Enhanced Authentication System Performance Based on Keystroke Dynamics using Classification algorithms

Asma Salem[1*], Ahmad Sharieh[1], Azzam Sleit[1] and Riad Jabri[1]
[1] Computer Science Department, KASIT, University of Jordan
Amman - Jordan
[e-mail: Asma_salem85@yahoo.com, sharieh@ju.edu.jo ;azzam.sleit@ju.edu.jo; jabri@ju.edu.jo]
*Corresponding author: Asma M. Salem

Received May 12, 2018; revised December 6, 2018; accepted February 14, 2019; published August 31, 2019

Abstract

Nowadays, most users access internet through mobile applications. The common way to authenticate users through websites forms is using passwords; while they are efficient procedures, they are subject to guessed or forgotten and many other problems. Additional multi modal authentication procedures are needed to improve the security. Behavioral authentication is a way to authenticate people based on their typing behavior. It is used as a second factor authentication technique beside the passwords that will strength the authentication effectively. Keystroke dynamic rhythm is one of these behavioral authentication methods. Keystroke dynamics relies on a combination of features that are extracted and processed from typing behavior of users on the touched screen and smart mobile users. This Research presents a novel analysis in the keystroke dynamic authentication field using two features categories: timing and no timing combined features. The proposed model achieved lower error rate of false acceptance rate with 0.1%, false rejection rate with 0.8%, and equal error rate with 0.45%. A comparison in the performance measures is also given for multiple datasets collected in purpose to this research.

Keywords: Security, biometric authentication, keystroke dynamics, behavioral authentication.

1. Introduction

Nowadays, mobile technologies are developing in fast manners. The mobiles continue to suffer from high security risks. These days most internet activities are being done through mobile devices. It is well known that the common way to authenticate users is using passwords [1-3]. While the techniques using password is not enough since it is subject to be guessed, stolen or forgotten and many other problems. Therefore, we are in a need to introduce an additional authentication technique that could strengthen the password [1][2].

Multiple techniques were provided in the literature; second factor authentication is already applied and used to strengthen the authentication mechanism and enhance the security. It is commonly known that authentication using biometric is expensive to establish; as we need extra hardware and inapplicability through internet. This made the behavioral based systems one of these cheap and easy to establish this mechanism through internet [3][4].

Keystroke dynamics (KSD) is one of these behavioral based systems. In which it is stated that the typing behavioral of users is should be unique. Is it possible to establish authentication systems that relying on the typing behavior of their users.

This study introduces the use of KSD on android touch screen mobile devices. We have proposed a programmed prototype for a software keyboard application, which developed especially for collecting timing and non-timing information. This prototype enables users to type freely any designated text consisted of any complex combination, which consists of text, numbers, and special characters. Our proposed method collecting a combination of various features; they are mainly divided in to two main categories: timing and non-timing featured. Using neural networking and machine learning models, we successfully classified users based on their typing. Different experiments will be implemented using timing and non-timing features. Proofing that the KSD is providing a significant role in authenticating user's performance based on their typing behavior.

This paper is ordered as the following, Section two will highlight the background. Techniques and methods will be proposed in section three. Experiments and results will be discussed in Section four and conclusion will be proposed in Section five.

2. Related Work

Authentication is simply the process of determining whether someone is the one who declared to be for the system [3][4]. User authentication could be achieved by using many techniques; something that the user knows (e.g., password), carries (e.g., credit card), or has (e.g., iris); and recently, a new mechanism for user authentication is based on how he/she behaves (e.g., typing behavior) [5].

Multiple features could be extracted from tying behavior. Timing and non-timing features are known in these systems. These features are characterized by up and down events. The Events recording in which when and how each key on the keyboard is pressed or released. Collecting features during typing could be statically collected or dynamic. Static collection

could be done by storing predefined profile of typing in a database that could be used later in comparisons between the predefined typing text and the stored profile. While dynamic one is continuous authentication is dynamically setting the profile of users during their typing [5][6].

Multiple methods and techniques were provided in the literature to classify users typing behavior; statistical mechanisms were used early to classify users. Using the mathematical equations is somehow good but not sufficient. Neural networks and machine learning techniques were also proposed. The better method was the less error that gotten from the system [6][7].

2.1 KSD Features

Typing behavior is about collecting typing user behavior by extracting features during user typing on keyboards. These features are divided into two main categories timing and non-timing features [4-13].

