IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 27, 2020, accepted August 17, 2020, date of publication August 26, 2020, date of current version September 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3019467

Continuous User Authentication Featuring Keystroke Dynamics Based on Robust Recurrent Confidence Model and Ensemble Learning Approach

ANUM TANVEER KIYANI[1], ABOUBAKER LASEBAE[1], KAMRAN ALI[1], (Member, IEEE), MASOOD UR REHMAN[2], AND BUSHRA HAQ[3]

[1]Faculty of Science and Technology, Middlesex University, London NW4 4BT, U.K.
[2]James Watt School of Engineering, University of Glasgow, Glasgow G12 8QQ, U.K.
[3]Department of Computer Science, Baluchistan University of Information Technology, Engineering, and Management Sciences, Quetta 83700, Pakistan

Corresponding author: Anum Tanveer Kiyani (ak1933@live.mdx.ac.uk)

ABSTRACT User authentication is considered to be an important aspect of any cyber security program. However, one-time validation of user's identity is not strong to provide resilient security throughout the user session. In this aspect, continuous monitoring of session is necessary to ensure that only legitimate user is accessing the system resources for entire session. In this paper, a true continuous user authentication system featuring keystroke dynamics behavioural biometric modality has been proposed and implemented. 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 which employs a novel perception of two thresholds i.e., alert and final threshold. Proposed methodology classifies each action based on the probability score of ensemble classifier which is afterwards used along with hyper-parameters 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. However, it tends to lock out imposter user more quickly if it reaches the alert threshold. Moreover, system has been validated with two different experimental settings and results are reported in terms of mean average number of genuine actions (ANGA) and average number of imposter actions(ANIA), whereby achieving the lowest mean ANIA with experimental setting II.

INDEX TERMS Continuous user authentication, keystroke dynamics, ensemble learning, behavioural analysis, biometrics, security systems.

I. INTRODUCTION

In In modern networks, the security of critical computer systems is highly susceptible to different attacks at the user level, system level or network level precisely. Subsequently, in the user level attacks i.e., masquerade attacks, intruder exploits the legitimate user rights for unauthorized 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 [1]. Consequently, security of critical cyber security system is mainly reliant on authentication or identification principles [2]. 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 [3]. This is referred to as static user authentication (SUA). However, if a person leaves its system or phone unattended or forgets to log out from authenticated session of any critical application that contains

The associate editor coordinating the review of this manuscript and approving it for publication was Marina Gavrilova.

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 system or application after initial log-in to ensure that the legitimate user is using the system for entire session. This is referred to as continuous user authentication (CUA) [4].

A robust continuous user authentication (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. Secondly, CUA should authenticate the user continuously on every single activity that user is performing. In order to meet these requirements, one possible way is to use behavioural biometrics e.g., keystroke 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 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. 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. In terms of implementation, there are numerous advantages for the usage of KD as a recognition method [5] 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. However, keystroke dynamics cannot substitute the traditional initial login methods but KD can provide an additional security layer which incessantly validates the user identity during the session. 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. Accordingly, most of the preceding research had employed the analysis based on a fixed number of actions or fixed time period which can be called as episodic authentication where system records the keystroke 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. On the contrary, a true continuous authentication system inclines to verify the identity of user after each keystroke action [6]. 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 [7] and different discrete features are extracted from those captured keystroke timestamps.

In this paper, we aim to implement a true continuous authentication system which can authenticate the user based on each keystroke action as shown in Fig.1. The main contributions of this work are:

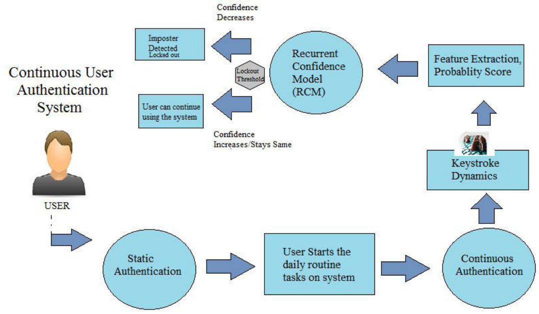
Continuous User Authentication System
USER
Confidence Decreases
Imposter Detected Locked out
Lockout Threshold
User can continue using the system
Confidence Increases/Stays Same
Recurrent Confidence Model (RCM)
Feature Extraction, Probablity Score
Keystroke Dynamics
Static Authentication
User Starts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

FIGURE 1. Framework of continuous user authentication.

- A robust-recurrent confidence model has been proposed which tends to authenticate the user on each single action performed on system. The system has been validated with keystroke dynamics, however, it can be implemented on any behavioural biometric modality.
- The proposed robust-recurrent confidence model uses a novel approach of detecting and locking out the imposter user once it crosses the alert threshold.
- The 2-phase methodology has been implemented for continuously authenticating the users which treats the input features as a key-press sequence instead of generating the keystroke profiles based on mean and standard deviations of key and key-pairs as found in literature.
- The two different experimental settings have been formulated which include the combination of divergent approaches in order to validate the proposed system methodology.

The rest of this paper is structured as follows. Section II presents the background and related work. Section III introduces a proposed system model for continuous user authentication. Section IV contains the detailed discussion on applied methodology and results. Afterwards, conclusion and further research are presented in Section V.

II. LITERATURE REVIEW

This section presents the speculative basis and preceding research works 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. Preliminary research on CUA using keystroke dynamics was conducted in 1995 by the group of researchers [8] and some notable results were presented.

The presently available keystroke dynamics datasets can be specifically categorized into two types, namely, short text and long text, as shown in Fig. 2. The short texts datasets are predominantly based on passwords thereby mostly appropriate for studying the static authentication [9]. On the other hand,

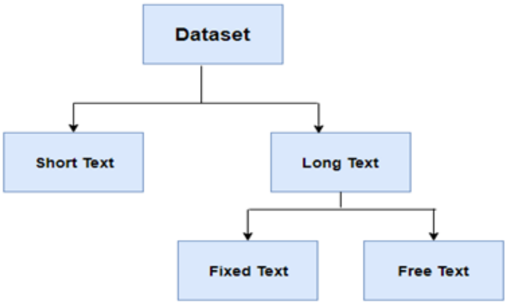

FIGURE 2. **Keystroke dynamics dataset classification.**

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 [10].

