	Sowndarya Krishnamoorthy
PLACE OF BIRTH:	Chennai, TamilNadu, India.
EDUCATION:	Bachelor of Technology in Information Technology, Anna University, Chennai, TamilNadu, India, 2010.
	Master of Science Co-op in Computer Science, University of Windsor, Windsor, Ontario, Canada, 2018.