Continuous User Authentication Featuring Behavioural Biometrics

Anum Tanveer Kiyani

Faculty of Science & Technology

Middlesex University, London

A thesis submitted for the degree of

Doctor of Philosophy
A thesis submitted to Middlesex University in partial fulfilment of the requirements for the degree

April 2021

I would like to dedicate this thesis to my loving parents, my family!

Acknowledgements

First and foremost, I am grateful to God for providing me with this opportunity and granting me the capability to proceed successfully. I would like to express my sincere appreciation and gratitude to the following people for helping me complete this thesis.

I am deeply grateful to Dr. Aboubaker Lasebae, my director of studies and supervisor, for giving me the opportunity to work under him. He has guided me and encouraged me to carry on through these years and has contributed to this thesis with a significant impact. Dr. Aboubaker has been very supportive and gave me the freedom to pursue various topic without objection. My gratitude for his contribution to my future career is immeasurable.

I would also like to thank my other academic supervisor, Dr. Kamran Ali for his invaluable insights and suggestions. Dr. Kamran Ali has always given me a hand by spending his valuable time in exploring different ideas and concepts. At many stages, during this research project, I benefited from his advice, particularly so when exploring innovative ideas. I am forever indebted for his enthusiasm, guidance and unrelenting support throughout this process.

Middlesex University has provided me with a very stimulating environment in what concerns the extraordinary quality of its academic staff, and that experience will leave a mark beyond this thesis.

I extend my thanks and gratitude to my family who have always been a significant source of support and encouragement. I would like to dedicate this thesis to my mother Azra Zatoon. This accomplishment would not have been possible without her. Thank you for always supporting me and believing me. Thank you for teaching me respect, confidence and proper etiquette. I would like to express my heartfelt thanks and gratitude to my brothers and sisters for always listening when I wanted to talk, for the invaluable support and concern.

Finally, to Sohaib Ali Kiani, my husband – there are no words to express how much you have helped me during these years. Without you, this would have been merely impossible. Thank you all for the constant unconditional support and encouragement you have provided me during this journey.

Abstract

A user authentication method consists of a username, password, or any other related credential. These methods are mostly used only once to validate the user's identity at the start of session or sometimes after regular interval of time which can lead to security loopholes. However, one-time verification of user's identity is not resilient enough to provide adequate security all over the session. Such authentication methods are required which can continuously verify that only genuine user is using the system resources for entire session.

In this thesis, a true continuous user authentication system is proposed and implemented using behavioural biometrics i.e., keystroke and mouse dynamics which tends to authenticate the user on each single action. Behavioural biometrics are used since these can passively provide the perpetual information about the user's behaviour when interacting with the system. Moreover, a novel idea of continuously establishing the identity of user without prior claim at the start of session is also investigated in this research.

Different types of system architectures were formulated based on baseline or traditional machine learning and deep neural network techniques. Baseline methods used the statistical features based on mean and standard deviation along with the traditional machine learning classifiers to authenticate the user. On the other hand, recurrent neural networks take the behavioural data input as a sequential time-series and extract features based on raw data events using recurrent neural networks. In particular, system frameworks are designed to lock out the imposter user as quickly as possible along with the optimal effort of avoiding the false lock out of genuine users.

This research is examined with thorough and vigorous experiments and validated with two types of behavioural biometric modalities. Overall, the impact of this research is twofold: **i)** it provides a potential solution framework for a true continuous user authentication system which re-verifies te identity of user on each action and **ii)** it presents a new possibilities of establishing the user's identity on each action without the earlier affirm of any identity associated with the current user of system at start of session.

Contents

List of Figures

List of Tables

Acronyms

ANN	Artificial Neural Network
ANGA	Average Number of Geniune Actions
ANIA	Average Number of Imposter Actions
CUA	Continuous User Authentication
CUI	Continuous User Identification
CNN	Convolutional Neural Network
CER	Crossover Error Rate
CDF	Cumulative Distribution Function
CTC	Connectionist Temporal Classification
DCS	Dynamic Classifier Selection
DD	Down-Down
DU	Down-Up
DT	Decision Trees
EER	Equal Error Rate
E2E	End-to-End
ETS1	Evaluation Threat Scenario 1

ETS2	Evaluation Threat Scenario 2
ETS3	Evaluation Threat Scenario 3
FAR	False Acceptance Rate
FRR	False Rejection Rate
FMR	False Match Rate
FNMR	False Non Match Rate
FER	Failure to Enroll
GRU	Gated Recurrent Unit
KD	Keystroke Dynamics
KDR	Keystroke Dynamics Recognition
KNN	K-Nearest Neighbour
LSTM	Long Short Term Memory
MD	Mouse Dynamics
MDR	Mouse Dynamics Recognition
MFA	Multi-Factor Authentication
PUA	Periodic User Authentication
PUA/I	Periodic User Authentication/Identification
RBF	Radial Basis Function
RCM	Recurrent Confidence Model
ReLU	Rectified Linear Unit
RF	Random Forest
R-RCM	Robust Recurrent Confidence Model

RNN	Recuurent Neural Network
SUA	Static User Authentication
SVM	Support Vector Machine
UD	Up-Down
UU	Up-Up
WCF	Weighted Classifier Fusion
XGBoost	Gradient boosting Decision trees
3FA	3 Factor Authentication

Chapter 1

Introduction

1.1 Introduction

With the advancement of technological revolution, computer systems and networks have become an imperative requisite in almost all aspects of human life at substantially greater rate. For instance, computer systems are controlling communication services, banking, aviation, medical, business and personal operations along with saving the confidential and important information into its databases. However, this escalating reliance on computer systems has divulged novel security threats to online confidential data and information. In this regard, security of computer systems and networks is susceptible to different attacks at the user level, system level or network level precisely.

Network and system level attacks include denial of service, malware and man-in-the-middle attacks while the common user level attacks are masquerade or imposter attacks. Subsequently, in the user level attacks i.e., masquerade attacks, intruders exploit the legitimate users' rights for unauthorised access to some confidential information. One of the main factors responsible for this kind of attack is vulnerable authentication which fosters the likelihood of impersonation by intruders as legitimate users. Hence, security of the critical cyber security systems is mainly reliant on the authentication or identification principles Dee et al. (2019).

Generally, authentication refers to the aptitude of individual to substantiate that he is the one who he claims to be. On the other hand, identification is the the process where system establishes the identity of user without any prior claim of identity.

1.1.1 Static User Authentication (SUA)

Traditionally, user is authenticated using password, usernames or any other related information to ensure whether the user is the one claiming to be while accessing a system or network. Subsequently, resources of session are allocated upon authentication and user can use session for which it has been authenticated until logged out or for some fixed period of time Shen et al. (2017). This is referred to as *Static User authentication (SUA).*

Static user authentication (SUA) can be based on one, two or three factors of authentication in which every factor increases the security of the system. *One-factor authentication* refers to the knowledge that user knows such as password, username, or PIN. *Two-factor authentication* incorporates the knowledge factor with the possession or something user owns, for instance token or identity card. And, the *three-factor authentication* integrates the identity element or biometrics of the end-user as well in terms of something you are in order to contribute as an additional security layer. Velásquez et al. (2018).

In most critical security systems, all three factors of authentication are incorporated to make a three-layer secure system to get access to system and the results show lesser risk of imposter to get access to system resources by exploiting any loophole Kiyani et al. (2020).

1.1.2 Continuous User Authentication (CUA)

The problem arises when system resources are still not considered to be secure for the whole session of user even after the successful implementation of three factor authentication at the start of session. For instance, if a person leaves its system or phone unattended or forgets to log out from authenticated session of any critical application that contains sensitive information, then an attacker can easily takeover as a legitimate user. For that reason, one-time validation of the user's identity is not strong enough for providing resilient security throughout the user's work session in high-risk security environments. Ultimate possible solution to this problem can be continuous monitoring of the system or application after the initial log-in to ensure that the legitimate user is using the system for the entire session. This is referred to as *Continuous User Authentication (CUA)* Alotaibi et al. (2019).

CUA might not substitute the SUA which is used at the start of session to get access to system but CUA is used as an extra layer of security after the legitimate user has got access to system resources to ensure the same user is using the system for whole session. The main aim of CUA is to detect and lock out the imposer user to avoid or lessen the damage caused to system resources and confidential information for which only legitimate user has the access privilege. In this aspect, CUA method should validate the identity of user on each action performed on system in order to detect imposter user as soon as possible and to avoid the false lockout of genuine user to substantially greater extend.

1.1.3 Continuous User Identification (CUI)

On the other hand, if SUA might be removed from the start of session then system needs to identify the current user from a given set of users. In this case, the identification of given user on each action would be referred to as Continuous User Identification (CUI) Kochegurova and Martynova (2020).

1.1.4 Periodic User Authentication/Identification (PUA/I)

Periodic User Authentication/Identification (PUA/I) is an approach in which system identifies the user based on fixed block of actions or after fixed period of time. In this approach, system waits until the user performs the given fixed number of actions or fixed time frame before the authentication or identification techniques can be applied to verify the users identity.

1.2 Problem Statement

The existing computer systems are mainly reliant on user authentication methods at initial login and system resources with privileged permissions are allocated to authorised user for the whole session without re-validating that whether the current user is the one who has been authenticated earlier or not. More specifically, the system resources are allocated to any user for the duration of whole session until user logs out of system after initial login. This type of static authentication mechanisms could be appropriate for low risk security systems, however, it can fosters the security threats in high risk security systems where confidentiality of data, information and system resources is the main priority.

A vulnerable authentication method escalates the risk of session hijacking and masquerade attacks on an open session of user who forgets to log out or leave the system unattended for shorter or longer period of time. In this case, continuous user authentication (CUA) ought to be an imperative pre-requisite for high risk security environments where only having static user authentication (SUA) at the start of session is not impermeable enough to monitor that only legitimate user is accessing the confidential information and system resources for the whole session.

A robust CUA system should meet two basic requirements. *Firstly*, it should not disturb the user while it is performing any tasks on system and work passively by gathering the behavioural information of users. This requirement rules out the knowledge and possession based authentication methods where user needs to be actively involved in providing the credentials i.e., password, PINS or tokens for authentication purpose. *Secondly*, CUA should authenticate the user continuously on every single activity that user is performing.

With the advancement of technology, more sophisticated approaches related to physiological and behavioural human characteristics have been utilised to make computing resources more secure which include biometrics i.e., fingerprints, face, iris, keystroke dynamics and mouse recognition. Biometric technology has been broadly used in physical security systems, however, the integration of biometrics for daily usage of computer systems has been comparatively low. The main reason for restricted adoption of biometrics is dependence on special purpose hardware for biometric data collection. Therefore, it can eventually make the incorporation of biometric security more expensive than the actual cost of computer resources. Most of the vendors are integrating these biometric data collection devices into their products. However, still enormous devices are scarce with such incorporated devices which has restricted the implementation of biometric technology for day to day computer security to a greater extent.

In this regard, another interesting class of biometrics named behavioural biometrics Alotaibi et al. (2019) have gained popularity nowadays. These behavioural biometrics scrutinize the user behaviour while individual is interacting with computer systems in order to validate its identity. Moreover, most of the behavioural biometrics i.e., keystrokes or mouse dynamics, do not require any extra hardware to collect biometric data thereby implementation and operational cost is less as compared to physiological biometrics which require special devices like fingerprint scanner or iris recognition devices etc.

In order to meet the requirements of true CUA, one possible way is to use behavioural biometrics e.g., Keystroke dynamics and Mouse dynamics which may play an important role to validate the user's identity throughout the session by distinguishing one user from another. Moreover, most of behavioural biometrics do not require users to present biometrics identification while preforming important tasks and tends to authenticate the user on each single action.

Analysing the user behaviour for continuous authentication or identification is a challenging task owing to the insufficient information and large intra-class disparities of data recorded by the computer input devices. Accordingly, most of the preceding research works had employed the analysis based on PUA so the system records the data for fixed block size and then afterwards analyse the data to decide if it belongs to genuine user or not. However, this approach can give room to imposer user to perform illegitimate activities on system. On the contrary, a true CUA method inclines to verify the identity of user after each single action. The basic concept of periodic and continuous user authentication is illustrated in Fig 1.1.

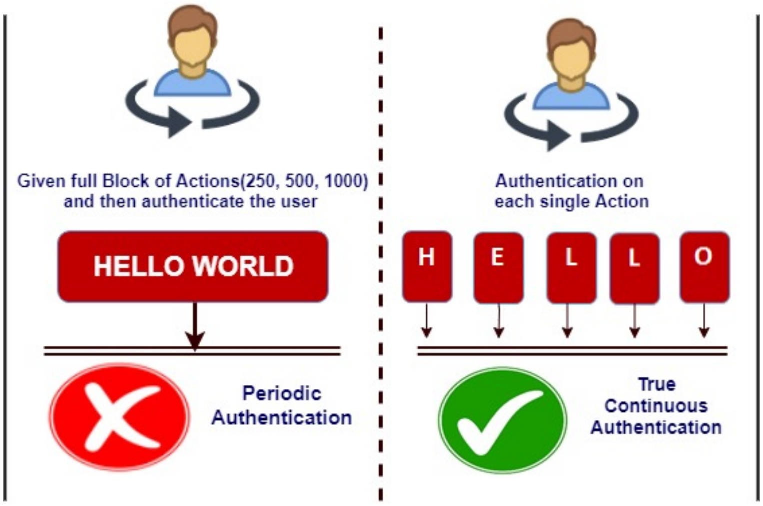

Given full Block of Actions(250, 500, 1000) and then authenticate the user
HELLO WORLD
Periodic Authentication
Authentication on each single Action
H
E
L
L
O
True Continuous Authentication

Figure 1.1: Periodic and Continuous User Authentication

Since the behavioural biometrics mostly depict the regular user behaviour while interacting with the relevant device, therefore these characteristics mostly rely on the hardware specification of devices, background context and user's emotion or age. The scholarly works, presented in literature review in domain of CUA using behavioural biometrics, mostly rely on statistical features based on mean and standard deviation of those features. These approaches had considered to maintain the static database of the relevant extracted features. However, this approach has few shortcomings: *Firstly* behavioural biometrics tend to change gradually with time or based on configuration and specification of different hardware devices. Therefore, the main disadvantage of maintaining a static database of users populated with statistical features could affect and decrease the performance or accuracy of system over time. *Secondly*, behavioural data i.e., keystroke and mouse dynamics, represents a sequential events of time-series which can contain hidden information regarding the specific behaviour of user which cannot be represented with statistical feature profiles of users as well as traditional classification methods cannot mine these type of features to distinguish one user from other.

1.2.1 Goal of this Research

This work aspires to investigate new techniques for continuous user authentication (CUA) in order to provide optimum security for the computing devices. With this aspect, the new possibilities are analysed for continuous user authentication or identification to overcome the drawbacks of the state of the art CUA systems. Additionally, the probability of rejecting a legitimate user and probability of accepting an attacker is investigated by the proposed CUA system. With this background, the following goals have been addressed in order to build a robust and reliable CUA/CUI system:

- **Continuous:** System should be able to monitor the users' computer usage incessantly, except the periods of deliberate pauses in keystrokes and mouse movements and actions by the relevant user, during its authenticated session on computer.
- **Real-time:** System could assess the user on each and every action performed on computing device. In order to do so, it should take into account or accumulate the confidence of user on all the previous actions as well. It should preferably decide, based on accumulated performance, if user can continue using the system or not.
- **Unobtrusive:** System should work in background without disturbing the user while he/she is performing the daily routine tasks on computer. For instance, user should not required to provide data or credentials after every set time duration in order to authenticate itself. System should work on continuous behavioural data collected from infused devices without requiring any additional equipment or attentions.
- **Robust:** System should be able to capture the distinct features of each user and trust the genuine user while identify the imposter with low errors.

1.2.2 Key Research Questions

In order to propose new architectural designs for CUA / CUI, some of the key research questions that were addressed in this thesis are:

1. Devising ways of continually verifying system users.
2. Investigating the impact of true continuous user authentication compared to state of art periodic authentication on security.
3. Establishing the identity of user without prior claim of identity at start of session.
4. Analysing the performance of the system using raw behavioural biometric data as compared to statistical user profiles.
5. Assessing the impact on system performance by using Deep Neural Networks as compared to traditional Baseline Methods.

1.3 Research Aim and Objectives

With the huge advances in the technology, issues of system hacking and theft of confidential information are escalating owing to the vulnerabilities in the security of the critical applications. Most of these susceptibilities are due to the one-time validation of user's identity by utilising the conventional user authentication practices like usage of passwords, tokens and PINs. For critical security systems, a continuous monitoring system is needed which can authenticate user on each action performed on system. In this aspect, the perception of employing behavioural patterns of user as biometric credential to escalate security is being investigated.

Alongside addressing the main issues highlighted above, the ultimate research aim is:

"To propose and implement an efficient continuous user authentication system using behavioural biometrics"

In order to implement efficient continuous authentication system, some of the key research objectives which have been addressed are:

1. To critically examine the constraints and drawbacks of state of art continuous authentication systems (CUA), and uncover the new possibilities for CUA to overcome the challenges within the traditional CUA.
2. To present a true CUA employing a proposed recurrent confidence module authenticating the user.
3. To analyse and implement the continuous user authentication using keystroke dynamics with baseline approach and techniques.
4. To propose the deep learning techniques in contrast to baseline approach to validate CUA with keystroke dynamics.
5. To analyse the continuous user authentication with behavioural biometrics based on traditional statistical features versus proposed temporal features.
6. To propose a method to establish the user identity continuously without prior claim of identity at start of session.
7. To investigate mouse Dynamics modality over CUA in order to explore CUA in comparison to other behavioural biometrics.
8. To apply baseline approach and deep learning techniques to validate CUA with mouse dynamics.

1.4 Contribution of this Thesis

The primary contributions made in this thesis are:

- Continuous authentication problem is not new in the research, however, the preceding research conducted in this domain had mostly focussed on periodic user authentication based on fixed block of actions which can give room to imposter user to perform illicit activities. In contrast, a true continuous user authentication mechanism based on each action is proposed and implemented in this work.

- A true CUA system has been proposed and implemented, consisting of robust recurrent confidence model (R-RCM). It takes into account each keystroke or mouse activity performed by individual in order to incessantly decide the legitimacy of user on each action. Moreover, the proposed R-RCM model uses a novel approach of detecting and locking out of imposter user once it crosses the alert threshold. The proposed system has been validated with keystroke and mouse dynamics.

- A continuous user identification (CUI) is studied and novel approach based region labelling and an idea of inserting blank label in area of low user confidence has been introduced for the first time.

- The usability and efficiency of various baseline and deep learning architectures in combination with newly proposed R-RCM model for CUA using keystroke and mouse dynamics have been investigated. Different novel system architectures are formulated.

- Combination of continuous user authentication and periodic user authentication is also studied for the time using deep neural networks.

- Behavioural data is utilised as a time-series and hidden behavioural features are also extracted along with per action based features.

- The recurrent neural network (RNN) is employed which to the best of our knowledge has not yet been studied for CUA using mouse dynamics and for CUI.

- Performance metrics have been proposed for true CUA system in terms of normalized portion of average genuine and imposter actions which can enhance the usability of these experimental results in future work done in domain of CUA/ CUI.

1.5 Thesis outline

In this thesis, the primary focus is how to validate the identity of user continuously on each action performed by user throughout the session. Specifically, the importance of performing the continuous authentication without disturbing the user during its session is recognised and different possibilities are explored to avoid the false lock out of genuine user as well as fast detection of imposter user to limit the damage caused to system resources. Additionally, novel techniques are researched to establish the identity of user without prior claim of identity at start of session. In this aspect, different continuous user authentication or identification schemes have been proposed and implemented based on baseline and deep neural networks. The proposed techniques have been validated with extensive experiments with different behavioural biometric modalities. The primary contributions of this work along with thesis structure are given below:

Chapter 2, details an in-depth literature review on continuous user authentication methods. It discusses the challenges and issues while designing the system for CUA methods, details the factors that caused unnecessary hindrance of the previous CUA schemes. Moreover, this chapter entails the importance of behavioural biometrics for CUA along with the challenges, feature selection and requirement of a true continuous user authentication system by exploiting a single action based authentication method.

Chapter 3, The continuous user authentication is introduced based on novel proposed Robust Recurrent Confidence Model (R-RCM) which tends to authenticate the user on each action in order to make the system a true CUA method. In this chapter, dataset based on keystroke dynamics and mouse dynamics have been studied along with the formulation of different dataset split strategies in order to investigate the effect of time intervals on behavioural data which has been collected during different sessions. The detailed structure of proposed R-RCM is given which can be applied to any biometric modality to achieve the true essence of CUA system. Moreover, the alternative performance measures for CUA systems are discussed based on Average Number of Genuine Actions (ANGA) and Average Number of Imposter Actions (ANIA) in terms of normalized mean actions which can make it easy to compare the work done in domain of CUA for future research.

Chapter 4, A Continuous User Authentication (CUA) using keystroke dynamics with proposed baseline techniques is presented. In this chapter, keystroke dynamics are used to implement a CUA system and system architecture is designed based on two phase methodology which integrates ensemble learning approach along with the new proposed Robust Recurrent Confidence Model (R-RCM). Different experimental settings have been formulated to achieve the optimal system performance.

Chapter 5, Continuous User Authentication (CUA) using keystroke dynamics with deep learning techniques are presented. In this chapter, different deep neural networks based on Long Short Term Memory (LSTM) have been trained using keystroke dynamics data. Moreover, novel different system architectures are proposed for a true CUA system based on deep neural networks and proposed R-RCM model.

Chapter 6 Continuous User Identification (CUI) is investigated without the involvement of static user authentication with usernames and passwords at start of session. The proposed R-RCM model is integrated with deep neural network to continuously establish the identity of user on each action. End-to-End (E2E) deep neural model using GRU + R-RCM are investigated for the first time for CUI problem.

Chapter 7, Continuous user authentication using Mouse Dynamics with baseline and Deep learning techniques is tested. In this chapter, mouse dynamics biometric modality is validated with our proposed baseline and deep neural network methods. Different experiments are conducted to evaluate the system performance. Moreover, mouse dynamics is also a less explored and emerging modality for CUA. In this regard, the formulated deep learning models have been tested for the first time in research for mouse dynamics using CUA.

Chapter 8, It discusses the conclusion, research findings and further work.

1.6 Publications

- Anum Tanveer Kiyani, Aboubaker Lasebae, Kamran Ali, Masood Ur Rehman, and Bushra Haq. Continuous user authentication featuring keystroke dynamics based on robust recurrent confidence model and ensemble learning approach. IEEE Access, 8:156177–156189, 2020.

- Anum Tanveer Kiyani, Aboubaker Lasebae, and Kamran Ali. Continuous user authentication based on deep neural networks. In 2020 International Conference on UK-China Emerging Technologies (UCET), pages 1–4. IEEE, 2020.

- Anum Tanveer Kiyani, Aboubaker Lasebae, Kamran Ali, and Masood Ur-Rehman. Secure online banking with biometrics. In 2019 International Conference on Advances in the Emerging Computing Technologies (AECT), pages 1–6. IEEE, 2020.

- Anum Tanveer Kiyani, Aboubaker Lasebae, Kamran Ali, Faisal Ahmad, Ahmed Alkhayyat. Continuous User Authentication Featuring Mouse Dynamics based on Ensemble Learning and Recurrent Neural Networks. IEEE Transactions on Information Forensics and Security, 2021. (Under Review)

- Anum Tanveer Kiyani, Aboubaker Lasebae, Kamran Ali. Continuous User Identification using Keystroke Dynamics based on Recurrent Neural Networks and region labelling approach utilising CTC. IEEE Transactions on Information Forensics and Security, 2021. (Under Review)

Chapter 2

Continuous User Authentication - An Overview

This chapter provides the background to the underlying concepts depicted in detail in the forthcoming chapters. It aspires to highlight the key concepts of user authentication, static and continuous user authentication, behavioural biometrics along with security and privacy aspects of computing devices. Moreover, it also gives an insight to the background of biometrics recognition systems. As a precursor to the following chapters, this chapter tends to build the prospect for a secure continuous authentication method by exploring the security loopholes which can be used by imposter users to get access to system resources and it is aimed to provide an in-depth insight into the key issues to be addressed in order to build a robust continuous user authentication system.

2.1 User Authentication - Background

User authentication is an important factor of computer and network security to ensue that only legitimate user has access to confidential information and critical system resources. Formally, User authentication can be defined as:

"When the user claims who he/she is and the system accepts (or declines) his/her claim."

The graphical representation of user authentication is shown in Fig 2.1 below:

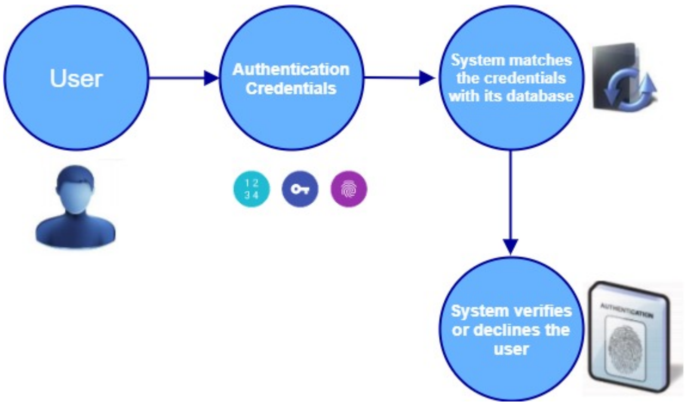

User
Authentication Credentials
1 2 3 4
System matches the credentials with its database
System verifies or declines the user
AUTHENTICATION

Figure 2.1: User Authentication Process

Presently, authentication systems incorporate multiple factors to verify the identity of user with associated credentials. Multi factor authentication (MFA) systems are considered to be more secure as compared to single factor authentication mechanisms since it provides more layers of security which given user has to pass through in order to get access to system Ometov et al. (2018). The basic concept of MFA is illustrated in Fig 2.2. Mostly, three factor authentication (3FA) is used which includes:

- ***Knowledge based factor***: It relates to something the user knows, such as a password, PIN or security questions.
- ***Possession based factor***: It relates to something the user has, such as cards, smartphones, or identity tokens.
- ***Biometric based factor***: It relates to something the user is, i.e., biometric data or behaviour pattern.

2.1.1 Types of Authentication

In this thesis, three types of authentication methods are discussed including:

- ***Static User Authentication (SUA)***: The system validates the identity of user once at the start of session i.e., login time.

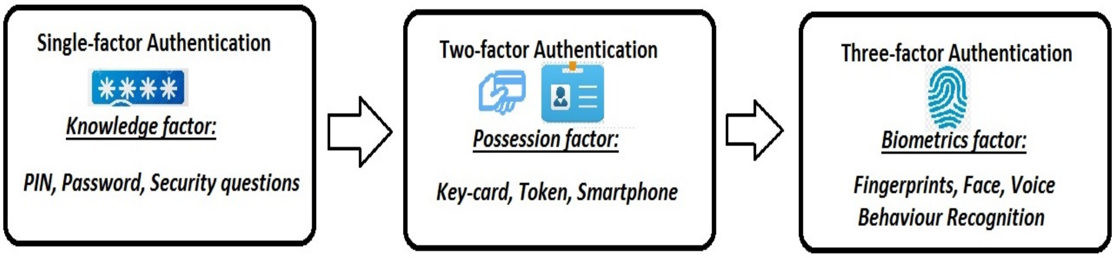

Single-factor Authentication
****
Knowledge factor:
PIN, Password, Security questions
Two-factor Authentication
Possession factor:
Key-card, Token, Smartphone
Three-factor Authentication
Biometrics factor:
Fingerprints, Face, Voice
Behaviour Recognition

Figure 2.2: Multi Factor Authentication (MFA)

- ***Periodic User Authentication (PUA)***: The system re-validates the identity of user after fixed block of actions or time intervals during the user active session.
- ***Continuous User Authentication (CUA)***: The system re-validates the identity of user at each action incessantly throughout the user active session.

The SUA methods define the process of initial identity validation at start of session or login time by utilising one or multi factor authentication scheme. However, these methods are still vulnerable owing to the fact that even though a strongest initial login is applied to give access to system resources. But if the user leaves the system unattended after the initial authentication or forgets to logout the authorised session, then any other individual can use the system on behalf of legitimate user and steal the important information.

On the other hand, periodic and continuous user authentication intends to verify the identity of user after the initial login to ensure that only legitimate user is using the system for the whole session. However, PUA validates the user's identity after fixed time intervals or fixed block sizes in contrast to CUA which can authenticate user on each activity or action. The key requirement for both PUA and CUA is that authentication process should not disturb the user while he/she is performing important tasks on system. For instance, user should not has to provide authentication credentials after some fixed interval to time in order to verify that he/she is the same user who has been authentication at the login time. This prerequisite rules out the incorporation of knowledge and possession based authentication credentials for PUA and CUA based authentication methods. Because for both, knowledge and possession based, authentication methods require the user to be actively engaged in authentication process by typing the password, PIN or by providing the token or card credentials.

On the other hand, some types of biometrics i.e., behavioural biometrics do not require the user to be actively engaged in providing the biometric data. Therefore, it can be used to periodically or continuously authenticate the user throughout the active session.

2.2 Biometric Recognition System

A biometric system is a system that takes biometric data from an individual, extracts the feature set and compares that feature set with the one that is stored in the database. It can be called as a pattern recognition system. Biometric systems follow two modes of operation namely identification and authentication. In the identification phase, the person identity is established by the system by searching the identity of the user in system database without any prior claim of identity by user. Subsequently, authentication phase of user verifies the user identity with the assistance of particular evidence provided such as username/password to determine whether this evidence or credential is counterpart of this username or not as given in registered database. Kiyani et al. (2020).

2.2.0.1 Types of Biometrics

Biometrics can be evidently divided into two categories named physiological and behavioural methods Yang et al. (2018) as shown in Fig 2.3.

- ***Physiological Biometrics***: Fingerprint, Facial, Iris, Retinal, and Hand geometry recognition.

- ***Behavioural Biometrics***: Keystroke dynamics, Mouse dynamics, signature, gait and sweat pores recognition.

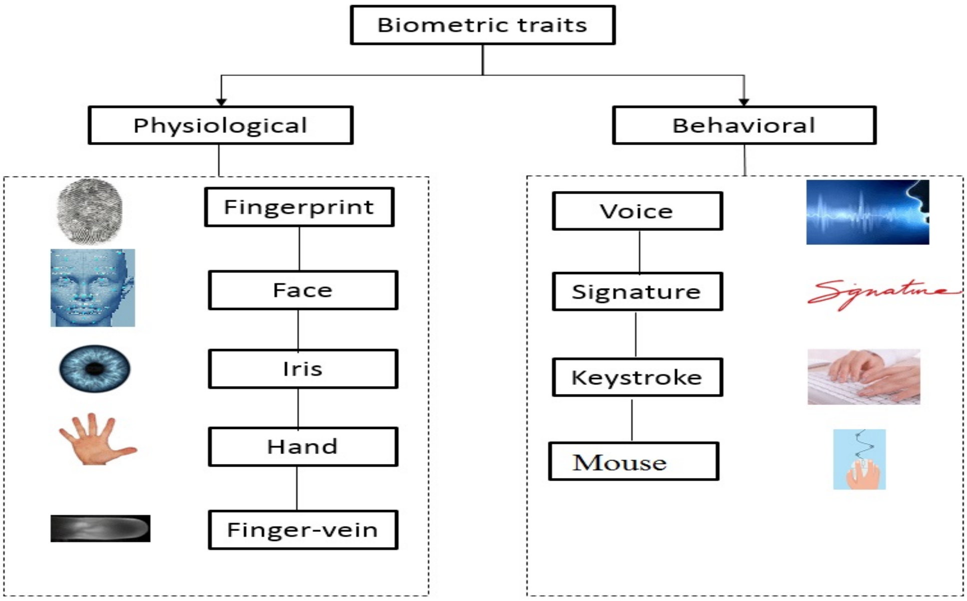


Figure 2.3: Biometric Traits Segregation Yang et al. (2018)

2.2.0.2 Phases of Biometric System

Biometric recognition system basically operates in the same manner regardless of the type of biometric used. However, collection of sample or its storage can differ in accordance to the nature of the biometric trait being used for authentication. The process of conventional biometric recognition system is illustrated in Fig 2.4 and it includes the following phases:

- Sample Collection
- Feature Extraction
- Template saved in the Database
- Matching or Comparison

- ***Sample Collection***

  Biometric recognition scheme instigates with the gathering of biometric data or sample which is meant to be used for authentication purpose. Biometrics data is collected using different devices depending upon the nature

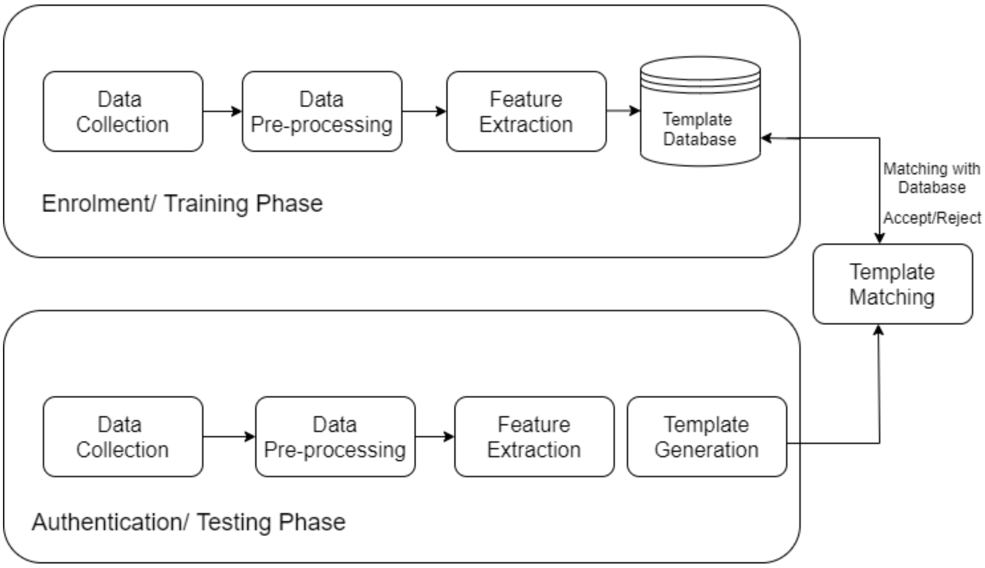


Figure 2.4: Phases of Biometric Recognition System

of biometric characteristic. For instance, a fingerprint scanner will be used for collection of fingerprints samples from users. On the other hand, behavioural biometric data can be collected using the keyboard and mouse devices. The collection step is mandatory since in any biometric system the user must have to register with the system so that every time for user authentication these samples will be used as template. Moreover, collecting several samples of a biometric characteristic from a single user and then selecting the quality one is often considered to be a good approach Singh et al. (2019). The rationale behind this strategy lies in the fact that poor sample can result in false rejection owing to the slight discrepancy which might be a consequence of different face expressions or the different pressure of fingers on scanner at the time of authentication and registration.

- ***Feature Extraction***

  After sample collection, different techniques are applied on the sample to extract the template. In computer systems, the whole sample is not stored in the database instead the important features of sample are extracted in the form of mathematical code and it is referred to be the template.

- ***Storage of Model pattern or template***

After template generation, it ought to be stored along with the other identifiers such as username or ID number in order to recover it in matching phase for comparison with the reference input for user authentication. Accordingly, these templates can be saved in any of the three locations:

- Back-end Database
- Smart card
- Biometric device

Back-end database is used when large numbers of templates are required to store, however, additional server is needed for this purpose. Subsequently, sometimes biometric device itself is used to store the templates; in this case, matching process can be faster as system will not have to wait longer for the server to retrieve the stored templates. Nevertheless, this practice is rational for small amount of data only. Often smart cards are also used to save biometrics onto it but this stratagem can be risky in case if card get lost. Among all these storage location, mostly backend system database is considered to be safe.

- ***Matching***

  When user wants to authenticate himself to access particular information or services of the application or system then biometric samples are taken from the intended persons at that time, these are known as live biometrics. Analogous to the registration phase, these samples are pre-processed and features are extracted followed by query template generation. Subsequently, this query template will be compared against the saved template or model pattern in the database. If both templates match and system returns the match score that is above the given threshold, then user is authenticated that he is the one who he is claiming to be thereby granting access to the specified services. Correspondingly, if the template will not match then authentication will fail and user would not be allowed to access the particular service.

2.2.1 Performance Measurement of a Biometric System

Performance of the biometric system depends upon the different measures such as:

- **False Acceptance Rate (FAR)**: It is the probability that the access would be given to the unauthorised user due to inaccurate classification.
- **False Rejection Rate (FRR)**: It is the probability that the access would be denied for the authorised user due to inaccurate classification.
- **Failure to Enroll Rate (FER)**: It refers to the percentage of users who could not accomplish the registration phase completely.
- **Equal Error Rate (EER)**: EER is a metric which assesses the data classification performance for any biometric model Yaacob et al. (2020). Subsequently, EER is considered to be a point where False rejection rate (FRR) and False acceptance rate (FAR) overlap each other as shown in Fig 2.5.

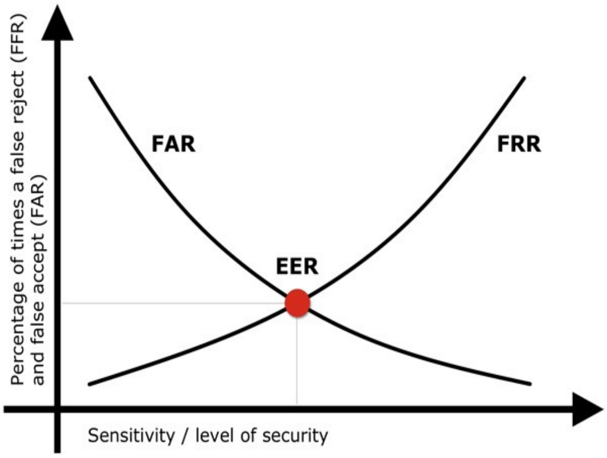


Figure 2.5: Equal Error Rate (EER) Yaacob et al. (2020)

2.3 Continuous User Authentication with Behavioural Biometrics

Behavioural biometrics are considered to be a type of biometric which depicts the certain behavioural patterns of given user according to the biometric trait which can be used to distinguish one user from the other. Keystroke dynamics and Mouse dynamics are interesting types of behavioural biometrics which have received special attention in recent years in biometric user authentication research. The advantage of using keystroke and mouse dynamics lies in the fact that:

- There is no need of any extra hardware to capture the behavioural data which makes it a cheaper solution to authenticate user with these biometric traits.
- Secondly, these behavioural biometrics can be employed to continuously monitor the legitimacy of user owing to its characteristic of passive verification that runs in background without disturbing the user while performing important tasks. On the other hand, physiological biometrics i.e., fingerprints require the user to be actively engaged in verification process. But there are few physiological biometrics i.e., facial recognition which require less user involvement in verification process but its accuracy depends on external factors like angle of camera, user posture, room light etc. In contrast, behavioural biometrics are completely transparent to perform authentication method passively.
- Moreover, it is difficult to spoof keystroke and mouse dynamics because typing patterns on keyboard will always differ for each user from other which makes it difficult for imposter user to perfectly emulate the way someone types.