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 [11]. Subsequently, numerous different features can be extracted 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.

User templates are created by calculating the mean and standard deviation of each key hold time and key digraph flight and latency [12]. On the other hand, some research studies [6] 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 [13] had employed the combination of key digraph, trigraph, error corrections and word per minute features to build the user profiles. Additionally in some studies [14] 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. [15]. Moreover, in [16] researchers had combined the timing features with non-timing features i.e., pressure, position, finger placement and finger choice for tying behaviour analysis.

It has been observed that most of the research work had built statistical user profiles based on mean and standard deviation of specific keys and key-pairs. However, 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.

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.

The researchers in [17] 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 is 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 have used the block size of 700-900 keystrokes to form each sample probe to identify the user which gives enough possibility to imposter for unauthorized access. Some other research works have also implemented the relative distance and absolute distance including [18] with sliding window of fixed n-graph latency features, [19] with 600 block size and duration of 2,3,4 and 5-graph features, [20] with 150 block size and di-graphs features, [21] with 100-1000 block size and di-graph latency features precisely, and [22] with block size of 250 actions and duration, digraph latency. The researchers in [23] 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 have 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.

Other statistical methods used for classification of keystroke dynamics in literature includes Euclidean distance [24], scaled Euclidean distance [25], scaled Manhattan distance with Mean of Horners Rules [26], Mahalanobis distance [21], and Bhattacharyya distance with Gaussian mixture model [27]. In addition, other statistical techniques such as Hidden Markov Model [28], Kolmogorov-Smirnov Test (KS-test) [29], and Bayesian Classification [30], were also employed to find the level of similarity between keystroke samples. It has been noticed that most of the research works in CUA domain had considered the block of actions to authenticate the user. However, a true CUA system should authenticate user on each single action. In this regard, this research work proposed the recurrent confidence model which authenticates the user on each single action and decides the legitimacy of user in combination with previous actions confidence.

Machine learning has also been exploited in recent times where some of the works have presented interesting results. A constructive example of this is presented in [31] 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 has never occurred while training the network. Another research work in [32] had implemented Decision trees with statistical feature profiles and used the block size of 1000 actions. Moreover, in [33] kernel ridge regression a truncated RBF kernel has been used with 900 words block size and trigraph latency feature profile.

In most preceding works, commonly used features for CUA with keystroke dynamics are digraphs. However, if we consider the real continuous authentication 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 so the optimal approach used in this research is fusion of monographs as well as digraphs to achieve better results.

Afterwards, the classification algorithms generally report the performance in terms of false acceptance rate(FAR), false rejection rate(FRR) [34] and equal error rate(EER) for biometric systems. [35]. 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 lock it out respectively. Based on our 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. In this aspect, this research uses the performance metrics as describe by researchers in [25] in form of ANGA and ANIA.

III. SYSTEM METHODOLOGY

This section presents the architecture and implementation of proposed CUA system which combines the SUA as well.

A. DATASET

In this research, the keystroke dataset provided by University of Buffalo [36] has been used. The baseline dataset is collected from 75 subjects in 3 separate sessions and the statistics of dataset is presented in table 1. There are 28 days in average time intervals between sessions. The dataset is based on long-text and it is the mixture of fixed and free style texts.

TABLE 1. Dataset statistics.

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

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

- $UserId(e)$ – 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.3 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 sequence and feed to training classifiers to build the reference templates for each user which can be used for authentication of user upon validation.

Given a tuple ($UserId'$, $SessionId'$, $DownTime$, $UpTime$, $KeyCode$) we group keyboard events into sequences:

$$\begin{aligned} &Sequence(UserId', SessionId') \\ &= \{e \mid \forall e \in E, s.t.UserId(e) = UserId' \text{ and } SessionId(e) \\ &= SessionId' \\ &\quad \text{and } DownTime(e) = DownTime \\ &\quad \text{and } UpTime(e) = UpTime \\ &\quad \text{and } KeyCode(e) = KeyCode'\} \end{aligned}$$

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

$$\begin{aligned} e_i < e_j \text{ if } &DownTime(e_i) < DownTime(e_j) \text{ or} \\ &DownTime(e_i) = DownTime(e_j) \text{ and} \\ &UpTime(e_i) < UpTime(e_j) \end{aligned}$$

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. We have defined the split range rule as follows:

$$\begin{aligned} SplitRange = [&SplitRange_0, SplitRange_1), \\ &SplitRange_0 \geq 0, \quad SplitRange_1 \leq 1, \end{aligned}$$

hence, $SplitRange_0 < SplitRange_1$.

In accordance to split range rule we have applied the following split strategy:

- T=train, $SessionId(E)$, $SplitRange = [0.0, 0.6]$
- X=test, $SessionId(E)$, $SplitRange = [0.6, 0.8]$
- V=val, $SessionId(E)$, $SplitRange = [0.8, 1.0]$

B. FEATURE ENGINEERING

Let's say we have a sequence of $M + U$ keystrokes where U is the context length and M is the length of keystroke sequence. Sequences of a defined length $M + U$ have been sampled to generate input features and target user ids *(x,y)*