Timing Features: The events time in which user press or release the key on the keyboards. They can be extracted with a special timer on keyboard, which pick up when a key is pressed and/or released [8][9]. There are two main event times: pressing time which is the time stamp recorded when the key is held down (D) and releasing time: which is the timestamp recorded when the key is released up (U). So, the timing features are extracted by capturing all time stamps of each event. These events could be consecutives of two or more events. The simplest two consecutive events could be one of the following combinations (Down-Up, Up-Up, Down-Down, and Up-Down) [4-9].

Non-Timing Features: These are related for and other non-related timing features; such as finger pressure, finger position on the screen, as well as the size of key surface touched by finger, typing speed, …, etc. For every key pressed or released and based on touch screen properties, we can extract many other features such as finger placement angle, finger identifier (which finger used to type), …, etc. [9-13].

2.2 Performance measures

Like any authentication system, we need a performance measures to evaluate and compare any two or more systems. The classification capabilities could be compared based on performance of the system each time the user access to the system. Based on the literature we have the following measures: -

- False Rejection Rate (FRR) describes the percentage of true users of the system, which is rejected by wrong classification of their features. This measure is known in statistics as Type I error [6][8].
- False Acceptance Rate (FAR) describes the percentage of false users of the system that is accepted by wrong classification of their features. This measure is known in statistics as Type II error [2][8].
- Equal Error Rate (EER) is the most used performance measure of biometric systems in the literature is the EER. It describes the state of the system where percentage of true users of the system who are rejected by wrong classification of their features (FRR), and the percentage of false users of the system who are accepted by wrong classification of their features (FAR) are being equal. As the all describing an error rate the lower of these values the better performance of the system [3-11].

2.3 Benchmarking, datasets, and challenges

A number of studies have been performed in the area of keystrokes analysis since it is founded on the field of biometric based systems [10-15]. Many authors proposing good contribution trials for existing unique datasets benchmarks [14][15]. It was difficult to compare the performance among different researches based on KSD, so different public datasets were published in this field to provide one unique evaluation research based on one reference. These datasets can be used by the community to evaluate researches results and asses in comparative analysis [16][17]. The first public dataset benchmark was coined by the Carnegie Mellon School (CMS) in 2009. It is well known by CMU-Keystroke Dynamics dataset, since they introduced their dataset for timing information only for personal computers and regular keyboards. Much other non-timing information was not included [18].

KSD used in the literature for more than twenty years [19]; and their analysis shown a promising performance with novel results. For example, the authors in [19] got 0% errors in authentication systems by using timing information only. Their datasets were built by their own and based on KSD derived from regular keyboards in personal computers. A dedicated dataset for mobile and touch screen devices are still few in literature and they were not covering all the provided non-timing features we mentioned in the literature. So, we are in bad needs for public touch screen devices benchmarks [20-28].

2.4 Typing rhythm mode

Based on the literature, the various researches were being classified upon the typing rhythm test mode; static and dynamic.

Static text: in which the user typing is being predefined during typing trials. This method is early determined a predefined text that the user asked to type multiple times [3]. Changing the text each time implies repeating enrollment process for the user to the system. Although this procedure was a robust technique in most researches, the continues security is not provided here. Any change in the text typed by a user, the overall procedure for building the user profile will be repeated from scratch [4].

Dynamic text: these dynamic modes are situations that are more real, as the user was being freely typing on the keyboard. This text mode verification sometimes called a continuous authentication in literature as the systems can classify the behavior of a user, whatever the text was being typed [1-8].

2.5 KSD Analysis Approaches

KSD analysis approaches can be based on the extracted features. These features are categorized into timing and non-timing categories. Behavioral authentication features can be extracted, preprocessed, and measured in multiple experimental methods, such as using the statistical techniques and could be carried on static text and dynamic text. Other techniques are the neural networks (NNs), which could be also carried on static text as well as dynamic text. Analyzing these features in extensive and efficient manner will maximize soon the performance of authentication in computerized systems [6].

In **Table 1**, we have summed up the latest researches done in the literature. Each research chosen were being done using mobile devices with touch screen and Android operating system based. The classification was being divided upon multiple criteria's such as the features were being used, the text mode and the classification techniques. The reported results of the

performance measures use the measure (EER), which was the common measure between all reported results.

Table 1. Performance comparisons between recent studies.