Hence, keystroke and mouse dynamics are considered to be important biometric trait which can be utilised to distinguish one user from the other to lessen the risks of security attacks on confidential information.

2.3.1 CUA using Keystroke Dynamics

This section presents the speculative basis and preceding research works done in domain of keystroke dynamics for Continuous user Authentication (CUA) leading to the proposed system. Most of the preceding studies in the domain of keystroke dynamics had normally focused on the static user authentication (SUA) while the work done on continuous user authentication (CUA) is relatively far less. However, nowadays CUA is getting more prevalent owing to the security concerns of systems and applications as more people are dependent on computers and mobile devices for daily routine tasks including office work, online shopping, online banking and much more.

The presently available keystroke dynamics datasets can be specifically categorised into two types, namely, short text and long text, as shown in Fig 4.1. The short texts datasets are predominantly based on passwords thereby mostly appropriate for studying the SUA. Mhenni et al. (2019). On the other hand, the long texts datasets are further divided into two categories i.e., fixed text and free text. In this regard, former is based on pre-defined texts where user has to mimic the already provided tasks. On the contrary, the latter refers to the pattern in which users are given complete independence to employ any random text of any length without any constraints Alsultan et al. (2016).

Keystroke Dynamics Recognition (KDR) system is mostly based on two main events associated with the user's typing rhythm i.e., key down and key up events where former occurs when user presses a key while latter is recorded as soon as user releases that respective key Patel et al. (2016).Preliminary research on CUA using keystroke dynamics was conducted in 1995 by the group of researchers Shepherd (1995) and some notable results were presented.

These two keystroke events can be used to extract numerous different features in order to make the unique feature set of the user. In this aspect, the most frequently used features in the literature are single key hold time and key digraph latency which is the duration between the given two consecutive keystrokes as shown in Fig 2.7.

User templates are created by calculating the mean and standard deviation of

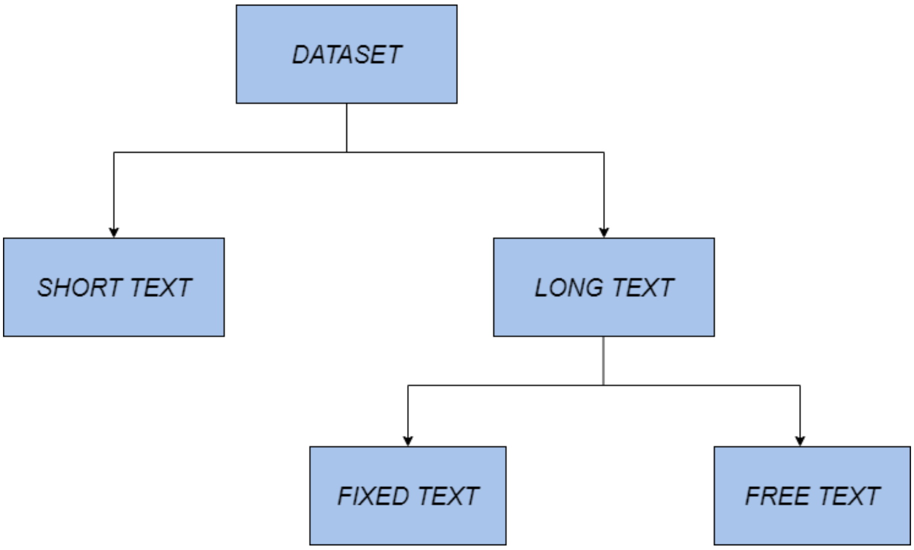


Figure 2.6: Keystroke Dynamics Dataset Classification

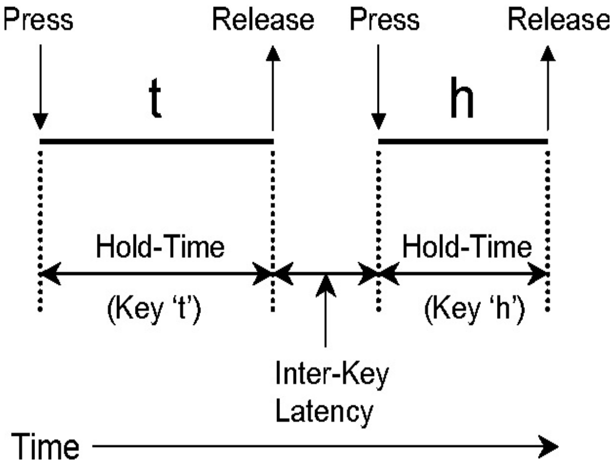


Figure 2.7: Keystroke Hold time and Key Digraph Latency

each key hold time and key digraph latency times Foresi and Samavi (2019) . On the other hand, some research studies Bours and Barghouthi (2009) had featured the mean and standard deviation of only those digraphs which had occurred least number of times in order to build the inimitable feature set. Moreover, the researchers in Di Tommaso et al. (2019) had employed the combination of key digraph, trigraph, error corrections and words per minute features to build the user profiles. Additionally in some studies Senathipathi and Batri (2014) feature set had been extended to include digraphs, trigraphs and some additional allied n-graphs. While some researchers had used the specific words which are common in English i.e., the, an, and, to, etc., to extract the features set Curtin et al. (2006). Moreover, in Salem and Obaidat (2019) researchers had combined the timing features with non-timing features i.e., pressure, position, finger placement and finger choice for tying behaviour analysis.

Once the feature set had been extracted, the next step followed is the classification. Many classification techniques had been used for continuous authentication including traditional statistical methods, pattern recognition and even more complex machine learning methods.

2.3.2 Traditional Statistical Distance Methods

The summary of research works performed on CUA problem with traditional statistical distance methods is presented in Table 2.1 while the detailed results are discussed below.

The researchers in Gunetti et al. (2005) conducted the free-text studies with digraphs, trigraphs and n-graphs as statistical features and it was essentially dependent on two underlying distance measures namely relative measure and absolute measure. The former was used to calculate the degree of disorder whereas the latter referred to the measurement of absolute distance between two keystroke samples and achieved the good results. However, they had used the block size of 700-900 keystrokes to form each sample probe to identify the user which gives enough possibility to imposter for unauthorised access and achieved the FMR of 0.005% and FNMR 4.833%.

Some other research works had also implemented the relative distance and absolute distance including: Huang et al. (2017) with sliding window of fixed n-graph latency features and achieved the FMR = 1% , FNMR = 11.5%. Kolakowska (2011) with 600 block size, duration of 2,3,4 and 5-graph features and reported the FMR 4.09% , FNMR 5.17%. Pinto et al. (2014) with 150 block size, duration of digraph RP , PP for (2-4) graphs and achieved FMR 2% , FNMR 2%. Ayotte et al. (2019) with 1000 block size, di-graph latency features and reported the EER= 3.6 % precisely, and Ferreira and Santos (2012) with block size of 250 actions, duration of digraph latency and reported the EER = 1.4%.

The researchers in Ferrari et al. (2018) had presented an adaptive continuous authentication scheme by building the statistical profiles of users using the single key, UD and DU features for only selected keys and key-pairs. They had reported the results for fixed window sizes i.e., 35, 50, 65, 80, for authentication as well as updating the statistical profile by using Euclidean distance, Manhattan distance and cosine similarity metrics. The optimal results achieved were FAR= 8.33 %, FRR = 40.54% with Euclidean distance while FAR= 17.24 %, FRR = 29.72% for Manhattan distance with window size of 50.

Other statistical methods used for classification of keystroke dynamics in literature were Euclidean distance Ferrari et al. (2018) with reported results as FAR=8.33% and FRR=40.54%.

Moreover, researchers in Locklear et al. (2014) employed Manhattan distance and reported the EER of 4.55-13.37% with varying time based blocks ranging from 30 seconds to 3.5 minutes. Moreover, the researchers in Kim and Kang (2020) used three types heterogeneous features based on time, accelerator and coordinate to generate the feature set and Kolmogorov-Smirnov statistic had used as classification technique. Different varying length of keystroke sets are used to achieve the EER = 1%.

2.3.3 Traditional and Advanced Machine Learning Methods

Machine learning techniques had also been exploited in recent times for CUA domain using KD where some of the works presented interesting results. The summary of research works performed on CUA problem with machine learning is

Work	Users	Features	Block	Method	Results
Gunetti et al. (2005)	40	2, 3, 4-graph latency	700-900	R- and A-distances	FMR 0.005% FNMR 4.833%
Huang et al. (2017)	56	n-graph latency	1 min sliding window	R- and A-distances	FMR 1% FNMR 11.5%
Kolakowska (2011)	10	2, 3, 4 and 5-graph latency	600	R- and A-distances	FMR 4.09% , FNMR 5.17%
Pinto et al. (2014)	10	Digraph RP, PP for 2-4graphs	150	R- and A-distances	FMR 2% FNMR 2%
Ayotte et al. (2019)	103	di-graph latency	1000	R, A, Mahalanobis distances	EER 3.6%
Ferreira and Santos (2012)	60	di-graph latency	250	R- and A-distances	EER 1.4%
Ferrari et al. (2018)	60	di-graph latency	Window size=50	Euclidean distance	FAR= 8.33 %, FRR = 40.54%
Locklear et al. (2014)	489	digraph latency	Time blocks of var.lengths	Manhattan distance,Fisher score	EER 4.55-13.37%
Kim and Kang (2020)	50	3-heterogeneous features	keystroke sets	Kolmogorov-Smirnov statistic	EER 1%

Table 2.1: Scholarly Works based on Traditional Distance Methods using Keystroke Dynamics

presented in Table 2.2 while the detail results are discussed below.

A constructive example of Machine learning techniques for CUA with KD was presented in Ahmed and Traore (2013) with neural networks implementation. They had used 500 keystroke block size with digraph features and employed the strategy of predicting the timing of digraphs in testing which had never occurred while training the network and achieved the FAR= 0.0152%, FRR = 4.82% and EER = 2.13%. Another research work in Alsultan et al. (2017) had implemented Decision trees with statistical feature profiles and used the block size of 1000 actions and reported the FMR= 1.1% and FNMR= 28%. Moreover, in Wu et al. (2016) kernel ridge regression a truncated RBF kernel had been used with 900 words block size and trigraph latency feature profile and reported EER of 1.39%.

Subsequently, support vector machine technique had also been exploited by researchers in Çeker and Upadhyaya (2016) with varying digraph sets for implementing CUA and acheived the EER of 0.0- 2.94% with different sets of digraphs. The researchers in Manandhar et al. (2019) had implemented an architecture named Spy Hunter for CUA using KD which utilised two 1-class support vector machines classifiers. They had used a single key hold time and digraph latency to build the feature vector and block size of 6 actions are used to classify a user after each block. The resultant FAR reported was 2.05% and FRR was 2.0%

Additionally, random forest classifier had been used in Ayotte et al. (2020)

with block size of 200 keystroke actions and the resultant EER as reported was 7.8%.

Moreover, the researchers in Porwik et al. (2021) had implemented the competitive selection ensemble classifier approach based on Random Forests (RF), Bayes Net (BN) , decision trees , Support Vector Machine (SVM), Random Tree (RT) and RIDOR RIpple-DOwn Rule learner (Ridor). They showed that employing an ensemble approach as compared to stand alone classifiers can improve the accuracy of system because keystroke dynamics being a weak behavioural biometric modality suffers from behavioural invariability issue. They had reported the FAR= 0.10% and FRR = 0.22% with ensemble classifier.

Lu et al. (2020) had employed the deep neural architecture consisting of convolutional neural network(CNN) and Recurrent neural network (RNN) on the free text dataset and achieved EER= 4.77%.

Work	Users	Features	Block	Method	Results
Ahmed and Traore (2013)	53	Duration, digraph latency	500	Neural Network	FAR 0.0152% FRR 4.82% EER 2.13%
Alsultan et al. (2017)	30	Statistical features	1000	Decision Trees	FAR 1.1% FRR 28%
Wu et al. (2016)	200	Trigraph latency	900 words	Kernel Ridge Regression	EER 1.39%
Çeker and Upadhyaya (2016)	34	Digraph latency	14 Digraph set	Support vector machine	EER 0.0 - 2.94%
Manandhar et al. (2019)	20	hold time, Digraph latency	6 block actions	one class Support vector machine	FAR = 2.05% FRR=2%
Ayotte et al. (2020)	103	Digraph latency	200	Random forest classifier	EER 7.8%
Porwik et al. (2021)	150	Digraph latency	—	Ensemble Classifier	FAR 0.10% , FRR 0.22%
Lu et al. (2020)	75	Digraph latency	sliding window	CNN, RNN	EER 4.77%

Table 2.2: Machine Learning Scholarly Works for CUA with Keystroke Dynamics

2.4 Continuous user Authentication with Mouse Dynamics

This section presents the background knowledge and preceding scholarly works in the domain of mouse dynamics leading to the proposed system. The most significant scholarly works done in domain of mouse dynamics are listed in Table 2.3.

The preliminary research on mouse dynamics had initially been started with the successful implementation of user verification system Syukri et al. (1998) on the basis of signatures drawn via mouse with the resultant identification rate of 93%. Afterwards, mouse biometric technology had been firstly introduced by researchers in Gamboa and Fred (2004*a*) as a method of non-signature based user verification system by utilising the general behavioural patterns of individuals while working with mouse.

A continuous authentication approach had been presented which considered each movement as an action and extracted features from each mouse stroke. The feature space consisted of 63 dimensional feature vector including spatial and temporal parameters such as angle, velocity and acceleration. However, the best subset feature set was chosen, by employing greedy feature selection procedure, for each user to reduce this feature space. Statistical model was used to generate the resultant authentication decisions based on mean classification scores of sequence of mouse actions. Data was collected from 50 users working under free environment and experiments were conducted on sequence of 1 action, 50 actions and 200 actions which had produced an EER rate of 48.9%, 2% and 0.2% respectively. The researchers in Ahmed and Traore (2005), Ahmed and Traore (2007) had first time converted the mouse data into meaningful distinct seven feature sets which were aggregated afterwards into 39-dimensional global feature vector. The 3-layer artificial neural network (ANN) had been trained for behavioural comparison of genuine and imposter users which achieved the false acceptance rate (FAR) and false rejection rate (FRR) of 2.4649% and 2.4614% respectively.

Scholarly Work	No. of Users	Features	Number of actions	Classification Model	Main Results
Gamboa and Fred Gamboa and Fred (2004*a*)	50	Statistical	1, 50 , 200 action	Statistical model	EER:48.9%, 2%, 0.2%
Pusara et al. Pusara and Brodley (2004)	11	Statistical	1000 actions	Decision Tree	FAR 1.75%, FRR 0.43%
Ahmed and Traore Ahmed and Traore (2007)	22	Statistical	2000 actions	Artificial Neural Network	EER: 2.46%
Feher et al. Feher et al. (2012)	25	Statistical	30 mouse actions	Random Forest	EER: 8.53%
Zheng et al. Zheng et al. (2016)	30	Statistical	20 mouse actions	SVM	EER: 1.3%
Hinbarji et al. Hinbarji et al. (2015)	10	Statistical	100 mouse curves	Artificial Neural Network	EER: 9.8%
Antal et al. Antal and Egyed-Zsigmond (2019)	10	Statistical	13 mouse actions	Random forest	EER: 2.6%
Mondal et al. Mondal and Bours (2017*a*)	53	Statistical	1 mouse action	SVM, ANN	ANGA= 2265 , ANIA=252
Antal et al. Antal and Fejér (2020)	10	Time-series	128 mouse events per block	CNN, Transfer Learning	accuracy=0.93

Table 2.3: Most Important Existing Scholarly Works on Mouse Dynamics

A re-authentication scheme had been presented Pusara and Brodley (2004) which converted the raw mouse data into processed data points representing the summary of mouse events encompassing configurable window size. In this aspect, mouse data was collected from 11 users under free working environment and personalized model had been generated for each user by employing the decision tree classifier to report the average FAR of 1.75% and average FRR of 0.43%.

Subsequently, new types of feature set for mouse dynamics recognition had been assembled Feher et al. (2012) based on the hierarchy of mouse actions where low-level features were used to construct the higher level features. Behavioural analysis had been done by training the Random forest classifier with newly assembled hierarchical features and an EER of 8.53% was achieved with window size of 30 mouse actions.

Zheng et al. Zheng et al. (2016) employed angle based statistical features for user authentication purposes. In this regard, data had been collected from 30 users of different ages, professions and educational background. Moreover, SVM had been used as a classification method. The best performance reported, for a block of 20 mouse actions, was EER rate of 1.3%.

The research work presented in Hinbarji et al. (2015) collected general behavioural mouse usage data from 10 users and employed mouse movement curves as basic feature for user authentication problem. Neural network was trained as a binary classifier for each user and undertaken 100 mouse curves i.e., approximately 5.6 min of mouse data, to obtain the EER of 9.8%.

Antal et al. Antal and Egyed-Zsigmond (2019) proposed a user re-authentication method involving the data of 10 users. Segmentation of data had been done to generate mouse events including mouse move, point and click, drag and drop(as shown in Fig 2.8), followed by extraction of statistical features from each mouse event. Random forest is employed as a classifier model and EER of 2.6% had been achieved by using the window size of 13 mouse actions.

The researchers in Mondal and Bours (2017*a*) presented the work on CUA which authenticated user on each action. The dataset involved 53 participants and employed SVM & ANN as a classification method. The system performance was presented in terms of average number of genuine actions (ANGA) and average number of imposter actions (ANIA). Moreover, ANGA had been reported as 2265 actions and ANIA as 252 actions on average.

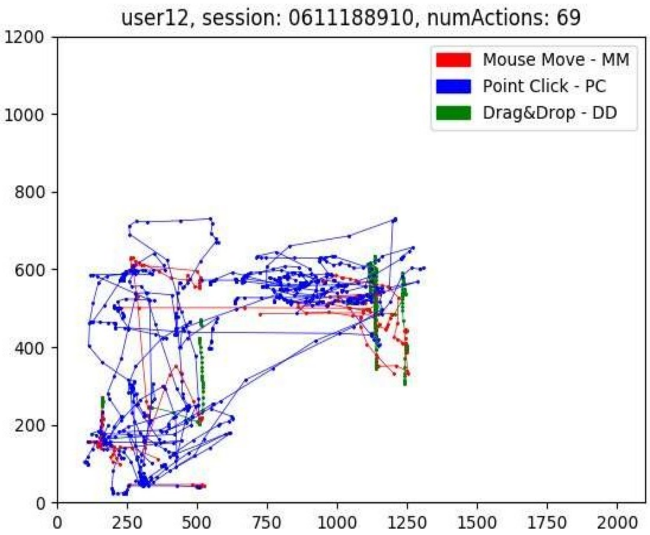


Figure 2.8: Mouse Events Signature of a User
Antal and Egyed-Zsigmond (2019)

The work in Antal and Fejér (2020) had proposed an authentication method based on convolutional neural network and transfer learning approach. The features were extracted based on time series of mouse events instead of formulating the statistical features. In this regard, Balabit public dataset was used for performance evaluation. The optimal accuracy reported was 0.93.

2.5 Continuous user Authentication Challenges

The following challenges for a true CUA system have been indicated after reviewing the state of art works:

1. **Authentication using fixed blocks of action or fixed time interval**

   It has been observed that most of the research works in CUA domain using keystroke or mouse dynamics had considered the block of actions (200, 1000, 2000 etc.) or fixed time intervals (1 min sliding window etc.) to authenticate the user. The main disadvantage of this type of authentication approach is that it gives room to imposter user to perform illicit activities on system because identity of user would not be checked for given block size, hence this type of method can be basically called as periodic user authentication (PUA). However, the PUA method might be insufficient to guarantee security depending on the risks in a particular environment. During the long pauses of un-authentication, the impact of intruder taking the session as a substitution of legitimate user could be inauspicious and discernible. Therefore, the PUA method is less secure and can cause the damage to confidential information or resources of system.

In order to mitigate this issue, an authentication method is needed which can re-verify the identity of user on each single activity or action in order to fulfil the requisite of a true continuous user authentication method.

2. **User behaviour can vary on each action**

   Biometric characteristics belonging to behavioural category are more likely to alter over time as compared to physiological features. Since the behavioural biometrics mostly depict the regular user behaviour while interacting with the relevant device, therefore these characteristics mostly rely on the hardware specification of devices, background context and user's emotion or age. Depending on external factors user behaviour on each action can change and it become difficult to validate the user's identity on the basis of one single action which is an important requirement of a true CUA system. For instance, a legitimate user can deviate from its normal behaviour on certain actions owing to the external factors i.e., distraction or noise in background. Therefore, it can affect the performance of system. In order to solve this issue, a true CUA system is needed which ought to validate the identity of user on each action but it should also keep in account the previous confidence of user's genuineness.

   In this regard, this research work proposes a recurrent confidence model which authenticates the user on each single action but decides the legitimacy of user in combination with previous actions' confidence.

3. **Continuous user Identity Establishment**

   It has been observed in literature review that most of the CUA approaches had integrated the static user authentication in the start of session. However, the work done on continuous identification based on establishing the identity of user without involving SUA is relatively far less.

This research work explores the new possibilities of identifying the user without prior claim of its identity or without involving SUA at the start os session.

4. **Performance metrics for a true CUA system**

   The PUA method, as studied in literature review, generally report the performance in terms of false acceptance rate (FAR), false rejection rate (FRR) and equal error rate (EER) Bakelman et al. (2012) for CUA biometric systems. However for true CUA, the identity of user should be checked on each single action and performance measure should depend on how many actions imposter or genuine user has performed before system detects it or falsely locks it out respectively. Based on general understanding, the number of actions executed by different users within a particular time frame substantially relies on individual's explicit behaviour patterns and this factor is distinctive among different users. For example, a person with fast typing speed would be able to perform more actions on system resulting in more damage to system resources as compared to a user with slow typing speed within any given time period.

   Therefore, it has been decided to report the performance of proposed CUA system in terms of action domain instead of considering the time complexity of identifying the imposter users.

5. **Static Feature Database**

   It has been noticed in literature review as listed in Table 2.1 and 2.2 that most of the existing research works in keystroke and mouse dynamics domain have considered the statistical feature extraction process to generate the feature vector for each user. Researchers have generally extracted features based on mean and standard deviation of specific keys and key-pairs or raw mouse events including direction, angle of curvature, distance, mouse click and drag timings. This type of method has achieved satisfactory results, however, it has few shortcomings. The statistical profiles contains the mean and variance of keys and key pairs. This type of approach is better suited for fixed text, on the contrary, for the free text where user must be using some key-pairs which are missing from statistical profile can lead to low accuracy.

In contrast, this research work has considered the approach of taking keystroke dynamics data as sequential series and analysing the user behaviour serially instead of measuring on statistical profile.

Moreover, in most preceding works, commonly used features for CUA with keystroke dynamics are digraphs as listed in Table 2.1 and 2.2. However, if the real continuous authentication is considered which authenticates the user on each single action then in this case monographs have special place since it tends to authenticate the user on each single action instead of after two actions performed within given time frame thus leaving no room for imposter user. But digraphs had seen to give more better results Bours and Mondal (2015*a*) so the optimal approach used in this research is fusion of monographs as well as digraphs to achieve better results.

With regards to mouse dynamics feature sets, researchers have used their own experiments and understanding to choose the features. For instance, some research works have considered only mouse actions, some have considered the distance between mouse curves or other basic mouse events. But there is still raw mouse data which has been ignored that can represent the instinct properties of each user's mouse usage behaviour and can be helpful to distinguish one user from other with greater accuracy.

Therefore, it has been decided to consider the behavioural keystroke and mouse dynamics data as a set of chronological time-series for the proposed CUA system which can also utilise all the hidden properties of data.

6. **Behavioural Biometric data contains hidden unique behavioural patterns**

   Keystroke dynamics and mouse dynamics are behavioural biometric modalities hence these tend to change more with time, age or background context in comparison to physiological biometrics. Therefore, maintaining a static database containing the mean and standard deviation of statistical features can lead to low system performance over time. Keystroke and mouse dynamics data are more like a sequential series containing some hidden properties as well Tse and Hung (2020). For example, it can be general mouse usage behaviour of a user that when he wants to open a file document on system he always used to double click the mouse left button. On the other hand,there can be another user who has the habit of firstly clicking the right button of mouse to go to the option of properties and then afterwards choosing the OPEN option from the dialogue box. This sort of hidden features or combination of hidden features can be used to differentiate users from each other. Subsequently, the traditional classification algorithms cannot mine this kind of hidden features and these cannot be stored into statistical feature profiles.

   In this work, a method is proposed and implemented which can preserve the necessary information about keyboard and mouse usage behaviour of user and tends to update the features with time. The proposed approach uses the deep learning techniques and incorporates the proposed R-RCM to make it a true CUA system.

2.6 Summary

In this chapter, preliminary research works done in continuous user authentication (CUA) using behavioural biometrics have been reviewed followed by finding the gaps in current scholarly works. It can be summarise that most of the research works had considered the approach of periodic user authentication based on fixed block of actions or fixed time intervals which does not depict the true insight of CUA which ought to authenticate user on each action. Moreover, feature extraction techniques are mostly based on traditional statistical methods and researcher's own experiments which sometimes tends to omit the hidden properties of normal user behaviour. In order to overcome these issues, a basis of a true CUA system is presented in next chapter based on proposed robust recurrent confidence model which can authenticate the user on each action incessantly.

Chapter 3

Continuous User Authentication based on novel Robust Recurrent Confidence Model(R-RCM)

Chapter 2 has given an overview of existing research into the domain of continuous user authentication (CUA) based on behavioural biometrics especially keystroke and mouse dynamics. It has highlighted the challenges and requirements which need to be considered in order to overcome the loopholes present in the existing CUA systems. In this chapter, a novel model is proposed and explicated which identifies the unique features of behavioural biometric modalities and defines the requirements of a true CUA system. Moreover, different continuous authentication scenarios are also explained. This chapter presents the baseline of this thesis work and explains the requirements along with the main components of proposed CUA system.

3.1 Introduction

User authentication is imperative to the security requirement in computer systems. Most of the computer systems are mainly reliant on conventional one-time validation of user's identity methods, for instance, passwords, usernames, fingerprints or facial recognition. However, this approach cannot validate the user's identity throughout the given session thereby fosters the security threats in high risk security environments. In contrast, the Continuous User Authentication (CUA) validates individuals on each action performed by them incessantly. Existing CUA systems are mostly built around the usage of behavioural biometrics of user to validate its identity on each action. However, there are number of issues associated with current CUA systems which limit the performance in detecting imposter users. These limitations include:

- Applying the authentication algorithms on set/block of fixed actions hence giving room to imposter users.
- Variability of behaviour biometric data between training and testing samples
- Variability of behaviour biometric data depending on background context.
- Classification techniques cannot mine the hidden unique features of user.

In this research, a true continuous user authentication system was investigated and implemented after identifying the limitations of existing CUA systems. The generic framework of proposed CUA system is shown in Fig3.1.

3.2 System Description

This section presents the architecture of proposed CUA system in more detail. The main components of the proposed CUA system are:

- **Dataset:** It consists of the subjects/individual who are initiator of any activity on target system and behavioural data collected from those subjects while interacting with system.
- **Sensor:** It is the device through which training and testing behavioural data of user is collected and translated into a signal which is readable for an observer. In case of this proposed CUA system, these sensor devices are keyboard and mouse through which users' behavioural data is collected.

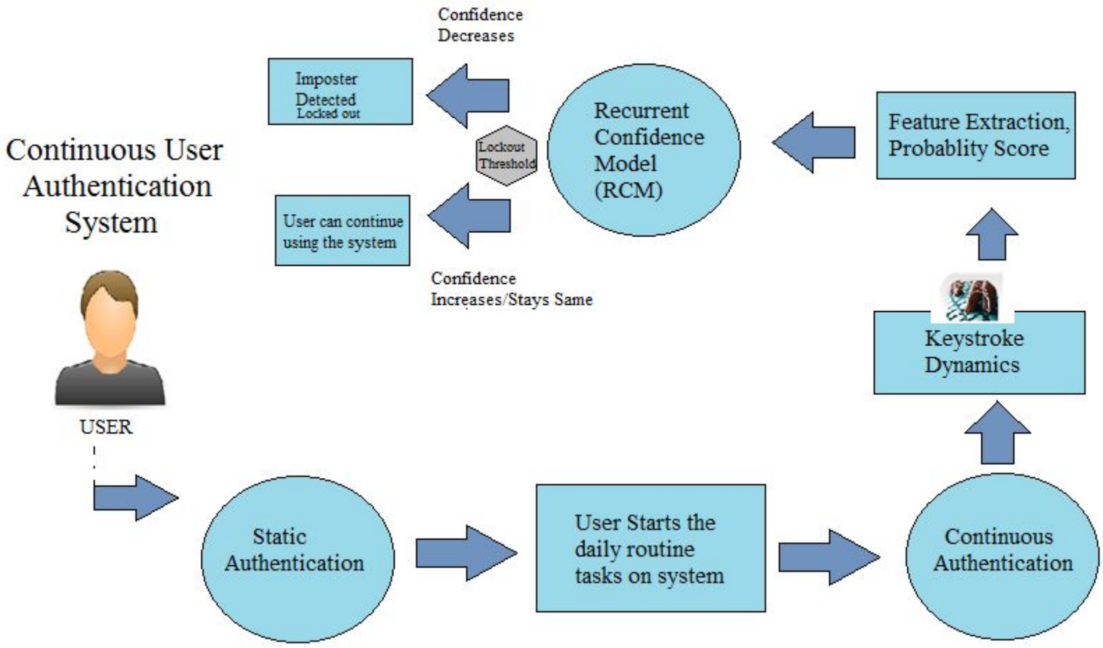

Confidence Decreases
Imposter Detected Locked out
Lockout Threshold
Recurrent Confidence Model (RCM)
Feature Extraction, Probablity Score
Continuous User Authentication System
User can continue using the system
Confidence Increases/Stays Same
Keystroke Dynamics
USER
Static Authentication
User Starts the daily routine tasks on system
Continuous Authentication

Figure 3.1: General Framework of Proposed CUA System

- **Feature Extraction:** It is a process of extracting different distinct features from raw dataset which can be utilised to distinguish one user from other.

- **Detector Unit:** It is the process of comparing the training feature profiles with testing data and performing the data classification measurements i.e., traditional machine learning or advanced deep learning, to calculate the error in order to detect imposter.

- **Recurrent Confidence Unit(RCM):** It is basically a component of detector unit which validates the identity of user on each action.

- **Response Unit:** It is the process of taking appropriate response based on calculation of detector unit in order to detect the intruders. In the proposed CUA system, response can be of two types:

  - Making the system alert regarding the risk of imposter attack
  - Locking out the detected imposter from system

- **Scenario/Threat Models:** It is an additional component of proposed CUA system which scenario different threat possibilities.
- **Performance Measure:** System performance for the proposed CUA system is discussed in terms of action based detection of imposter users instead of time based detection.

Each of the component of proposed CUA system as listed above are discussed in more detail below:

3.2.1 Dataset

Dataset mainly consists of biometric data collected from the subjects under study. Subjects are normal users which are basically the instigators of any action on a system. These subjects can be either authorised or unauthorised where the former are permitted to access the system resources with their biometric data already saved in user database while the latter are external to system having no registered data in the system. More specifically, unauthorised users are mostly the intruders who want to access the system resources by proliferating the security loopholes or by masquerading the authorised user behaviour.

In this research, the keystroke and mouse dynamics dataset provided by University of Buffalo Sun et al. (2016) have been used. The baseline datasets are collected from 75 subjects in 3 separate sessions. The statistics of keystroke and mouse datasets are presented in table 3.1 and table 3.2 respectively.

Property	**Mean ± Std**
Total Users	75
Keystrokes per User	16348 ± 1766
Total Keystrokes	1.2M
Up Time[t] - Down Time[t] (Hold Time,ms)	119 ± 18
Up Time[t] - Up Time[$t-1$] (UU,ms)	501 ± 383
Down Time[t] - Down Time[$t-1$] (DD,ms)	501 ± 383

Table 3.1: Keystroke Dynamics Dataset Statistics

There are 28 days in average time interval between each of the three sessions. The keystroke dynamics dataset is based on long-text and it is the mixture of fixed and free style texts.

Property	**Mean ± Std**
Total Users	75
Actions per User	27600 ± 7336
Total Actions	2.1M

Table 3.2: Mouse Dynamics Dataset Statistics

3.2.1.1 Dataset Split

For the analysis of CUA system, the data of a user is split into 3 parts. The first part is used to create the template, i.e. train the classifier to build a model. The second part of the data will be used for testing for parametric adjustments and validation part is used for final evaluation of unseen data. The validation data of a user is used action by action and each action will determine a change in confidence of user being genuine or imposter.

Two types of split ranges are considered which include *Across Session* and *Across Sequence* of given keystroke mouse sequences. However, in both types of split ranges, the following rule is considered:

$SplitRange = [SplitRange_0, SplitRange_1)$,
$SplitRange_0 \geq 0$, $SplitRange_1 \leq 1$,

hence, $SplitRange_0 < SplitRange_1$.

The two dataset split ranges used in this research are given below:

1. ***Across Sequence Split***

    - T=train, $SessionId(E)$, $SplitRange = [0.0, 0.6]$
    - X=test, $SessionId(E)$, $SplitRange = [0.6, 0.8]$

- V=val, $SessionId(E)$, $SplitRange = [0.8, 1.0]$

2. ***Across Sessions Split***

- T=train, $SessionId(E) = [0, 1]$, $SplitRange = [0.0, 0.7]$
- X=test, $SessionId(E) = [0, 1]$, $SplitRange = [0.7, 1.0]$
- V=val, $SessionId(E) = [2]$, $SplitRange = [0.0, 1.0]$

The ***Across Sequence Split*** takes whole data of all the 3 sessions altogether and then apply the partition rule for training testing and validation. This dataset split strategy can be utilised to study generic user behaviour. On the other hand, the ***Across Sessions Split*** takes the user' first two data sessions [0,1] for training and testing purpose with the proportion of 70% and 30%. The third session [2] is used for the validation purpose. There is an average difference of 28 days between each session, therefore, difference between 1st and 3rd session is approximately 56 days or 2 months, hence this dataset split strategy can be utilised to study the effect of time difference in user' behaviour.

3.2.2 Feature Analysis

Feature extraction refers to the depiction of biometric data and different variables which can be used to distinguish one user from the other. In this regard, feature selection is considered to be an active research area in different domains. The main notion of feature selection technique is to extract a subset of features which has more importance in prediction than the whole raw data. However, opponents of feature selection techniques believe that feature selection is based on researchers' individual experiments and the main disadvantage could be the ignorance of some hidden information which can only be represented with the help of raw data. In this research work, two types of feature analysis techniques have been used:

- Feature selection which are representative of a user's behaviour
- Feature analysis from raw dataset

In feature selection techniques, different statistical features have been extracted to make the user profile. On the other hand, feature analysis from raw dataset has treated the raw data as a time series. A time-series is a sequence of events arranged by chronological order and each new event shows some relationship with previous event.

3.2.3 Detector Unit

The detector unit is composed of two components i.e., classification process and our newly proposed confidence model for a true CUA system.

3.2.3.1 Classification Process

The first phase of detector unit basically performs classification process to get the error rate in order to detect any intruder activity based on given behavioural biometric data. This is a complex step and where it is performed on both training and testing sets / registration and identification steps respectively. In this work, two types of classification algorithms are applied to authenticate user continuously.

- Traditional Machine Learning/ Baseline Techniques
- Deep Neural Networks

The first method is based on traditional machine learning algorithms which treats the input as independent vectors and this method is referred as *baseline method* throughout this work while the second technique is based on deep learning which treats the data input as time series to extract the hidden distinct features of user where each event is dependent on the previous event and can predict the new event as well.

Since this research is focused on validating the user's identity on each action in order to propose a true CUA system, therefore classification is performed on action-by-action basis. The classification score of each action is given as an input to the newly proposed Robust Recurrent Confidence Model (R-RCM) which is the second phase of detector unit and explained below in more detail.

3.2.3.2 Robust Recurrent Confidence Model (R-RCM)

Most of the work done in CUA systems, as observed in literature review, consider the sliding window approach with block of actions. In that case, system waits until the block is filled up with specified number of actions and only then the legitimacy of user is decided based on full block of actions. However, this approach gives room to imposter users to do the damage to sensitive information for the given action block size. In this regard, it was proposed to use the robust Recurrent Confidence Model (R-RCM) which considers each and every action of user in order to decide if a particular user is legitimate or not. However, each action does not make this decision alone but R-RCM takes into account the confidence generated by previous actions as well. When considering behavioural biometrics, even genuine users can deviate from their normal behaviour owing to the changing background context and similarly imposter users can behave exactly as the genuine users on some of the actions. Hence, the typing behaviour of any user is never completely stable all the time that's why deciding the legitimacy of user on single action leads to low accuracy. But since no two users can ever type exactly in the same manner to each other and at some points the behaviour of imposter will differ from the normal behaviour of genuine user noticeably and is quite enough to differentiate between the two users in order to detect the imposters. To implement this strategy, the author used the concept of "recurrent confidence in the genuineness" of the current user.

In Bours and Barghouthi (2009) researchers had used the similar approach of trust model for CUA based on threshold function. The results of the study showed that the trust level escalates or lessens based on the scaled Manhattan distance between the legitimate user reference template and current typing actions. In this research work, a *Robust recurrent confidence model (R-RCM)* is proposed which keeps track of previous confidence value and tends to lock out a user from the system once it reaches the final lockout threshold. Confidence value depends on the fused classifier score from the detector unit.