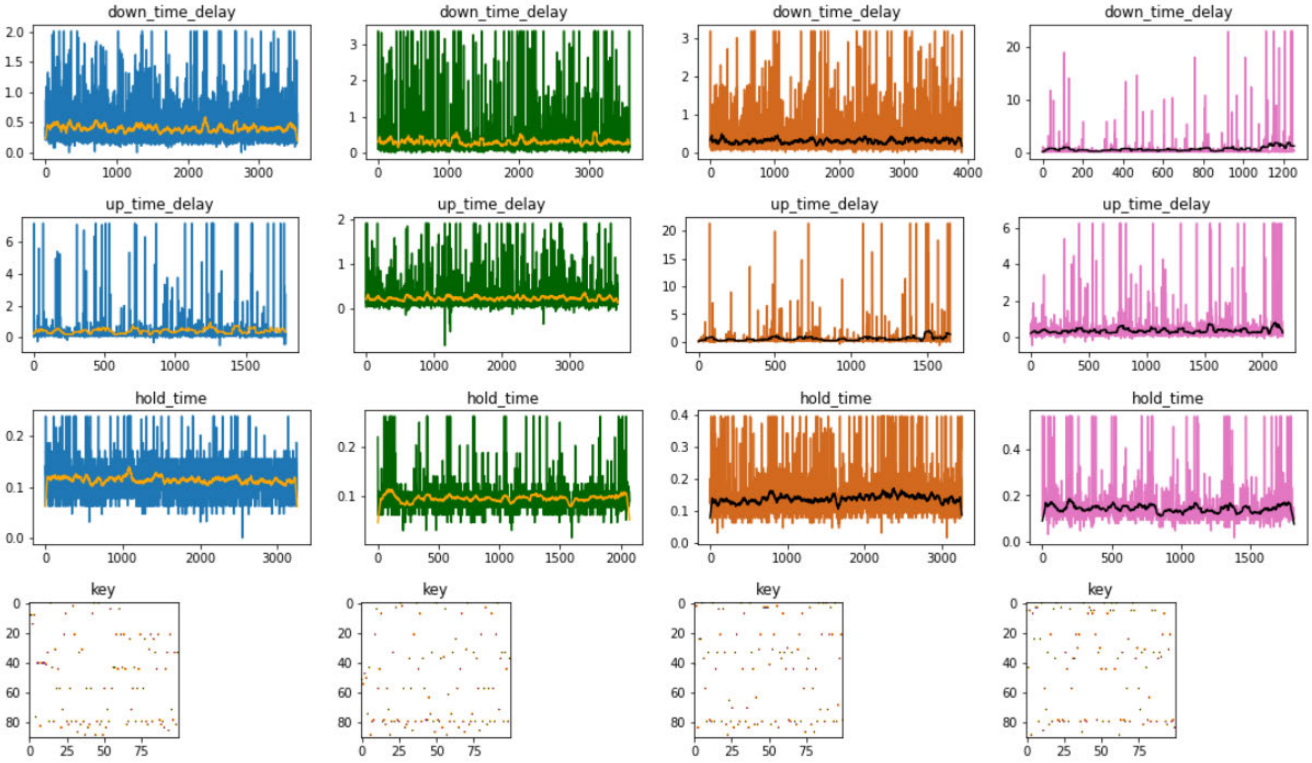

FIGURE 3. Keystroke distinct features for 4 Different users.

with *T time steps* in total. Moreover, sequences are sampled with *U = 1* and *M = 512*, a sequence of keyboard actions with monographs and digraph features. In the experiments, the following features have been considered to generate from the raw keystroke sequence.

- *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
  - $Down - Up\ Time$: Total time duration of first key press to second key release.
  - $Down - Down\ Time$: The time between first key press and second key press
  - $Up - Down\ Time$: The time between first key release and second key press
  - $Up - Up\ Time$: The time between first key release and second key release of a particular key digraph.

The one attribute and five main features are utilized 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 via one hot encoding. To apply a classification algorithm, input data has been processed to obtain numerical feature series as given follows: For $\forall t = \overline{0, M-1}, p = t-1$, we have:

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

The graphical representation of the keystroke dynamics feature extraction process is shown in Fig.4. 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 sequence of keystrokes by pausing for 2000ms after each keypress hence leaving no feature for system to authenticate the user successfully.

C. ROBUST RECURRENT CONFIDENCE MODEL(R-RCM)

Most of the work done in CUA systems, as observed in literature review, considers 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, we have proposed the robust Recurrent confidence Model (R-RCM) which considers each and every action of user in order to decide if user is legitimate or not. However, each action itself does not make this decision but R-RCM takes into account the confidence generated by

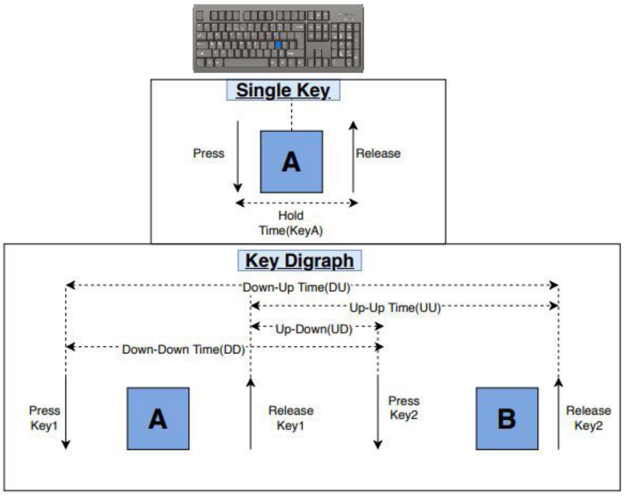

FIGURE 4. Keystroke dynamics features representation.

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 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 same manner to each other and at some point the behaviour of imposter user 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, we used the concept of "recurrent confidence in the genuineness" of the current user.

In [6] researchers had used the similar approach of trust model for CUA based on threshold function. They showed that the trust level escalates or lessens based on the scaled Manhattan distance between the legitimate user reference template and current typing actions. However, the same concept had been used by [37] where the trust level variation depends on the probability score of current action. In this paper, we are proposing a Robust recurrent confidence model*(R-RCM)* which keeps track of previous confidence value and tends to lock out the user from system once it reaches the final lockout threshold. Confidence value depends on the fused classifier score from ensemble classifier.

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

As stated above, CUA cannot substitute the SUA so once user logs in to 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 user can use the system without any restraint, however if the confidence of user goes below the given final threshold then user will be locked out of the system.

In this research, 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.5.