Re.	Features extracted	Classification Technique and text mode	EER %
[5]	Timing: Time Non-Timing: Pressure, Size	Neural Networks (Static text mode)	2.3%
[3]	Timing: Time Non-Timing: Pressure, Size, XY position	Neural Networks (Static text mode)	2.8%
[7]	Timing: Time Non-Timing: Pressure, Size, angle	Neural Networks (Dynamic text mode)	3.65%
[13]	Timing: Time Non-Timing: Pressure, Size, XY position	Neural Networks (Static text mode)	5.43%
[6]	Timing: Time Non-Timing: Pressure, Size, XY position	Neural Networks (Dynamic text mode)	8.10%
[2]	Timing: Time Non-Timing: Pressure, Size	Statistical (Dynamic text mode)	10%
[1]	Timing: Time Non-Timing: Pressure, Size	Statistical (Dynamic text mode)	12.2%
[10]	Timing: Time Non-Timing: Speed, Distance	Neural Networks (Static text mode)	13.6%

3. KSD Methods and techniques

3.1 Authentication phases Implementation

To authenticate users using KSD on mobile phones, we have implemented our procedure into two important phases [6-9]. The first phase, called the enrollment; which sometimes called the profile building in the literature. The typing rhythm was being collected in many trials to choose the most similar profile for the user typing behavior. In this phase, the user asked to enter many trials (i.e 10 times) and the average of the data is collected and stored in the database as user signature. In the enrollment phase, it is good to note that many researches were excluded the first two trials for typing, as the users want some time to feel familiar with the designated keyboard. Outlier's signature resulted from varied typing behavior for the user at each trial. These outliers are most likely to appear at the beginning of typing trials.

The second phase is the actual authentication process. Here, the user asked again to type his cadential to be compared and matched with the stored one. At each time the user authenticated, he/she asked to enroll few times to the systems to be compared with the stored profile in the

database. Enrollment and Authentication phases are described in **Fig. 1** and **Fig. 2** respectively [11][12].

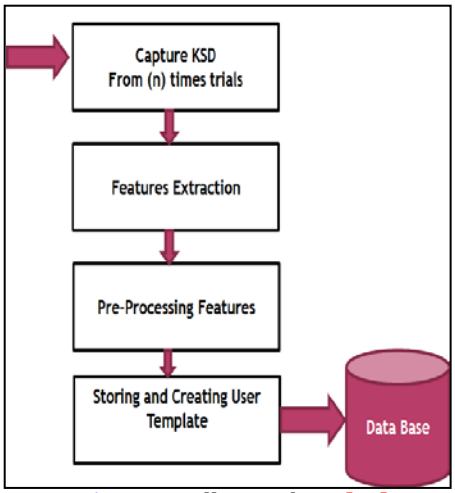

Fig. 1. Enrollment phase [13]

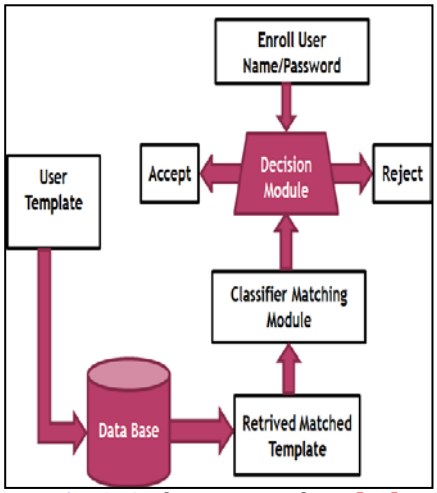

Fig. 2. Authentication phase [13]

3.2 System design interface

To provide our KSD based system, we have developed a software virtual keyboard, that being installed easily on each device used to authenticate users. This keyboard was dedicated for all Android platform mobile devices with touch screen. All the Experiments in this research were performed using Sony Xperia tablet Z, and with Android version (5.1.1).

Fig. 3 describes the developed interface for the application and the soft keyboard. The keyboard enable users to type text (A-Z) (a-z), numbers (0-9), and special characters such as (!@#$%^&*) and others [13].

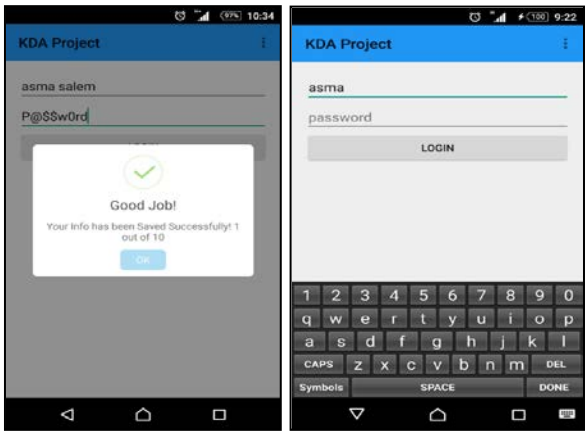

Fig. 3. KSD application interface [13]

In **Table 2,** we have summed up the main Android call functions that used in developing the soft keyboard. Each call was dedicated to extract one feature at each Key Up and Key Down events.