A Novel Approach of Robust Recurrent Confidence Model(R-RCM)

As stated above, CUA cannot substitute the SUA so once user log onto the system using the SUA credentials then confidence of user is set to 1.00, which is the maximum value of confidence. On each action, R-RCM calculates the confidence of user based on the classifier score of performed action. If the current action is performed according to genuine user's behaviour then user earns points and confidence increases while if the performed action does not match the genuine user then user loses points and confidence decreases. During the active time, if the confidence of user remains higher than the given final threshold then the user can use the system without any restraint, however if the confidence of a user goes below the given final threshold then the user will be locked out of the system. In this research work, two types of RCM are proposed and system performance is assessed for both types.

- **Experimental Setting I: Simple RCM**

  Continuous authentication is an important aspect of a computer security, however, it might not substitute the static authentication completely and it can be considered as an additional layer of security. Firstly user logs onto computer system using the SUA credential e.g., username and password. After the initial login, confidence related to genuineness of user is set to its maximal value i.e., 1.00. Thereafter, RCM calculates the confidence of user on each action based on the classifier score of performed action and few other parameters. Subsequently, if action is performed in accordance to the behavioural pattern of legitimate user then confidence increases. In contrast, if behavioural pattern of performed action deviates from normal user training patterns then confidence decreases. During the user's active session, if confidence goes lower than final threshold then user will be locked out by the system.

- ***Experimental Setting II: R-RCM with Alert Threshold***

  In this setting, the two thresholds namely alert threshold $T_i = D$ and lockout threshold T_f have been employed to make the system more secure. The system has implemented the concept of alert threshold where if the user's confidence level is going down incessantly and reaches the alert threshold

T_i then the user loses confidence points more than usual in order to lock it out as soon as possible as shown in Fig.3.2.

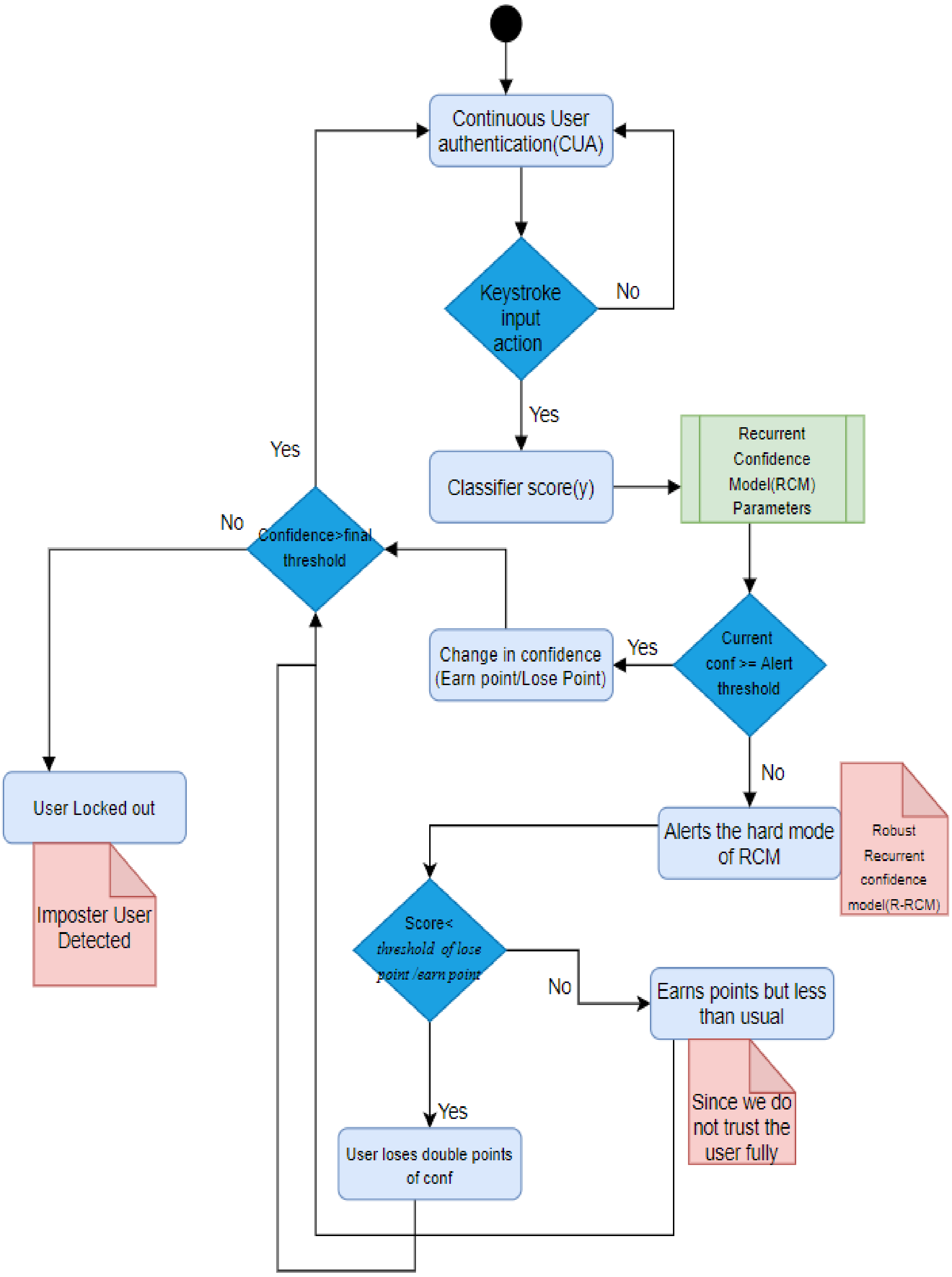

Continuous User authentication(CUA)
Keystroke input action
No
Yes
Classifier score(y)
Recurrent Confidence Model(RCM) Parameters
Yes
No
Confidence>final threshold
Change in confidence (Earn point/Lose Point)
Yes
Current conf >= Alert threshold
No
User Locked out
Imposter User Detected
Alerts the hard mode of RCM
Robust Recurrent confidence model(R-RCM)
Score< threshold of lose point /earn point
No
Earns points but less than usual
Since we do not trust the user fully
Yes
User loses double points of conf

Figure 3.2: Robust Recurrent Confidence Model (R-RCM)

The recurrent confidence is determined by the classification score of the current action performed by the user along with other 5 parameters as shown in algorithm 1. The parameter H denotes the threshold value between lose or earn points precisely. In this aspect, if the classification score of the current action $\hat{y_t}$ is greater than this threshold (H) then $\Delta Conf_i > 0$, i.e., user earns points, and vice versa. Furthermore, the parameter Z is the width of sigmoid for this function, while the parameters M and N are the maximum value of the points earned or lost respectively. Parameter D is alert threshold which checks if user is losing confidence points consistently and reached the alert threshold. If this is the case then system switches to its more hard mode of operation where it checks if current confidence is lower than alert threshold and current action $\hat{y_t} < H$ then it makes the user lose more points on each action hence making it lock out quicker so that it can only make lesser damage on system. However, it is probable that sometimes genuine user behaves in unusual way owing to the background context thereby reaches the alert threshold by losing confidence points. In this case, R-RCM checks on each action if current confidence is less than alert threshold but the $\hat{y_t} > H$, then it means user would earn points on this current action but still model does not trust user completely and grants points less than expected. Since if it would be genuine user than despite of getting less points than usual it would gradually achieve the highest score.

Algorithm 1 Robust-Recurrent Confidence Model

```
Initialization
Static Authentication, Confidence set to 1.00
After 1st action, probability of genuineness of user calculated by set classifiers
```

Phase 1 – Data Input

```
$\hat{y}_t$ —- probability of user genuineness at given time step t
H —- represents the threshold value between lose point and earn point
Z —- the width of the sigmoid for this function
M —- the maximum value for points earned
N —- the maximum value for points lose
D —- Alert borderline threshold $T_i$
T —- Lockout Threshold $T_f$
```

Phase 2 – Change in confidence

```
    begin
if $conf_i \geq D$ then
$\Delta Conf_i = min\left(-N + (\frac{2N}{1+\exp\left(-\frac{\hat{y}_t-H}{Z}\right)}), M\right)$
    $RecurrentConf = min(max(RecurrentConf_{i-1} + \Delta Conf_i), 0), 1.00)$
else if $(conf_i < D)$ and $(\hat{y}_t < H)$ then
$\Delta Conf_i = min\left(-N + (\frac{2N(1-H)}{1+\exp\left(-\frac{\hat{y}_t-H}{Z}\right)}), M\right)$
    $RecurrentConf = min(max(RecurrentConf_{i-1} + \Delta Conf_i), 0), 1.00)$
else if $(conf_i < D)$ and $(\hat{y}_t > H)$ then
$\Delta Conf_i = min\left(N + (\frac{3N}{1+\exp\left(-\frac{\hat{y}_t-H}{Z}\right)}), M\right)$
    $RecurrentConf = min(max(RecurrentConf_{i-1} + \Delta Conf_i), 0), 1.00)$
end if
    End
```

The concept of R-RCM has been elaborated more in Fig.3.3 and Fig.3.4.

In Fig.3.3, when training sample of genuine user has been compared with its own validation sample, it can be noticed that how the recurrent confidence level

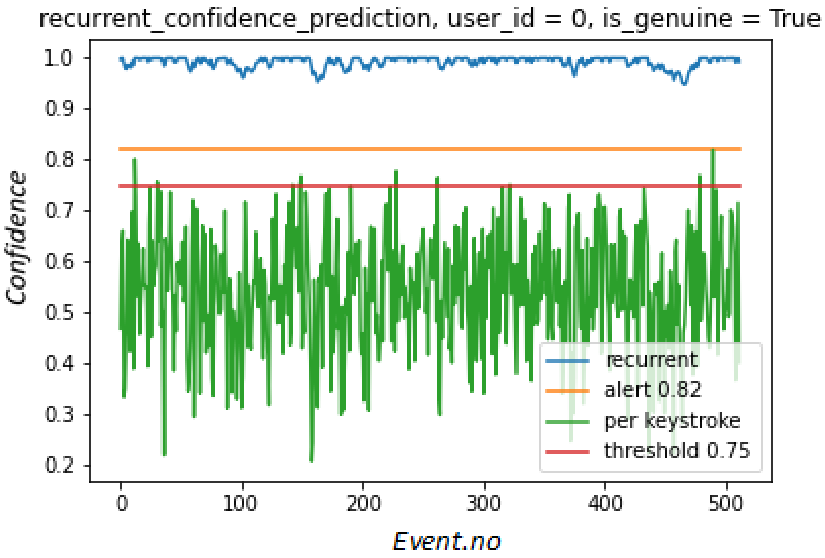


Figure 3.3: Confidence value for genuine user tested with the genuine test data

is varying on each action. Sometimes it goes down due to points lost but again it attains its maximum value and never drops down to the final lockout threshold.

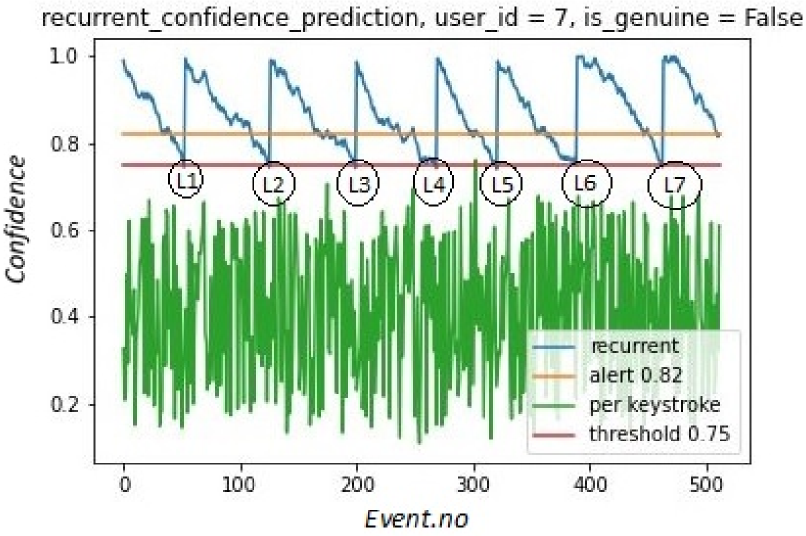


Figure 3.4: Confidence value for genuine user tested with the imposter test data

However, Fig.3.4 shows that when genuine user's training sample is compared against the validation data of an imposter user, then the confidence level drops

7 times below the lockout threshold (L1, L2, L3, L4, L5, L6, L7) within 500 user actions. But it can be discerned that alert threshold is set at 0.82 and as soon as confidence reaches the alert threshold, system locks out the user as quickly as possible due to the hard mode of R-RCM. For simulation purposes, it is assumed that the users are again using the SUA to access the system after every lock out and their maximum confidence of 1.00 is re-gained.

3.2.4 Response Unit

The response unit is considered to be the decision module of proposed system. The output from R-RCM is treated as an input for decision module. In this aspect, if the current confidence of user is higher than the final threshold then user can continue using the system. However, if the current confidence is lower than the final threshold then user is considered to be an imposter user and in this case system takes the action by immediately locking out the user from system.

3.2.5 Evaluation Threat Scenarios

Different Evaluation threat scenarios can be designed in order to assess the system performance. The system has trained binary classifier for each user with genuine and imposter classes in order to distinguish an activity of genuine user against other users. Accordingly, the data of genuine and imposter samples have been considered in equal proportion in order to avoid the classifier biasness. In this regard, three threat scenarios have been designed for evaluation namely internal, external and hybrid which are explained below:

Suppose system has been given a set U of $N = \mathsf{U}$ users and in total each scenario has N cases. For each scenario, firstly system needs to select:

- g – A Genuine user,
- I_1 – Impostors set available for train and test,
- I_2 – Impostors set available for validation.

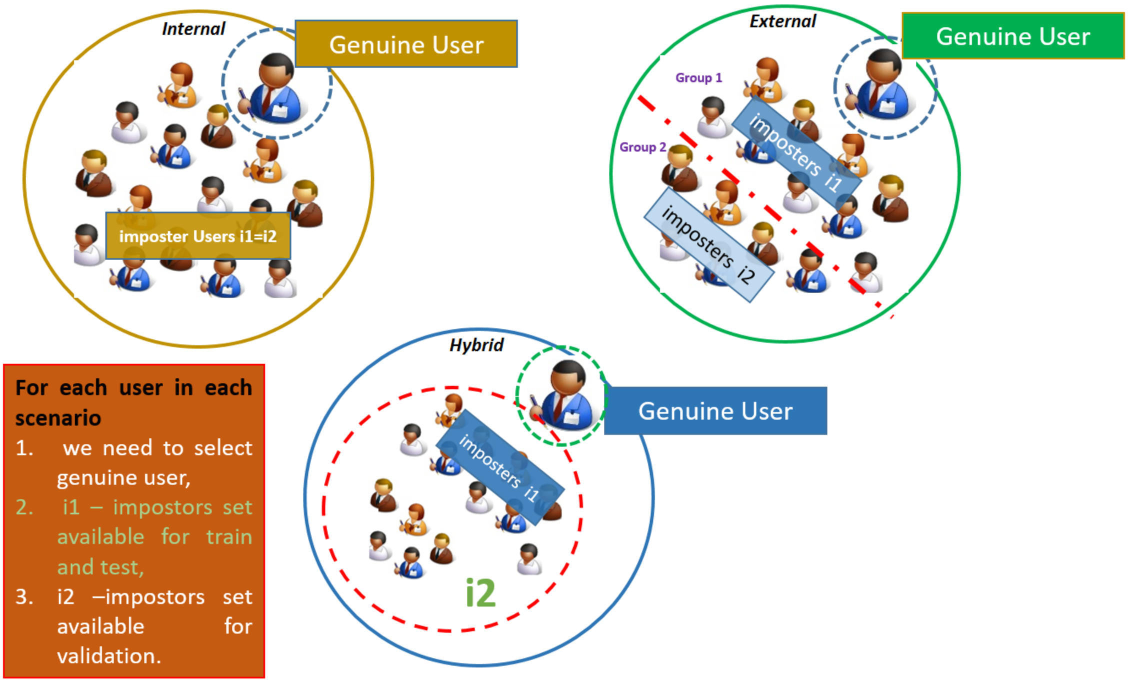
1. *Internal Threat Scenario* $ETS1$: Each of N users is selected as a genuine user g.The rest of the users are assigned to $I_1 = I_2 = U \setminus g$ as shown in Fig.3.5. Accordingly, it is assumed that system has training samples of all the users in the given organisation/dataset.

2. *Hybrid Threat Scenario* $ETS2$: Each of N users is selected as a genuine user g. First M users that do not include g are assigned to I_1. $I_2 = U \setminus g \setminus I_1$ as shown in Fig.3.5. It is assumed that rest of the users are added to organisation after the training process and system does not have any training samples of these newly added users for the first M users. While the validation is done on all the users so $I_2 = U \setminus g$.

3. *External Threat Scenario* $ETS3$: U is split into groups of M users. If $N \mod M \neq 0$, then system pads a set of users in a ring like fashion, such that $U' = \{u_0, u_1, ..., u_N, u_{N+1}, u_{N+M-N \mod M}\}$ and U $\mod M = 0$. For every group, each of M users is picked up consequently as a genuine user g while the rest of users are assigned to I_1. Users not present in the group are assigned to I_2 as shown in Fig.3.5. In such a case validation set of impostor users doesn't include any of users used during the training and testing at all.

3.2.6 Performance Measure

The classification algorithms generally report the performance in terms of False Acceptance Rate (FAR), False Rejection Rate (FRR) Giot et al. (2011) and Equal Error Rate(EER) for biometric systems Bakelman et al. (2012). However, for a true CUA the identity of user should be checked on each single action and performance measure should depend on how many actions imposter or genuine user have performed before system detects it or falsely lock it out respectively. Based on our understanding the number of actions performed by different users within a particular time frame considerably depends on individual's unique behaviour patterns and this factor is distinctive among different users. For instance, a person



For each user in each scenario

1. we need to select genuine user,
2. i1 – impostors set available for train and test,
3. i2 –impostors set available for validation.

Figure 3.5: Three Evaluation Scenarios

with fast typing speed would be able to execute more actions on system resulting in more harm to system resources in comparison to a user with slow typing speed within any given time period.

Therefore, it has been decided to report the performance of proposed CUA system in terms of action domain instead of considering the time complexity of identifying the imposter users. In this aspect, this research uses the performance metrics as describe by researchers in Bours and Mondal (2015*a*) in form of Average Number of Genuine Actions (ANGA) and Average Number of Imposter Actions (ANIA). However, their results were demonstrated in the exact number of actions which makes it difficult to compare the research work results because of different amount of validation data in any experimental work. On the other hand, the method used in this research to calculate the normalized average/mean portion of genuine and imposter actions would make it easy to compare the results regardless of amount of validation data in any future research work done for true continuous user authentication system.

The performance measure of implemented CUA system has been evaluated based on the following performance metrics:

- $ANIA$: Normalized portion of Average Number of Imposter Actions

- $ANGA$: Normalized portion of Average Number of Genuine Actions

- EER: Equal Error Rate

ANGA and ANIA: In general, if imposter user i, when validated against the template of genuine user g, is locked out L times after performing respectively A_1,A_2,......, A_L actions before each lockout. Then, the normalized imposter actions over the total sampling sequence actions A_T are defined as:

$$ANIA = \frac{\sum A_L}{L * A_T}, \tag{3.1}$$

The ANGA are calculated in the same way where genuine user g is validated against the template of genuine user itself and the genuine actions are calculated which it can perform against its own reference template before false lockout.

$$ANGA = \frac{\sum A_L}{L * A_T}, \tag{3.2}$$

Moreover, the total systems' ANIA and ANGA are calculated as follows:

$$SystemANIA = \frac{\sum(MeanANIA * User)}{TotalUsers}, \tag{3.3}$$

$$SystemANGA = \frac{\sum(MeanANGA * User)}{TotalUsers}, \tag{3.4}$$

More specifically, if the ANGA and ANIA are demonstrated with exact number of genuine and imposter actions respectively instead of giving the normalized portion of actions then it can be calculated as follows:

$$Exactgeniune/imposteractions = Normalizedactions * Totalvalidationdata \tag{3.5}$$

Equal Error Rate (EER): EER has also been used in this research work. EER is a metric which assesses the data classification performance for any model. Subsequently, EER is considered to be a point where False rejection rate (FRR) and False acceptance rate (FAR) overlap each other. In this aspect, FRR is the probability that system wrongly classifies the genuine user as an imposter user and denies access to system resources. Whereas, FAR is the possibility that system incorrectly classifies the imposter user as a genuine user and gives access to system resources. The EER is calculated by using the Eq 3.6 given below:

$$EER = \frac{FRR + FAR}{2}, \tag{3.6}$$

3.2.6.1 User Categories

The ideal CUA system should not lock out genuine user and detect the imposter user quickly hence ANGA should be high while ANIA should be as low as possible respectively. However, sometimes situation can vary from ideal conditions hence four different categories are formulated for all system users based on two rules given below:

Let's assume, there are M users hence there are total M cases. Two properties have been allotted for each of M cases as given below:

- The 1st property indicates if ANGA = 100% or not, for given genuine user g when tested against its own reference training sample.
- The 2nd property indicates if ANIA > 40% or not, for all the imposters i who have been tested against this genuine user g training sample.

Based on these two rules explained above, the following four categories have been formulated:

1. **Very Good**, ANGA = 100% and ANIA $\leq$ 40%
2. **Good**, ANGA < 100% and ANIA $\leq$ 40%
3. **Bad**, ANGA = 100% and ANIA > 40%
4. **Ugly**, ANGA < 100% and ANIA > 40%

The detailed framework containing all the components of proposed CUA system is shown in Fig 3.6.

3.3 Summary

This chapter highlighted the concept and baseline of our proposed continuous user authentication (CUA) system. Subsequently, all the components of proposed CUA have been delineated in detail including the dataset analysis, feature extraction, classification, and performance measure techniques. Moreover, this chapter also presents the explanation of our proposed Robust Recurrent Confidence Model (R-RCM) which tends to authenticate the user on each and every action. Additionally, different threat scenarios are explained and designed comprehensively which can be employed to assess the system performance. In this aspect, the next chapter applies the proposed CUA system framework, described in this chapter, on keystroke dynamics and evaluates the system performance with baseline classification techniques.

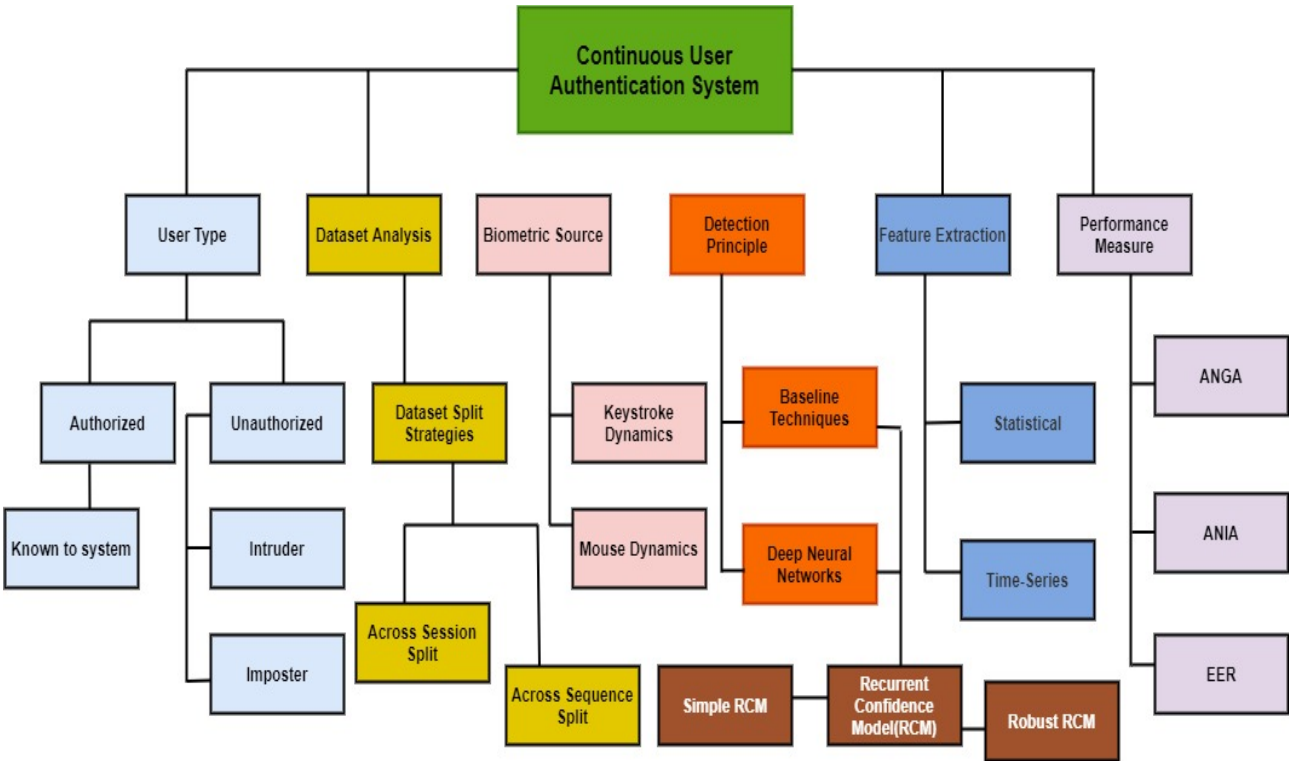


Figure 3.6: Components of Proposed CUA System

Chapter 4

Continuous User Authentication using Keystroke Dynamics with Baseline/Traditional Machine Learning Techniques

In this chapter, a true continuous user authentication system featuring keystroke dynamics behavioural biometric modality has been proposed and implemented using baseline techniques. A novel method of authenticating the user on each action has been presented which decides the legitimacy of current user based on the confidence in the genuineness of each action. The 2-phase methodology, consisting of ensemble learning and robust recurrent confidence model (R-RCM), has been designed. Proposed methodology classifies each action based on the probability score of ensemble classifier which is afterwards used, along with hyperparameters of R-RCM, to compute the current confidence in genuineness of user. System decides if user can continue using the system or not based on new confidence value and final threshold.

4.1 Introduction

Keystroke dynamics recognition (KDR) can be referred as a behavioural biometrics which comprises of evaluating the computer user's distinct typing patterns followed by recognition of person's identity based on these patterns. Moreover, most of behavioural biometrics i.e., the keystroke dynamics do not require users to present biometric identification while preforming important routine tasks and also tends to authenticate the user on each single key press action.

In terms of implementation, there are numerous advantages for the usage of keystroke dynamics (KD) as a recognition method Ali et al. (2017) since these are practical and inexpensive where no additional hardware component is required in order to capture the KD biometrics as oppose to other biometrics which require special hardware like fingerprints, iris and facial biometrics.
Analysing the user behaviour for continuous authentication is a challenging task owing to the insufficient information and large intra-class disparities of data recorded by the computer input devices. The KD based authentication system works on the basis of keystroke timing information which is captured by keyboard with the assistance of specifically designed software Vyazigin et al. (2019) and different discrete features are extracted from those captured keystroke timestamps.

4.2 System Methodology

This section presents the architecture and implementation of proposed CUA system using keystroke dynamics and it also combines the static user authentication (SUA) as an initial login mechanism.

4.2.1 Dataset Analysis

Keyboard usage is typically undertaken in a sequential manner key-press by key-press. More formally, a keystroke sequence is a consecutive ordering of a set of events (E) that occur within a specified interval of time. Each event $e \in E$ has the following properties:

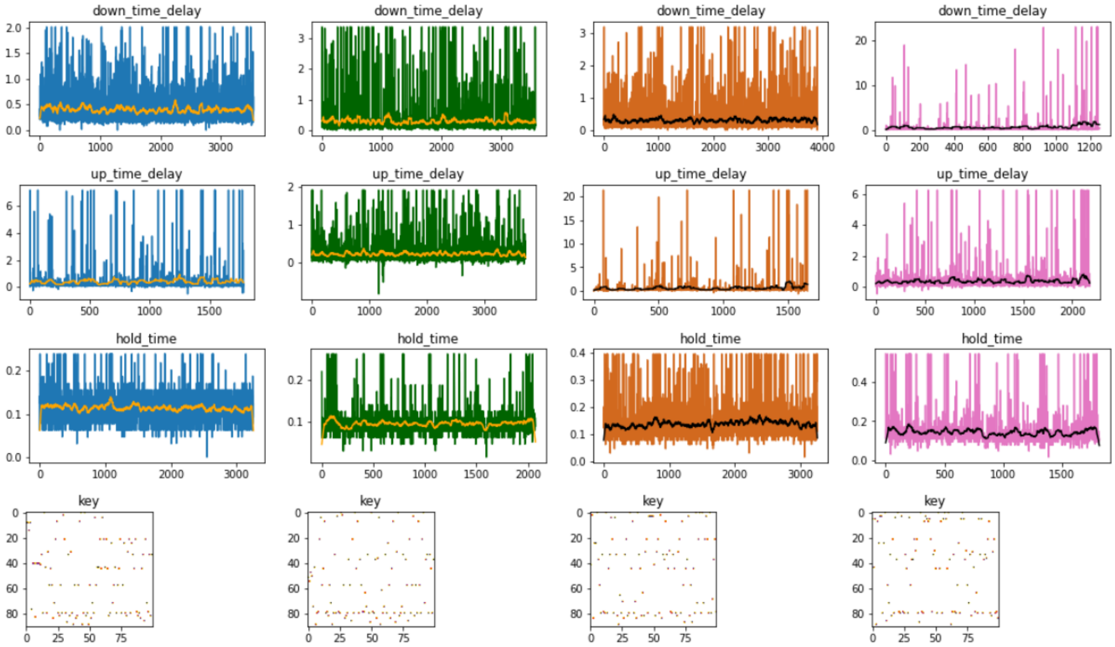
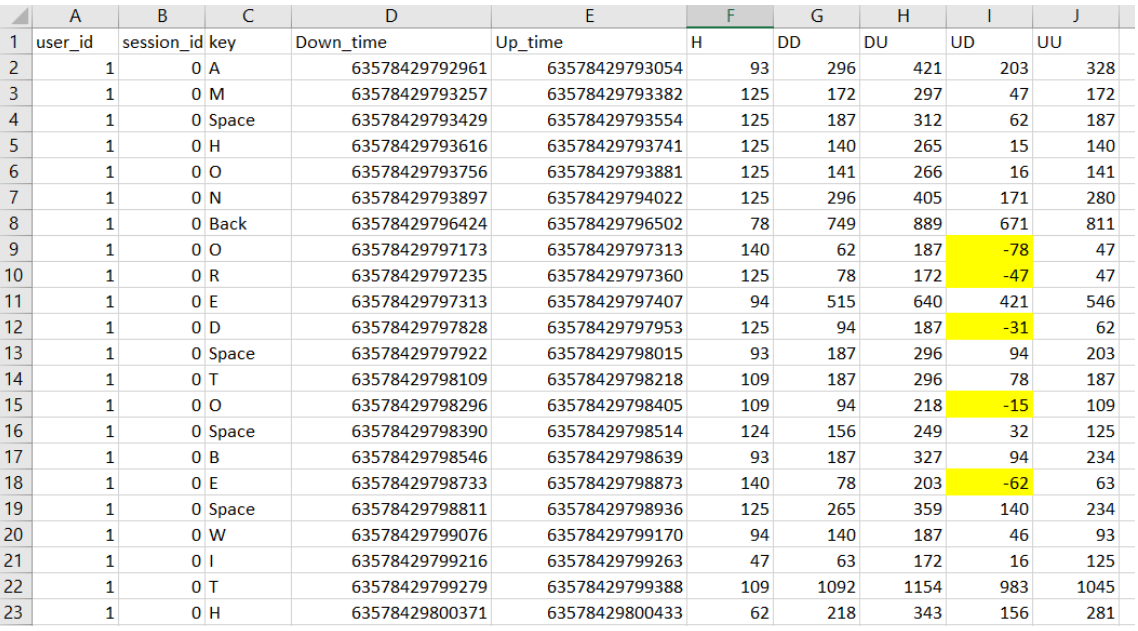


Figure 4.1: Keystroke Distinct Features for 4 Different Users

- $UserId(e)$ – id of the user that has performed an action
- $SessionId(e)$ – id of actions sequence that event belongs to
- $DownTime(e)$ – a key absolute down time (milliseconds) during the action
- $UpTime(e)$ – a key absolute up time (milliseconds) during the action
- $KeyCode(e)$ – a key code that the user has pressed

Fig.4.1 shows the down-time, up-time, key monograph (also known as hold time) and pressed keys features for four different users. It can be noticed that keystroke features provide substantial distinctive patterns for each user. The distinctive features can be generated for each action or sequence and feed to training classifiers to build the reference templates for each user which can be used for authentication of user upon validation.

Fig 4.2 shows an example of the keystroke dataset structure. Each row contains User ID, Session ID, Key-code, Down Time when key was pressed, Up Time when key was released, duration of a pressed key time to release time represented



	A	B	C	D	E	F	G	H	I	J
1	user_id	session_id	key	Down_time	Up_time	H	DD	DU	UD	UU
2	1	0	A	63578429792961	63578429793054	93	296	421	203	328
3	1	0	M	63578429793257	63578429793382	125	172	297	47	172
4	1	0	Space	63578429793429	63578429793554	125	187	312	62	187
5	1	0	H	63578429793616	63578429793741	125	140	265	15	140
6	1	0	O	63578429793756	63578429793881	125	141	266	16	141
7	1	0	N	63578429793897	63578429794022	125	296	405	171	280
8	1	0	Back	63578429796424	63578429796502	78	749	889	671	811
9	1	0	O	63578429797173	63578429797313	140	62	187	-78	47
10	1	0	R	63578429797235	63578429797360	125	78	172	-47	47
11	1	0	E	63578429797313	63578429797407	94	515	640	421	546
12	1	0	D	63578429797828	63578429797953	125	94	187	-31	62
13	1	0	Space	63578429797922	63578429798015	93	187	296	94	203
14	1	0	T	63578429798109	63578429798218	109	187	296	78	187
15	1	0	O	63578429798296	63578429798405	109	94	218	-15	109
16	1	0	Space	63578429798390	63578429798514	124	156	249	32	125
17	1	0	B	63578429798546	63578429798639	93	187	327	94	234
18	1	0	E	63578429798733	63578429798873	140	78	203	-62	63
19	1	0	Space	63578429798811	63578429798936	125	265	359	140	234
20	1	0	W	63578429799076	63578429799170	94	140	187	46	93
21	1	0	I	63578429799216	63578429799263	47	63	172	16	125
22	1	0	T	63578429799279	63578429799388	109	1092	1154	983	1045
23	1	0	H	63578429800371	63578429800433	62	218	343	156	281

Figure 4.2: Structure Example of Keystroke Dataset

as H or hold time for single key, as well as 4 digraph latencies for the 1st key and 2nd key represented as DD, DU,UD and UU. Moreover, it can be observed that UD latencies can be negative value (highlighted as yellow in Fig 4.2). This means user was still holding the 1st key after pressing the 2nd key that's why the timing value of UD can be computed as negative value.

For the analysis of CUA system, the data of a user is split into 3 non-overlapping parts. The *training part T* is used to train the classifier to build a model. The *testing part X* is used for testing the parametric adjustments and *validation part V* is used for final evaluation of unseen data. The validation data of a user is used action by action and each action will determine a change in confidence of user being genuine or imposter.

The proposed architecture is tested with two types of dataset split strategies i.e., Across Session Split and Across Sequence Split, as described in Chapter 3 section 3.2.1.1.

4.2.2 Feature Engineering

Let's say, there is an event of $M + U$ keystrokes where U is the context length and$M$ is the length of keystroke event. Keystroke events are sampled with $U = 1$ and $M = 1$, an event of 1 keyboard action with monographs and digraph features. In the experiments, the following features have been considered to generate from the raw keystroke events.

- *Key Monograph Action* : It represents the key hold time of any key which is calculated by subtracting the key up time from key down time.
- *Key Digraph Action* : Where the features are
    1. $Down - Up\ Time(DU)$: Total time duration of first key press to second key release.
    2. $Down - Down\ Time(DD)$: The time between first key press and second key press.
    3. $Up{-}Down\ Time(UD)$: The time between first key release and second key press.
    4. $Up - Up\ Time(UU)$: The time between first key release and second key release of a particular key digraph.

Table 4.1 shows how these features are used in the reference feature template of user. The system stores the mean and standard deviation(σ) of duration of each monograph and digraph occurring during enrolment or training phase. Also, the key-codes along with their hit counts are stored as well where hit count refers to the number of times each key or key digraph is pressed during enrolment or training phase.

The graphical representation of the keystroke dynamics feature extraction process is shown in Fig.4.3. In this study, time difference considered between two key actions ought to be below 2000ms, since higher timing difference than 2000ms does not represent the normal typing pattern. Moreover, it has been considered necessary to include key monographs in the analysis of true CUA since ignoring the monographs can give room to imposter users to type the full

Features	Components of Reference Template
Monograph	Key-code Hit count $Duration_{Mean,\sigma}$
Digraph	Key-code(1), Key-code(2) Hit count $DD_{Mean,\sigma}$, $DU_{Mean,\sigma}$, $UD_{Mean,\sigma}$, $UU_{Mean,\sigma}$

Table 4.1: Structure of Reference Feature Template

sequence of keystrokes by pausing for 2000ms after each keypress hence leaving no feature for system to authenticate the user successfully.

4.2.3 System Architecture

Let's say, there are total N users. System needs to identify each user per action based on given sequence of keyboard actions. More formally, there is:

$S = \{(x, y)\} \subset \mathbb{R}^{A \times T} \times \{1, \ldots, N\}^T$,

where x_t – keyboard action properties at a time t, $y_t \in \{1, \ldots, N\}$ – user who has taken the action, T – total amount of actions to classify, A – action vector dimension. The implemented system predicts a user identity y_t per time step t, which in the simplest case equals to an indicator whether it is a genuine user action or not.