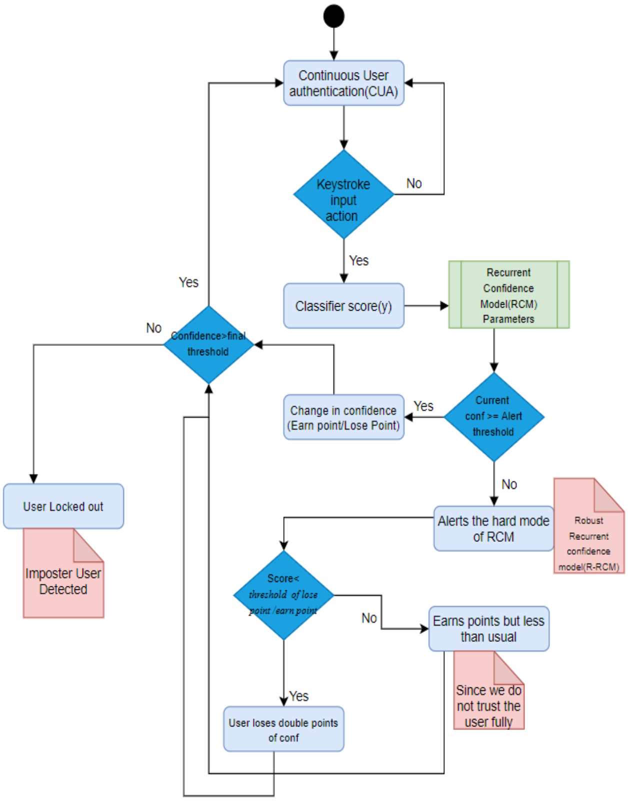
Continuous User authentication(CUA)
Keystroke input action
No
Yes
Classifier score(y)
Recurrent Confidence Model(RCM) Parameters
Yes
No
Confidence>final threshold
Change in confidence (Earn point/Lose Point)
Yes
Current conf >= Alert threshold
No
User Locked out
Imposter User Detected
Alerts the hard 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 5. 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 will 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.6 and Fig.7.

In Fig.6, when training sample of genuine user has been compared with its own validation sample, it can be noticed that how the recurrent confidence level 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.

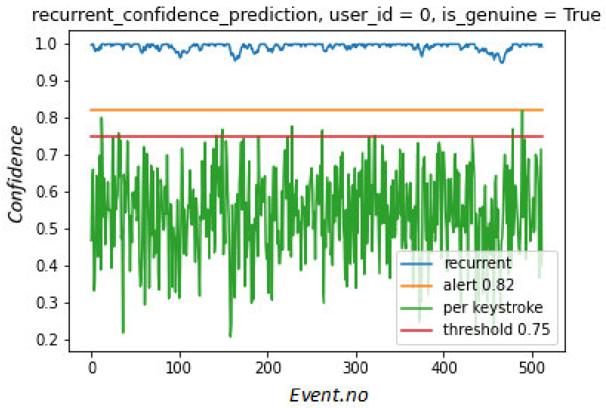

FIGURE 6. **Confidence value for genuine user tested with the genuine test data.**

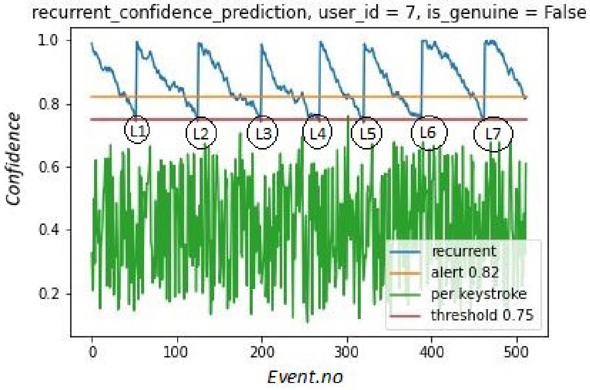

FIGURE 7. **Confidence value for genuine user tested with the imposter test data.**

However, Fig.7 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, we have assumed that after every lock out the user is again using the SUA to access the system and its maximum confidence of 1.00 is re-gained.

D. SYSTEM ARCHITECTURE

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

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

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,

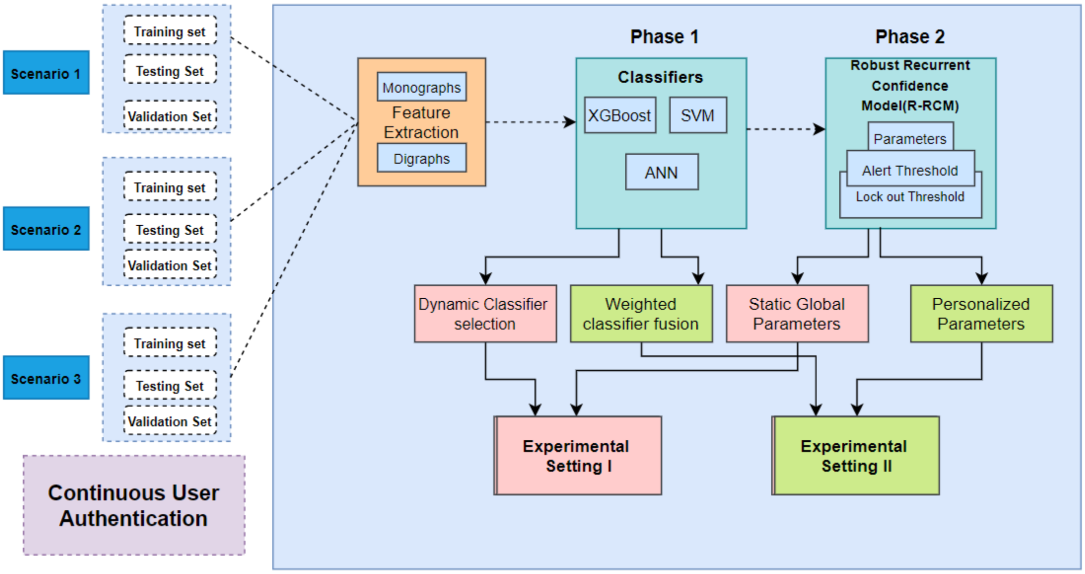

FIGURE 8. The system Architecture.

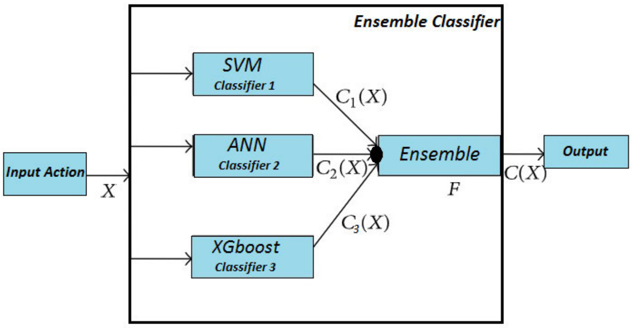

FIGURE 9. Ensemble classifier approach.