Table 2. Function calls events for Android System [13]

Feature in Android function call	Descriptionm
getEventTime(),	Extract time in millisecond (***msec)*** at each UP/Down event.
getPressure()	Extract pressure level for finger when pressed or released the key on the touch screen, It was varied between (0-1) for Sony Xperia tablet Z device.
getSize(),	Extract Size of finger on the Key surface when pressed or released the key on the touch screen, It was varied between (0-1) for Sony Xperia tablet Z device.
getX(), getY().	Extract the X and Y coordination for the key when it is pressed or released.

3.3 Neural network classifier (NN)

Multilayer Perceptron (MLP) is one of the chosen classifiers that being used in this research. A feed forward neural network classifier maps sets of input data onto a set of outputs. The structure of MLP consists of multiple layers and each layer consisted of multiple nodes [13][22]. MLP has main features such as:

a. Layers

The MLP consists of three or more layers. MLP usually used as a fully connected network (mesh). Each node, applying some weight on the input and propagate the result to the next layer [13]. The number of nodes should be determined each time we carried out the experiment. The notation of MLP classifier (x,y,z) represents the number of neurons in each layer with three layers structure. For example, if we used the notation MLP (10, 10, 20) to represent MLP structure; then we have three layer with 10, 10, and 20 nodes in each layer respectively.

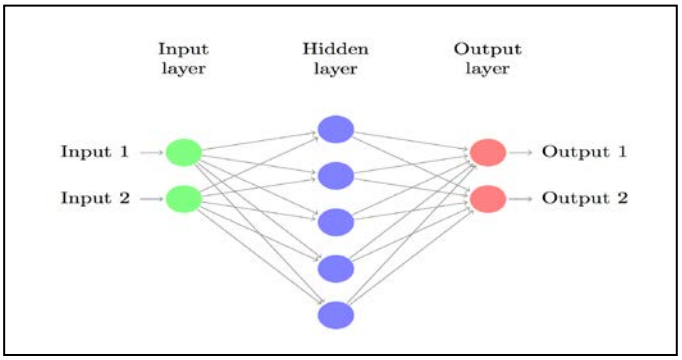

Fig. 4. MLP classifier architecture [24]

Fig. 4 shows MLP simplest architecture with one hidden layer. This simplest architecture was used in this paper. The left side represents the input layers where the features are inserted to the classifier, the middle part is the hidden layer and the right side represents the output of the classifier. In this work, the architecture was extracted from WEKA tool. We have used it in this paper.

b. Weights

Weights are being configured initially in random way, then, they should be adapted using the training set that flows from input layers to the output layer in the network. They continued to be adapted until we reach a specific error threshold. There are many other parameters such as

gradient, momentum (value should be between 0 and 1, while the default value always = 0.2), and learning rate (value should be between 0 and 1, while the default value = 0.3) they are affecting the behavior of MLP classifier [22]. **Fig. 5** describes the flow of the MLP algorithm in Pseudocode.

```
Input: The Features vector for each user

    Start with random initial weights (i.e. , uniform random in [-3,+3])
    Do
    {
    For all patterns P
      {
        For All output Nodes j
        {
          Calculate activation (j)
          Error_j= Target value_j_for_Pattern_p= Activation_j
          For  all input Nodes i to output node j
           {
             Delta_weight = learning constant = Error_j* Activation_i
             Weight= weight * Delta_weight
           }
        }
      }
    }
     Until Error is sufficiently small or "Time_out"
Output : The User ID identification results.
```

Fig. 5. MLP classifier Pseudo code [25]

3.4 Decision Trees

Decision Trees and Random Trees providing a simple and fast way of learning a function that maps data x to outputs y, where x can be a mix of categorical and numeric variables and y can be categorical for classification, or numeric for regression. A Decision Tree is a tree (and a type of directed, acyclic graph) in which the nodes represent decisions (a square box), random transitions (a circular box) or terminal nodes, and the edges or branches are binary (yes/no, true/false) representing possible paths from one node to another [28-36]. The specific type of decision tree used for machine learning contains no random transitions. To use a decision tree for classification or regression, one grabs a row of data or a set of features and starts at the root, and then through each subsequent decision node to the terminal node. The process is very intuitive and easy to interpret, which allows trained decision trees to be used for variable selection or more generally, feature engineering [28-36]. **Fig. 6** describes the simple random forest (RF) architecture. **Fig. 7** describes the pseudocode of the classification algorithm.