Subsequently, this research work implements a 2-Phase system methodology for continually authenticating the user with keystroke biometric modality, as shown in Fig.7.3, and discussed below:

4.2.3.1 1st PHASE , BASELINE CLASSIFIERS

The proposed system uses three performance threat evaluation scenarios namely *ETS1, ETS2 and ETS3* described in *chapter 3, section 3.2.5.* In each scenario, score of the classifiers, for per action, decides whether it is genuine or belongs to an impostor user. In this regard, ensemble learning approach consisting of three clas-

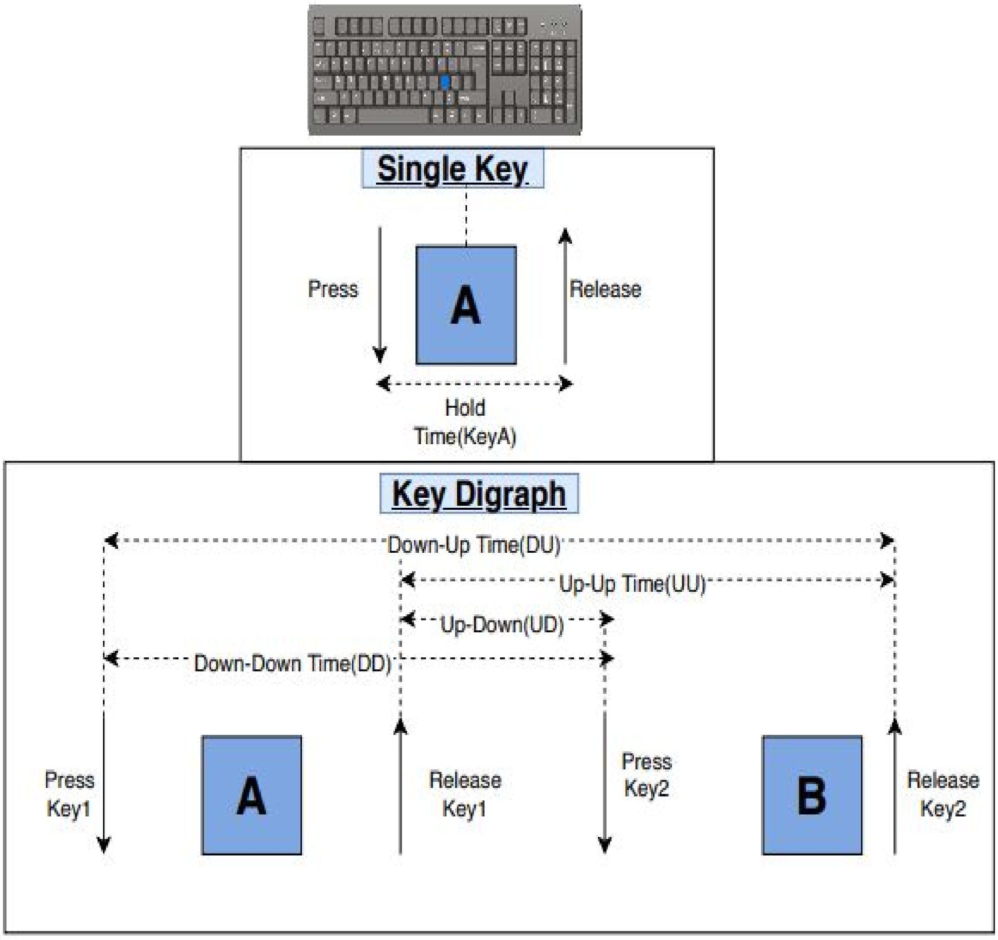


Figure 4.3: Keystroke Dynamics Features Representation

sifiers including *Support Vector Machine(SVM)*, *Artificial Neural Network(ANN)* and *Gradient Boosting Decision Trees(XGBoost)* has been used where an output score is produced according to ensemble classifier rule based on input scores of all three classifiers as shown in Fig. 4.5

The proposed system employs two types of ensemble rules including:

- Dynamic Classifier Selection (DCS)
- Weighted Classifier Fusion (WCF)

DCS Mendialdua et al. (2015) reflects the tendency to extract a single best classifier at train-test split for each action which is the most likely to produce

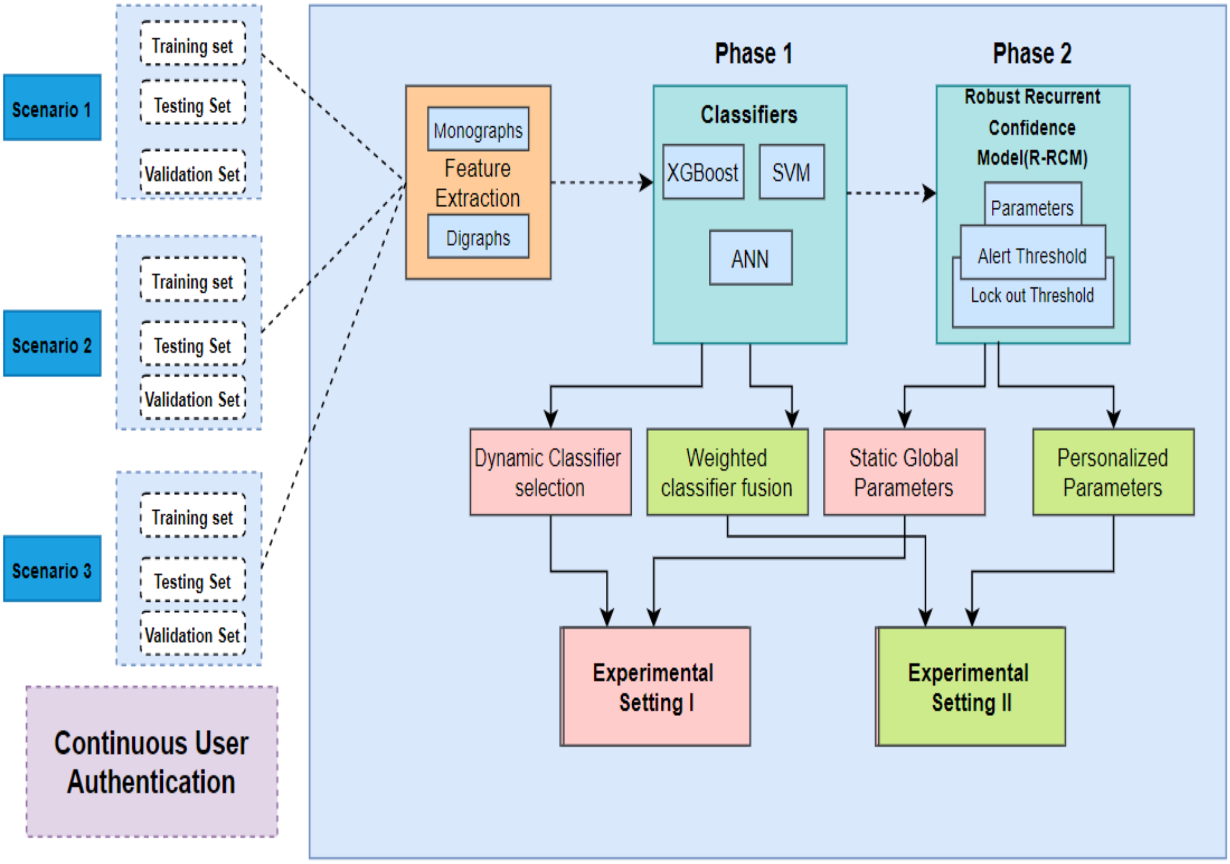


Figure 4.4: The System Architecture

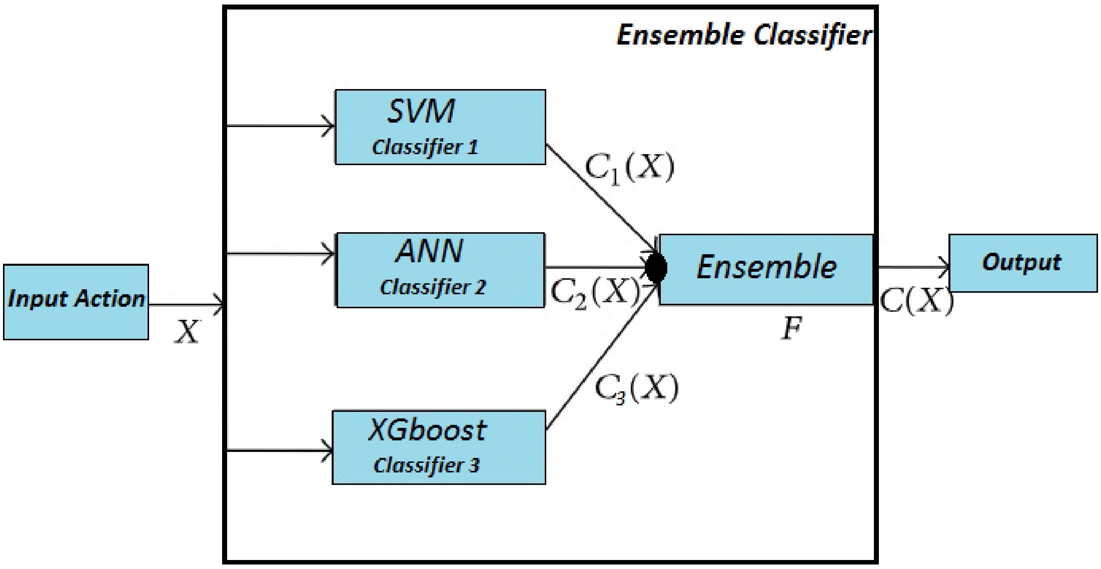


Figure 4.5: Ensemble Classifier Approach

the correct classification label for an input sample at validation split. However, the WCF Mi et al. (2016) relates to approach where all the classifier scores goes to the weighted fusion module and an output score is a weighted sum of input scores of all the three classifiers as shown in Eq: 7.1.

$$\hat{y}_t(c_t|W) = \frac{\sum_{i=0}^{K-1} W_i c_{ti}}{\sum_{i=0}^{K-1} W_i} \tag{4.1}$$

where c_{ti} – input scores, K – amount of classifiers, W_i – input score weights and the value of these weights have been optimised with genetic algorithm Weile and Michielssen (1997) and $\hat{y}_t(c_t|W)$ – fused score which will be used as a raw confidence score in the second phase for each action.

The architectures and implementation details of three classifiers used in ensemble method are listed below:

- ***Support Vector Machine(SVM)***

Support vector machine (SVM) is a supervised learning model Hsu et al.

(2003) which is usually used for classification and regression problems. Labelled sets of data are provided, during training of model, for each class and on the basis of which it categorises the new unseen data. Moreover, it works on the principle of creating hyperplane on high dimensional space in order to classify the data points. Hyperplanes are considered to be optimal decision boundaries which assist to classify data correctly into classes on each side of hyperplane as shown in Fig.4.6. Moreover, hyperplane shape depends on kernel which can be linear or non-linear in order to fit data correctly into distinct classes. In this framework, Scikit-learn library in python has been used to implement SVM classifier with radial basis RBF kernel.

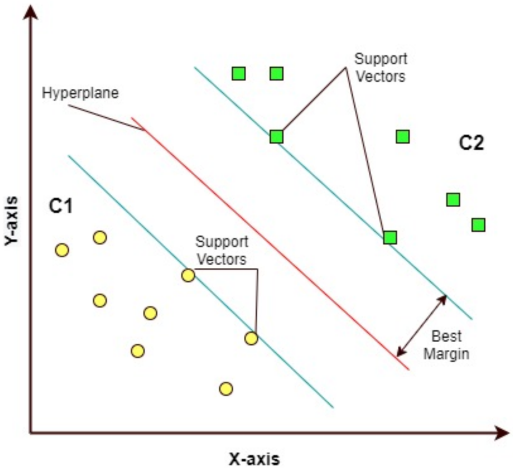


Figure 4.6: Support Vector Machine

- ***Artificial Neural Network (ANN)***

Artificial Neural Network (ANN) is an arrangement of multiple Artificial neurons which can be used for classification and regression problems Yao (1999). In this framework, the neurons consist of a linear activation function with a 2-layer Feed-Forward neural network. Moreover, KERAS is used for the ANN classifier and Adam optimizer is implemented which is efficient to optimize the cost function and also it reduced the ANN training time.

Different regularization parameters are tested to maintain the training time of classifier.

- ***XGBoost***

  XGBoost stands for “Extreme Gradient Boosting”, and it is more efficient implementation that sklearn Random Forest, also XGBoost uses gradient boosted decision trees which has more speed than traditional decision tree algorithm Chen and Guestrin (2016). Generally, gradient boosting is a technique which makes a traditional algorithm closer to neural networks as it permits substantially more control over capacity. XGBoost is implemented since it is fast when compared to other implementations of gradient boosting.

4.2.3.2 2$^{\text{nd}}$ PHASE, Recurrent Confidence Function

In this research, a novel robust recurrent confidence Model (R-RCM), described in *Chapter 3, Section 3.2.3.2*, has been proposed and implemented. The model computes the variation in confidence for each action by employing some parameters and returns the system confidence to indicate the genuineness of the current user. The parameters can be:

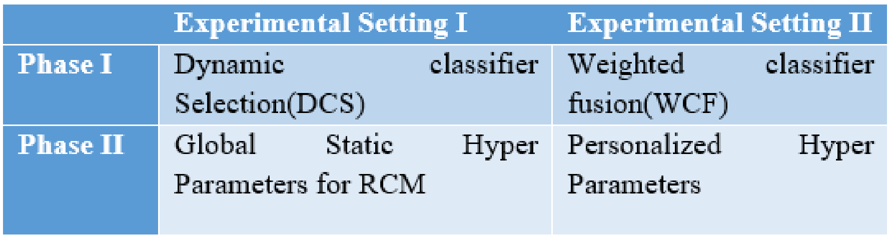
- *Global Static Parameters*
- *User Specific Parameters*

In order to analyse the performance, system has been tested using both global static parameters as well as personalising the parameter of RCM. These parameters are optimised by employing the genetic algorithm Weile and Michielssen (1997) to find the optimal value for each user based on their train-test split samples.

The following discrete values are used for new samples introduction into an epoch, or samples mutation. Logarithmic scale for Z, M, and N values has been applied to achieve better convergence. W_0, W_1 and W_2 of Eq.7.1 are being normalized afterwards to have a weighted average. Since just sampling values within those boundaries does not imply that W_0, W_1 and $W_2 = 1$. Cross validation procedure is implemented, so that all of the samples are evaluated at train split. And only a portion of the best is being preserved from the previous population. The other samples are obtained via either crossover or mutation procedures. A best sample per epoch is being preserved with its train and test performance. Afterwards, a sample with the best test performance is picked up. A final result is obtained for validation split.

- $H = 0 + k * 100/99, k = \overline{0, 99}$
- $Z = 2.0^{-7+k*14/13}, k = \overline{0, 13}$
- $M = 2.0^{-7+k*14/13}, k = \overline{0, 13}$
- $N = 2.0^{-7+k*14/13}, k = \overline{0, 13}$
- $W_0 = 0 + k * 100/99, k = \overline{0, 99}$
- $W_1 = 0 + k * 100/99, k = \overline{0, 99}$
- $W_2 = 0 + k * 100/99, k = \overline{0, 99}$

The proposed system methodology has been validated in this work by formulating two experimental settings as shown in Fig.7.3. These settings combine the divergent approaches for the output of ensemble classifiers and parameters of R-RCM in order to test the system from different perspectives as shown in Fig.4.7.



	Experimental Setting I	Experimental Setting II
Phase I	Dynamic classifier Selection(DCS)	Weighted classifier fusion(WCF)
Phase II	Global Static Hyper Parameters for RCM	Personalized Hyper Parameters

Figure 4.7: Two Divergent Experimental settings for Proposed 2-Phase Methodology

4.3 Results and Discussion

The programming language used throughout for this architecture is Python 3.4. Keras interface with tensorflow is employed to execute the neural network computations precisely. Scikit-learn is used to train the SVM. Moreover, XGBoost is an enhanced distributed gradient boosting library which is employed to train machine learning algorithms for Gradient Boosting framework. The results attained from the experiments are discussed in this section.

Here, some excerpts of experimental results are presented based on 512 action sequence where the user has been authenticated on each action. However, in practice validation has been done on whole of validation split data (20%) or session [2] depending on the dataset split strategy. Aggregated results are provided in tabular form below but here for sake of understanding only some samples of results are shown in order to visualise the user categories.

- *Good*: Fig.4.8 shows an excerpt of a genuine user sample where the validation set of user was used against its own reference set on the right side of figure while the left part shows the validation of an imposter sample against the same genuine user sample. It can be noticed that genuine user has been locked out for the given sequence sample, so ANGA can be calculated using Eq. 3.2

  $ANGA = \frac{320}{1*512} = 0.625$ or 65% so, $ANGA < 100\%$

  Similarly, ANIA can be calculated using Eq. 3.1

  $ANIA = \frac{480}{8*512} = 0.117$ or 12% so, $ANIA < 40\%$

  In this example, genuine user has been locked out at least once but the given imposter validated against this genuine user's reference sample has been detected before performing 40% of actions hence this genuine user falls in good category. More precisely, the ANIA & ANGA are taken in terms of normalized number of actions as a portion of actions in relation to a total sequence length for this example i.e., 512 then it can be inferred that this imposter had performed 60 actions on average before detection for the given genuine user.

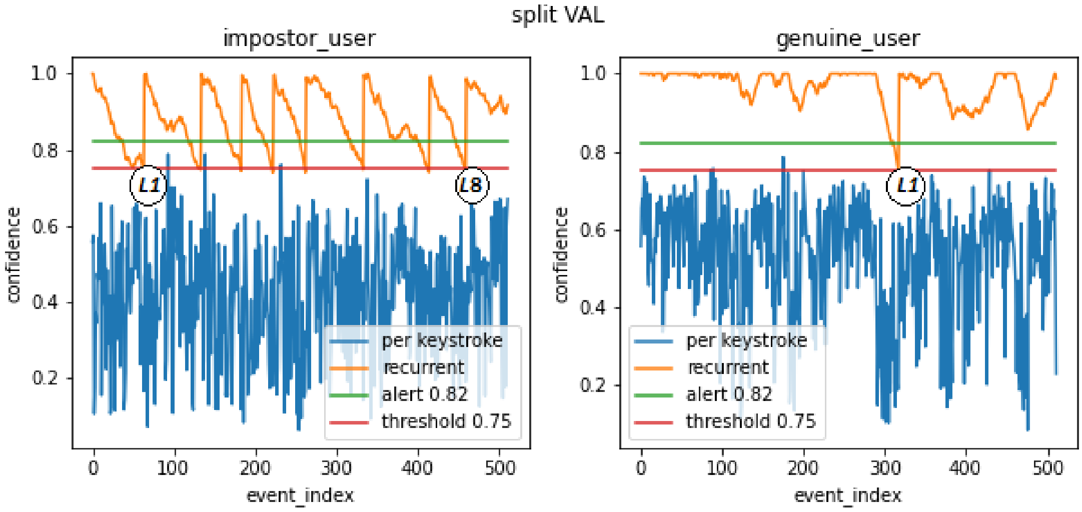


Figure 4.8: Genuine user validated with its own reference set(right) and with imposter set(left)

- *Very Good*: Fig.4.9 shows another excerpt of validation sample which specifies that genuine user has never been locked out for the given sequence sample making the ANGA=100% while the imposter user has been locked out 24 times (L1-L24) in the given sequence sample hence the ANIA of this example, according to Eq. 3.1, is 0.04 or 4.0% , so it can be concluded that ANIA < 40%.

  More specifically, if the ANIA & ANGA are taken in terms of normalized number of actions as a portion of actions in relation to a total sequence length 512 then it can be assumed that this imposter had performed 21 actions on average before detection for the given genuine user and it falls in very good category.

- *Bad*: Similarly, Fig.4.10 shows another excerpt of validation which indicates that genuine user has never been locked out for the given sequence sample

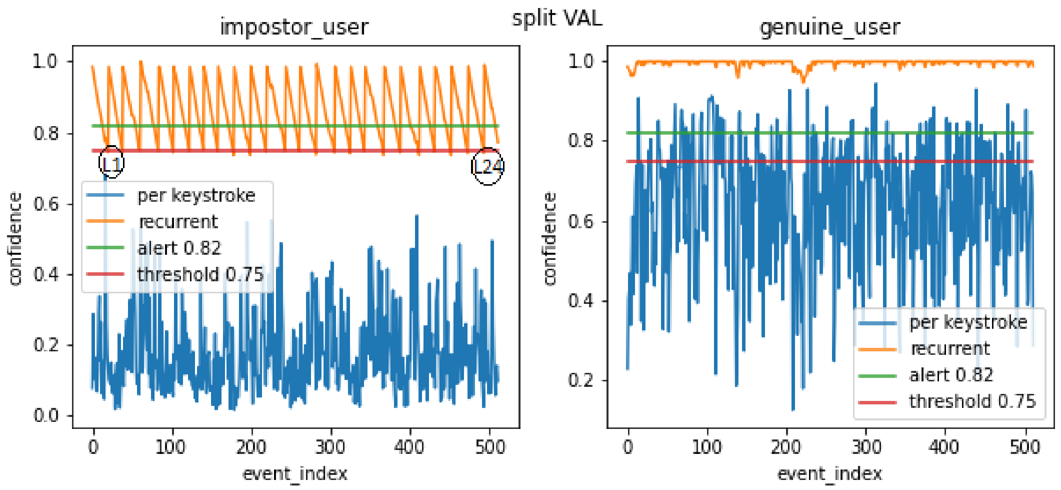


Figure 4.9: Genuine user validated with its own reference set(right) and with imposter set(left)

making the ANGA=100% while the imposter user has been locked out 2 times only(L1-L2) in the given sequence sample hence the ANIA of this example, according to Eq. 3.1, is 0.5 or 50% , so it can be said that ANIA > 40%. More precisely, if the ANIA & ANGA are taken in terms of normalized number of actions then it can be assumed that this imposter had performed 256 actions on average before detection for the given genuine user and it falls in bad category.

- *Ugly*: Fig.4.11 shows the genuine user has been locked out so ANGA < 100% while the imposter user has not been detected before performing 50% of actions, according to Eq. 3.1, on average hence ANIA > 40%.

Now, the aggregated results for all the 75 users are reported in tabular form for the following four settings:

1. Across Session split–Experimental Setting I: Dynamic Classifier Selection with global Static RCM
2. Across Session split–Experimental Setting II: Weighted Classifier fusion with Personalized RCM

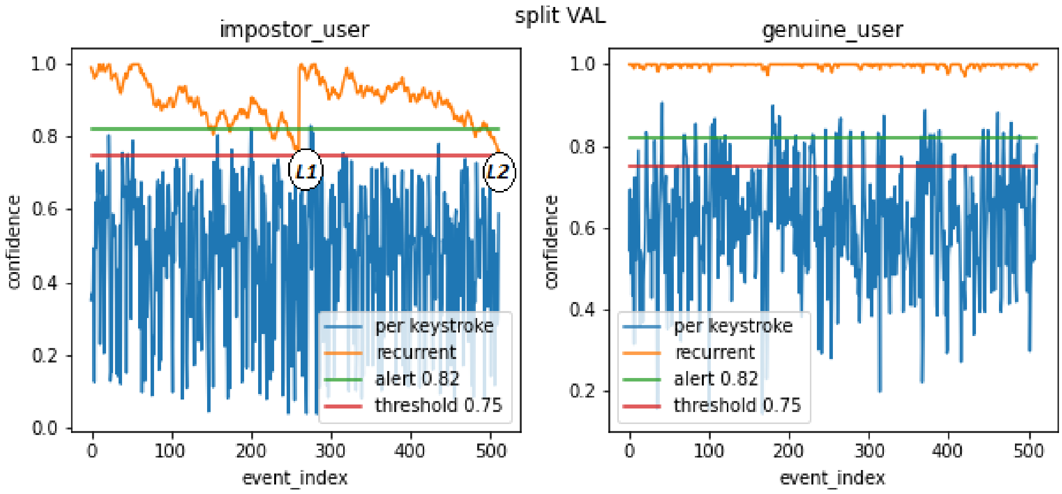


Figure 4.10: Genuine user validated with its own reference set(right) and with imposter set(left)

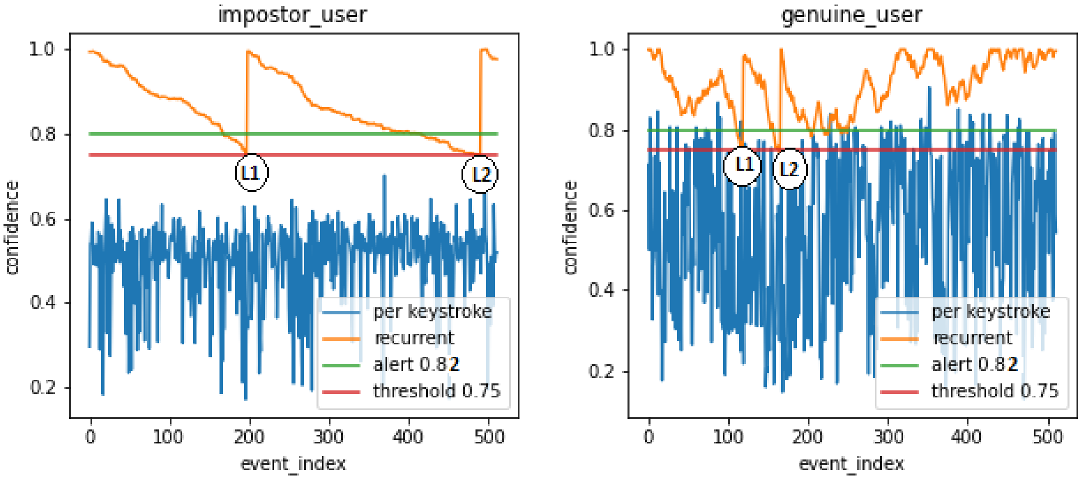


Figure 4.11: Genuine user validated with its own reference set(right) and with imposter set(left)

3. Across Sequence split–Experimental Setting I: Dynamic Classifier Selection with global Static RCM
4. Across Session split–Experimental Setting II: Weighted Classifier fusion with Personalized RCM

4.3.1 Across Session Split

In this setting, the dataset is divided across sessions as discussed in Chapter 3 section 3.2.1.1. Session 0 and 1 are used for training and testing purpose while the session 2 is used for generalisation of our methodology on unseen data. For our session split, the system is tested with both settings given below:

4.3.1.1 Experimental Setting I: Dynamic Classifier Selection with global Static RCM

The aggregated result for all the users are reported in tabular form along with the ANGA & ANIA. It can be observed from the Table 4.2 that:

In scenario 1, 33 participants qualify for the 'very-good' category, where the mean of ANGA is 1.00 actions which represents that no genuine participant has been locked out leaving the ANGA 100%, whereas the mean of ANIA is 0.24 which indicates that all the imposters for these 33 genuine users has been detected before performing 0.24% of actions. Subsequently, the 22 users fall in good category with ANGA 0.55 and ANIA being 0.32 , 8 users fall in bad category with ANGA 1.00 (never locked out) and ANIA 0.48 and 12 users fall in bad category with ANGA 0.61 and ANIA 0.59.

In scenario 2, there are 31 users in very good category with ANIA 0.20, the 11 users fall in good category with ANGA 0.63 and ANIA being 0.18 , 15 users fall in bad category with ANIA 0.48 and 18 users fall in bad category with ANGA 0.67 and ANIA 0.65.

In scenario 3, there are 45 users in very good category with ANIA 0.24, the 22 users fall in good category with ANGA 0.22 and ANIA being 0.20 , 4 users fall in bad category with ANIA 0.49 and 4 users fall in bad category with ANGA 0.73 and ANIA 0.71.

Overall, if the system is evaluated based on the number of actions performed by imposter and genuine user before detection and false lockout respectively, then it can be assumed that scenario 3 has performed well with the most lowest ANIA and scenario 2 has performed well with highest ANGA for keeping the most of genuine user logged in for the whole of testing sessions and not locked out falsely.

Category	Scenario	Users	Normalized ANGA	Normalized ANIA
Very Good	1	33	1.00	0.24
Good	1	22	0.55	0.32
Bad	1	8	1.00	0.48
Ugly	1	12	0.61	0.59
System Total	1	**75**	**0.80**	**0.34**
Very Good	2	31	1.00	0.20
Good	2	11	0.63	0.18
Bad	2	15	1.00	0.48
Ugly	2	18	0.67	0.65
System Total	2	**75**	**0.86**	**0.36**
Very Good	3	45	1.00	0.24
Good	3	22	0.22	0.20
Bad	3	4	1.00	0.49
Ugly	3	4	0.73	0.71
System Total	3	**75**	**0.75**	**0.26**

Table 4.2: Across Session Split–Experimental Setting I

4.3.1.2 Experimental Setting II: Weighted Classifier fusion with Personalized RCM

It can be observed from the Table 4.3 that: In scenario 1, 11 participants qualify for the ‘very-good’ category, the mean of ANIA is 0.20 which indicates that all the imposters for these 11 genuine users have been detected before performing 0.20% of actions. Subsequently, the 64 users fall in good category with ANGA 0.71 and ANIA being 0.35. In scenario 2, there are 26 users in very good category with ANIA 0.23, the 45 users fall in good category with ANGA 0.81 and ANIA being 0.31 , 4 users fall in ugly category with ANIA 0.41. In scenario 3, there are 15 users in very good category with ANIA 0.15, 52 users fall in good category with ANGA 0.71 and ANIA being 0.24 , 8 users fall in ugly category with ANIA 0.42.

Overall, if the system is evaluated based on the number of actions performed by imposter and genuine user before detection and false lockout respectively, then it can be assumed that scenario 3 has performed well again with the most lowest ANIA and scenario 2 worked well for keeping the most of genuine user logged in for the whole of testing sessions and not locked out falsely.

Category	Scenario	Users	Normalized ANGA	Normalized ANIA
Very Good	1	11	1.00	0.20
Good	1	64	0.71	0.35
Bad	1	0		
Ugly	1	0		
System Total	1	**75**	**0.75**	**0.31**
Very Good	2	26	1.00	0.23
Good	2	45	0.81	0.31
Bad	2	0		
Ugly	2	4	0.94	0.41
System Total	2	**75**	**0.88**	**0.29**
Very Good	3	15	1.00	0.15
Good	3	52	0.71	0.24
Bad	3	0		
Ugly	3	8	0.69	0.42
System Total	3	**75**	**0.77**	**0.24**

Table 4.3: Across Session Split–Experimental Setting II

4.3.1.3 Analysis for setting I and setting II (Across Session split)

The aggregated results of DCF with static RCM parameters(setting I) are reported in Table 4.2 and weighted fusion with personalised parameters optimised with genetic algorithm(setting II) in Table 4.3.

It can be noticed that for static global RCM, there are users in all three scenarios who are falling in bad as well as ugly categories which means there are users for which the imposters could not be caught up even after performing more than 40% of actions. In the worst case(scenario 3), there are users for which imposters could not be detected even after performing 71% of actions on average.

On the other hand, if setting II with personalised parameters is observed then most of users are falling in either very good or good category but still there are some portion of users falling in ugly category. In the worst case (scenario 3), there are users for which imposters could not be detected even after performing 42% of actions on average. However, it is still lower than worst case of setting I. Secondly, setting II has lowest system ANIA and highest ANGA as compared to setting I in almost all scenarios. Hence, it can be assumed that setting II has performed well in detecting the imposter users since it includes the personal parameters of each user optimised by genetic algorithm.

4.3.2 Across Sequence Split

In this setting, dataset is divided across sequences as discussed in Chapter 3 section 3.2.1.1.

4.3.2.1 Experimental Setting I: Dynamic Classifier Selection with global Static RCM

It can be observed from the Table 4.4 that, *For scenario 1,* 71 participants qualify for the very-good category where the mean of ANGA is 1.00 actions which represents that none of genuine participant has been locked out leaving the ANGA 100%, whereas the mean of ANIA is 0.22 which indicates that all the imposters for these 71 genuine users has been detected before performing 0.22 or 22% of actions. Subsequently, 4 users fall in bad group where ANGA is again 100% showing the genuine user itself is not locked out when exposed to its own validation data and mean ANIA is 0.41 which indicates that all the imposters had been locked out only after performing 41% of actions for given validation data.
In scenario 2, there are 49 users in very good category with ANIA 0.27 (27% actions) which is quite high while 11 users fall in good group where mean of ANGA is 0.97 (97% actions) and ANIA is 0.26 (26% actions). And, 15 users fall in bad category with ANIA 0.42 (42% actions).
In scenario 3, it can be noticed that 49 users in very good category with mean ANIA 0.24 (24% actions) while 15 users fall in good group where mean of ANGA is 0.96(96% actions) and ANIA is 0.31(31% actions). 11 users fall in bad category

with ANIA 0.44(44% actions).

Overall, the system performance has been evaluated based on the number of actions performed by imposter before detection and average number of actions performed by genuine users before false lockout then it can be assumed that scenario 1 has performed well with the most lowest ANIA and highest ANGA as well.

Category	Scenario	Users	Normalized ANGA	Normalized ANIA
Very Good	1	71	1.00	0.22
Good	1	0		
Bad	1	4	1.00	0.41
Ugly	1	0		
System Total	1	**75)**	**1.0**	**0.23**
Very Good	2	49	1.00	0.27
Good	2	11	0.97	0.26
Bad	2	15	1.00	0.42
Ugly	2	0		
System Total	2	**75**	**0.99**	**0.30**
Very Good	3	49	1.00	0.24
Good	3	15	0.96	0.31
Bad	3	11	1.00	0.44
Ugly	3	0		
System Total	3	**75**	**0.99**	**0.28**

Table 4.4: Across Sequence Split–Experimental Setting I

4.3.2.2 Experimental Setting II: Weighted Classifier fusion with Personalized RCM

It can be observed from Table 4.5 that: *In scenario 1*, 8 participants qualify for the 'very-good' category, where the mean of ANGA is 1.00 actions which represents that none of the genuine participant has been locked out leaving the ANGA 100%, whereas the mean of ANIA is 0.05 which indicates that all the imposters for these 8 genuine users have been detected before performing 0.05% of actions. Subsequently, the 67 users fall in good category with ANGA and

ANIA being 0.80 and 0.09 (9% actions) respectively.
In scenario 2, there are 4 users in very good category with ANIA 0.28 which is quite high as compare to ANIA of scenario 1 while the 71 users fall in good group where mean of ANGA is 0.75 and ANIA is 0.10.
In scenario 3, it can be noticed that 23 users are falling in very good category with mean ANIA 0.15 which is better than scenario 2 while the 52 users fall in good group where mean of ANGA is 0.72 and ANIA is 0.12 actions.

Overall, the system performance has been evaluated based on the number of actions performed by imposter before detection then then it can be assumed that scenario 1 has performed well with the most lowest ANIA and highest ANGA as well. And secondly, scenario 3 worked well for keeping the most of genuine user logged in for the whole of testing sessions and not locked out falsely even once.

Category	**Scenario**	**Users**	**Normalized ANGA**	**Normalized ANIA**
Very Good	1	8	1.00	0.05
Good	1	67	0.80	0.09
Bad	1	0		
Ugly	1	0		
System Total	1	**75**	**0.82**	**0.09**
Very Good	2	4	1.00	0.28
Good	2	71	0.75	0.10
Bad	2	0		
Ugly	2	0		
System Total	2	**75**	**0.76**	**0.11**
Very Good	3	23	1.00	0.15
Good	3	52	0.72	0.12
Bad	3	0		
Ugly	3	0		
System Total	3	**75**	**0.80**	**0.13**

Table 4.5: Across Sequence Split–Experimental Setting II

4.3.2.3 Analysis for setting I and setting II

Aggregated results of DCS with static RCM parameters (setting I) and weighted fusion with personalized parameters optimised with genetic algorithm (setting II) given in Table 4.4 and Table 4.5 respectively are referred here. First of all, it can be noticed that for static global RCM there are users in all three scenarios who are falling in bad categories which mean there are some genuine users against which the imposters could not be caught up even after performing more than 40% of actions. On the other hand, in setting II with personalised parameters, it can be observed that all of users are falling in either very good or good category where all the imposters have been caught before performing 40% of actions which also means that none of the imposter got undetected. More precisely in setting II, the only worst case has been observed in scenario 2, where imposters could have performed 28% actions on average before detection. Except this case, on average most of the imposters had been detected before performing 8% of actions in setting II. Hence, it can be concluded that proposed setting II has performed well in detecting the imposter users since it includes the personal parameters of each user for R-RCM optimised by genetic algorithm as well as weighted classifier fusion approach.

If the System ANIA are computed for scenario I in relation to the portion of users falling in each category for both experimental settings then the system's ANIA can be calculated with the equation 3.3 as follows::

- ***Experimental Setting I***

  $$\textit{System ANIA}= \frac{(0.22*71)+(0.41*4)}{75} = 0.23 \text{ or } 23\%$$

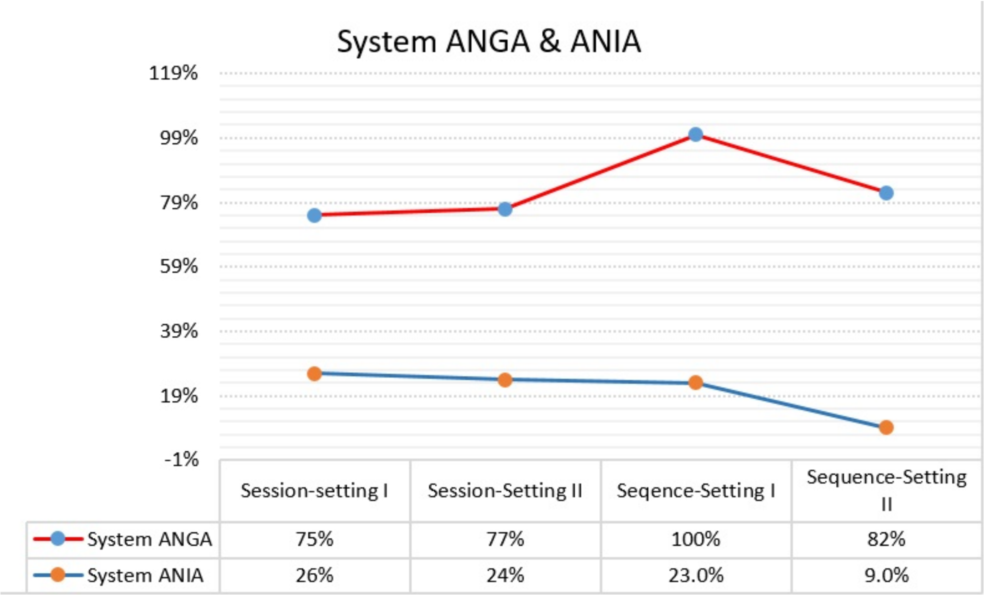
- ***Experimental Setting II***

  $$\textit{System ANIA}= \frac{(0.05*8)+(0.09*67)}{75} = 0.09 \text{ or } 9\%$$

It can be noticed that the System's ANIA for our experimental setting II has been the lowest as compared to our setting I.