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.8, and discussed below:

1) 1st PHASE, BASELINE CLASSIFIERS

The proposed system uses three performance evaluation scenarios namely *ES1, ES2 and ES3* described in *section F*. In each scenario, score of the classifiers, for per action, decides whether it is genuine or belongs to an impostor. In this regard, ensemble learning approach consisting of three classifiers 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. 9

The proposed system employs two types of ensemble rules including dynamic classifier selection(DCS) [38] and weighted classifier fusion(WCF) [39]. DCS reflects the tendency to extract a single best classifier at train-test split for each action which is the most likely to produce the correct classification label for an input sample at validation split. However the WCF relates to approach where all the classifier scores goes to the weighted fusion module, where an output score is a weighted sum of input scores of all the three classifiers as shown in Eq: 1

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

where c_{ti} – input scores, K – amount of classifiers, W_i – input score weights and the value of these weights have been optimized with genetic algorithm [40], $\hat{y}_t(c_t|W)$ – fused score which will be used as a raw confidence score in the second phase for each action.

2) 2nd PHASE, RECURRENT CONFIDENCE FUNCTION

In this research, a novel robust recurrent confidence Model(R-RCM), described in *section C*, 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* or *user specific*. In order to analyse the performance, system has been tested using both global static parameters as well as personalizing the parameter of RCM. These parameters are optimized by employing the genetic algorithm [40] 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.1 are being normalized afterwards to have a weighted average.

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

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

The proposed system methodology has been validated in this work by formulating two experimental settings as shown in Fig.8. 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.

E. PERFORMANCE MEASURE

To evaluate the performance of true CUA system, this research uses the performance metrics as describe by researchers in [25].

- *ANIA*: Average Number of Imposter Actions
- *ANGA*: Average Number of Genuine Actions

However, the system considers the keystrokes as a sequential series, so we take the mean of ANGA and ANIA for each sequence and report the results in terms of Mean ANGA and Mean ANIA over all the testing samples. 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, we define the normalized imposter actions over the total sampling sequence actions A_T as:

$$ANIA = \frac{\sum A_L}{L * A_T}, \tag{2}$$

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.

$$ANGA = \frac{\sum A_L}{L * A_T}, \tag{3}$$

For an efficient CUA system, ANIA should be as low as possible while ANGA should be high. In ideal situation, genuine user should never be locked out by the system and imposter user should be detected as soon as possible but in reality situation may vary. Therefore, the four categories are defined based on ANGA and ANIA for all the system users given as follows: Suppose we have N users, each of N cases is assigned two attributes. The first one indicates whether ANGA $=$ 100% or not. The second one indicates whether ANIA $>$ 40% or not.

- Very Good, ANGA $=$ 100% and ANIA $\leq$ 40%
- Good, ANGA $<$ 100% and ANIA $\leq$ 40%
- Bad, ANGA $=$ 100% and ANIA $>$ 40%
- Ugly, ANGA $<$ 100% and ANIA $>$ 40%

F. EVALUATION SCENARIOS

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 evaluation scenarios namely internal, external and hybrid are designed which are explained below:

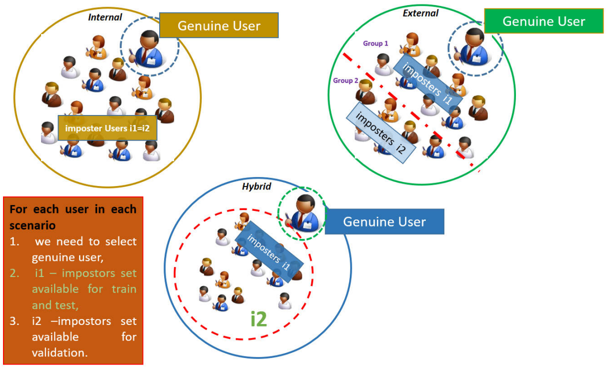
Internal
Genuine User
imposter Users i1=i2
External
Genuine User
Group 1
Group 2
imposters i1
imposters i2
For each user in each scenario
1. we need to select genuine user,
2. i1 – impostors set available for train and test,
3. i2 –impostors set available for validation.
Hybrid
Genuine User
imposters i1
i2

FIGURE 10. Three evaluation scenarios.

Suppose system has been given a set U of $N = |U|$ users and in total each scenario has N cases. For each scenario, firstly system needs to select g – genuine user, I_1 – impostors set available for train and test, I_2 – impostors set available for validation.

- *Internal Scenario*: Each of N users is selected as a genuine user g.The rest users are assigned to $I_1 = I_2 = U \setminus g$ as shown in Fig.10. Accordingly, it is assumed that system has training samples of all the users in the given organization.
- *Hybrid Scenario*: 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.10. It is assumed that rest of the users are added to organization 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 $I_2 = U \setminus g$.
- *External Scenario*: 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 $U' = \{u_0, u_1, \ldots, u_N, u_{N+1}, u_{N+M-N \mod M}\}$ 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.10. In such a case validation set of impostor users doesn't include any of users used during the training and testing at all.

IV. RESULTS AND DISCUSSION

The programming language used throughout this work 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 our experiments will be discussed in this section.

Here, we present some excerpts of our results 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%) and aggregated results are provided in tabular form but here for sake of understanding only some samples of results are shown in order to visualize the user categories.

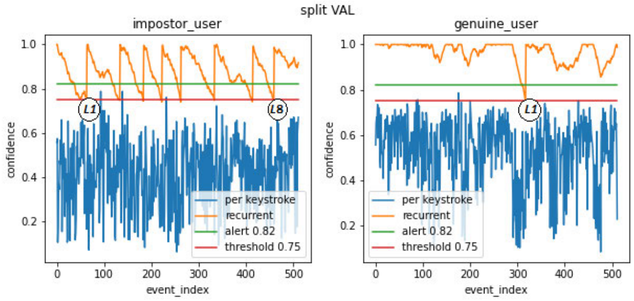

FIGURE 11. Genuine user validated with its own reference set(right) and with imposter set(left).