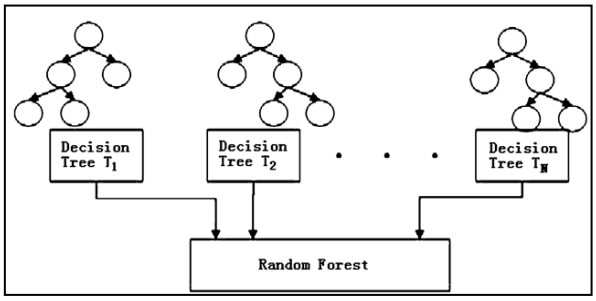

Fig. 6. RF classifier architecture [26]

Table 3. Trees Classifier important parameters [28]

Feature	Description
batchSize	The preferred number of instances to process.
maxDepth	The maximum depth of the tree, 0 for unlimited.
seed	The random number seed used for selecting attributes.
breakTiesRandomly	Break ties randomly when several attributes look equally good.
numFolds	Determines the amount of data used for backfitting.
minNum	The minimum total weight of the instances in a leaf.
numTrees	The number of trees to be generated.
numFeatures	The number of attributes to be used in random selection.

Random forests (RF): it is the generalized form of random tree or decision tree. It is known from the literature that it is more efficient than the random trees. At each candidate tree, it is doing a split in the learning process, with a random subset of the features. This process sometimes is called "feature bagging". If one or a few features are very strong predictors in the classification decision for the target output, these features will be selected in many of the generated trees, causing them to become correlated and dominated in the forest domain. The stronger features are, the less error for classification results will be gotten. Main features are described briefly in **Table 3** [28-30]

Input: The Features vector for each user
Do
Procedure Random Forest in Pseudocode
1: For 1 to T do
2: Draw n points Di with replacement from D
3: Build full decision/regression tree on Di
4: But : each split only consider k features, picked uniformly at random
New features for every split
5: Prune tree to minimize out-of-bag error
6: End for
7: Average all T trees
8: end procedure
Output : The User ID identification results.

Fig. 7. RF classifier Pseudocode [27]

4. Experimental Classification Results and Analysis

These analyses examined the KSD use as a behavioral authentication in multiple experiments. The study focused on enhancing security level and strengthening access control using artificial neural networking model based on MLP, random tree and random forest classifiers. The proposed work was trained and tested using real dataset samples we have collected and stored in several datasets [13].

4.1 Experiment one

The main goal for this experiment is to examine the training factor effect of the user on his/her typing behavior performance measures. We highlighted this by finding out the relationship between the learning rate of the model and the training factor improvement of the user (this would be implemented in the experiment by the trials numbers). The classification capability of the model is a good indication of the model in this experiment to reach the desired goal.

Experiment setup details are described in **Table 4**. We have conducted the experiments twice. Each time, we have changed the trial numbers of typing for volunteers. The first session was conducted with 10 trials and the second one was with 30 trials.

Table 4. Experiment 1 setup

Experiment Requirements	Description
Number of volunteers and samples	Five users provided 50 and 150 samples respectively.
Password template	"P@ssw0rd", static template.
Device	Sony Xperia tablet Z, with Android version 5.1.1. It was used for training and testing.
Local machine specification	Toshiba Laptop with core i7 processor and 6 GB RAM.
Toolkit for NN	WEKA 3.6 windows version.
WEKA-MLP parameters tuning	- Learning rate = 0.3 , Momentum = 0.2 , Number of Hidden layers =1 - Total number of layers =3 , with MLP (10,10,20) - Test Option =Cross-Validation, Folds =10.

Table 5 provides KSD performance measures in term of error ratios such as FRR and FAR. Comparative analyses of performance measures for experiment in two sessions are provided. **Fig. 8** clearly describes that the training factor has a significant role in the classification capability of the model. It is clear that the classification with respect of True Positive (TP) is improved in the second session and the error rate measured in FAR, FRR and EER. EER is decreased by increasing the training factor for each user.

Table 5. Experiment 1 result using our random datasets

Sessions	TP	FAR	FRR	EER
Session 1-10 trials (Exp-10)	85.5%	3.60%	14.56%	9.1%
Session 2-30 trials (Exp-30)	91.3%	2.2%	8.68%	5.44%