4.3.3 Analysis on Across Session and Across Sequence

Best scenario results of both split strategies i.e., Across session (scenario 3) and Across sequence (scenario 1) are compared as shown in Fig 4.12. It can be observed that across sequence setting I & II has highest ANGA and lowest ANIA as compared to across session split strategy. Moreover precisely,if sequence split setting I and II are compared, then setting II has the lowest ANIA as compared to setting I, hence it can be assumed that setting II has worked well in detecting the imposter users in only performing 9% of actions. More formally, when two CUA systems are compared then the system with lowest ANIA is considered optimal from the perspective of security. However, if system's ANGA is taken into account then experimental setting I has performed well but as stated earlier, if two CUA systems are compared then the system which detects imposter users faster is considered the best one so in experimental setting II ANGA can be a trade-off for such environments where confidentiality and integrity of data and resources are main priorities.



	Session-setting I	Session-Setting II	Seqence-Setting I	Sequence-Setting II
System ANGA	75%	77%	100%	82%
System ANIA	26%	24%	23.0%	9.0%

Figure 4.12: ANGA & ANIA percentage for all the four experiments

Secondly, across sequence split strategy has overall performed well as compared to across session split strategy. As discussed before, since there is an average gap of 28 days between the capture of each session and according to this it means that session 2 has been recorded approximately after 56 and 28 days in comparison to first two (0,1) sessions. Moreover, it has turned out during experimentation of session split strategy that users exhibit different patterns between sessions, and a model with high train and test performance, drops it drastically at validation split. On the other hand, considering Sequence split strategy has reduced a gap between training and validation data during experiments. So it can be assumed that time gap between training and validation data have temporal effects which can affect the behaviour of user while using the keyboards. It can be one reason of lower performance of session split as compared to sequence split which has been researched more in detail in chapter 5 with deep learning methods.

4.4 Summary

The true CUA system works on authenticating the user based on the typing behaviour which distinguishes one user from the other. The implemented system has focussed on the dilemma of validating the user's identity on each and every action instead of authenticating on blocks of actions thereby lessening the risk of imposter activity to a greater extent. A two phase system methodology has been implemented and results are reported in terms of normalized portion of ANGA and ANIA. Subsequently, the combination of monographs and digraphs features have been used thereby leaving no room for imposters to do illicit activity in between the digraph features. The ensemble learning approach including SVM, ANN and XGboost is used to increase the accuracy score of each action. Since keystroke biometric is a weak modality and integration of multiple diverse classifiers has escalated the confidence in classification of each action thereby increasing the system performance. Moreover, the two dataset split strategies have been tested with different experimental strategies and it has been found out the system performance is lower in session split strategy as compared to sequence split strategy because users exhibit different patterns between sessions, and a model with high train and test performance, drops it drastically at validation split. One reason could be the time difference between training and validation data collection sessions. To do more research on it, the deep learning methods are applied in chapter 5 with session split strategy to compare the results with baseline session dataset split findings.

Chapter 5

Continuous User Authentication using Keystroke Dynamics Based on Deep Neural Networks

Behavioural biometrics of users tend to change depending on different factors including time, background context, age or hardware specification. As described in the aforementioned chapter, when the validation data of user is taken after the gap of few months from the training data then users show different behavioural patterns in validation data. This chapter aims to study the effect on system performance with the implementation of deep learning classification techniques which can treat the data as a sequential series and learns the hidden features from raw data without depending over the statistical user profiles.

5.1 Introduction

A true continuous authentication system, based on keystroke dynamics, is presented which tends to validate the user on each action. Keystroke Dynamics Recognition (KDR) is a behavioural biometrics method which can incessantly authenticate the user by analysing the typing rhythm of each individual. Continuously authenticating the user on each action based on behavioural information is substantially challenging work because sometimes behavioural data collected by explicitly configured software Vyazigin et al. (2019) is scarce along with having huge intra-class disparities to some extent.

Moreover, the keystroke dynamics is a behavioural biometrics hence it alters gradually with time or based on background context more as compared to physiological traits. Subsequently, sustaining the static database for any given user can lead to lessened accuracy over time. One solution can be perpetual addition of new keystroke samples after specific time. However, the major drawback of this process is enormous memory consumption especially if the system is dealing with substantially huge number of users. Moreover, traditional algorithms process the keystroke input as a vector, however, keystroke input samples are more similar to a sequence. Mostly, it is assumed in traditional neural networks that all inputs are autonomous of each other.

Several traditional statistical methods had been exploited for classification of KDR system including Euclidean distance Bours and Mondal (2015*a*), scaled Manhattan distance with Mean of Horners Rules Chandranegara et al. (2020) and Mahalanobis distance Ayotte et al. (2019). In addition, machine learning techniques aiming on prediction strategies are also adopted for CUA using KD. In this regard, K-Nearest Neighbor (KNN) classifiers Shikder et al. (2017), Decision trees Alsultan et al. (2017), Support vector machine (SVM) Çeker and Upadhyaya (2016) and artificial neural networks (ANNs) Ahmed and Traore (2013) had been applied for KD classification problems.

The major limitation of these methods is their difficulty to substantially and appropriately tackle the non-linear discrete problem scenarios. Exceptionally, Artificial neural networks (ANNs) are frequently considered to elucidate the non-linear association among varied data dimensions. However, conventional neural networks process the input as vector where all inputs are independent of each other. In contrast, keystroke patterns are more of a chronological sequence Tse and Hung (2020) which can contain hidden features. For instance, if a person commits mistake while typing and correct it, then this behaviour will be saved in keystroke sequence. This sort of information cannot be stored in statistical profiles and the conventional classification methods also cannot mine that hidden properties which can also uniquely distinguish one user from the other.

Therefore, in order to make CUA system scalable and more reliable, this research work has used recurrent neural network (RNN) which productively learns time-series data and generates the high quality features.

5.2 System Methodology

This section presents the system methodology and architecture of proposed system based on deep learning methods. Two types of architectures have been proposed:

- Hybrid Deep learning and R-RCM Model
- Integrated LSTM Per Frame and Per Sequence

5.2.1 Dataset Analysis

As discussed in chapter 4, it had turned out during experiments of session split strategy for dataset that users exhibit different patterns between sessions and a model with high train and test performance drops it drastically at validation split. On the other hand, sequence spilt strategy has reduced the gap between training and validation data during experiments. It has been assumed that time gap between training and validation data could affect the user to show different behaviour. In this aspect, recurrent neural network has been used in this chapter which can effectively learn the time-series data.

Dataset Split: For both proposed architectures, session spilt strategy is used, as discussed in chapter 3 section 3.2.1.1, where session [0,1] are used for training and testing while session [2] is used for final validation of model.

5.2.2 Hybrid Deep Learning Model

The proposed hybrid deep learning and R-RCM model utilises the recurrent neural network (RNN) Medsker and Jain (2001) to learn the hidden features of keystroke behavioural data and it integrates the proposed robust recurrent confidence model, as discussed in chapter 3 section 3.2.3.2, to make the system continual. Different architectures with RNN have been experimented in order to make CUA system robust and more secure. The different stages of the proposed model are discussed below:

5.2.2.1 Keystroke Sequence Sampling

Keystroke patterns can be assumed as a sequential series comprising of key-press and key-down events. Formally, a keystroke sequence is a chronological organisation of set of events (E) representing a time series.

Let's suppose, there is tuple $(UserId', SessionId', DownTime, UpTime, KeyCode)$ of keystroke events and those events are assembled to make up a sequence:

$Sequence(UserId', SessionId') = \{e | \forall e \in E, s.t.$ where
$UserId(e) = UserId'$,
$SessionId(e) = SessionId'$,
$DownTime(e) = DownTime$,
$UpTime(e) = UpTime$,
$KeyCode(e) = KeyCode'$,
$\}$

Formally, the order of actions is imposed by the following sorting criterion:

$e_i < e_j$ if

$$DownTime(e_i) < DownTime(e_j) \text{ or}$$
$$DownTime(e_i) = DownTime(e_j) \text{ and}$$
$$UpTime(e_i) < UpTime(e_j)$$

The proposed model portions the whole data into defined length keystroke sequences so that RNN can learn the keystroke time-series.

The one attribute and five main features are utilised in the CUA system, namely *key-codes*, *monograph* durations, *digraph latencies* i.e., *DD, DU, UD and UU. Key Code* belongs to a limited set of values with a power equal to C and it is transformed with one hot encoding. To apply a classification algorithm, input data has been processed to obtain numerical feature series as given follows: For $\forall t = \overline{0, M-1}, p = t - 1$, there is:

- $X_{t0} = KeyCode_t$
- $X_{t1} = KeyCode_p$
- $X_{t2} = (UpTime_t - DownTime_t)$
- $X_{t3} = (UpTime_t - UpTime_p)$
- $X_{t4} = (DownTime_t - DownTime_p)$
- $X_{t5} = (UpTime_t - DownTime_p)$
- $X_{t6} = (DownTime_t - UpTime_p)$

5.2.2.2 Architecture

Lets assume, there are M subject users and system needs to classify each user based on given action and sequence containing keyboard actions. So, formally there is:
$KS = \{(a, b)\} \subset \mathbb{R}^{X \times T} \times \{1, \ldots, M\}^T$,

where a_t –represents a keyboard action at a particular time-step t, $b_t \in \{1, \ldots, M\}$ – user who executed the current keystroke, T – total number of actions or time-step, X – action vector dimension precisely. The proposed system predicts the identity of user b_t per time step t, and decides on each action if it belongs to genuine user or not.

5.2.2.3 Recurrent Neural Network (RNN)

A recurrent neural network (RNN) is mostly used for the problems containing time-series data thereby it can be employed for keystroke dynamics data owing to its sequential nature consisting of organised timestamps for each action. Generally, RNNs contain loops which permit the perseverance of information and each loop permits information to pass through it onto the next loop. Each loop receives the input information at any time step and produces the output value which would be passed to next loop transferring the information from one step of network to another. RNN can be considered as various replicas of same network connected together in a sequential manner in order to transfer information to subsequent successor as shown in Fig.5.1.

Basic RNNs are easy to understand with simple architecture as compared to other neural network models Tse and Hung (2020). However, it is difficult to train basic RNNs owing to its instinct problem called as exploding or vanishing gradients which substantially hinders learning of long data sequences.

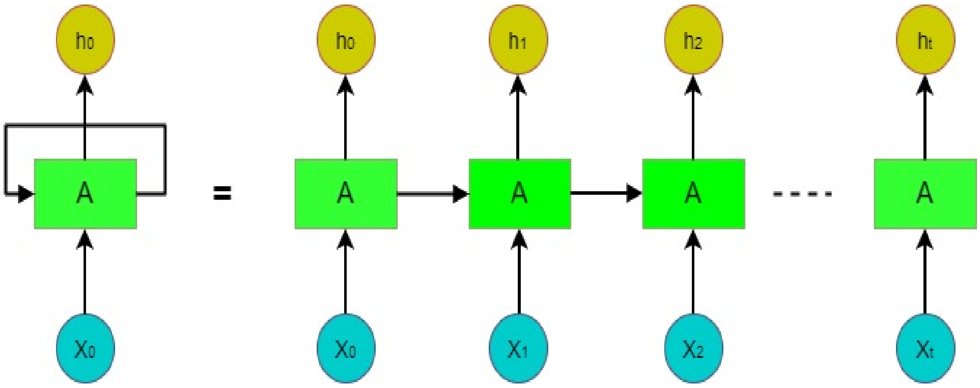


Figure 5.1: Unrolled Recurrent Neural Network

Long Short Term Memory (LSTM)

In this research work, the more refined form of RNNs known as long short-term memory (LSTM) network has been implemented in order to tackle the problem of diminishing gradients hence making it appropriate for effectively learning long term dependencies. More formally, the architecture of basic RNNs have series of simple iterating neural network units and each unit contains a single tanh layer Xiaofeng et al. (2019). In contrast, the core approach of LSTM is that, along with tanh layer, it contains a cell or vector known as gate which functions as a memory. This vector stores and modifies the information on each step hence network can write or remove/modify the information to/from the memory through computational steps. The LSTM model is defined in Xiaofeng et al. (2019) as below:

$$
\begin{gathered}
k_i = \sigma(W_kkx_i + b_kk + W_hkh_i - 1 + b_hk) \\
d_i = \sigma(W_idx_i + b_id + W_hdh_i - 1 + b_hd) \\
g_i = tanh(W_kgx_i + b_kg + W_hgh_i - 1 + b_hg) \\
o_i = \sigma(W_kox_i + b_ko + W_hoh_i - 1 + b_ho) \\
c_i = d_ic_i - 1 + k_ig_i \\
h_i = o_itanh(c_i)
\end{gathered}
\tag{5.1}
$$

where k_i, d_i, g_i and o_i are the input, forget, cell and output gates respectively, c_i is the cell state at time i, h_i is the hidden state at time i, x_i is the input at time i and σ is the sigmoid activation function.

The basic LSTM model containing one layer investigates each individual part of data sequence on each timestep. However, LSTM model which consists of two or more layers investigates each part of sequence on first layer and aggregates the results from each timestep to generate the final output. Afterwards, the second layer of model receives a hidden state from first layer as an input and updates its memory cell accordingly to generate an output for subsequent layer.

5.2.2.4 Architecture 1: LSTM and Robust recurrent confidence model (R-RCM)

The proposed framework compares three settings in experiments:

- Setting I: Only LSTM network
- Setting II: LSTM merged with our proposed RCM (without alert threshold)
- Setting III: LSTM merged with our proposed R-RCM (with alert threshold)

The system framework has been shown in Fig.5.2. The reference deep network architecture employs one LSTM layer, three fully connected dense time-distributed layers and final layer is the output layer predicting the probability of given sequence action as shown in Fig.5.2. Moreover, a dropout layer is configured between every two layers in order to lessen the menace of over fitting.

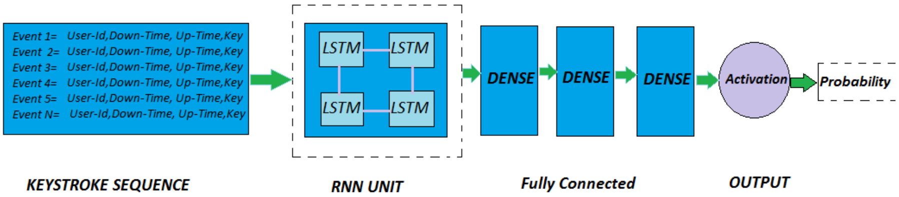

Event 1= User-Id,Down-Time, Up-Time,Key
Event 2= User-Id,Down-Time, Up-Time,Key
Event 3= User-Id,Down-Time, Up-Time,Key
Event 4= User-Id,Down-Time, Up-Time,Key
Event 5= User-Id,Down-Time, Up-Time,Key
Event N= User-Id,Down-Time, Up-Time,Key
LSTM
LSTM
LSTM
LSTM
DENSE
DENSE
DENSE
Activation
Probability
KEYSTROKE SEQUENCE
RNN UNIT
Fully Connected
OUTPUT

Figure 5.2: Framework of RNN Network Layers

The architecture of implemented LSTM is given table 5.1 below:

Layers	Explanation
Input Layer	3-dimensional
LSTM Layer	128 hidden units
batch Normalization	128 hidden units
3-fully connected time distributed	128 units
activation (Activation)	76 units
dropout	to avoid overfitting

Table 5.1: LSTM Network Structure

The raw data in the form of series of monographs and digraphs are feed into the LSTM which generates the probability output based on each action. In second and third approach, RCM or R-RCM function has been merged into LSTM respectively, it receives the output from LSTM output layer and applies hyper parameters to decide if user can continue using the system or should be locked out based on final threshold of RCM or R-RCM.

5.2.2.5 Architecture 2: LSTM per Frame and LSTM per Sequence

The proposed architecture integrates the two classification approaches to get the probability based on per action as well as per sequence. Let's say, there is a sequence of $M + U$ keystrokes where U is the context length and M is the

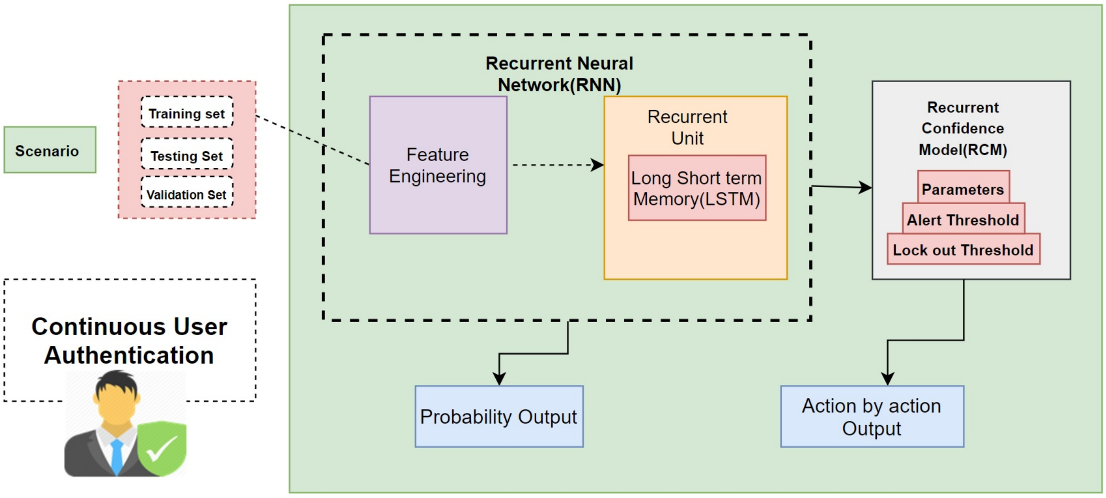


Figure 5.3: The System Architecture 1

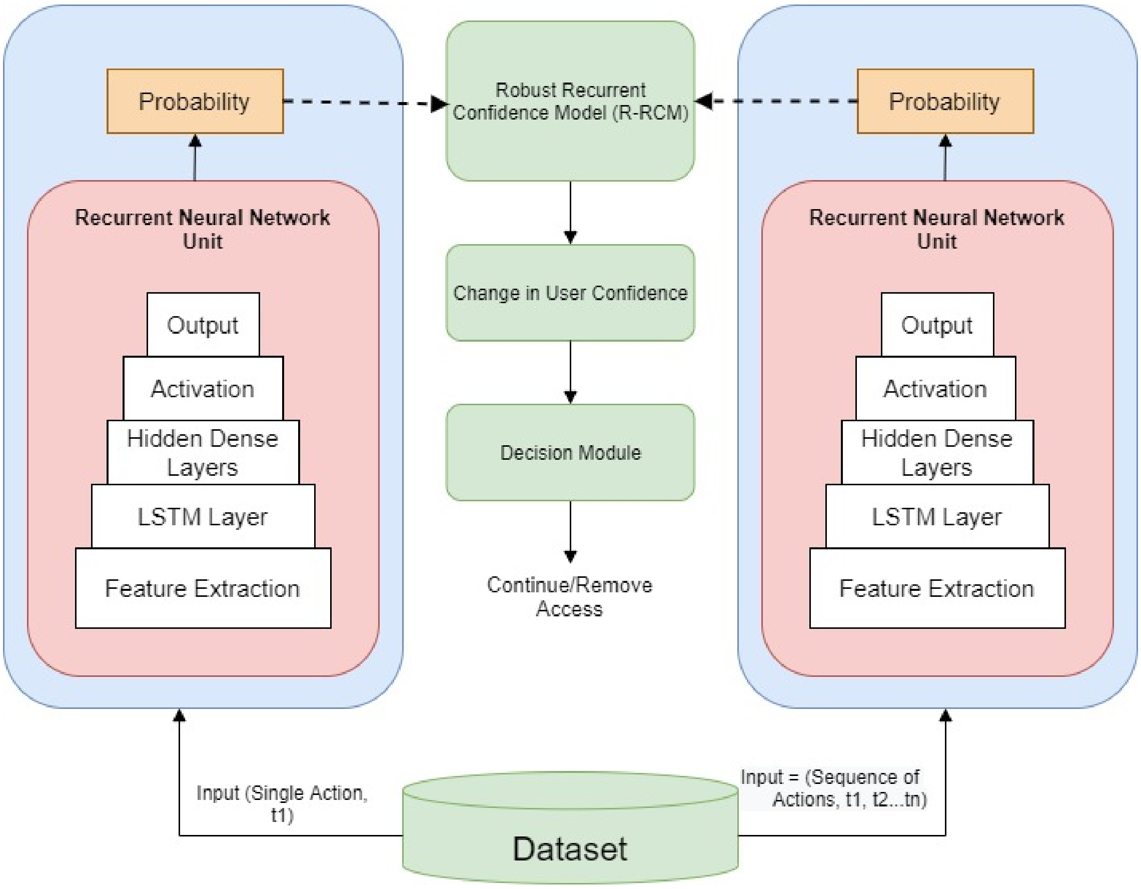


Figure 5.4: The System Architecture 2

length of keystroke sequence. Sequences of a defined length $M + U$ have been sampled to generate input features and target user ids *(x,y)* with T *time steps* in total.

There are two setups in practice:

- $U = 1$ and $T = M = 1$, a single keyboard action with monograph and digraph features for per frame classification.
- $U = 1$ and $T = M = 64$, a sequence of keyboard actions with monograph, digraph and n-graph features for per sequence classification.

The system architecture has been shown in Fig.5.4. The first presented model based on per frame classification takes the input data as each action and extracts the action based features containing monographs and digraphs. On the other hand, the second model based on sequence classification segregates the keystroke data into fixed length keystroke sequences and generates the input patterns according to the timing features of keystrokes containing monographs, digraphs and n-graphs. The fixed length sequence used in this study are based on 64 time-steps which contain enough hidden features for the behavioural patterns of given user.

Afterwards, the processed sequence containing the monograph, digraph or n-graph features are fed into the LSTM network that has been effectively trained to extract the unique behavioural patterns of user from given sequence and then fully joined with dense layers to produce the final probability output.

The probability output for both models goes to the recurrent confidence model which changes the confidence level according to probability output and its hyper parameters. It must be noted that output from per frame LSTM would go into the R-RCM on each action, while the output from per sequence would go into the R-RCM after 64 actions. Therefore, per frame output makes the changes in user's confidence after each action whereas per sequence output makes the change in confidence of user after 64 actions. In a case, when the user's confidence becomes low than the final threshold before the user has completed the 64 actions then the user would be lock out of system without waiting the user to complete the 64 action window.

The core notion of integrating the sequential approach is that model can learn the hidden behavioural features from a given sequence and generates an output according to the unique behavioural pattens which cannot be extracted through per frame. Moreover, it is assumed that input keystroke sequence would never be entirely different for given user, conversely, it would gradually alter from time to time based on external factors. Accordingly, LSTM network is trained so that it can learn the unique features of current input sequence while remembering the prior input features owing to its memory cell structure.

Formally speaking, this architecture combines the continuous and periodic authentication owing to classification based on per action and per sequence strategy respectively. It combines the advantage of per action features which specifies the user behaviour on each action with per sequence features which can depict the unique hidden user behaviour based on general computer usage.

5.3 Results and Discussion

The performance metrics described in chapter 3 section 3.2.6 for CUA system have been used which includes normalized portion of ANGA and ANIA and EER. Additionally, the four categories based on ANGA and ANIA as describe in chapter 3 section 3.2.6.1 are also used to assess the system performance.

The *External Threat Scenario* $ETS3$ (scenario 3) is used for training and testing purpose as it represented the worst performance in detecting the imposter users in chapter 4 with baseline techniques. (see tables 4.2 & 4.3.) The reason for worst performance could be that all the imposters used in validation of model had not been used for training, so all the imposters in this scenario are assumed to be external to the organisation for final validation. This scenario is chosen to work with deep learning methods to check if the advanced deep learning techniques can detect the external imposters whom data samples are not included in the training.

5.3.1 Results in terms of ANGA and ANIA

For Architecture 1, some extracts of the results based on 512 actions are shown in Fig 5.5 where genuine user sample has been validated with its own data(right side) and with imposter data (left side).

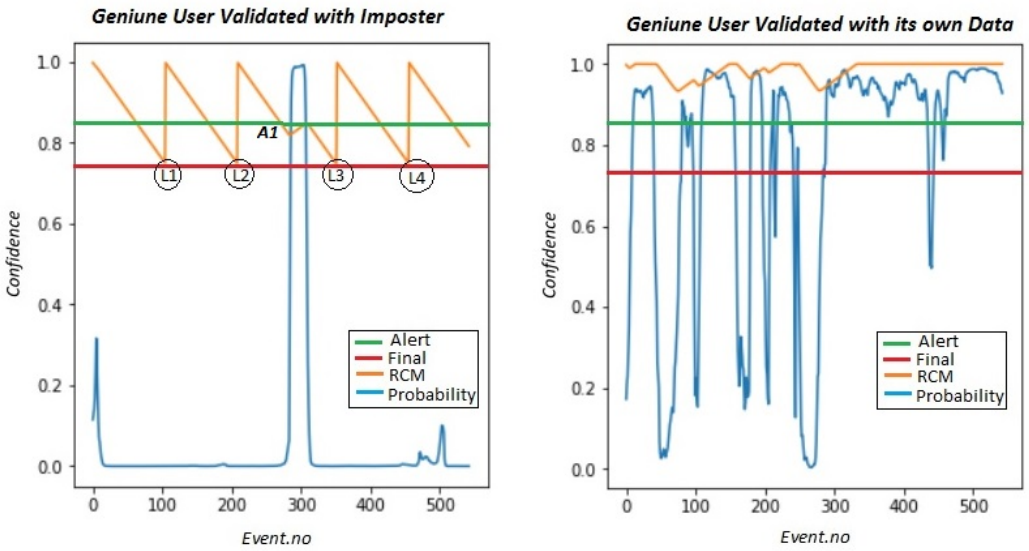


Figure 5.5: Genuine user validated with its own reference set(right) and with imposter set(left)

It illustrates the combination of LSTM and R-RCM output (setting 3), it can be noticed on left side part that the user has been logged out of system 4 times during 512 actions. For testing purposes, user's confidence is set to its highest level after each lockout assuming it has again used the SUA credentials to login back to system. It can be noticed during the third attempt, user has crossed the alert threshold marked by A1 in Fig.5.5. After few actions, it can be observed that probability of actions drastically increased from LSTM output which shows that actions are probably done by genuine user. However, the R-RCM has not increased the confidence of user rapidly instead it granted points less than usual because the overall confidence is still lower than alter threshold. Afterwards, user's probability again dropped from LSTM output and R-RCM locked out the user as soon as possible in order to limit the damage caused by imposter attempt.

5.3.1.1 Aggregated Results for all three settings of Architecture 1

Here, the aggregated results for 75 users are presented in tabular form for all the three settings. Table 5.2 shows the results of only LSTM model where 60% or 45 users are falling in very good category while 40% or 30 users are falling in good category with ANIA being 0.20 and 0.28 respectively.

Category	Scenario	Users	Normalized ANGA	Normalized ANIA
Very Good	3	0.60 (45)	1.00	0.20
Good	3	0.40 (30)	0.83	0.28
Bad	3	0		
Ugly	3	0		
System Total	3	**1.0(75)**	**0.93**	**0.23**

Table 5.2: Architecture 1 (setting I): Aggregated Results of LSTM only

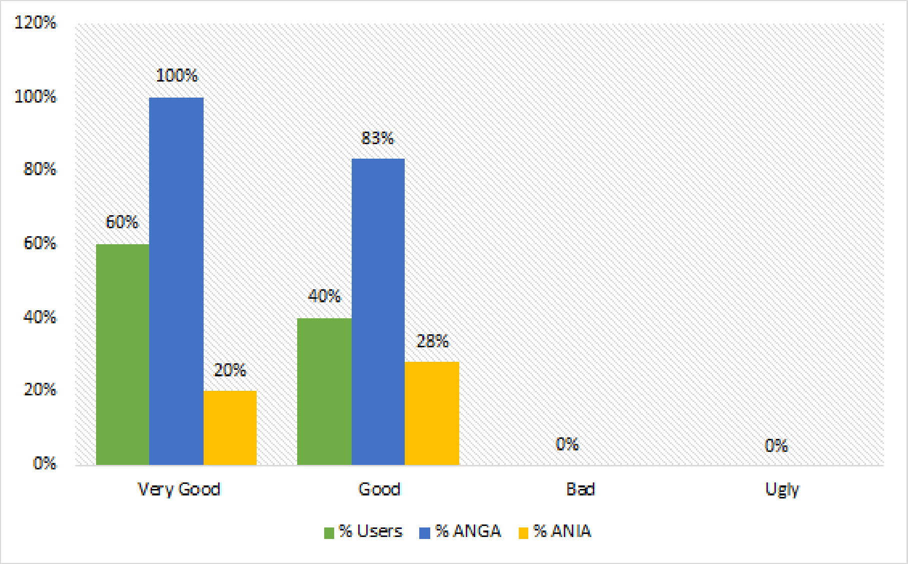


Figure 5.6: LSTM results represented in percentage

On the other hand, Table 5.3 shows the results of LSTM + RCM where 54 users are falling in very good category whereas only 21 users are in good category with ANIA 0.17 and 0.18 respectively.

Category	Scenario	Users	Normalized ANGA	Normalized ANIA
Very Good	3	0.72 (54)	1.00	0.17
Good	3	0.28 (21)	0.90	0.18
Bad	3	0		
Ugly	3	0		
System Total	3	**1.0(75)**	**0.97**	**0.17**

Table 5.3: Architecture 1 (setting II): Aggregated Results of LSTM-RCM

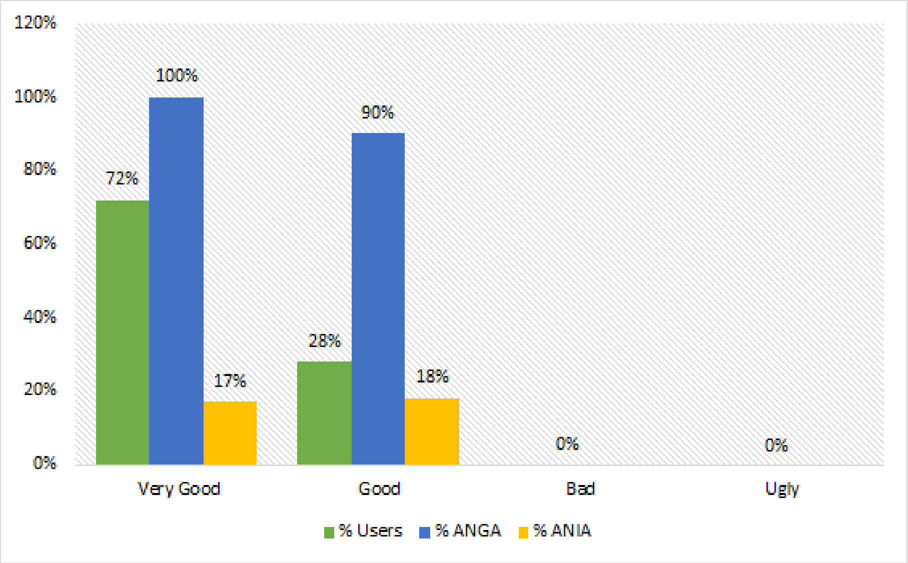


Figure 5.7: LSTM-RCM results represented in percentage

Table 5.4 shows the results of LSTM + R-RCM approach which contains the alert threshold to detect the imposter users quickly. It can be observed that system ANIA has reduced to 0.04 with this approach and imposter users have been detected by system more quickly.

Category	Scenario	Users	Normalized ANGA	Normalized ANIA
Very Good	3	0.88 (66)	1.00	0.04
Good	3	0.12 (9)	0.88	0.05
Bad	3	0		
Ugly	3	0		
System Total	3	**1.0(75)**	**0.98**	**0.04**

Table 5.4: Architecture 1 (setting III): Aggregated Results of LSTM-Robust RCM

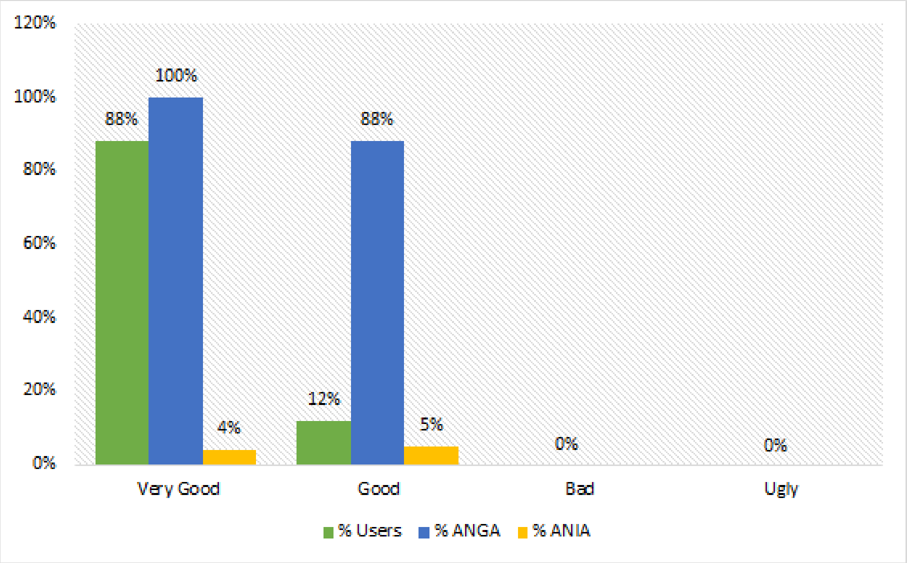


Figure 5.8: LSTM- Robust RCM results represented in percentage

5.3.1.2 Aggregated Results of Architecture 2: LSTM per Frame and per Sequence

The collective results of Architecture 2 have been listed below in Table 5.5. It can be observed that only 3 users have been falsely locked out of system and imposter users have been detected by system by only performing 0.016 actions.

Category	Scenario	Users	Normalized ANGA	Normalized ANIA
Very Good	3	0.96 (72)	1.00	0.016
Good	3	0.04 (3)	0.91	0.03
Bad	3	0		
Ugly	3	0		
System Total	3	**1.0(75)**	**0.99**	**0.016**

Table 5.5: Architecture 2: Aggregated Results of Integrated LSTM per Frame and per Sequence

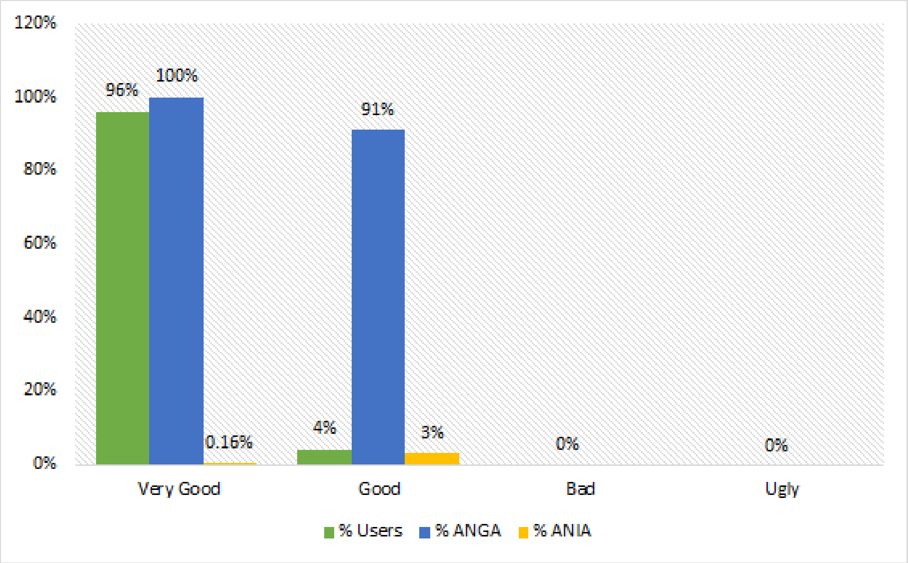


Figure 5.9: Integrated LSTM per frame and per sequence results represented in percentage

Comparison with Baseline Methods:

Moreover, if the system total of scenario no 3 results of baseline session split strategy given in chapter 4 table 4.3 is compared with the deep learning approaches then it can be observed in Table 5.6 that system ANGA has increased with LSTM and LSTM-R-RCM methods as compared to baseline while system ANIA has decreased with LSTM and LSTM-R-RCM methods as compared to baseline technique.

Methodology	Scenario	Normalized System ANGA	Normalized System ANIA
Baseline(Session split)	3	**0.77**	**0.24**
Architecture 1: LSTM (Session split)	3	**0.93**	**0.23**
Architecture 1: LSTM + RCM (Session split)	3	**0.97**	**0.17**
Architecture 1: LSTM + R-RCM (Session split)	3	**0.98**	**0.04**
Architecture 2: Integrated (Session split)	3	**0.99**	**0.016**

Table 5.6: Comparison of Baseline Session split with Deep Learning Methods

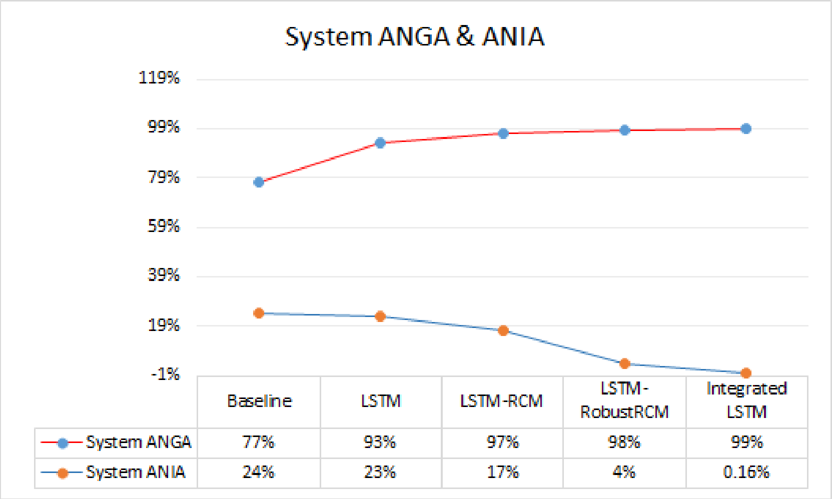


Figure 5.10: Comparison of baseline and deep learning methods

Fig 5.10 illustrates the System's ANGA and ANIA for the four proposed methodologies and baseline method which is discussed in chapter 4 table 4.3. It can be observed that System ANGA has substantially increased from simple baseline method to robust deep learning Integrated LSTM R-RCM which includes the two thresholds to authenticate user as well as includes two classification methods. Similarly, it can be noticed that system's ANIA has substantially decreased from simple baseline to robust deep learning model.