- *Good*: Fig.11 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

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

Similarly, ANIA can be calculated using Eq. 2

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

In this example, geniune user has been locked out at least once but the given imposter validated against this geniune user's reference sample has been detected before performing 40% of actions hence this geniune 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.

- *Very Good*: Fig.12 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. 2, 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.

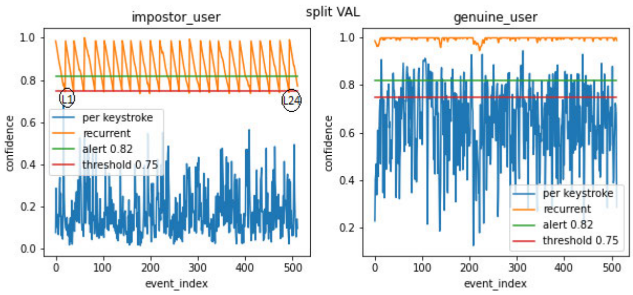

FIGURE 12. Genuine user validated with its own reference set(right) and with imposter set(left).

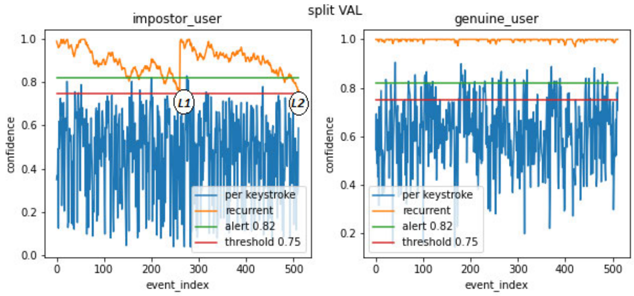

FIGURE 13. Genuine user validated with its own reference set(right) and with imposter set(left).

- *Bad*: Similarly, Fig.13 shows another excerpt of validation which indicates 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 2 times only(L1-L2) in the given sequence sample hence the ANIA of this example, according to Eq. 2, 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.14 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. 2, on average hence ANIA>40%.

Now, the aggregated results for all the users are reported in tabular form for both of experimental settings as below:

A. EXPERIMENTAL SETTING I: DYNAMIC CLASSIFIER SELECTION WITH GLOBAL STATIC RCM

It can be observed from the table.2 that, *For scenario 1,* 95% of 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 95% genuine users has been detected before performing 0.22 or 22% of actions.

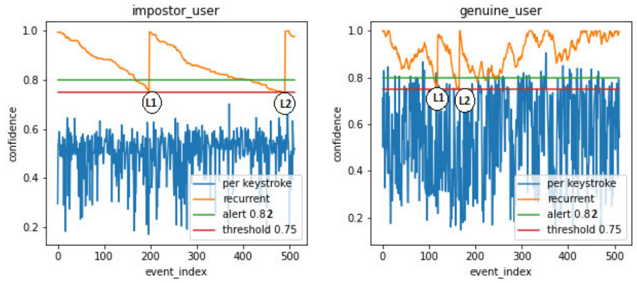

FIGURE 14. Genuine user validated with its own reference set(right) and with imposter set(left).

TABLE 2. Aggregated results of experimental setting I.

Group Name	Scenario	%. of Users	ANGA-Mean	ANIA-Mean
Very-Good	1	0.95	1.00	0.22
Good	1	0		
Bad	1	0.05	1.00	0.41
Ugly	1	0		
Very-Good	2	0.65	1.00	0.27
Good	2	0.15	0.97	0.26
Bad	2	0.20	1.00	0.42
Ugly	2	0		
Very-Good	3	0.65	1.00	0.24
Good	3	0.20	0.96	0.31
Bad	3	0.15	1.00	0.44
Ugly	3	0		

Subsequently, 5% 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 65% users in very good category with ANIA 0.27 (27% actions) which is quite high while 15% users fall in good group where mean of ANGA is 0.97(97% actions) and ANIA is 0.26 (26% actions).And, 20% users fall in bad category with ANIA 0.42(42% actions).

In scenario 3, it can be noticed that 65% users in very good category with mean ANIA 0.24 (24% actions) while 20% users fall in good group where mean of ANGA is 0.96(96% actions) and ANIA is 0.31(31% actions). 15% 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.

B. EXPERIMENTAL SETTING II: WEIGHTED CLASSIFIER FUSION WITH PERSONALIZED RCM

It can be observed from table.3 that: *In scenario 1*, 10% 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 2 genuine users has been detected before performing 0.05% of actions. Subsequently, the 90% users fall in good category with ANGA and ANIA being 0.80 and 0.09 (9% actions) respectively.

TABLE 3. Aggregated results of experimental setting II.

Group Name	Scenario	% of Users	ANGA-Mean	ANIA-Mean
Very-Good	1	0.10	1.00	0.05
Good	1	0.90	0.80	0.09
Bad	1	0		
Ugly	1	0		
Very-Good	2	0.05	1.00	0.28
Good	2	0.95	0.75	0.10
Bad	2	0		
Ugly	2	0		
Very-Good	3	0.30	1.00	0.15
Good	3	0.70	0.72	0.12
Bad	3	0		
Ugly	3	0		

In scenario 2, there are 5% users in very good category with ANIA 0.28 which is quite high as compare to ANIA of scenario 1 while the rest 95% fall in good group where mean of ANGA is 0.75 and ANIA is 0.10.

In scenario 3, it can be noticed that 30% users are falling in very good category with mean ANIA 0.15 which is better than scenario 2 while the rest 70% 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.

1) ANALYSIS FOR SETTING I AND SETTING II

We are referring to the aggregated results of DCS with static RCM parameters (setting I) and weighted fusion with personalized parameters optimized with genetic algorithm (setting II) in table 2 and III respectively. 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 personalized 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. If we see 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 optimized by genetic algorithm as well as weighted classifier fusion approach.

More specifically, the system's ANIA can be calculated with the following equation:

$$SystemANIA = \frac{\sum(MeanANIA * User\%)}{TotalUsers\%}, \tag{4}$$

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:

- *Experimental Setting I*
  $System\ ANIA = \frac{(0.22*0.95)+(0.41*0.05)}{1.00} = 0.23 \text{ or } 23\%$
- *Experimental Setting II*
  $System\ ANIA = \frac{(0.05*0.10)+(0.09*0.90)}{1.00} = 0.09 \text{ or } 9\%$

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

V. CONCLUSION

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 mean ANGA and ANIA.