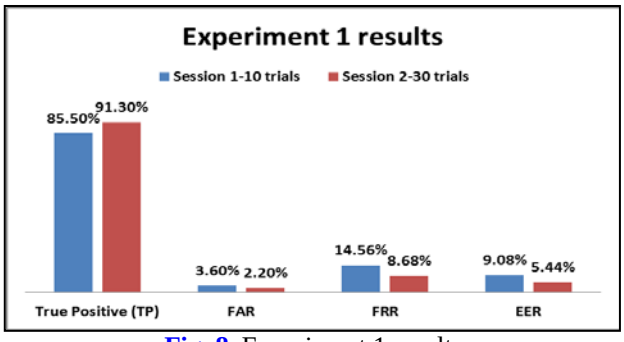

Fig. 8. Experiment 1 results

4.2 Experiment two

In this experiment, seven volunteers were asked to type multiple complex passwords. We have improved user's behavioral performance of typing on KSD keyboard more and more; by providing multiple complex passwords' templates. We used complex passwords combinations in our work to reflect the nature of typing real password in a public application. Users sometimes feeling confused to write their real passwords for testing experimental work. We

have many users didn't accept typing their real passwords even for research use, while others were being comfortable to share some of their password's combinations to public. Many applications on the internet forced users to use complex passwords combinations. These combinations were provided with passwords templates as shown in **Table 6**. Note that we have used the same experiment 1 setup for this experiment also.

Table 6. Experiment 2 setup

Experiment Requirements	Description
Number of volunteers and samples	Seven users
Password template 10 different templates	Predefined complex templates : P@ssw0rd -A$m@1234- K@LK0t@@ ASMA1234- N@d!A963 -AhL@m123 M0HH@M@D-B@s!m@12- N@b!L123-R!d@B456
Number of trials	(6-9) trials for predefined templates

In this experiment, the MLP classifier reached a significant performance levels in term of (EER). This is clearly described in **Table 7** for higher trials provided by the users. See **Fig. 9** which describes experiment 2 results.

Table 7. Performance measures for experiment 2

Sessions	TP	FAR	FRR	EER
Session 3-6 trials (Exp-6)	87.1%	2.1%	12.85%	**7.48%**
Session 4-9 trials (Exp-9)	97.8%	0.4%	2.2%	**1.3%**

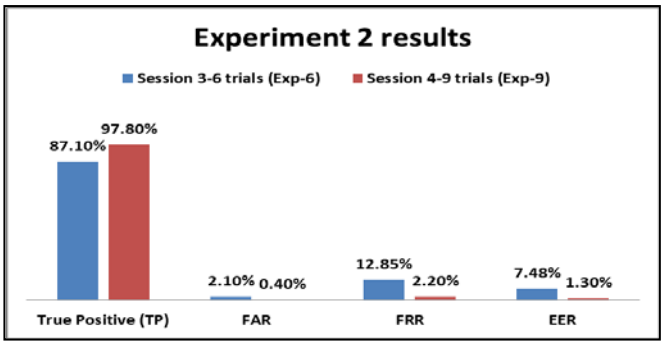

Fig. 9. Experiment 2 results

4.3 Experiment three

In this experiment, we have raised the training factor of the MLP classifier for comparative analysis manners. Although, this will increase the cost of the implementation of the classification, we still have the chance to propose the model implementation with minimum typing effort provided from the volunteer's side. They have really feeling annoyed to type many times of the same template more than thirty times. We have deduced the analysis to do some tradeoff system implementation issue. The new learning factor is increased to be (0.5) and the momentum is (0.4). The reduction in error rate was noticed in the new obtained results. The reduction in experiment 1 is highlighted in **Table 8** and **Table 9**.

Table 8. Performance measures comparisons for experiment 1

Sessions	TP	FAR	FRR	EER
Session 1-10 trials (Learning rate 0.3 , Momentum 0.2)	85.5%	3.60%	14.56%	**9.1%**
Session 1-10 trials (Learning rate 0.5, Momentum 0.4)	89.3%	2.7%	10.7%	**6.7%**

The second experiment was also being deduced using the new learning rate, and the error rate was reduced as highlighted in **Table 9**.

Table 9. Performance measures comparisons for experiment 2

Sessions	TP	FAR	FRR	EER
Session 3-6 trials (Learning rate 0.3, Momentum 0.2)	87.1%	2.1%	12.9%	**7.5%**
Session 3-6 trials (Learning rate 0.5, Momentum 0.4)	89%	1.8%	11%	**6.4%**

4.4 Experiment four

In this experiment, we used the tree classifiers for comparative analysis manners. W deduced the analysis to do some performance improvements. The implementation carried out according to specific configurations summed up in **Table 10**. The result of experiment 4 is highlighted in **Table 11**.

Table 10. Experiment 4 setup

Experiment Requirements	Description
Number of volunteers and samples	Seven users
Password template 10 different templates	Predefined complex templates :- P@ssw0rd -A$m@1234- K@LK0t@@- ASMA1234 N@d!A963 -AhL@m123 M0HH@M@D- B@s!m@12 N@b!L123 -R!d@B456
Number of trials	(6-9) trials for predefined templates.
WEKA- classifiers RF , RT parameters tuning	Batch size =100 Max depth unlimited Seed =1

The fourth experiment was also being deduced using the new classifier and the error measures were reduced to significant rates as highlighted in **Table 11**.