5.3.2 Results in terms of EER

Equal error rate (EER) has also been calculated to evaluate the results with previous research works. EER is a metric which assesses the data classification performance for any model. In this work, EER has been calculated for the optimal methodologies and the results are shown in Table below:

Methodology	EER %
LSTM-Robust RCM	3.2%
LSTM Integrated	1.04%

Table 5.7: Results in terms of EER

5.3.2.1 Comparison with Previous Research

Overall, deep learning models have worked well in avoiding the false lockout and quick detection of imposter thereby escalating the ANGA and lessened the ANIA respectively. Since deep learning method does not depend on statistical features which can change over time so it can retain the previous information. More specifically, if the ANGA and ANIA are demonstrated with exact number of actions instead of giving the normalized portion of actions then it can be calculated as follows using Eq 3.5:

$$Exactgeniune/imposteractions = Normalizedactions * Totalvalidationdata$$

The exact number of imposter actions for our optimal experimental setting i.e., robust LSTM R-RCM model and integrated LSTM, can be calculated with the above formula.

Exact imposter actions are of this research are given in Table 5.8

Methodology	Exact Imposter actions
Previous Research Bours and Mondal (2015*b*)	547
LSTM-Robust RCM	240
LSTM Integrated	96

Table 5.8: Exact Imposter actions for Keystroke Dynamics

It means the imposter users have been locked out of system only after performing 96 keyboard actions. Moreover, if the optimal experimental setting results i.e., 96 imposter actions are compared with previous scholarly work, then researchers in Bours and Mondal (2015*b*) had also used ANGA & ANIA as performance metric for CUA system. The ANIA reported for their optimal settings was 547 keyboard actions which is quite higher than the results achieved in this work. However, their results were demonstrated in the exact number of actions which makes it difficult to compare the work because of different amount of validation data in any experimental work. On the other hand, the method used in this research to calculate the normalized average portion of genuine and imposter actions will make it easy to compare the results regardless of amount of validation data in any future research work done for true continuous user authentication system.

Additionally, if EER of this research work is compared with previous researches given in table 2.1, 2.2 then it can be observed that most of the research works have used the block size of actions then the researchers in Kim and Kang (2020) achieved the EER of 1% but they had utilised the varying keystroke sets instead of single keystroke action to authenticate the user. Moreover, another notable work presented in Lu et al. (2020) had also used the RNN for authenticating the users but researchers have used the sliding window approach consisting of sequence of block actions i.e., 50 actions and achieved the EER of 4.77%. In contrast, the results provided in this research have used the single action and achieved the lowest error rate (EER) of 1.04% precisely.

5.4 Summary

This chapter focuses on a true continuous authentication system, based on keystroke dynamics, which tends to validate or identify the user on each action by using the recurrent neural network (RNN). RNN has been used to exploit the sequential nature of keystroke data. Different Architectures have been experimented with RNN and the proposed Robust recurrent confidence model (R-RCM) to authenticate the user on each action along with learning the hidden behavioural features which can be represented through keystroke sequential series.

Chapter 6

Continuous User Identification Using End-to-End Deep Neural Model

This chapter aims to implement a novel strategy of continuous identification without the prior claim of user's identity at the start of session. All the proposed frameworks for continuous user authentication relies on static user authentication with usernames and passwords at the start of session. In this study, a novel method is proposed which can perform identification of the user continuously without the need of static user authentication at start of session. In this regard, an end-to-end deep learning model is trained which effectively learns the user identity on each action.

6.1 Introduction

End-to-end model is a method which trains the entire model simultaneously instead of discretely training its different constituents M Jomaa et al. (2020). Subsequently, end-to-end model lessens the human intervention for training of model and eradicates the requisite of separate schemes in order to integrate the multiple models. Moreover, a major principle of end-to-end training is that the model itself decides which features are important for the classification task and it does not depend upon the experiments or choice of researchers as shown in Fig6.1.

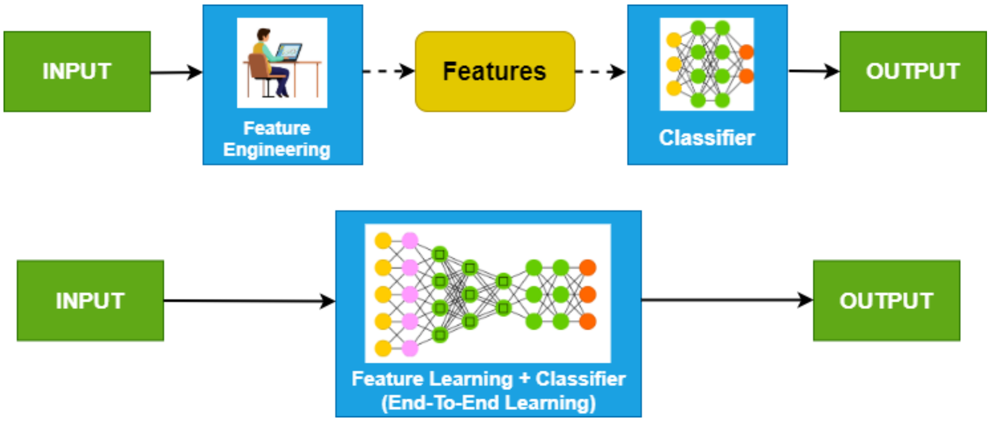


Figure 6.1: Comparison of Traditional Classification Techniques and End-to-End Deep Learning

The loss and accuracy are the common metrics for evaluating the performance of a classifier deep neural network. Loss is the difference between the output of the model (the model prediction) and the expected output. Accuracy is the number of correct predictions divided by the total number of predictions Paulsen et al. (2020).

6.2 End-to-End Deep Identification Model (E2E)

End-to-End Deep Identification Model (E2E) is based on multi-class classification problem which needs to identify any current user from a set pool of users. Subsequently, one model is trained which learns to differentiate among all the given users in dataset. In this aspect, deep learning model is used which has been trained in end-to-end manner. The proposed recurrent confidence model (R-RCM) is also part of this model to make the identification of user continuous. The core idea is to eliminate the need of static user authentication (SUA) in the start of session by employing the authentication credential like username and passwords. To best of our knowledge, continuous identification without the incorporation of SUA has been studied for the first time in this domain.

6.2.1 Architecture

Let's say, there is $D = \{(x, y)\}$, where x_{tj} is j–th feature value at a time step t, and y_t is an id of user that performs an action at time step t, $t = \overline{0, T-1}$.

6.2.2 Sequence Sampling

Two novel sequence sampling approaches are proposed and investigated for continuous user identification:

1. Sequences of single genuine user is fed into the model for training purpose at one time.

2. Sub-sequences of multiple users are stitched together to make a single sequence and model is trained to make correct recognition between transition of user identity from one subsequence to another.

User sequences are stitched together to train the model to learn the transition between different user identity correctly.

Given a set of subsequences $S = (S_0, S_1, \ldots, S_{r-1})$, it is required to alter their *DownTime* and *UpTime* so that they would occur sequentially, and do not exhibit anomaly patterns, otherwise model learns that in truth keyboard actions are independent short sequences rather than a single long sequence.

6.2.3 Gated Recurrent Unit (GRU): Model Training

Recurrent Neural Network (RNN) is trained to learn the identity of user on each action. Gated Recurrent Unit (GRU) is used as a recurrent unit this time which is considered to be computationally more efficient than LSTM Yang et al. (2020). Efficient performance of GRU was the reason of its selection over LSTM since joint model E2E was trained for 75 users that is why more computationally more compatible solution was required for the problem.

Firstly, network is trained for 10 users only where sequence samples from different users have been stitched together. The training network was set up as a classification problem with multiple classes, i.e., the network predicts which user a sample belongs to. The target labels, consisting of integers from 0 to 9 corresponds to user ids. Afterwards, model is extended by training 20 users, then to 40 users and then successfully to 75 users.

The rectified linear unit (ReLU) is used as activation function for the hidden layers and the Softmax function was used for the output layer. ReLU is considered to be non-linear function which is frequently used as activation function for deep neural networks owing to its low computational cost on GPU hardware. Softmax is used for the activation function of the output layer during pre-training. The architecture of reference deep learning model is given in Table 6.1:

Layers	**Explanation**
Input Layer	3-dimensional
GRU Layer	256 hidden units
GRU Layer	256 hidden units
batch Normalization	256 hidden units
time distributed	76 units
activation (Activation)	76 units
ctc	blank regions
dropout	to avoid overfitting

Table 6.1: GRU Network Structure

6.2.3.1 Region Labelling Approach

It is assumed if $|G(y_t)| >= 1$, i.e. there are multiple genuine users sequence or genuine single user sequence, then having more actions before classifying all of them at once increases overall accuracy. In such case, a novel approach of three different types of regions are introduced in order to recognise the user's identity and these regions are defined below:

- **Blank Region** Blank regions are those regions where in general confidence is always low, due to model not seeing enough samples, it is a beginning of each user sequence stitch or when user has just started using the system in case of single user sequence. For such regions, a previous label or blank label crossentropy loss must be enforced, or CTC loss that learns a transition between such. $BlankRegion$ represents first $MinDetectLength$ actions of a new user that are likely to make classifier uncertain about its prediction.

In such a case a separate class $N+1$ is being introduced, its called *Blank* as described in Graves et al. (2006), $y'_t = N+1, \forall t \in BlankRegion(G(y_t))$.

$$\begin{aligned} &BlankRegion(G(y_t)) = \{ \\ &\qquad g_{ij} | \forall j = \overline{0, MinDetectLength - 1}, \forall i = \overline{0, |G(y_t)| - 1} \\ &\qquad\qquad \} \end{aligned} \tag{6.1}$$

- **LabelShort Region** is the one right after a region of lower confidence, it is where it is safe to enforce crossentropy loss given that blank region has enough length, and it can be guaranteed that as of now user actions are classifiable without contradictions.

$$\begin{aligned} &LabelShortRegion(G(y_t)) = \{ \\ &g_{ij} | \forall j = \overline{MinDetectLength, MinDetectLength + ShortLabelLength - 1}, \\ &\qquad \forall i = \overline{0, |G(y_t)| - 1} \\ &\qquad\qquad \} \end{aligned} \tag{6.2}$$

- **LabelLong Region** is introduced mainly to account for a recurrent model, which is unwrapped before gradient estimation, and as such having an error at later actions per single user might have a worse gradient propagation comparing to earlier ones. Since at those points a main concern model should have is to keep to a previous prediction and monitor whether keystrokes with different dynamics are introduced, and in such a case it must collect enough evidence before switching to another user id prediction.

$$
\begin{aligned}
&LabelLongRegion(G(y_t)) = \{\\
&\qquad g_{ij} | \forall j = \overline{MinDetectLength + ShortLabelLength, |g_i| - 1},\\
&\qquad\qquad \forall i = \overline{0, |G(y_t)| - 1}\\
&\}
\end{aligned}
\tag{6.3}
$$

The illustration for different regions are shown in Fig 6.2. it can also be observed in Fig 6.2 that two user's sequences are stitched together in ground truth (left side) which means user 8 would start the system and user 4 would take over after 350 actions.

In the second part of figure (right side), it can be seen that the model has been relaxed in start and it has not been enforced to start prediction from 1st action. So in first few actions model predict the blank class which is discussed above which means it does not predict any user and this part has been labelled in Fig 6.2 as *BLANK REGION*. Subsequently, after that is the label short region where model learns to get confident with the user identity by accumulating enough samples and this part has been labelled in Fig 6.2 as *LABEL SHORT REGION*. Afterwards, as soon as model gets confident the label long region starts and it continues until the users is changed and this part has been labelled in Fig 6.2 as *LABEL LONG REGION*. Moreover, it can also be observed in Fig 6.2 that when user has been changed after 350 actions then model has again started over by predicting the blank class followed by label short and long regions.

Moreover, model is trained to insert maximum of 32 *emptyset* to mark the blank region, it means if imposter user has started using the system then within 32 actions it can be locked out of system since if model cannot identify the user from given set of genuine users then it will lock out the unknown user from system after 32 *emptyset*. Similarly, the SHORT REGIONS contain maximum of 20 actions where the model gets confident with its prediction. However, if model gets imposter actions after starting predicting the user identity then within next 20 action it can lock out the user. LABEL LONG REGION contains actions until the user is changed. The summary of maximum actions for each region is given in Table 6.2 below:

Regions	Maximum Actions
BLANK REGION	32
LABEL SHORT REGION	20
LABEL LONG REGION	Until Identity of User changes

Table 6.2: Maximum Actions for Different Regions Approach

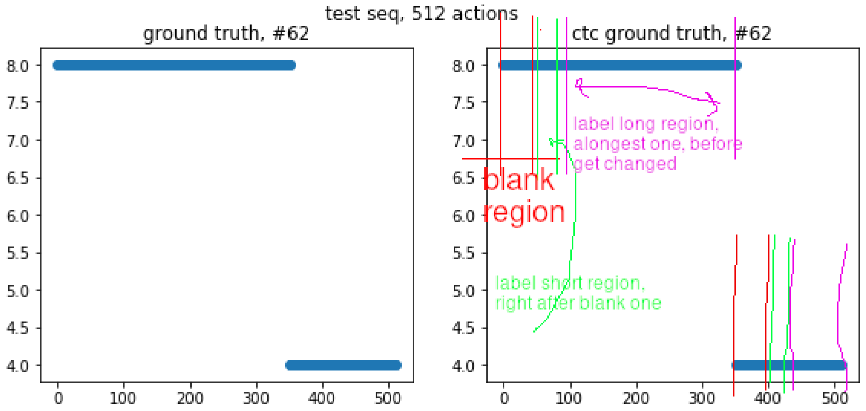


Figure 6.2: Illustration of proposed regions

6.2.3.2 GRU Training: Loss Functions

In practice, deep learning identification model is trained with sampled crossentropy loss and CTC loss and experimented to increase ANGA and decrease ANIA.

Crossentropy is being sampled for $LabelShortRegion$, and $LabelLongRegion$. On the other hand, $BlankRegion$ uses CTC loss instead of crossentropy, and is targeted to predict sequence $\emptyset UserId$. The $\emptyset$ is not inserted everywhere, but condition loss to align a sequence above in start of user session only until the model gets confident with user's identity. It allows to learn an adaptive amount of samples needed to give a first predict. Since just per frame accuracy is much lower than say an aggregated one across 32 actions or less.

6.3 Results and Discussion

For the system evaluation, results are reported with system's ANGA, ANIA and EER . To define these metrics for identification problem let's assume:

If there is a classifier among N users, $Genuine$ – a set of genuine users, I – a set of impostor users, then the following notion of $TruePositive$, $TrueNegative$, $FalseNegative$, $FalsePositive$ are defined as:

- y_t – ground truth values(True Labels/User Ids)
- $\hat{y}_t$ – predicted user ids
- $G(\hat{y}_t) = \{g_0, g_1, \ldots, g_{h-1}\}$ – set of continuous, disjoint, fully covering $\{1, \ldots, T\}$ subsequences.

  Taking these parameters in consideration, it can be defined:

1. $$\begin{aligned}&TruePositive(Genuine, y_t, \hat{y}_t) = \{\\ &\qquad g_j \cap g_i | \forall i = \overline{0, |G(y_t)|}, \forall j = \overline{0, |G(\hat{y}_t)|}, \text{ s.t.}\\ &\qquad y_{g_{j0}} \in Genuine \text{ and } y_{g_{i0}} \in Genuine\\ &\}\end{aligned}$$

2. $$\begin{aligned}&FalseNegative(Genuine, y_t, \hat{y}_t) = \{g_j \cap g_i | \forall i = \overline{0, |G(y_t)|}, \forall j = \overline{0, |G(\hat{y}_t)|}, \text{ s.t.}\\ &\qquad y_{g_{j0}} \notin Genuine \text{ and } y_{g_{i0}} \in Genuine\\ &\}\end{aligned}$$

3. $TrueNegative(Genuine, y_t, \hat{y}_t) = TruePositive(I, y_t, \hat{y}_t)$

4. $FalsePositive(Genuine, y_t, \hat{y}_t) = FalseNegative(I, y_t, \hat{y}_t)$

Given the above definitions:

- $$ANGA(Genuine, I, y_t, \hat{y}_t) = \\ |\cup TruePositive(Genuine, y_t, \hat{y}_t)|/|TruePositive(Genuine, y_t, \hat{y}_t)|$$

- $$ANIA(Genuine, I, y_t, \hat{y}_t) = \\ |\cup FalseNegative(I, y_t, \hat{y}_t)|/|FalseNegative(I, y_t, \hat{y}_t)|$$

The following Fig 6.3, presents the pre-processed ground truth stitched multiple user sequences from user 1 and 5, which have been trained to represent the transition between two user ids for the model.

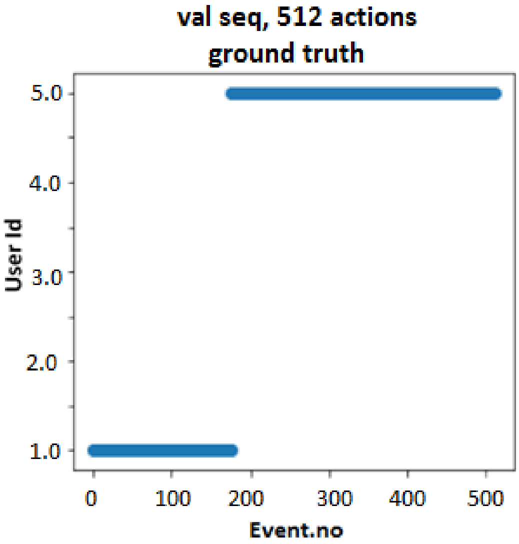


Figure 6.3: Pre-processed ground truth stitched multiple user sequences

Fig 6.4 shows the user id per action taken as argmax from softmax layer. Particularly in this case, the heuristic of inserting blank region is not applied and model is forced to predict a user id from a very first action. As a result, it can be noticed that quite often a user sequence is broken by a short sequence of incorrect user id, as well as in general has pretty short noisy predictions from user 2,4,7 and 9.

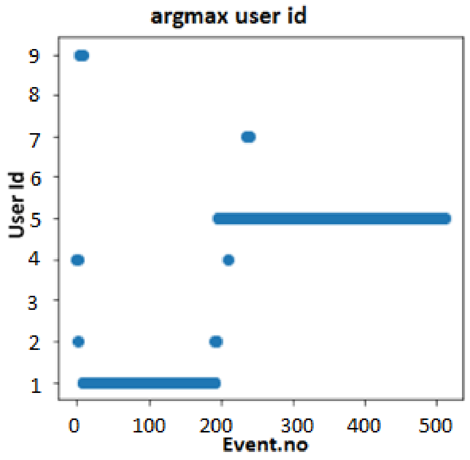


Figure 6.4: Model Predictions from Softmax GRU

Fig 6.5 presents a continuous user id prediction for a recurrent end-to-end model which has been trained with sparse crossentropy loss only. It can be observed that in the start model was not confident with any user id. However, model started predicted user 1 (colour labelled as orange) correctly and the confidence of user 1 started increasing after few actions which represents the LABEL SHORT REGION and model gets confident with identity of user for rest of action which is LABEL LONG REGION until the identity of user changes and next sequence from different user has started. Again, it can be observed that after 180 actions, model give noisy predictions until it starts identifying the user correctly (colour labelled as brown).

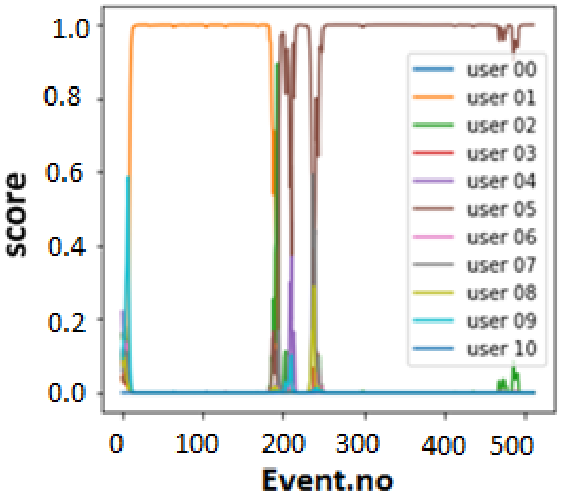


Figure 6.5: Recurrent Predictions from GRU-Robust RCM

Another similar example can be seen in Fig 7.6 where the ground truth user sequences are from user 2 until 90 action events then onwards user 9 action sequence is stitched. However, the argmax output start predicting from the first action since the methodology of inserting our black region is not enforced yet and model gives the noisy predictions from user 1, 3, 4 and 6 before it has started predicting the correct user which is 2. Moreover, for the rest of sequence still noisy predictions are present.

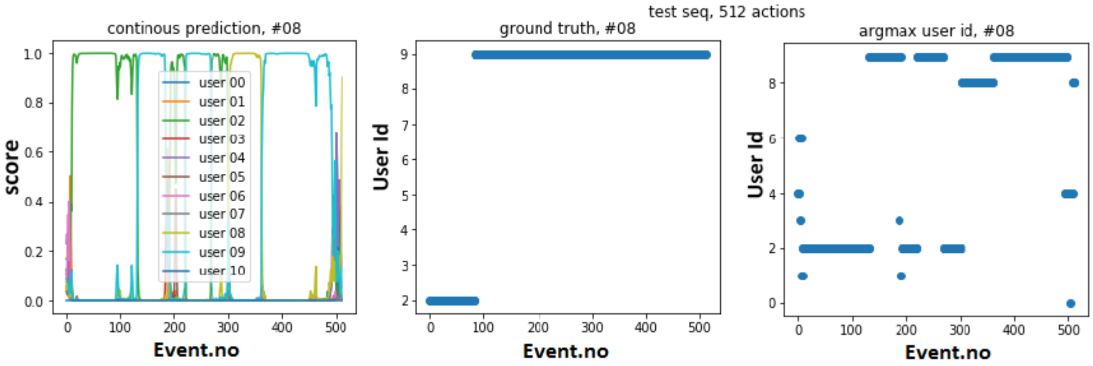


Figure 6.6: Recurrent deep learning identification

An objective for CTC loss was that, crossentropy one does not penalize these noisy short predicts, hence increases the imposter predictions or ANIA and lock out the genuine user hence decreases the ANGA. Therefore, the idea of inserting blank labels with CTC loss has been experimented.

The CTC loss and idea of inserting blank regions is implemented for multiple user cases in such way that model is conditioned to predict a transition sequence with just 3 labels, a previous stitch user id, *emptyset* and a next stitch user id. But in practice model does not predict *emptyset* in the transition area, but switches to a new user at some point. It allows model to keep a previous predict as long as becomes confident that a user has changed and it does provide a new predict.

A final model has been trained with sampled cross-entropy for LabelShortRegion and LabelLongRegion and sampled CTC loss for BlankRegion.

The following Fig 7.4 shows the single user (user3) sequence where the CTC has been implemented for blank region insertion in start of user identity. It can be noticed that model is relaxed in the start to delay the predict until it becomes confident with the user prediction. It can be observed that an extra empty class 20 has been added and model predicts the class 20 in the start which means it inserts *emptyset* in extra class instead of giving noisy predictions. Afterwards when model gets confident with the user identity it starts predicting the correct identity of user 3 for rest of the sequence.

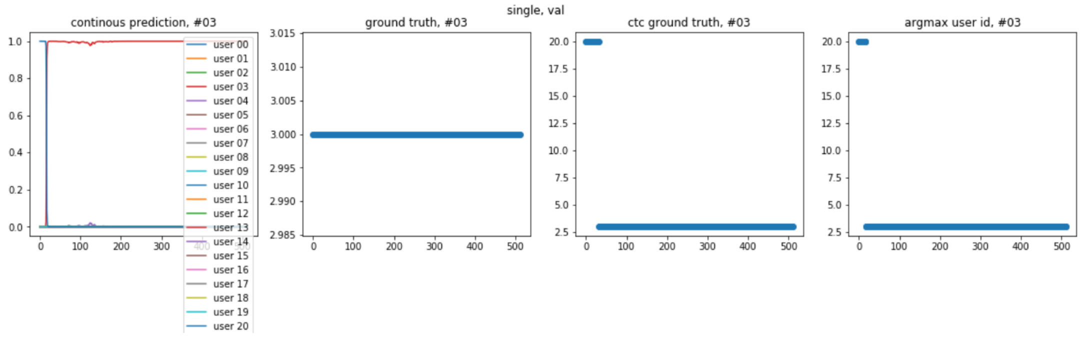


Figure 6.7: Single User Identification and Authentication

Moreover, if the first plot is looked closely, also shown below in Fig 6.8, then it can be noticed that confidence value of blank class(class 20) is highest which is illustrated with blue colour. However, once the model starts predicting the correct user identity for user no 3 then the confidence of blank class is dropped and confidence score increases for user 3 which is illustrated with red colour.

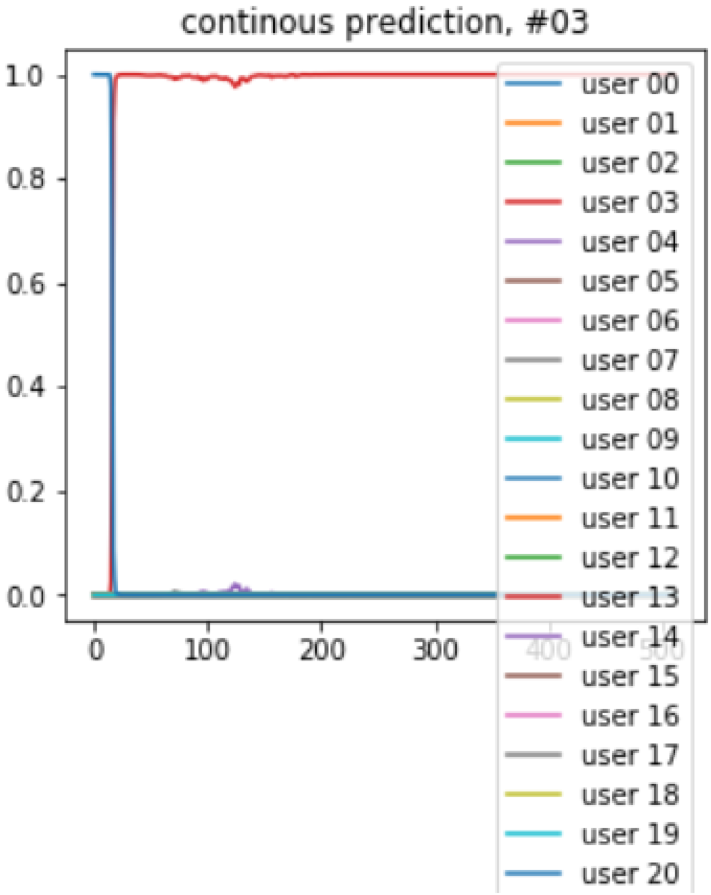


Figure 6.8: Recurrent Continuous Output of E2E Model

Fig 7.7 shows another example where the CTC has been implemented for blank region insertion in start of user identity when model is unsure of user's identity. It can be noticed in plot 4 that model has predicted the blank class through argmax output and then after few actions started predicting the correct class 15. However, the recurrent confidence plot 1 shows that as soon as the model started predicting the class 15 (coloured brown), its confidence decreased twice which could be owing to low probability score from softmax classification. But eventually the confidence increased and stayed same for rest of the sequence.

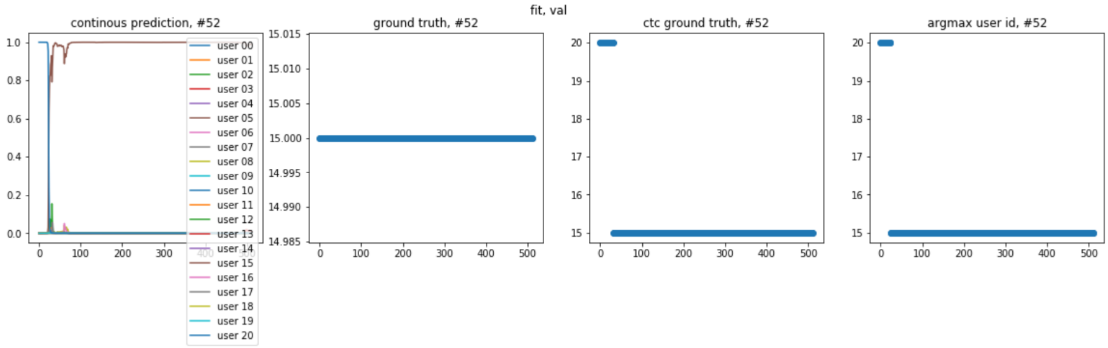


Figure 6.9: CTC Blank Region for User Identification

6.3.1 Aggregated Results for End-to-End Model

Now the aggregated result for all the users in tabular form are reported in Table 6.3.

Firstly, the model is trained for 10 users, then 20, 40 and finally extended to 75 users. The results are reported for 20, 40 and 75 users in table 6.3.

It is observed that out of total 20 users 18 are correctly identified by system for whole validation set while 2 genuine users are falsely locked out by the system. For 40 users, again 3 genuine users are falsely locked out by the system. For 75 users, 73 users are correctly identified by system for whole validation set while rest of 2 users are falsely locked out by the system. It can be observed in table 6.3 that system ANGA and ANIA are optimal for 40 and 75 users and the accuracy achieved by e2e model for 75 users is 93%.

Category	Total Users	Normalized ANGA	Normalized ANIA	Accuracy E2E Model
System Total	**20**	0.98	0.015	94.6%
System Total	**40**	0.99	0.009	90.3%
System Total	**75**	0.99	0.007	93%

Table 6.3: Aggregated Results of Recurrent End-to-End Model

The exact number of actions for imposter detection when calculated with Eq 3.5 are 42 actions.

The model is trained for 60 epoch and the following Fig 6.10 shows the comparison of model trained with 20 users (v42), 40 users (v43) and then 75 users (v44) precisely. On the left side, the loss on each epoch is shown for three configurations while the right figure shows the accuracy on each epoch for three configurations. It can be observed that model performance for v42=20 users on validation was highest i.e., 94.6% on last 10 epoch, however it was lowest in the start. Subsequently, the optimal model performance was for v44 = 75 users configuration on validation with high accuracy and low loss because of step-by-step continuous training.

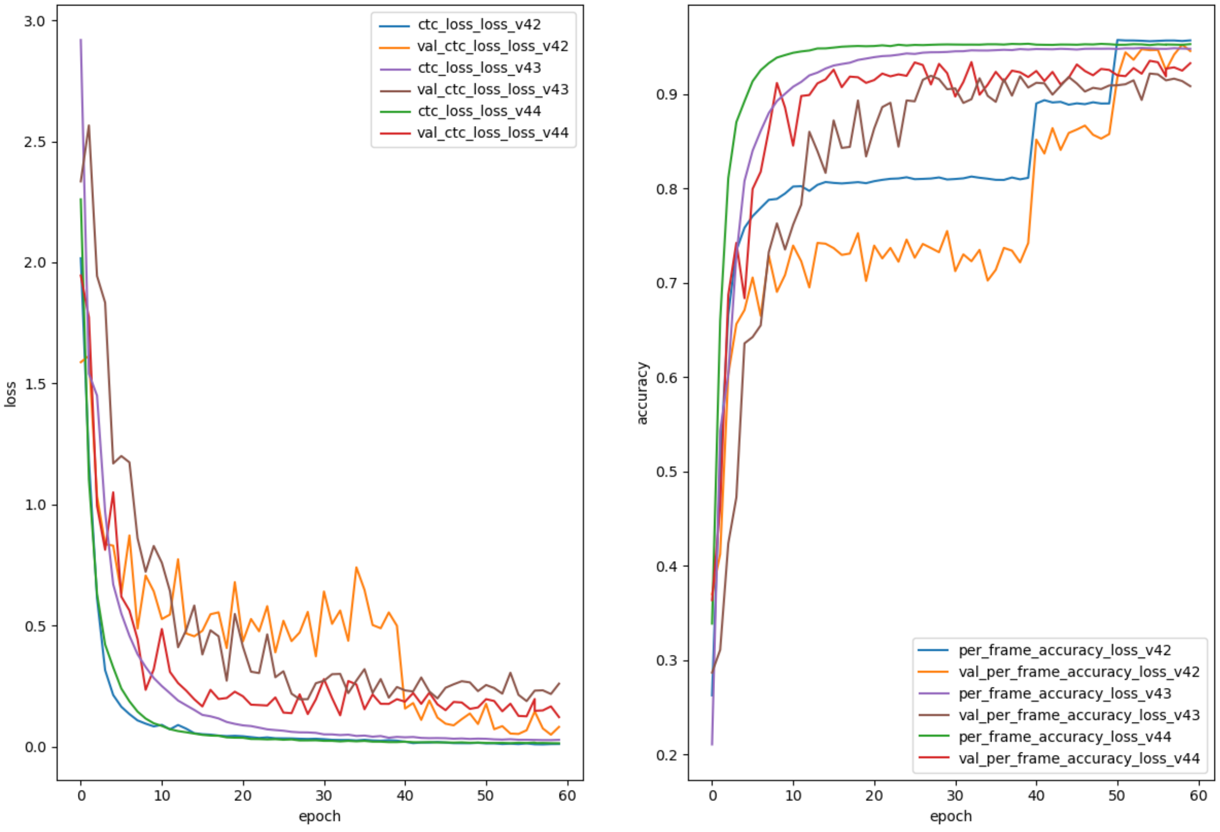


Figure 6.10: Accuracy and CTC loss Comparison for 20, 40, 75 users

6.3.2 Results in terms of EER

Equal error rate (EER) has also been calculated to evaluate the results with previous research works. EER is a metric which assesses the data classification performance for any model. In this work, EER has been calculated for the optimal methodologies and the results are shown in Table 6.4 below:

Methodology	**EER %**
End-to End Model	1.2%

Table 6.4: Results in terms of EER

6.3.3 Comparison with Previous Research

The work done in CUI domain is relatively far less. Any research directly related to this research could not be found, therefore, our continuous user identification results could not be compared directly to any research. However, researchers in Motlagh (2015) proposed a CUI system based on keystroke dynamics. But their CUI system was supposed to work after the user was locked out by continuous authentication system (CUA) using the same keystroke dynamics samples. Two distance based classifiers named as euclidean and manhattan distance methods were used. The best results achieved were accuracy rate of 60% after 50 actions and 72% after 1000 actions to correctly identify the user which is locked out by system. In comparison, the CUI system implemented in this research does not depend on CUA and the accuracy rate of user identification achieved in this research is 93% on each action which is quite higher than the previous research results of 72% accuracy rate on 1000 actions.

6.4 Summary

This chapter investigated a novel approach of continuous user identification (CUI) which does not require the static user authentication with the help of usernames and passwords and tends to predict the user identity in minimal actions performed on system. The methodologies given in this chapter can be applied to any behavioural biometric modality.

Chapter 7

Continuous User Authentication using Mouse Dynamics with Baseline and Deep Learning Techniques

In this chapter, a continuous user authentication method has been proposed which can authenticate the user on each action by distinguishing the normal behaviour of user from abnormal actions. The proposed method uses the mouse dynamics, a behavioural biometric modality, which depicts the mouse usage behaviour of user. Moreover, this chapter implements the four different methodologies, based on baseline and deep neural networks, in order to improve the system performance. Each of the methodology incorporates the proposed recurrent confidence model which enables the system to validate user's identity based on each action. The system performance has been evaluated based on normalized portion of genuine and imposter actions where ideal system requires the former as high as possible by avoiding the false lockout while latter ought to be as low as possible by rapidly detecting the imposter user.

7.1 Introduction

Mouse dynamics is an interesting type of behavioural biometrics which can be used for user authentication purposes. Subsequently, mouse dynamics is an emerging research technique but it is still less explored behavioural biometric modality in terms of user authentication. Mouse dynamics recognition (MDR) includes extraction of mouse movement features and mining them to build unique signature profiles which can be eventually used to differentiate one user from others Pilankar and Padiya (2016). MDR method has two main advantages: Firstly, it does not require usage of any special hardware device to collect data hence it is an inexpensive approach. Secondly, it can be used to continuously monitor the identity of user based on each action, which is referred to as continuous user authentication (CUA) Salman and Hameed (2018).

For critical security systems, a continuous monitoring system is a requisite which can authenticate the user on each and every action performed on system. In this chapter, a true CUA system based on mouse dynamics has been proposed and implemented which tends to authenticate user on each mouse action by employing the proposed robust recurrent confidence model (R-RCM). The proposed R-RCM model uses a novel approach of detecting and locking out of imposter user once it crosses the alert threshold. Moreover, two types of system models have been proposed based on machine learning or baseline approach and deep learning techniques for CUA using mouse dynamics in detail. The recurrent neural network (RNN) is employed which to the best of our knowledge has not yet been studied for CUA using mouse dynamics.

7.2 System Methodology

In this chapter, two different techniques have been implemented consisting of baseline and deep learning methods.

The mouse dynamics dataset recorded by University of Buffalo Sun et al. (2016) has been studied for this work. The statistical properties of dataset are shown in chapter 3.

For proposed methods, the dataset for each user is partitioned into three non-overlapping sets named training set (T), testing set (X), and validation set (V) precisely. Training and testing sets are used to build the classifier reference model and parametric adjustments respectively whereas validation set is used on action by action basis for final evaluation of reference models determining the genuineness of user. The split strategy based on session spilt as discussed in chapter 3 section 3.2.1.1 has been used where the first two sessions are used for training while session three is used for validation.

7.2.1 CUA Featuring Mouse Dynamics using Baseline Approach

A novel baseline authentication method has been proposed which verifies the user on each mouse action as compared to histogram based approach that deals with accumulation of multiple activities before the accurate decision, regarding the identity of user, can be made.