In this research, the robust recurrent confidence model(R-RCM) has been implemented which tends to lock out the imposter user as quickly as possible if it crosses the alert threshold. On the same hand, it keeps in account the fact that sometimes even genuine user deviates from normal behaviour owing to the background context and crosses the alert threshold. In this case, R-RCM increases the genuine user's confidence gradually and does not trust the user fully until its confidence level again goes up from the alert threshold and reaches the safe zone.

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, both proposed experimental settings, using the novel approach of R-RCM with alert threshold, have detected the imposter users faster and achieved the lowest mean ANIA as compared to previous scholarly works done in domain of true continuous user authentication. Additionally, experimental setting II has achieved the lowest system's ANIA and detected the imposter user as soon as it crosses the alert threshold.

REFERENCES

[1] I. Bhardwaj, N. D. Londhe, and S. K. Kopparapu, "Study of imposter attacks on novel fingerprint dynamics based verification system," *IEEE Access*, vol. 5, pp. 595–606, 2017.

[2] T. Dee, I. Richardson, and A. Tyagi, "Continuous transparent mobile device touchscreen soft keyboard biometric authentication," in *Proc. 32nd Int. Conf. VLSI Design, 18th Int. Conf. Embedded Syst. (VLSID)*, Jan. 2019, pp. 539–540.

[3] S.-S. Shen, T.-H. Kang, S.-H. Lin, and W. Chien, "Random graphic user password authentication scheme in mobile devices," in *Proc. Int. Conf. Appl. Syst. Innov. (ICASI)*, May 2017, pp. 1251–1254.

[4] S. Alotaibi, A. Alruban, S. Furnell, and N. Clarke, "A novel behaviour profiling approach to continuous authentication for mobile applications," in *Proc. 5th Int. Conf. Inf. Syst. Secur. Privacy*, 2019, pp. 1–6.

[5] M. L. Ali, J. V. Monaco, C. C. Tappert, and M. Qiu, "Keystroke biometric systems for user authentication," *J. Signal Process. Syst.*, vol. 86, nos. 2–3, pp. 175–190, Mar. 2017.

[6] P. Bours and H. Barghouthi, "Continuous authentication using biometric keystroke dynamics," in *Proc. Norwegian Inf. Secur. Conf. (NISK)*, vol. 2009, 2009, pp. 1–12.

[7] A. A. Vyazigin, N. Y. Tupikina, and E. V. Sypin, "Software tool for determining of the keystroke dynamics parameters of personal computer user," in *Proc. 20th Int. Conf. Young Spec. Micro/Nanotechnol. Electron Devices (EDM)*, Jun. 2019, pp. 166–171.

[8] S. Shepherd, "Continuous authentication by analysis of keyboard typing characteristics," in *Proc. Eur. Conv. Secur. Detection*, 1995, pp. 111–114.

[9] A. Mhenni, E. Cherrier, C. Rosenberger, and N. E. B. Amara, "Double serial adaptation mechanism for keystroke dynamics authentication based on a single password," *Comput. Secur.*, vol. 83, pp. 151–166, Jun. 2019.

[10] A. Alsultan, K. Warwick, and H. Wei, "Free-text keystroke dynamics authentication for arabic language," *IET Biometrics*, vol. 5, no. 3, pp. 164–169, Sep. 2016.

[11] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello, "Continuous user authentication on mobile devices: Recent progress and remaining challenges," *IEEE Signal Process. Mag.*, vol. 33, no. 4, pp. 49–61, Jul. 2016.

[12] A. Foresi and R. Samavi, "User authentication using keystroke dynamics via crowdsourcing," in *Proc. 17th Int. Conf. Privacy, Secur. Trust (PST)*, Aug. 2019, pp. 1–3.

[13] F. D. Tommaso, M. Guerra, F. Martinelli, F. Mercaldo, M. Piedimonte, G. Rosa, and A. Santone, "User authentication through keystroke dynamics by means of model checking: A proposal," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 6232–6234.

[14] K. Senathipathi and K. Batri, "An analysis of particle swarm optimization and genetic algorithm with respect to keystroke dynamics," in *Proc. Int. Conf. Green Comput. Commun. Electr. Eng. (ICGCCEE)*, Mar. 2014, pp. 1–11.

[15] M. Curtin, C. Tappert, M. Villani, G. Ngo, J. Simone, H. S. Fort, and S. Cha, "Keystroke biometric recognition on long-text input: A feasibility study," in *Proc. Int. MultiConf. Eng. Comput. Sci. (IMECS)*, 2006, pp. 1–5.

[16] A. Salem and M. S. Obaidat, "A novel security scheme for behavioral authentication systems based on keystroke dynamics," *Secur. Privacy*, vol. 2, no. 2, p. e64, Mar. 2019.

[17] D. Gunetti, C. Picardi, and G. Ruffo, "Dealing with different languages and old profiles in keystroke analysis of free text," in *Proc. Congr. Italian Assoc. Artif. Intell.* Berlin, Germany: Springer, 2005, pp. 347–358.

[18] J. Huang, D. Hou, and S. Schuckers, "A practical evaluation of free-text keystroke dynamics," in *Proc. IEEE Int. Conf. Identity, Secur. Behav. Anal. (ISBA)*, Feb. 2017, pp. 1–8.

[19] A. Kolakowska, "User authentication based on keystroke dynamics analysis," in *Computer Recognition Systems*. Berlin, Germany: Springer, 2011, pp. 667–675.

[20] P. Pinto, B. Patrão, and H. Santos, "Free typed text using keystroke dynamics for continuous authentication," in *Proc. IFIP Int. Conf. Commun. Multimedia Secur.* Berlin, Germany: Springer, 2014, pp. 33–45.

[21] B. Ayotte, J. Huang, M. K. Banavar, D. Hou, and S. Schuckers, "Fast continuous user authentication using distance metric fusion of free-text keystroke data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1–9.

[22] J. Ferreira and H. Santos, "Keystroke dynamics for continuous access control enforcement," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery*, Oct. 2012, pp. 216–223.