Table 11. Performance measures for experiment 4

Random Tree classifier				
Experiment	**TP**	**FAR**	**FRR**	**EER**
Session 3-6 trials (Exp-6)	90.2%	1.6%	9.8%	**5.7 %**
Session 4-9 trials (Exp-9)	97.1%	0.5%	2.9%	**1.7%**

Random Forest classifier				
Experiment	**TP**	**FAR**	**FRR**	**EER**
Session 3-6 trials (Exp-6)	96.2%	0.6%	3.8%	**2.2 %**
Session 4-9 trials (Exp-9)	99.2%	0.1%	0.8%	**0.45%**

This paper presents a novel technique for authenticating users using KSD based systems. Combining timing and non-timing features together, using several non-timing features. The proposed model (RF) achieved lower error rate of false acceptance with 0.1%, false rejection with 0.8%, and equal error rate with 0.45% compared to previous results of using MLP. These analyses were carried out on the same dataset and same selected features. We also, provided the use of strong complex passwords in static text mode. This complex combination is being more real and reflecting the nature of passwords templates being recommended to be used in authentication systems via applications on the internet.

Classifiers such as MLP provided good results. However, despite its power against larger and more complex datasets, they are extremely hard to interpret, and neural nets can take many iterations and hyper parameter adjustments before a good result is achieved. As well, one of the biggest advantages of using Decision Trees and Random Forests is the ease in which we can see what features or variables contribute to the classification or regression and their relative importance based on their location depth wise in the tree.

We have summed up our results in the last experiments as shown in **Fig. 10**. This figure shows that we provided a good promise for using and implementation an authentication system based on KSD using the random tree and forest classifiers. The RF reached a minimum EER in this experiment when the user provided nine trials for training set.

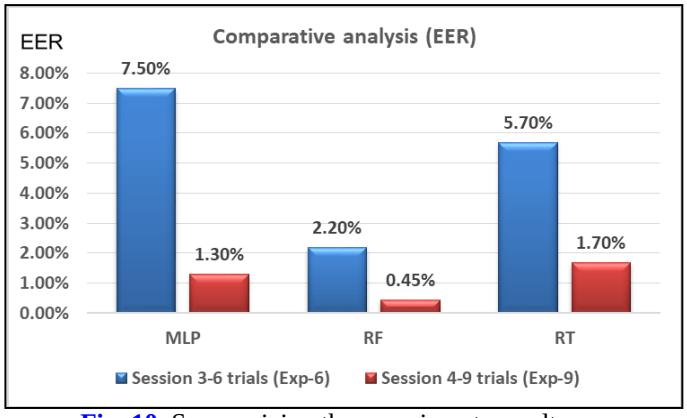

Fig. 10. Summarizing the experiments result

In addition, we have summed up our results to be compared with other researches have done in the literature. Although there was a shortage of available similar studies carried out using common techniques, feature extraction, or password combination, we still have the ability to compare the available similar techniques. **Table 12** described the comparison between

multiple researches from different perspectives. We have reached significant performance improvement and achieved lower error rate when comparing our technique with similar ones.

Table 12. Performance measures for recent researched based on KSD

Re.	Features and Input	Password template	Data set	EER %
Our work	**Timing: Time** **Non-Timing: Pressure, Size, XY position**	**Static, 8 character (Complex passwords)**	**Own data set**	**0.45 %**
[5]	Timing: Time Non-Timing: Pressure, Size	Static ("ABSDEFGH" alphabetic)	3 dataset 2 public 1 own	2.3%
[3]	Timing: Time Non-Timing: Pressure, Size, XY position	Static, 10 digits (PIN password)	Own data set	2.8%
[7]	Timing: Time Non-Timing: Pressure, Size, angle	Dynamic, 4-8 digit (PIN password)	Own data set	3.65%
[13]	Timing: Time Non-Timing: Pressure, Size, XY position	Static, 8 character (Complex passwords)	Own data set	5.43%
[6]	Timing: Time Non-Timing: Pressure, Size, XY position	Dynamic (Free Text)	Own data set	8.10%
[2]	Timing: Time Non-Timing: Pressure, Size	Dynamic, 4-8 digits (PIN password)	Own data set	10%
[1]	Timing: Time Non-Timing: Pressure, Size	Dynamic (Free Text)	Own data set	12.2%
[10]	Timing: Time Non-Timing: Speed, Distance	Static, 10-47 digits (Free Text)	Own data set	13.6%
[9]	Timing: Time only	Dynamic, 10 digits (PIN password)	Own data set	18%

5. Conclusion

This research proposed the use of KSD based as a second factor authentication for mobile users and smart phone devices. Besides the cheap cost provided by KSD since no extra hardware were installed, the KSD provides a significant performance for authenticating remote users through internet applications. This performance with acceptable level in performance as a second factor authentication model. The contribution of using the Timing and Non-timing in features in our analysis improved the security level for the authentication systems.