The proposed baseline approach consists of four main phases i.e., feature extraction, classification, recurrent confidence model and decision module as given below:

7.2.1.1 Feature Processing, Baseline Approach

Mouse dynamics is generally considered as a series of mouse events acquired from input device for a specific user for the duration of his interaction with any respective graphical user interface. In order to understand mouse usage behaviour of any user, mouse events should be identified from raw data stream. These events are deemed to be timely system's communication, regarding the current mouse cursor position and mouse clicks, to specified application which is designed for collecting the data. Generally, gathered data is a list of different mouse events e.g., mouse move, mouse button pressed and released as shown in Table 7.1. Mouse usage is assumed to be a chronological series of mouse move or click events. More formally, a mouse data is considered to be a time series consisting of sequential ordering of a set of events (E) that occur within a specified interval of time. Each

Event	Description
Mouse Down (D)	This event relates to the press of left or right mouse button by user.
Mouse Up (U)	This event relates to the release of left or right mouse button by user.
Mouse Move (M)	This event relates to the mouse movement by user.
Mouse Wheel(W)	This event relates to the movement of mouse wheel, if mouse has wheel.

Table 7.1: Mouse Dynamics Raw Data Events

event $e \in E$ has 1 target, 2 attributes and 3 basic features as defined below:

- $UserId(e)$ – id of the user that has performed an action.
- $SessionId(e)$ – id of session that event belongs to.
- $TaskId(e)$ – id of the task that user has been assigned.
- $Timestamp(e)$– – an absolute time (milliseconds) when action was performed.
- $MouseCoordinate(e)$–is expressed in pixels as a pair (x, y).
- $ActionType(e)$– given type of mouse action such as:
  - Mouse Move
  - Left Click
  - Right Click

In order to make a true CUA system which tends to authenticate the user on each mouse action, the features based on single mouse event have been extracted to make a feature vector for each user. The mouse events are listed in Table 7.1

and the features extracted from those events are given in Table 7.2. The features listed in Table 7.2 are extracted from single mouse events where:

- X_{i-1} = *X-Coordinate* of given action on starting point
- X_i = *X-Coordinate* of given action on ending point
- $Y_{i-1=}$ = *Y-Coordinate* of given action on starting point
- $Y_{i=}$ = *Y-Coordinate* of given action on ending point
- $TimeDelay$= Total duration to complete the action

The explanation of the extracted statistical mouse features Ahmed and Traore (2007) as shown in Table 7.2 is given below:

1. **Angle of mouse move**: Based on the x and y actions, angle θ is calculated which is the angle of path tangent with the relevant x and y axis.
2. **Direction of movement**: Direction feature is the direction of the end to end line. To reduce the possible direction values, the 8 main directions are used as defined by Ahmed and Traore (2007) (see Fig 7.1).
3. **Shift in X-coordinate**: It is the travelled distance in abscissa direction.
4. **Time delay:** The time interval between starting point and ending point of mouse movements.
5. **Travelled distance:** The distance between two adjacent positions of mouse click actions.
6. **Curve length**: It is defined as the total distance travelled in given one sequence of mouse event.
7. **Ratio of total length and total distance**: It is the ratio of curve length to distance travelled in one sequence of actions.
8. **Curve Speed**: It is the ratio of distance travelled between two adjacent points to the time taken for this distance.

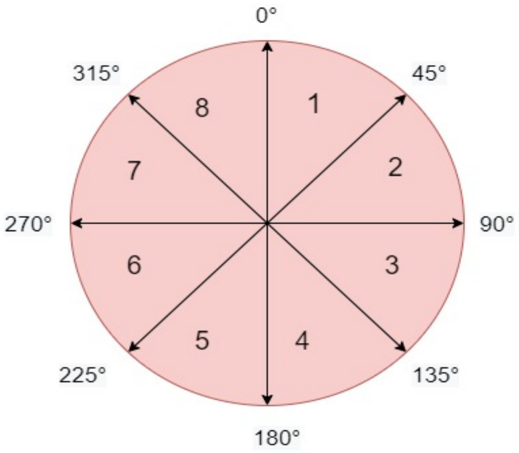


Figure 7.1: Direction of Mouse Move

It can be noticed that angle of mouse move, its direction (Fig.7.1), distance travelled, mouse curve length and speed features are extracted based on mouse events from raw data. Mean and standard deviation (σ) of these features are used to built a reference feature template for each user. Moreover, Fig.7.2 illustrates the cumulative distribution function (CDF) of different features for 2 different users. It can be observed that the investigated features are distinct for each user hence can be utilised to differentiate one user from the other.

CUA Feature	Description
Angle of mouse move	$\theta_i = atan\frac{(y_i - y_{i-1})}{(x_i - x_{i-1})}$
Direction	Shown in Fig.7.1
Shift in X coordinate	$ShiftX = x_i - x_{i-1}$
Time delay	$delay_i = timestamp_i - timestamp_{i-1}$
Distance	$dist_i = \sqrt((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2)$
Curve Length	$Curvelength_i = \sum_{k=1}^{i} dist_k$
Ratio of total length and total distance	$r = \frac{(Curvelength_i)}{(dist_i)}$
Curve Speed	$CurveSpeed_i = \frac{(dist_i)}{(timedelay_i)}$

Table 7.2: Mouse Dynamics Extracted Features for Baseline Approach

7.2.1.2 BASELINE CLASSIFIERS

Conventional pattern recognition system had mostly used a single classifier for classification purposes. However, it has been identified in recent researches Liang et al. (2014) that most samples which were incorrectly classified by some classifiers were not the same when experimented with other discrete classifiers. Therefore, fusing classification decisions from multiple discrete or complimentary classifiers can escalate the classification accuracy and system robustness as compared to having a single classifier. Moreover, mouse dynamics is considered to be a weak biometric modality since it depends on behaviour of user which can alter with time, background context or different hardware configurations. Hence, it is considered to use multiple classifiers approach consisting of Support Vector Machine

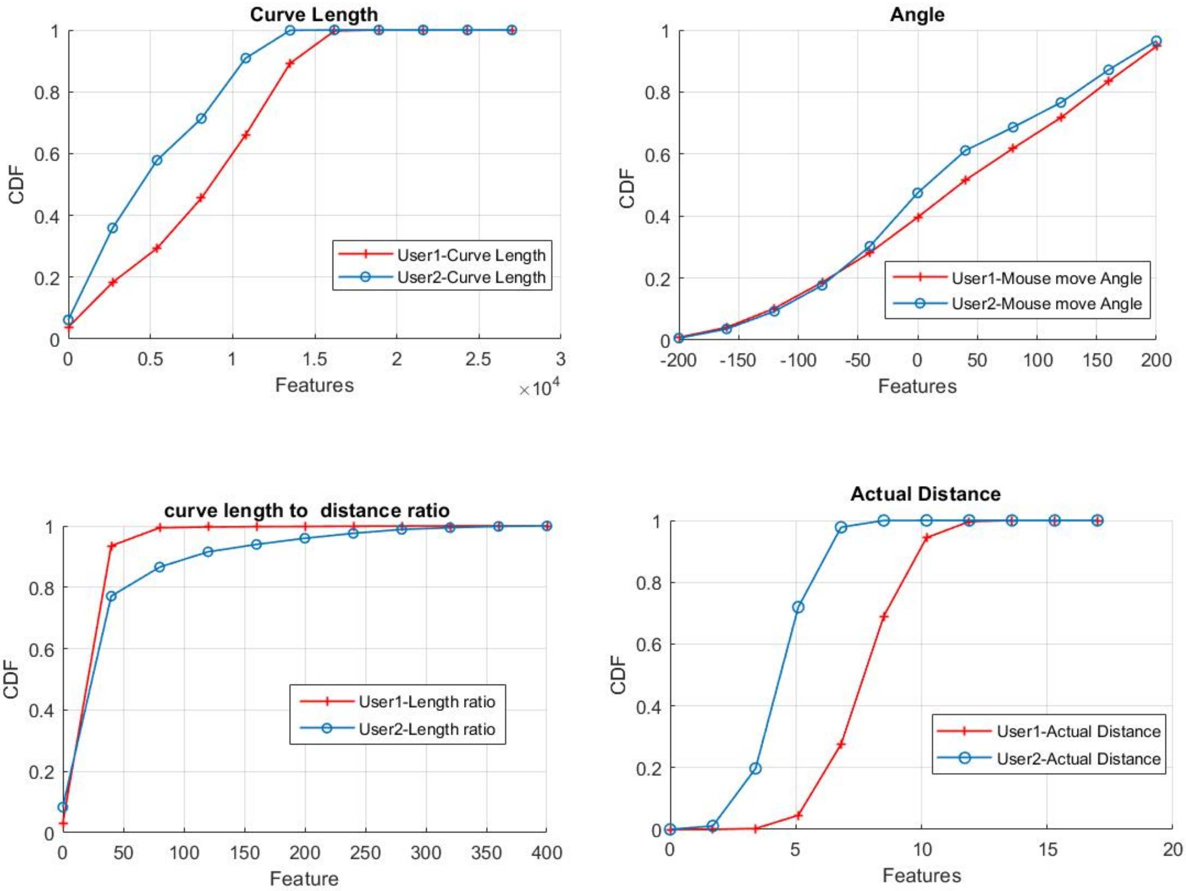


Figure 7.2: Cumulative Distribution Function (CDF) of Mouse dynamics features for User 1 & User 2

(SVM) and Decision Trees (DT).

i. Support Vector Machine(SVM)

Support vector machine (SVM) is a supervised learning model Hsu et al. (2003) which is usually used for classification and regression problems as explained in Chapter 4, Section 4.2.3.1. In the mouse dynamics frameworks, LibSVM has been used to implement SVM classifier with linear kernel.

ii. Decision Trees (DT)

Decision Trees (DTs) are known to be non-parametric supervised learning technique used for classification. Some if-then else decision rules are inferred from the data features and model is trained to predict target values based on these rules Song and Ying (2015).

The proposed system uses the weighted classifier fusion(WCF) Mi et al. (2016) classifier ensemble rule for fusing the score of both classifiers. The WCF refers to method of sending the scores of both classifiers as an input into the weighted fusion module which produces an output score consisting of weighted sum of input scores of both classifiers as shown in Eq: 7.1

$$\hat{y}_t(c_t|W) = \frac{\sum_{i=0}^{K-1} W_i c_{ti}}{\sum_{i=0}^{K-1} W_i}, \tag{7.1}$$

where c_{ti} – input scores, K – amount of classifiers, W_i – input score weights and the value of these weights have been optimised with genetic algorithm, $\hat{y}_t(c_t|W)$ – fused score which will be used as a raw confidence score in the second phase for each action.

7.2.1.3 Robust Recurrent Confidence Model(R-RCM)

The preceding works presented in continuous authentication systems mostly used fixed window based approach consisting of block of actions. Such systems prespecify the number of actions in each block and authentication decision can only be made after the block has been filled to its maximum value. Hence, the legitimacy of current user can only be decided once he or she has performed the specified number of actions i.e., 100, 500 or 1000 actions. However, this method can provide opportunity to imposter user to perform illicit activities on system or steal some confidential information in meantime. To overcome this issue, a robust recurrent confidence model (R-RCM) has been proposed and explained in Chapter 3 section 3.2.3.2 which tends to authenticate the user on each action. In this chapter, two types of RCM are used and the performance is assessed for both types. The two experimental settings for RCM are given below:

- ***Experimental Setting I: Simple RCM:*** RCM calculates the confidence of user on each action based on the classifier score of performed action and few other parameters. During the user's active session, if confidence goes lower than final threshold then user will be locked out by the system.
- ***Experimental Setting II: R-RCM with alert threshold:*** Two types of thresholds have been used named as final threshold and alert threshold. if the resultant confidence drops down the alert threshold during the user's activity but it is still above the final threshold, then R-RCM will work robustly in its hard mode in order to detect the imposter user as soon as possible. In this case, user can still continue the work because the confidence value is still higher than final threshold.

7.2.2 Mouse Dynamics CUA using Deep learning

Biometrics can be evidently divided into two categories named physiological and behavioural methods. In this aspect, biometric characteristics belonging to physiological category are less likely to alter over time as compared to behavioural features. Since the behavioural biometrics mostly depict the regular user behaviour while interacting with the relevant device, therefore these characteristics mostly rely on the hardware specification of devices, background context and user's emotion or age. Mouse dynamics, being a behavioural trait, tends to change gradually with time or based on configuration of different hardware mice. Therefore, maintaining a static database of users populated with statistical features could affect and decrease the performance or accuracy of system over time.

Moreover, mouse dynamics data is more like a sequential series containing some hidden properties as well. For example, it can be general mouse usage of a user that when he wants to open a file document on system he always used to double click the mouse left button. On the other hand, there can be another user who has the habit of firstly clicking the right button of mouse to go to the option of properties and then afterwards choosing the OPEN option from the dialogue box. This sort of hidden features or combination of hidden features can be used to differentiate users from each other.

Subsequently, the traditional classification algorithms cannot mine this kind of hidden features and these cannot be stored into statistical feature profiles. For that reason, the proposed system has implemented the recurrent neural network (RNN) which tends to effectively learns and mines the chronological data in order to build the dynamic user profiles. Furthermore, in comparison to previous research works which applied the periodic authentication approach as discussed in Chapter 2, this work adopted the real CUA system approach which authenticates the user on each action by integrating the proposed robust recurrent confidence model (R-RCM) with RNN.

7.2.2.1 Problem Formulation

Let's assume, there are total U users and their identity needs to be verified on each mouse action performed on system such as:

$S = \{(m, n)\} \subset \mathbb{R}^{Z \times A} \times \{1, \ldots, U\}^A,$

where m_t – Mouse action properties at a time t, $n_t \in \{1, \ldots, U\}$ – user who has taken the action, A – total amount of actions to classify, Z – action vector dimension.

7.2.2.2 Sequence Sampling

Given a tuple $(UserId', SessionId', TaskId',)$ mouse events are grouped into sequences:

$$
\begin{aligned}
Sequence(UserId', SessionId', TaskId', Timestamp,&\\
MouseCoordinate) = \{e|\forall e \in E, s.t.&\\
UserId(e) = UserId'&\\
\text{and } SessionId(e) = SessionId'&\\
\text{and } TaskId(e) = TaskId'&\\
\text{and } Timestamp(e) = Absolutetime'&\\
\text{and } MouseCoordinate(e) = MouseCoordinate(x,y)'\}&
\end{aligned}
$$

7.2.2.3 Bidirectional Long Short Term Memory (BiLSTM)

The more refined form of RNNs known as long short-term memory (LSTM) network has been used in this work in order to tackle the problem of diminishing gradients as discussed in Chapter 5 Section 5.2.2.3.

In this chapter, the more advanced type of LSTM named as bidirectional LSTM network is experimented. A Bidirectional LSTM, or BiLSTM, is a sequential model which comprises of two LSTMs: one taking the input in a forward direction, and the other in a backward direction hence provides additional training of data by processing the input data twice i.e., left to right and right to left. The previous scholarly results have shown that BiLSTM network outperforms the simple LSTM owing to additional training of data Siami-Namini et al. (2019). BiLSTM model has been integrated with proposed R-RCM to formulate a true CUA system and the system architecture has been illustrated in Fig.7.3.

7.2.2.4 Hybrid Bidirectional Long Short Term Memory (LSTM R-RCM)

The designed LSTM network architecture includes one BiLSTM layer, two fully connected dense layers, activation layer and output classification layer. The formulated model segregates the mouse dynamics raw data into fixed length mouse sequences as well as single frames and produces the input sequence consisting of mouse X coordinate, Y coordinate and timestamp for performed action.

The architecture of implemented BiLSTM is given Table 7.3 below:

Layers	**Explanation**
Input Layer	3-dimensional
BiLSTM Layer	256 hidden units
2-fully connected dense layers	256 units
activation (Activation)	90 units
dropout	to avoid over fitting

Table 7.3: BiLSTM Network Structure

It should be worth noting that the optimal approach of integrated per frame and per sequence as presented in Chapter 5 section 5.2.2.5 is utilised in both settings i.e., simple RCM and RCM with alert threshold. Subsequently, the processed input, either being per frame or per sequence, is fed into the bidirectional LSTM layers which is further processed by fully connected dense layers to produce the final classification score for given input sequence. Afterwards, the final classification output from LSTM structure based on per frame and per sequence is fed into our proposed R-RCM model as an input and it applies hyper-parameters to decide if user can continue using the system or not based on final threshold as described in Algorithm 2.

7.2.3 Performance Measure

The performance measure of implemented CUA system has been evaluated based on ANGA, ANIA and EER as discussed in chapter 3 section 3.2.6

Moreover the four categories and evaluation threat scenario 3 (ETS3) as discussed in chapter 3 3.2.5 and section 3.2.6.1 is utilised respectively to assess the system performance.

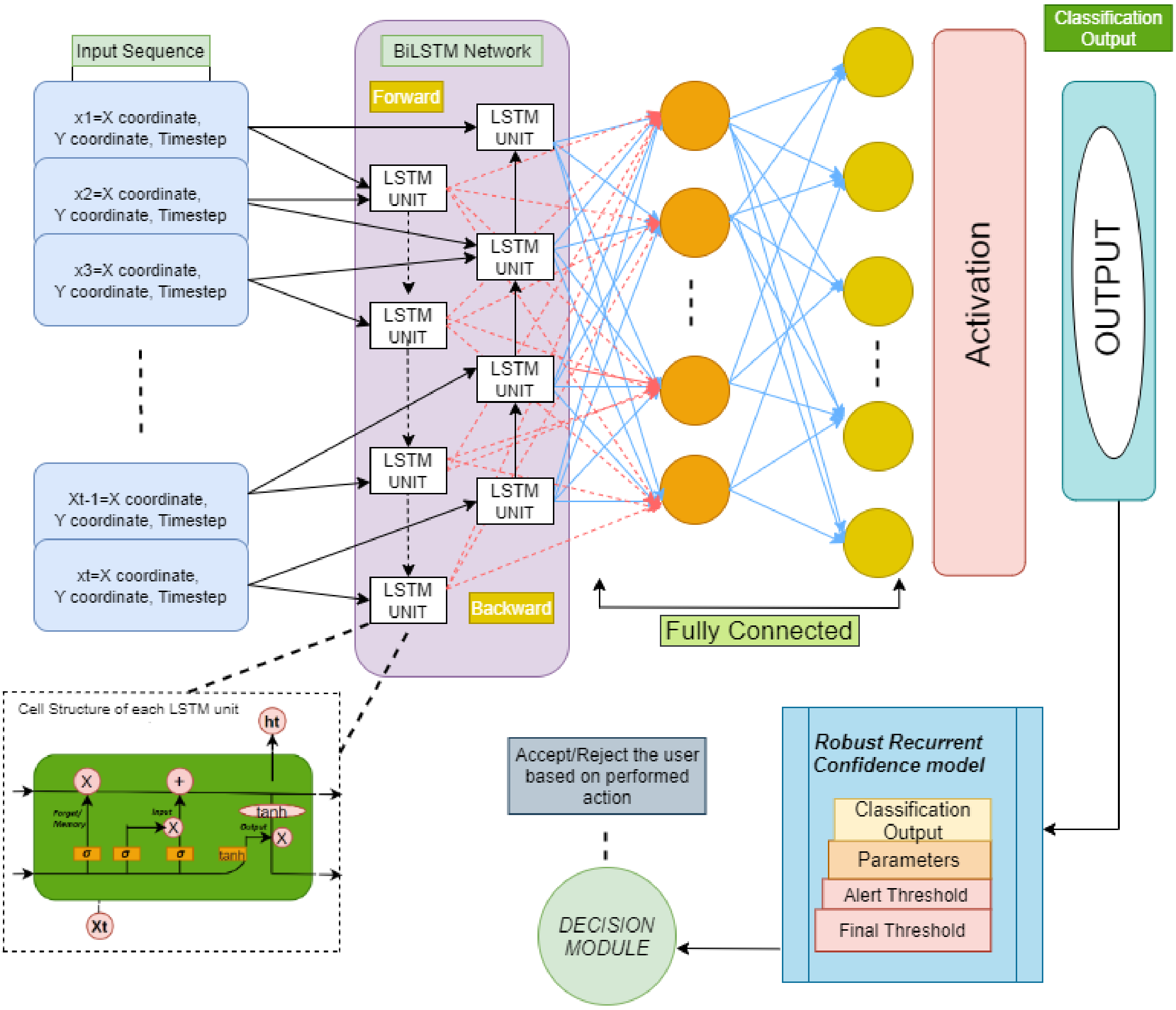

Input Sequence
x1=X coordinate, Y coordinate, Timestep
x2=X coordinate, Y coordinate, Timestep
x3=X coordinate, Y coordinate, Timestep
Xt-1=X coordinate, Y coordinate, Timestep
xt=X coordinate, Y coordinate, Timestep
BiLSTM Network
Forward
Backward
LSTM UNIT
Fully Connected
Activation
Classification Output
OUTPUT
Cell Structure of each LSTM unit
ht
X
+
Forget/ Memory
Input
Output
tanh
σ
Xt
Accept/Reject the user based on performed action
DECISION MODULE
Robust Recurrent Confidence model
Classification Output
Parameters
Alert Threshold
Final Threshold

Figure 7.3: Framework of Hybrid BiLSTM-R-RCM network

Algorithm 2 Framework of Hybrid BiLSTM-R-RCM network

```
—>Inputs: Mouse Dynamics raw data for each action: (X coordinate, Y coordinate, Timestep t) Mouse Dynamics raw data for each sequence: (X1 co-ordinate, Y1 coordinate, Timestep t1,X2 coordinate, Y2 coordinate, Timestep t2...., )
—>Outputs: Probability of user genuineness and user confidence
    Phase 1 – BiLSTM Model Training
Initialisation
 Split data into: Session [0,1] = 70% Training and 30% Testing data, Session [2] = Validation
—> Implement BiLSTM model to training data Procedure Bilstm(train, epoch, layers, option)
X ← train
Y ← train - X
[lstmodel]<—Sequential-Model([ sequenceInputLayer() bilstmLayer() fully-ConnectedLayer() softmaxLayer classificationLayer])
Loss <– crossentropy, optimiser <– Adam, MaxEpochs <– 60.
lstmodel.compile(LOSS, optimiser)
lstmodel.train(train, epoch, layers, option)
return model
    Phase 2 – Hybrid BiLSTM with R-RCM
Static Authentication, Confidence set to 1.00(Max)
Implement For loop which continues until user uses system
for <each $action \in \forall(Valset)$> do
    Find probability of per action and per sequence calculated by BiLSTM
    Send probability results P to R-RCM as input
    Apply Hyperparameters onto action and sequence probability result

    if $currentconfidence \geq AlertThreshold$ then
        Calculate new confidence

    else if ($currentconfidence < AlertThreshold$) and ($probabilityP < H$) then
        Calculate new confidence and user loses double confidence point than usual

    else if ($currentconfidence < AlertThreshold$) and ($probabilityresultP > H$) then
        calculate new confidence but only grant half of confidence point than usual
    end if
return confidence(c)

end for
```

7.3 Results and Discussion

The results attained from the experiments are discussed in this section. Firstly, few sample results have been presented for 512 action events in order to visualise the results for each category. Afterwards, the detailed results are given in tabular form which have been obtained from whole validation data split as discussed in section III.

- *GOOD*: A validation sample of a genuine user's has been shown in Fig.7.4. It shows the visualisation of two cases:
    - Genuine user training sample has been validated with its own validation set (right side)
    - Genuine user training sample has been validated with an imposter user's validation set (left side)

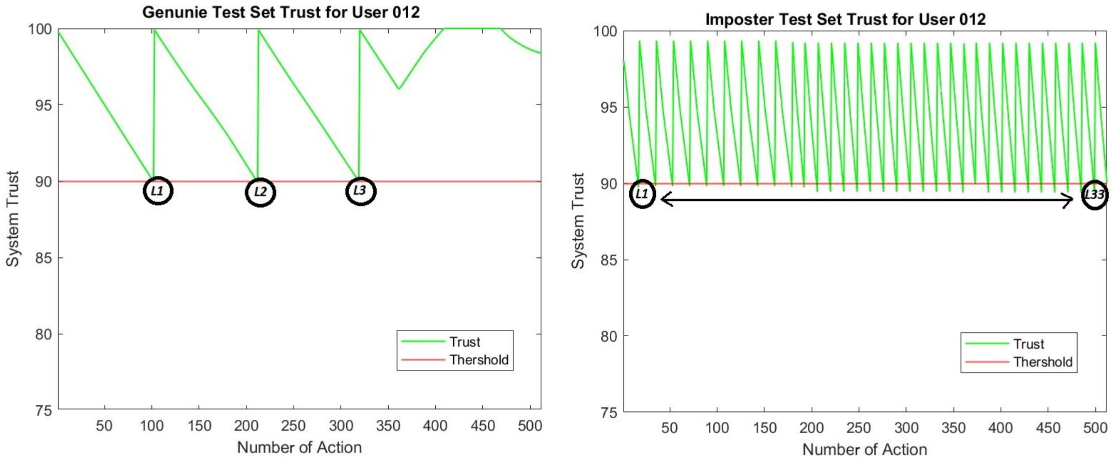


Figure 7.4: Genuine user validated with its own reference set(left) and with imposter set(right)

In both cases, it can be observed that genuine and imposter users have been locked out of the system so the ANGA & ANIA can be calculated using the equation 3.2 and 3.1 respectively.

$ANGA = \frac{320}{3*512} = 0.20$ or 20% so, $ANGA < 100\%$

Similarly, ANIA can be calculated: using Eq. 3.1

$ANIA = \frac{512}{33*512} = 0.03$ or 3% so, $ANIA > 40\%$

In this case, genuine user has been falsely locked out 3 times in given action sequence i.e., 512 mouse events, hence the normalized ANGA are less than 100%. Moreover, the imposter user has been detected before performing 40% of actions again the training sample of given genuine user, hence this user falls in good category. At this point, it should be noted that only 1 imposter user is shown against the given genuine user in Fig.7.4 in order to understand the user categorisation. However in practice, each genuine user has been validated with all the imposters user and then Mean ANIA relating to each imposter user is calculated to decide the user category for detailed results.

- $VeryGood$:

  Another validation sample has been shown in Fig.7.5 indicating that genuine user has never been falsely locked out of system so the ANGA=100%. However, the Fig.7.5 (left side) specifies that imposter user has been detected 42 times in the given action sequence hence the ANIA, according to Eq. 3.1, is 0.02 or 2.0% , so it can be concluded that ANIA < 40%. Hence this user falls in Very Good category in this example.

- *BAD*: Fig.7.6 illustrates another validation sample which shows that genuine user is not locked out even once (right side) hence ANGA = 100% for the given action sequence. However, it indicates that imposter user has been locked out 2 times (left side) and ANIA for this case is 0.5 or 50% according to equation 3.1. Hence, this user falls in Bad category because ANGA = 100% and ANIA > 40% i.e., genuine user has not been locked

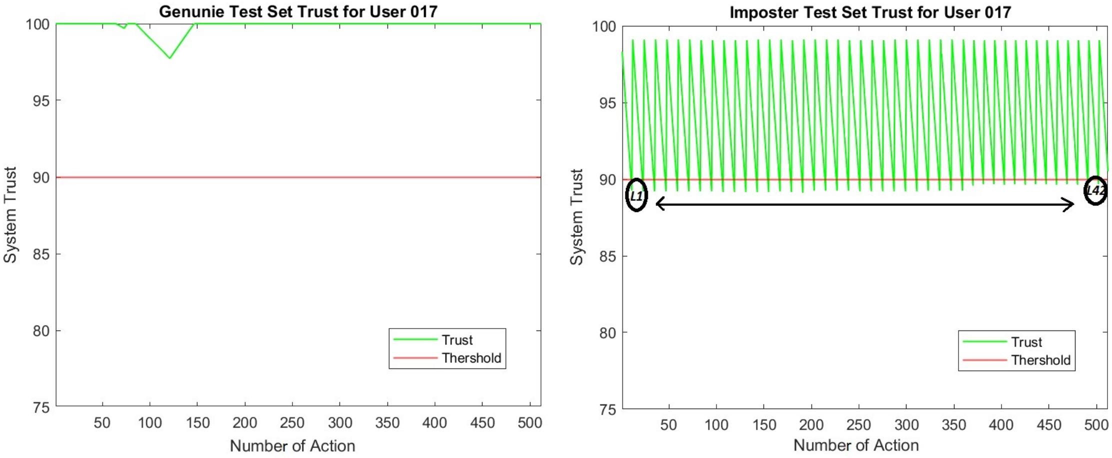


Figure 7.5: Genuine user validated with its own reference set(left) and with imposter set(right)

out even once but imposter user could not be caught even after performing 50% of actions on system.

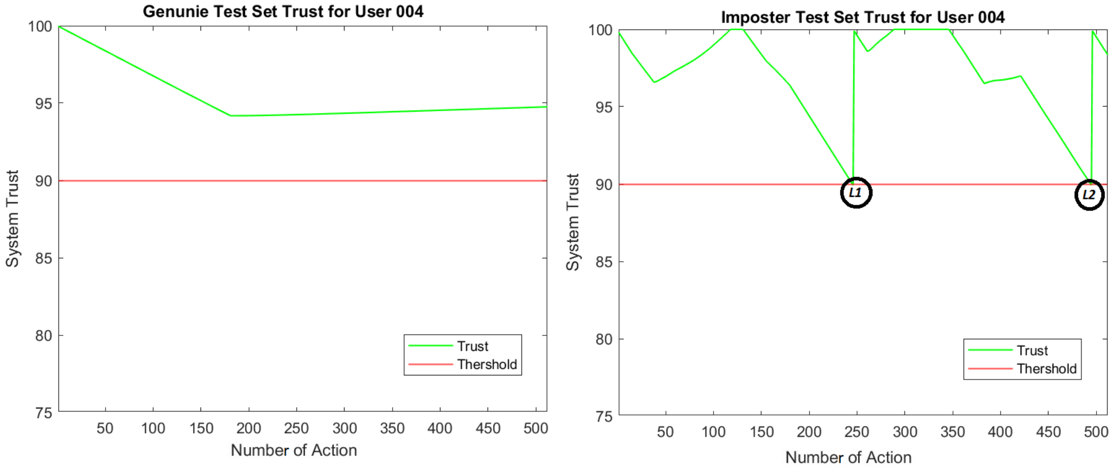


Figure 7.6: Genuine user validated with its own reference set(left) and with imposter set(right)

- *UGLY*: Fig.7.7 indicates the genuine user has been locked out in a given action sequence hence ANGA $<$ 100%. On the other hand, the imposter user has not been locked out even after performing 43% of actions, according to Eq. 3.1. Therefore, ANGA $<$ 100% and ANIA $>$ 40%.

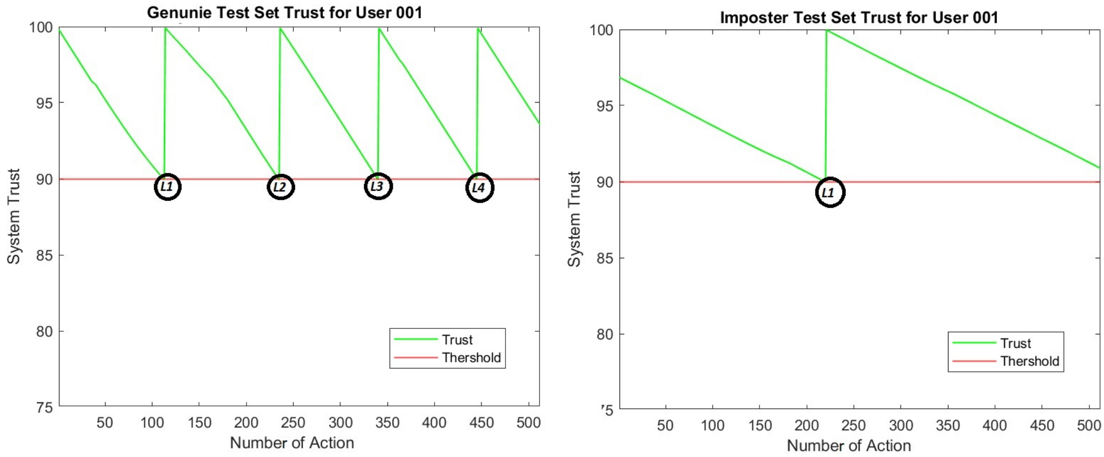


Figure 7.7: Genuine user validated with its own reference set(left) and with imposter set(right)

7.3.1 Aggregated Results for Baseline methods

Baseline method has covered two types of experimental settings:

- Baseline with Simple RCM
- Baseline with Robust RCM (R-RCM)

The detailed collective results for simple and robust baseline are reported below in table 7.4 and 7.5 respectively, while the graphical representation is shown in Fig 7.8 and Fig 7.9 respectively.

It can be observed in Table table 7.4 that 56 users are falling in very good category with mean ANGA being 100% (referred as 1 in this work) and mean ANIA are 0.16 which means all the imposters who have been tested against these 56 genuine users are locked out of system after performing 16% of actions out of total testing data as shown in Fig 7.8. Moreover, 7 users fall in good category with ANGA = 0.20 (20%) which means these genuine users have been falsely locked out of system after performing only 20% of actions on average. Furthermore, 9 and 3 users fall in bad and ugly categories with mean ANIA recorded as 42% and 41% which indicates that imposter users against these genuine users (9 + 3 = 12) could not be caught up by system even after performing 42% of actions. The system's

ANIA and ANGA are calculated by using the equation 5 and 6 respectively. The system's ANGA and ANIA are 0.88 and 0.20 respectively.

Category	Users	% Mean ANGA	% Mean ANIA
Very Good	56	1	0.16
Good	7	0.20	0.15
Bad	9	1	0.42
Ugly	3	0.07	0.41
System Total	**75**	**0.88**	**0.20**

Table 7.4: Aggregated Results of Simple Baseline

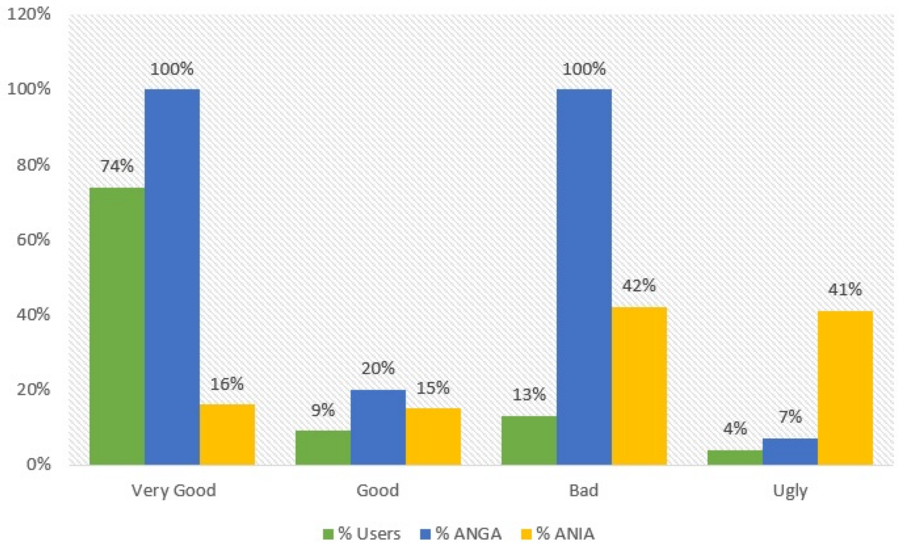


Figure 7.8: Simple Baseline results represented in percentage

On the other hand, table 7.5 shows that 65 users fall in very-good category with ANIA 0.07, while 8 users belong to good category with ANGA=0.41 & ANIA=0.27, there is no user in bad category, however, 2 users fall in ugly category with ANGA= 0.39 & ANIA=0.51. The system's ANGA and ANIA are 0.92 and 0.10 respectively.

Overall, the results for both settings of baseline indicate that:

- System's ANGA and ANIA have improved with the proposed robust baseline methodology. More specifically, it can be assumed that robust baseline

Category	Users	% Mean ANGA	% Mean ANIA
Very Good	65	1	0.07
Good	8	0.41	0.27
Bad	0		
Ugly	2	0.39	0.51
System Total	**75**	**0.92**	**0.10**

Table 7.5: Aggregated Results of Robust Baseline

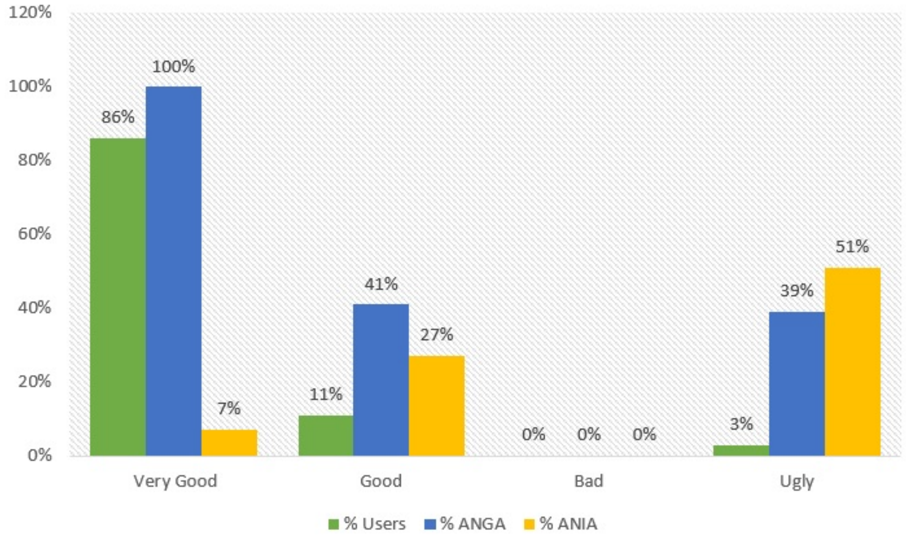


Figure 7.9: Robust Baseline results represented in percentage

methodology with alert threshold has performed well in order to detect imposters users faster and in less number of action as compared to simple baseline method.

- However, it is worth noticing that system ANGA has also improved with few points i.e., from 0.88 to 0.92 which indicates that robust baseline has reduced the false lock out of genuine users. But ANIA has improved more drastically i.e., from 0.20 to 0.10 in comparison to system ANGA.

- Moreover, there are less number of users in robust baseline who fall in bad and ugly categories (0 + 2= 2) as compared to simple baseline users who belong to bad and ugly categories (9 + 3= 12) which shows that simple baseline could not caught some of the users before performing 40%

of actions.