[23] C. Ferrari, D. Marini, and M. Moro, "An adaptive typing biometric system with varying users model," in *Proc. 32nd Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, May 2018, pp. 564–568.

[24] H. Lee, J. Y. Hwang, D. I. Kim, S. Lee, S.-H. Lee, and J. S. Shin, "Understanding keystroke dynamics for smartphone users authentication and keystroke dynamics on smartphones built-in motion sensors," *Secur. Commun. Netw.*, vol. 2018, pp. 1–10, Mar. 2018.

[25] P. Bours and S. Mondal, "Continuous authentication with keystroke dynamics," in *Recent Advances in User Authentication Using Keystroke Dynamics Biometrics-GCSR*, vol. 2, 2015, pp. 41–58.

[26] D. R. Chandranegara, H. Wibowo, and A. E. Minarno, "Combined scaled manhattan distance and mean of horner's rules for keystroke dynamic authentication," *Telkomnika*, vol. 18, no. 2, pp. 770–775, 2020.

[27] D. Escobar-Grisales, J. Vásquez-Correa, J. F. Vargas-Bonilla, and J. R. Orozco-Arroyave, "Identity verification in virtual education using biometric analysis based on keystroke dynamics," *TecnoLógicas*, vol. 23, no. 47, pp. 193–207, 2020.

[28] V. V. Phoha, S. Phoha, A. Ray, S. S. Joshi, and S. K. Vuyyuru, "Hidden Markov model ('HMM')-based user authentication using keystroke dynamics," U.S. Patent 8 136 154, Mar. 13, 2012.

[29] A. Ceffer and J. Levendovszky, "Kolmogorov-smirnov test for keystroke dynamics based user authentication," in *Proc. IEEE 17th Int. Symp. Comput. Intell. Informat. (CINTI)*, Nov. 2016, pp. 105–110.

[30] C. Chantan, S. Sinthupinyo, and T. Rungkasiri, "Comparing structure learning algorithms of Bayesian network in authentication via short free text," *Int. J. Comput. Appl.*, vol. 46, no. 3, pp. 19–24, 2012.

[31] A. A. Ahmed and I. Traore, "Biometric recognition based on free-text keystroke dynamics," *IEEE Trans. Cybern.*, vol. 44, no. 4, pp. 458–472, Apr. 2014.

[32] A. Alsultan, K. Warwick, and H. Wei, "Non-conventional keystroke dynamics for user authentication," *Pattern Recognit. Lett.*, vol. 89, pp. 53–59, Apr. 2017.

[33] P.-Y. Wu, C.-C. Fang, J. M. Chang, and S.-Y. Kung, "Cost-effective kernel ridge regression implementation for keystroke-based active authentication system," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3916–3927, Nov. 2017.

[34] R. Giot, M. El-Abed, B. Hemery, and C. Rosenberger, "Unconstrained keystroke dynamics authentication with shared secret," *Comput. Secur.*, vol. 30, nos. 6–7, pp. 427–445, Sep. 2011.

[35] N. Bakelman, J. V. Monaco, S.-H. Cha, and C. C. Tappert, "Continual keystroke biometric authentication on short bursts of keyboard input," Student-Fac. Res. Day, CSIS, Pace Univ., New York, NY, USA, Tech. Rep., 2012.

[36] Y. Sun, H. Ceker, and S. Upadhyaya, "Shared keystroke dataset for continuous authentication," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2016, pp. 1–6.

[37] I. Deutschmann, P. Nordstrom, and L. Nilsson, "Continuous authentication using behavioral biometrics," *IT Prof.*, vol. 15, no. 4, pp. 12–15, Jul. 2013.

[38] I. Mendialdua, J. M. Martínez-Otzeta, I. Rodriguez-Rodriguez, T. Ruiz-Vazquez, and B. Sierra, "Dynamic selection of the best base classifier in one versus one," *Knowl.-Based Syst.*, vol. 85, pp. 298–306, Sep. 2015.

[39] A. Mi, L. Wang, and J. Qi, "A multiple classifier fusion algorithm using weighted decision templates," *Sci. Program.*, vol. 2016, pp. 1–10, Jan. 2016.

[40] D. S. Weile and E. Michielssen, "Genetic algorithm optimization applied to electromagnetics: A review," *IEEE Trans. Antennas Propag.*, vol. 45, no. 3, pp. 343–353, Mar. 1997.

ANUM TANVEER KIYANI received the M.Sc. degree in network security and penetration testing from Middlesex University, London, U.K, where she is currently pursuing the Ph.D. degree with the Faculty of Science and Technology. Her research interests include biometric security, artificial intelligence, behavioural analysis, human–computer interaction, and cyber security.

ABOUBAKER LASEBAE is currently an Associate Professor and a Computer Communication and Networks Director of programmes. He is also leading contact in 5G and managing Huawei section with Middlesex University. He has published four books and published several journal and conference papers in the areas of computer networks, wireless networks, telecommunications, mobile communications, network security, cyber security, and performance analysis.

KAMRAN ALI (Member, IEEE) received the Ph.D. degree in disaster communication architecture funded project from Newton Fund/British Council Institute. He is pursuing his career in teaching and research in U.K. and Pakistan. He is currently with the Department of Computer Science, Middlesex University, London, U.K. His current research interests include D2D communication, wireless co-operative networks, disaster management systems, cluster and cloud computing. He is a Fellow of the Higher Education Academy (U.K.) and part of the technical program committees and organizing committees of several international conferences and journals.

MASOOD UR REHMAN received the M.Sc. and Ph.D. degrees in electronic engineering from the Queen Mary University of London, U.K. He is currently working as an Assistant Professor with the James Watt School of Engineering, University of Glasgow. He has contributed to a patent and authored/coauthored four books, seven book chapters, and more than 105 technical papers in leading journals and peer-reviewed conferences. His research interests include compact antenna design, radiowave propagation, channel characterization, and satellite navigation system antennas in cluttered environment.

BUSHRA HAQ received the M.S. degree from the Baluchistan University of Information Technology, Engineering, and Management Sciences (BUITEMS), where she is currently pursuing the Ph.D. degree. Since 2014, she has been with BUITEMS. Her research interests include machine learning, deep learning, software engineering, cloud computing, and so on.