7.3.2 Aggregated Results for Deep Neural Network methods

Deep Neural network (LSTM) has been tested with two types of experimental settings:

- Integrated LSTM with Simple RCM
- Integrated LSTM with Robust RCM (R-RCM)

The elaborated collective results for simple and robust deep neural networks are reported below in table 7.6 and 7.7 respectively, while the graphical representation is shown in Fig 7.10 and Fig 7.11 respectively.
It can be noticed in Table 7.6 that 72 users are falling in very-good category with mean ANIA 0.02 (2%) also shown in Fig 7.10, 3 users belong to good category with ANGA = 0.62 & ANIA = 0.04 and there is no user falling into bad and ugly category. The system's ANGA and ANIA are 0.98 and 0.02 respectively.

Category	**Users**	**% Mean ANGA**	**% Mean ANIA**
Very Good	72	1	0.02
Good	3	0.62	0.04
Bad	0		
Ugly	0		
System Total	**75**	**0.98**	**0.02**

Table 7.6: Aggregated Results of Integrated Hybrid Simple LSTM-RCM

On the other hand, Table 7.7 shows that 73 users fall in very-good category with ANIA 0.008, while 2 users belong to good category with ANGA = 0.80 & ANIA = 0.02 and there is no user falling into bad and ugly category. The system's ANGA and ANIA are 0.99 and 0.008 respectively.

Overall, the results for both settings of Hybrid LSTM R-RCM indicate that:

- System's ANGA and ANIA have improved with the proposed Integrated Hybrid LSTM R-RCM methodology. More specifically, it can be assumed

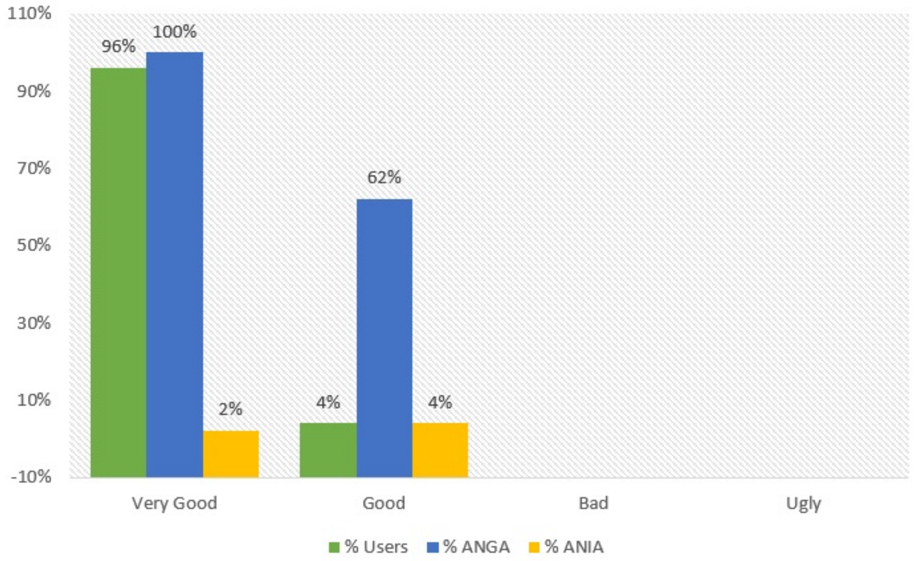


Figure 7.10: Integrated Hybrid Simple LSTM results represented in percentage

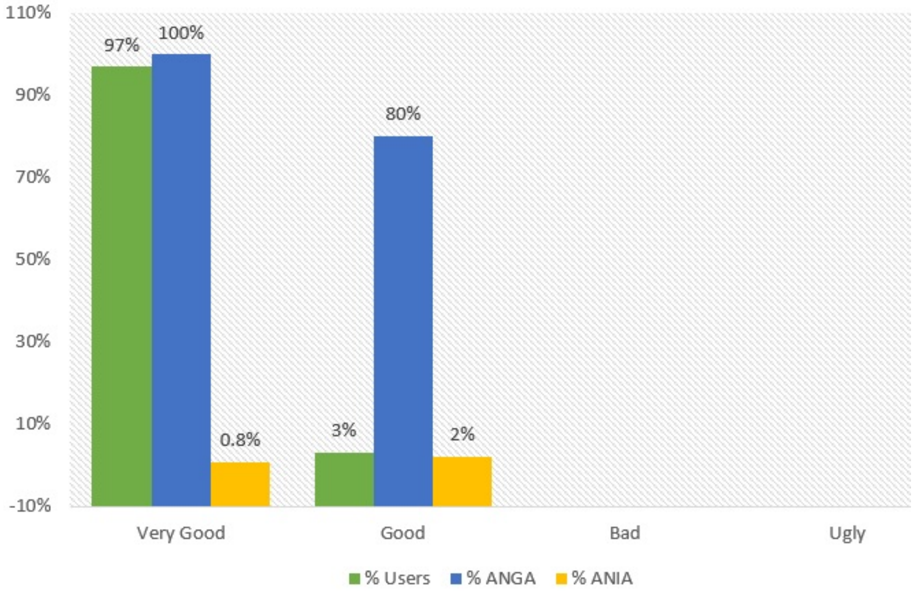


Figure 7.11: Robust LSTM results represented in percentage

Category	Users	% Mean ANGA	% Mean ANIA
Very Good	73	1	0.008
Good	2	0.80	0.02
Bad	0		
Ugly	0		
System Total	**75**	**0.99**	**0.008**

Table 7.7: Aggregated Results of Integrated Hybrid Robust LSTM-RCM (R-RCM)

that robust baseline methodology with alert threshold has performed well in order to detect imposters users faster and in less number of action as compared to simple LSTM method.

7.3.2.1 Results Analysis in terms of Equal Error Rate

Equal error rate (EER) has also been calculated to evaluate the results with previous research works by using the Eq 3.6. In this work, EER has been calculated for all the four methodologies and the results are shown in Table 7.8 below:

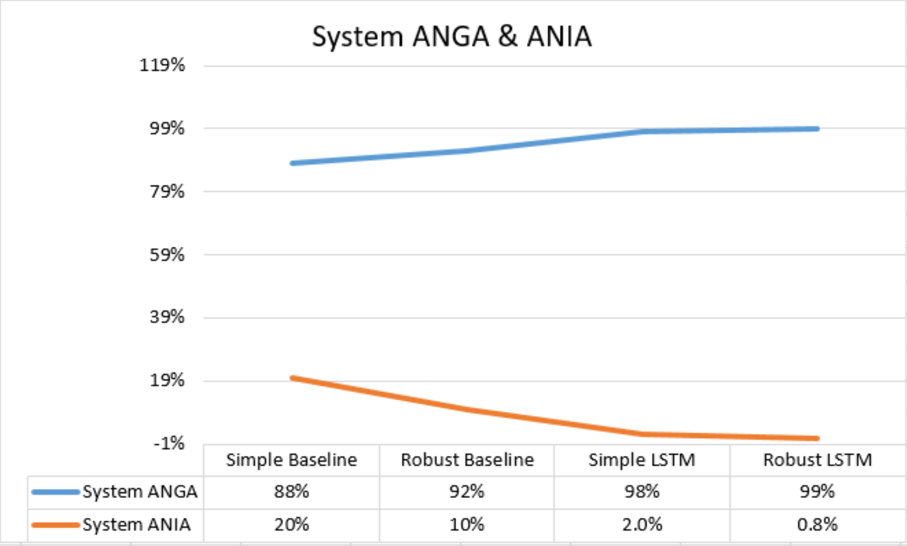
Category	EER
Baseline with simple RCM	9.65%
Baseline with Robust R-RCM	5.3%
Integrated Hybrid LSTM with simple RCM	2.1%
Integrated Hybrid LSTM with Robust R-RCM	1.3%

Table 7.8: EER Rate for Proposed Four Methods

It can be observed in Table 7.8 that the integrated hybrid LSTM with robust R-RCM has achieved the lowest EER, hence this can be considered as our optimal experimental setting. If the optimal setting results of this research are compared with previous scholarly work given in chapter 2 table 2.3 , then researchers in Gamboa and Fred (2004*b*) had achieved the EER of 0.2% with 200 mouse actions whereas they had reported 48% EER with 1 mouse action. Since our work has considered 1 mouse action for authentication then it can be said that our optimal experimental setting has achieved lowest EER when compared with previous research works done as given in chapter 2 Table 2.3.

7.3.2.2 Results Analysis in terms of Normalized ANGA and ANIA

Aggregated results of Robust Baseline and integrated Hybrid LSTM R-RCM in table 7.5 and 7.7 respectively are referred here. First of all, it can be noticed that for robust baseline there are 3% of users who fall into ugly category, however, for robust LSTM there are no users who belong to ugly category which means all the imposter users have been caught by system before performing 40% of actions. Hence, the deep learning approach has performed well in detection of imposter users more quickly as compared to baseline method.



	Simple Baseline	Robust Baseline	Simple LSTM	Robust LSTM
System ANGA	88%	92%	98%	99%
System ANIA	20%	10%	2.0%	0.8%

Figure 7.12: ANGA & ANIA percentage for all the four experiments

Fig 7.12 illustrates the System's ANGA and ANIA for our four proposed methodologies. it can be observed that System ANGA has substantially increased from simple baseline method to robust deep learning LSTM R-RCM which includes the two thresholds to authenticate user. Similarly, it can be noticed that system's ANIA has substantially decreased from simple baseline to robust deep learning model.

Overall, deep learning models have worked well in avoiding the false lockout and quick detection of imposter thereby escalating the ANGA and lessened the ANIA respectively. Since deep learning method does not depend on statistical features which can change over time so it can retain the previous information.

More specifically, if the ANGA and ANIA are demonstrated with exact number of actions instead of giving the normalized portion of actions then it can be calculated by using Eq 3.5 as follows:

$$Exactgenuine/imposteractions = \\ Normalizedactions * Totalvalidationdata$$

If the exact number of imposter actions for the optimal experimental setting i.e., integrated robust LSTM R-RCM model are calculated then Exact imposter actions are 72 which means the imposter users have been locked out of system only after performing 72 mouse actions.

Moreover, if optimal experimental setting results i.e., 72 imposter actions are compared with previous scholarly work, then researchers in Mondal and Bours (2017*b*) had also used ANGA & ANIA as performance metric for CUA system. The ANIA reported for their optimal settings was 252 mouse actions which is quite higher than the results achieved in this work (also shown in table 7.9).

Methodology	**Exact Imposter actions**
Previous Research Mondal and Bours (2017*b*)	252
Integrated LSTM-Robust RCM	72

Table 7.9: Exact Imposter actions for Mouse Dynamics

7.4 Summary

This chapter proposes a true continuous user authentication (CUA) method using the mouse dynamics. Moreover, four different types of system architectures have been formulated based on baseline and deep neural network techniques. Moreover, both methods i.e., baseline and deep neural network, uses the proposed RCM model with its two forms. Analysis of experimental results depicted that baseline and deep learning method when incorporated with robust RCM have performed well in detecting the imposter users quickly while avoiding the false lock out of genuine users. In this aspect, the experimental results also show that deep neural network approach based on integrated Hybrid LSTM-RRCM has achieved the lowest imposter actions and highest genuine actions before lock out and hence considered to be the optimal experimental setting for this work.

Chapter 8

Conclusions and Further Work

The preceding systems employed the analysis based on Period User Authentication (PUA) where system records the keystroke or mouse dynamics timings for fixed number of actions or fixed block size and then afterwards analyse the data to decide if it belongs to genuine user or not. These systems give room to imposter user to cause damage to confidential information and system resources. On the contrary, a true Continuous User Authentication (CUA) system inclines to verify the identity of user after each keystroke or mouse action.

8.1 Conclusions

Computer systems and networks are essential part of almost every aspect of human life. All the businesses, banking systems, government services, medical, aviation, communication, education and entertainment are mainly controlled by computer systems. Each organisation is effectively using computer systems to store important information and data including confidential financial transactions, employee records, personal and business emails and medical history. However, this escalating dependence on computers has excavate new computer security threats as well. Moreover, cybercrimes have also been escalated owing to the presence of imposter users who can masquerade the legitimate user in order to get access to system resources which can result into serious exploitation and obliteration of personal, governmental and commercial information. In order to preclude the imposters to steal those confidential information and files, one important factor is considered to be robust user authentication method.

Static user authentication (SUA) methods mainly consisting of usernames, passwords and PINs have been predominantly used for identification purposes in many computing systems. However, these methods are not directly connected to the genuine user hence any other individual can exploit these credentials, on behalf of legitimate user, to access the resources or confidential information for fallacious activities.

Moreover, these methods cannot verify the identity of user throughout the active session which can fosters security risks for system resources. Therefore, continuous monitoring of authorised user session is necessary to ensure that only legitimate user is accessing the system resources for entire session.

In this regard, the previous research in the domain of CUA had mostly focussed on strategy of periodic user authentication (PUA) which refers to the re-verifying of user identity on fixed block of actions i.e., 200, 1000, 2000 actions or fixed time period window. These methods possess security risk and gives chance to imposter user to cause damage to system.

Secondly, there are only few works done in domain of continuous user identification (CUI) where user's identity is established on each action without prior claim of any identity or without the involvement of SUA. These type of systems are important for forensic analysis.

Therefore, *this research aimed to address the above gap to design a true continuous authentication and identification system which can authenticate or identify the user on each single action or activity performed on computer system.* The ultimate goal was to investigate and implement a true CUA and CUI system which can work passively without disturbing the user while he/she is performing important tasks on system.

To achieve the above aim the following objectives were set:

1. **Critically examined the constraints and drawbacks of existing CUA systems**

   First objective was to critically examine the constraints and drawbacks of existing CUA systems and this research found that existing CUA systems are employing the approach of PUA system which is user verification based on fixed block of actions. Moreover, existing techniques mostly utilised the static database for reference features of user particularly populated with mean and standard deviation of different keys and key-pairs or mouse activities. Existing systems ignored the fact that behavioural biometrics tends to change with time, age or external factors hence maintaining the static feature database can lessens the system performance over time.

2. **Presented a true CUA based on a proposed recurrent confidence module**

   The second objective was to present a true CUA employing a proposed recurrent confidence module authenticating the user on each action. A robust recurrent confidence model (R-RCM) is proposed and implemented which tends to authenticate user on each and single activity by allocating a confidence value for user genuineness. However, it does not lock out the user on one action but it also keeps track of confidence values on previous actions as well. R-RCM also used the novel approach of two thresholds i.e., alert threshold and final threshold to provide more security and reliability to system. The perception of alert threshold is employed according to which if confidence of user is constantly going down and eventually reaches the alert threshold then system doubts the legitimacy of current user. In this case, if the probability of current new action shows that it belongs to imposter user. Then, user loses double confidence points on such actions thereby making it locked out of system quicker than usual in order to limit the damage caused by it. However, it is also known that sometimes genuine user can also deviate from normal typing behaviour owing to changing external factors and can reach the alert threshold. In this case, if probability of new action shows that it belongs to genuine user still system does not trust user fully and grant it confidence points less than usual.

3. **Implement a true continuous user authentication using keystroke dynamics with baseline approach**

The third objective was to analyse and implement a true continuous user authentication using keystroke dynamics with baseline approach. In this research, keystroke dynamics is used as a behavioural biometric modality to continuously authenticate the user on each key press and key release actions. Baseline or machine learning classification methods are used to find the probability of each keystroke. Baseline classification techniques are integrated with the proposed R-RCM model to formulate a true continuous authentication framework. Additionally, two types of dataset split strategies are designed to study the affect of time gap between different sessions of data collection. It has found out that system performance decreases when final validation of system would be done on data which have been collected with few months of gap from training data. Different experimental settings are tested to find the optimal system performance and to make the system more secure by locking out the imposter user as quickly as possible.

4. **Deep learning techniques were proposed in contrast to baseline approach to validate CUA with keystroke dynamics.**

The fourth objective was to propose the deep learning techniques in contrast to baseline approach to validate CUA with keystroke dynamics. Deep learning methods are used to train the keystroke data as sequential time-series data to learn unique hidden features which baseline methods cannot mine properly. In this regard, different frameworks are formulated with recurrent neural networks (RNN) which tends to learn the time series data efficiently. LSTM is used as a recurrent unit of RNN which can add to or remove from the previous important information about the user's unique features. LSTM is integrated with proposed R-RCM to make a true CUA system. Different experimental settings are formulated with LSTM and R-RCM to acheive the optimal system performance in comparison to preceding works done in this domain. Moreover, the system is tested with two types of dataset split strategies (as done in baseline method/objective. no 3). It has found out that deep neural network method reduces the gap between training and final validation data owing to the usage of integrated LSTM R-RCM methods, hence improved the system performance. In addition, an important architecture is introduced which integrated the per action/frame classification and per sequence classification hence it is facilitated with the advantages of both continuous as well as periodic user authentication to improve the system performance. To best of our knowledge, the combination of periodic and continuous user authentication is studied for the first time using deep neural network.

5. **CUA based on traditional statistical features versus proposed temporal features**

   The fifth objective was to analyse the continuous user authentication with behavioural biometrics based on traditional statistical features versus proposed temporal features. This research investigated both methods of populating the reference template database of user with mean and stand deviation of features and treating the behavioural data as a sequential time-series where each event has some connection with previous event. It had turned out the experimental framework which utilised the time-series data had shown improved system performance as compared to statistical features.

6. **A method was proposed to establish the user identity continuously without prior claim of identity at start of session**

   The sixth objective was to propose a method to establish the user's identity continuously without prior claim of identity at start of session. In this research, a novel approach of continuous user identification (CUI) is investigated based on RNN and R-RCM. A novel region labelling approach is proposed to correctly identify the current user from a given pool of users. The GRU is used as a recurrent unit along with CTC approach which is an idea of inserting blank labels on the regions of low user confidence and CTC has been implemented for user identification problem for the first time. The proposed framework has achieved a good performance in identifying the user and also to lock out the imposter users.

7. **Proposed architectures were validated with Mouse Dynamics biometric modality**

   The seventh and eighth objective was to investigate Mouse Dynamics modality over CUA in order to explore CUA in comparison to other behavioural biometrics. Mouse dynamics is investigated with the proposed architecture consisting of baseline and deep learning method. For the first time, Recurrent neural networks (RNN) are implemented for mouse dynamics modality to authenticate user on each action. Different proposed architectures are applied and optimal results are achieved in comparison to preceding scholarly works done in domain of CUA using mouse dynamics.

8.2 Limitations of this Work

The CUA system has been implemented successfully with the provided dataset by applying the novel contributed techniques and the performance of system has escalated to a greater extent, However, the current study has few limitations which are listed below:

- **Active attack scenario:** In this research work, zero effort attack scenario is considered, however, it is also interesting to know the performance of system under active attack scenario where an attacker tries to mimic the genuine user's behaviour. According to general understanding, it is quite difficult task to continuously imitate someone's typing or mouse usage behaviour successfully. Hence, the experiments for active attack scenario are complex and need extensive effort, focus and time from the participants. However, the experiment can be designed to further study the exact affect of active attack on system.
- **Dataset Availability**

  Continuous behavioural analysis of any user takes the data patterns continuously from the user. Hence, the huge amount of data is required to sufficiently train the system. However, the available datasets do not have an infinite amount of data which can be used to train deep learning classifiers. Moreover, the collection of data is done under controlled environment with limited time for each session which can have an impact on normal behaviour of user. In order to collect data under free environment for sufficient amount of time needs the careful handling of data under data protection laws which can be quite complex task to perform.

- **Implementation of CUA system in real environment**

  The CUA system continuously collect the behavioural information regarding the computer usage devices of users. Hence, it is very important for any organisation to get the consent of users regarding the passive collection of behavioural patterns before the proposed solution can be installed on systems. However, the risks associated with the unauthorised usage of computing resources outweighs the privacy concerns of users which can be explained to them before the system installation. In this regard, users need to be ensured that their data would be handled under the data protection laws.

8.3 Further Work

There are many areas related to the understudy research topic which can be further researched to take this work into new directions. In order to make the proposed CUA and CUI system more robust and to overcome some of the limitations, some of the following issues can be addressed:

- **Combination of Periodic and Continuous User Authentication**

  In this research, the integration of PUA and CUA is done where per frame and per sequence probabilities are utilised to authenticate the user. However, different architectures can be experimented to further explore this idea of combining PUA and CUA. For instance, the probability scores of PUA and CUA can be fused together to make the final decision instead of independently sending both probabilities to R-RCM to make the final decision.

- **Investigation of hardware effects on CUA**

  The investigation of hardware changes, for instance different keyboards with different key layout, can be used to collect behavioural biometric data. This behavioural data can be utilised to study the effect of hardware which is new to the user. For example, if the keyboard of users will be changed, then if there would be any change in performance of continuous authentication system.

- **CUA using Facial and Keystroke/Mouse Recognition**

  With increasing usage of computers, there is a growing concern about ensuring the security of users' personal information on these devices. The research can utilise face recognition for CUA to detect the mood of user if he/she is angry, sad, excited or normal and evaluate his/her keystroke in accordance to their current mood in order to avoid false lockout. For instance, if user is angry then final threshold can be keep low since in super angry mood user will press the keys with more pressure which can result in legitimate user false lockout.

- **System adaptability according to the Application used**
  It is assumed that users may behave differently for different applications e.g. the user's behaviour will change from playing games to typing important documents. These could be useful to adjust the lockout threshold to improve the system performance. As an example, when a user is playing games, then lockout threshold can be lowered.

- **CUI approach can be Explored Further**
  The novel architecture provided for CUI can be utilised for forensic analysis as well to detect the imposter users quickly. It is still an open research area and the proposed architecture can be explored further to improve the system.

- **LSTM integration with Convolutional Neural Network (CNN)**
  The proposed system can utilise the two powerful deep neural architectures named as LSTM and convolutional Neural Network (CNN). The different forms of LSTM have been used in this work, however, in future the combination of LSTM and CNN can be utilised to extract more advanced features and to improve the system performance.

- **CUA extension with touchscreen, stylus pens or voice recognition**

  The proposed system has utilised the keystroke and mouse dynamics to continuously authenticate the user. However, the proposed system can be further extended with new input methods i.e., touchscreen, swipe gestures, stylus pens or voice recognition. The proposed architectures can be applied to above mentioned new input biometric traits to validate the system architectures since the system is effectively trained to extract the hidden features from raw data and to apply proposed hybrid models to any biometric trait.

- **CUA using the alternative biometric trait for special users**

  The proposed system can be extended for special users or any user who cannot provide the keystroke or mouse dynamics due to injury, disability or unavailable/faulty system hardware. In this regard, the system can be extended in such a way that if user is unable to provide the keystroke or mouse dynamics at any given time, then alternative biometric traits for instance facial recognition or iris recognition could be utilised to continuous monitoring of the user's genuineness.

In conclusion, the work presented in this thesis aimed to use enhanced technologies and architectures to continuously authenticate and identify the users correctly thereby locking out the imposter users quickly to avoid the damage caused to system by intruders and also to avoid the false lock out of genuine users hence substantially escalating the genuine user's working productivity.

References

Ahmed, A. A. E. and Traore, I. (2005), Anomaly intrusion detection based on biometrics, *in* 'Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop', IEEE, pp. 452–453. 29

Ahmed, A. A. E. and Traore, I. (2007), 'A new biometric technology based on mouse dynamics', *IEEE Transactions on dependable and secure computing* **4**(3), 165–179. 29, 122

Ahmed, A. A. and Traore, I. (2013), 'Biometric recognition based on free-text keystroke dynamics', *IEEE transactions on cybernetics* **44**(4), 458–472. 27, 28, 84

Ali, M. L., Monaco, J. V., Tappert, C. C. and Qiu, M. (2017), 'Keystroke biometric systems for user authentication', *Journal of Signal Processing Systems* **86**(2-3), 175–190. 58

Alotaibi, S., Alruban, A., Furnell, S. and Clarke, N. L. (2019), A novel behaviour profiling approach to continuous authentication for mobile applications., *in* 'ICISSP', pp. 246–251. 3, 5

Alsultan, A., Warwick, K. and Wei, H. (2016), 'Free-text keystroke dynamics authentication for arabic language', *IET Biometrics* **5**(3), 164–169. 23

Alsultan, A., Warwick, K. and Wei, H. (2017), 'Non-conventional keystroke dynamics for user authentication', *Pattern Recognition Letters* **89**, 53–59. 27, 28, 84

Antal, M. and Egyed-Zsigmond, E. (2019), 'Intrusion detection using mouse dynamics', *IET Biometrics* **8**(5), 285–294. 29, 30, 31

Antal, M. and Fejér, N. (2020), 'Mouse dynamics based user recognition using deep learning', *Acta Universitatis Sapientiae, Informatica* **12**(1), 39–50. 29, 31

Ayotte, B., Banavar, M., Hou, D. and Schuckers, S. (2020), 'Fast free-text authentication via instance-based keystroke dynamics', *IEEE Transactions on Biometrics, Behavior, and Identity Science* **2**(4), 377–387. 27, 28

Ayotte, B., Huang, J., Banavar, M. K., Hou, D. and Schuckers, S. (2019), Fast continuous user authentication using distance metric fusion of free-text keystroke data, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops', pp. 0–0. 26, 27, 84

Bakelman, N., Monaco, J. V., Cha, S.-H. and Tappert, C. C. (2012), 'Continual keystroke biometric authentication on short bursts of keyboard input', *Proceedings of Student-Faculty Research Day, CSIS, Pace University* . 33, 51

Bours, P. and Barghouthi, H. (2009), Continuous authentication using biometric keystroke dynamics, *in* 'The Norwegian Information Security Conference (NISK)', Vol. 2009. 25, 44

Bours, P. and Mondal, S. (2015*a*), 'Continuous authentication with keystroke dynamics', *Norwegian Information Security Laboratory NISlab* pp. 41–58. 34, 53, 84

Bours, P. and Mondal, S. (2015*b*), 'Performance evaluation of continuous authentication systems', *IET Biometrics* **4**(4), 220–226. 100

Çeker, H. and Upadhyaya, S. (2016), User authentication with keystroke dynamics in long-text data, *in* '2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)', IEEE, pp. 1–6. 27, 28, 84

Chandranegara, D. R., Wibowo, H. and Minarno, A. E. (2020), 'Combined scaled manhattan distance and mean of horner's rules for keystroke dynamic authentication', *Telkomnika* **18**(2), 770–775. 84

Chen, T. and Guestrin, C. (2016), Xgboost: A scalable tree boosting system, *in* ‘Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining’, pp. 785–794. 67

Curtin, M., Tappert, C., Villani, M., Ngo, G., Simone, J., Fort, H. S. and Cha, S. (2006), ‘Keystroke biometric recognition on long-text input: A feasibility study’, *Proc. Int. MultiConf. Engineers & Computer Scientists (IMECS)* . 25

Dee, T., Richardson, I. and Tyagi, A. (2019), Continuous transparent mobile device touchscreen soft keyboard biometric authentication, *in* ‘2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)’, IEEE, pp. 539–540. 1

Di Tommaso, F., Guerra, M., Martinelli, F., Mercaldo, F., Piedimonte, M., Rosa, G. and Santone, A. (2019), User authentication through keystroke dynamics by means of model checking: A proposal, *in* ‘2019 IEEE International Conference on Big Data (Big Data)’, IEEE, pp. 6232–6234. 25

Feher, C., Elovici, Y., Moskovitch, R., Rokach, L. and Schclar, A. (2012), ‘User identity verification via mouse dynamics’, *Information Sciences* **201**, 19–36. 29, 30

Ferrari, C., Marini, D. and Moro, M. (2018), An adaptive typing biometric system with varying users model, *in* ‘2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)’, IEEE, pp. 564–568. 26, 27

Ferreira, J. and Santos, H. (2012), Keystroke dynamics for continuous access control enforcement, *in* ‘2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery’, IEEE, pp. 216–223. 26, 27

Foresi, A. and Samavi, R. (2019), User authentication using keystroke dynamics via crowdsourcing, *in* ‘2019 17th International Conference on Privacy, Security and Trust (PST)’, IEEE, pp. 1–3. 25

Gamboa, H. and Fred, A. (2004*a*), A behavioral biometric system based on human-computer interaction, *in* ‘Biometric Technology for Human Identification', Vol. 5404, International Society for Optics and Photonics, pp. 381–392. 29

Gamboa, H. and Fred, A. (2004*b*), A behavioral biometric system based on human-computer interaction, *in* 'Biometric Technology for Human Identification', Vol. 5404, International Society for Optics and Photonics, pp. 381–392. 141

Giot, R., El-Abed, M., Hemery, B. and Rosenberger, C. (2011), 'Unconstrained keystroke dynamics authentication with shared secret', *Computers & security* **30**(6-7), 427–445. 51

Graves, A., Fernández, S., Gomez, F. and Schmidhuber, J. (2006), Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks, *in* 'Proceedings of the 23rd international conference on Machine learning', pp. 369–376. 106

Gunetti, D., Picardi, C. and Ruffo, G. (2005), Dealing with different languages and old profiles in keystroke analysis of free text, *in* 'Congress of the Italian Association for Artificial Intelligence', Springer, pp. 347–358. 25, 27

Hinbarji, Z., Albatal, R. and Gurrin, C. (2015), Dynamic user authentication based on mouse movements curves, *in* 'International Conference on Multimedia Modeling', Springer, pp. 111–122. 29, 30

Hsu, C.-W., Chang, C.-C., Lin, C.-J. et al. (2003), 'A practical guide to support vector classification'. 65, 126

Huang, J., Hou, D. and Schuckers, S. (2017), A practical evaluation of free-text keystroke dynamics, *in* '2017 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)', IEEE, pp. 1–8. 26, 27

Kim, J. and Kang, P. (2020), 'Freely typed keystroke dynamics-based user authentication for mobile devices based on heterogeneous features', *Pattern Recognition* **108**, 107556. 26, 27, 101

Kiyani, A. T., Lasebae, A., Ali, K. and Ur-Rehman, M. (2020), Secure online banking with biometrics, *in* '2019 International Conference on Advances in the Emerging Computing Technologies (AECT)', IEEE, pp. 1–6. 2, 17

Kochegurova, E. and Martynova, Y. A. (2020), 'Aspects of continuous user identification based on free texts and hidden monitoring', *Programming and Computer Software* **46**(1), 12–24. 3

Kolakowska, A. (2011), User authentication based on keystroke dynamics analysis, *in* 'Computer Recognition Systems 4', Springer, pp. 667–675. 26, 27

Liang, S.-Y., Han, D.-Q. and Han, C.-Z. (2014), 'A novel diversity measure based on geometric relationship and its application to design of multiple classifier systems', *Acta Automatica Sinica* **40**(3), 449–458. 124

Locklear, H., Govindarajan, S., Sitová, Z., Goodkind, A., Brizan, D. G., Rosenberg, A., Phoha, V. V., Gasti, P. and Balagani, K. S. (2014), Continuous authentication with cognition-centric text production and revision features, *in* 'IEEE International Joint Conference on Biometrics', IEEE, pp. 1–8. 26, 27

Lu, X., Zhang, S., Hui, P. and Lio, P. (2020), 'Continuous authentication by free-text keystroke based on cnn and rnn', *Computers & Security* **96**, 101861. 28, 101

M Jomaa, R., Mathkour, H., Bazi, Y. and Islam, M. S. (2020), 'End-to-end deep learning fusion of fingerprint and electrocardiogram signals for presentation attack detection', *Sensors* **20**(7), 2085. 102

Manandhar, R., Wolf, S. and Borowczak, M. (2019), One-class classification to continuously authenticate users based on keystroke timing dynamics, *in* '2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)', IEEE, pp. 1259–1266. 27, 28

Medsker, L. R. and Jain, L. (2001), 'Recurrent neural networks', *Design and Applications* **5**, 64–67. 85

Mendialdua, I., Martínez-Otzeta, J. M., Rodriguez-Rodriguez, I., Ruiz-Vazquez, T. and Sierra, B. (2015), 'Dynamic selection of the best base classifier in one versus one', *Knowledge-Based Systems* **85**, 298–306. 63

Mhenni, A., Cherrier, E., Rosenberger, C. and Amara, N. E. B. (2019), 'Double serial adaptation mechanism for keystroke dynamics authentication based on a single password', *Computers & Security* **83**, 151–166. 23

Mi, A., Wang, L. and Qi, J. (2016), 'A multiple classifier fusion algorithm using weighted decision templates', *Scientific Programming* **2016**. 65, 126

Mondal, S. and Bours, P. (2017*a*), 'A study on continuous authentication using a combination of keystroke and mouse biometrics', *Neurocomputing* **230**, 1–22. 29, 30

Mondal, S. and Bours, P. (2017*b*), 'A study on continuous authentication using a combination of keystroke and mouse biometrics', *Neurocomputing* **230**, 1–22. 143

Motlagh, M. R. M. (2015), Continuous user identification, Master's thesis. 117

Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T. and Koucheryavy, Y. (2018), 'Multi-factor authentication: A survey', *Cryptography* **2**(1), 1. 15

Patel, V. M., Chellappa, R., Chandra, D. and Barbello, B. (2016), 'Continuous user authentication on mobile devices: Recent progress and remaining challenges', *IEEE Signal Processing Magazine* **33**(4), 49–61. 23

Paulsen, B., Wang, J. and Wang, C. (2020), Reludiff: Differential verification of deep neural networks, *in* '2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)', IEEE, pp. 714–726. 103

Pilankar, P. S. and Padiya, P. (2016), Multi-phase mouse dynamics authentication system using behavioural biometrics, *in* '2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)', IEEE, pp. 1947–1950. 119

Pinto, P., Patrão, B. and Santos, H. (2014), Free typed text using keystroke dynamics for continuous authentication, *in* 'IFIP International Conference on Communications and Multimedia Security', Springer, pp. 33–45. 26, 27

Porwik, P., Doroz, R. and Wesolowski, T. E. (2021), 'Dynamic keystroke pattern analysis and classifiers with competence for user recognition', *Applied Soft Computing* **99**, 106902. 28

Pusara, M. and Brodley, C. E. (2004), User re-authentication via mouse movements, *in* 'Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security', pp. 1–8. 29

Salem, A. and Obaidat, M. S. (2019), 'A novel security scheme for behavioral authentication systems based on keystroke dynamics', *Security and Privacy* **2**(2), e64. 25

Salman, O. A. and Hameed, S. M. (2018), Using mouse dynamics for continuous user authentication, *in* 'Proceedings of the Future Technologies Conference', Springer, pp. 776–787. 119

Senathipathi, K. and Batri, K. (2014), An analysis of particle swarm optimization and genetic algorithm with respect to keystroke dynamics, *in* '2014 international conference on green computing communication and electrical engineering (ICGCCEE)', IEEE, pp. 1–11. 25

Shen, S.-S., Kang, T.-H., Lin, S.-H. and Chien, W. (2017), Random graphic user password authentication scheme in mobile devices, *in* '2017 International conference on applied system innovation (ICASI)', IEEE, pp. 1251–1254. 2

Shepherd, S. (1995), 'Continuous authentication by analysis of keyboard typing characteristics'. 23

Shikder, R., Rahaman, S., Afroze, F. and Al Islam, A. A. (2017), Keystroke/mouse usage based emotion detection and user identification, *in* '2017 International Conference on Networking, Systems and Security (NSysS)', IEEE, pp. 96–104. 84

Siami-Namini, S., Tavakoli, N. and Namin, A. S. (2019), The performance of lstm and bilstm in forecasting time series, *in* '2019 IEEE International Conference on Big Data (Big Data)', IEEE, pp. 3285–3292. 129

Singh, M., Singh, R. and Ross, A. (2019), 'A comprehensive overview of biometric fusion', *Information Fusion* **52**, 187–205. 19

Song, Y.-Y. and Ying, L. (2015), 'Decision tree methods: applications for classification and prediction', *Shanghai archives of psychiatry* **27**(2), 130. 126

Sun, Y., Ceker, H. and Upadhyaya, S. (2016), Shared keystroke dataset for continuous authentication, *in* '2016 IEEE International Workshop on Information Forensics and Security (WIFS)', IEEE, pp. 1–6. 40, 119

Syukri, A. F., Okamoto, E. and Mambo, M. (1998), A user identification system using signature written with mouse, *in* 'Australasian Conference on Information Security and Privacy', Springer, pp. 403–414. 28

Tse, K.-W. and Hung, K. (2020), User behavioral biometrics identification on mobile platform using multimodal fusion of keystroke and swipe dynamics and recurrent neural network, *in* '2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE)', IEEE, pp. 262–267. 35, 84, 88

Velásquez, I., Caro, A. and Rodríguez, A. (2018), 'Authentication schemes and methods: A systematic literature review', *Information and Software Technology* **94**, 30–37. 2

Vyazigin, A. A., Tupikina, N. Y. and Sypin, E. V. (2019), Software tool for determining of the keystroke dynamics parameters of personal computer user, *in* '2019 20th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM)', IEEE, pp. 166–171. 58, 84

Weile, D. S. and Michielssen, E. (1997), 'Genetic algorithm optimization applied to electromagnetics: A review', *IEEE Transactions on Antennas and Propagation* **45**(3), 343–353. 65, 67

Wu, P.-Y., Fang, C.-C., Chang, J. M. and Kung, S.-Y. (2016), 'Cost-effective kernel ridge regression implementation for keystroke-based active authentication system', *IEEE transactions on cybernetics* **47**(11), 3916–3927. 27, 28

Xiaofeng, L., Shengfei, Z. and Shengwei, Y. (2019), 'Continuous authentication by free-text keystroke based on cnn plus rnn', *Procedia computer science* **147**, 314–318. 88, 89

Yaacob, M. N., Idrus, S. Z. S., Ali, W. N. A. W., Mustafa, W. A., Jamlos, M. A. and Abd Wahab, M. H. (2020), Decision making process in keystroke dynamics, *in* 'Journal of Physics: Conference Series', Vol. 1529, IOP Publishing, p. 022087. xi, 21

Yang, S., Yu, X. and Zhou, Y. (2020), Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example, *in* '2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)', IEEE, pp. 98–101. 104

Yang, W., Wang, S., Hu, J., Zheng, G., Chaudhry, J., Adi, E. and Valli, C. (2018), 'Securing mobile healthcare data: a smart card based cancelable finger-vein bio-cryptosystem', *IEEE Access* **6**, 36939–36947. xi, 17, 18

Yao, X. (1999), 'Evolving artificial neural networks', *Proceedings of the IEEE* **87**(9), 1423–1447. 66

Zheng, N., Paloski, A. and Wang, H. (2016), 'An efficient user verification system using angle-based mouse movement biometrics', *ACM Transactions on Information and System Security (TISSEC)* **18**(3), 1–27. 29, 30