Many datasets were being provided in the literature but the benchmarking needs in this field is major issue in these type of researches in order to provide a way to do fair comparative analysis among researches. The CMU benchmark exists so far which is not afforded to mobile

devices; because it targets personal computers with classical physical keyboards only with timing features. We are in a bad need to find, build a benchmark dataset that targets the touch screen devices, so we can make fair comparative analysis among researches.

References

[1] Alghamdi, S. J., & Elrefaei, L. A, "Dynamic user verification using touch keystroke based on medians vector proximity," in *Proc. of Computational Intelligence, Communication Systems and Networks (CICSyN), 2015 7th International Conference on*, pp. 121-126, 2015. Article (CrossRef Link)

[2] Tasia, C. J., Chang, T. Y., Cheng, P. C., & Lin, J. H, "Two novel biometric features in keystroke dynamics authentication systems for touch screen devices," *Security and Communication Networks*, 7(4), 750-758, 2014. Article (CrossRef Link)

[3] Jain, L., Monaco, J. V., Coakley, M. J., & Tappert, C. C, "Passcode keystroke biometric performance on smartphone touchscreens is superior to that on hardware keyboards," *International Journal of Research in Computer Applications & Information Technology*, 2(4), 29-33, 2014.

[4] Al-Obaidi, N. M., & Al-Jarrah, M. M, "Statistical median-based classifier model for keystroke dynamics on mobile devices," in *Proc. of Digital Information Processing and Communications (ICDIPC), 2016 Sixth International Conference on*, pp. 186-191, 2016. Article (CrossRef Link)

[5] Alshanketi, F., Traore, I., & Ahmed, A. A, "Improving Performance and Usability in Mobile Keystroke Dynamic Biometric Authentication," in *Proc. of Security and Privacy Workshops (SPW), 2016 IEEE*, pp. 66-73, 2016. Article (CrossRef Link)

[6] Draffin, B., Zhu, J., & Zhang, J, "Keysens: passive user authentication through micro-behavior modeling of soft keyboard interaction," in *Proc. of Mobile Computing, Applications, and Services*, pp. 184-201, 2014. Article (CrossRef Link)

[7] Zheng, N., Bai, K., Huang, H., & Wang, H, "You are how you touch: User verification on smartphones via tapping behaviors," in *Proc. of Network Protocols (ICNP), 2014 IEEE 22nd International Conference on*, pp. 221-232, 2014.

[8] Trojahn, M., Arndt, F., & Ortmeier, F, "Authentication with keystroke dynamics on touchscreen keypads-effect of different N-Graph combinations," in *Proc. of 3rd International Conference on Mobile Services, Resources, and Users (MOBILITY)*, pp. 114-119, 2013.

[9] Tsai, C. J., Chang, T. Y., Tsai, W. J., Peng, C. C., Chiang, M. L., & Wu, H. S, "Work in progress: A new approach of changeable password for keystroke dynamics authentication system on smart phones," in *Proc. of Communications and Networking in China (CHINACOM), 2014 9th International Conference on*, pp. 353-356, 2014. Article (CrossRef Link)

[10] Kambourakis, G., Damopoulos, D., Papamartzivanos, D., & Pavlidakis, E, "Introducing touchstroke: keystroke-based authentication system for smartphones," *Security and Communication Networks*, 9(6), 542-554, 2016. Article (CrossRef Link)

[11] D'Lima, N., & Mittal, J, "Password authentication using Keystroke Biometrics," in *Proc. of Communication, Information & Computing Technology (ICCICT), 2015 International Conference on*, pp. 1-6, 2015. Article (CrossRef Link)

[12] Frank, M., Biedert, R., Ma, E., Martinovic, I., & Song, D, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE transactions on information forensics and security*, 8(1), 136-148, 2013. Article (CrossRef Link)

[13] Salem, A., Zaidan, D., Swidan, A., & Saifan, R, "Analysis of strong password using keystroke dynamics authentication in touch screen devices," in *Proc. of Cybersecurity and Cyberforensics Conference (CCC)*, pp. 15-21, 2016. Article (CrossRef Link)

[14] Coakley, M. J., Monaco, J. V., & Tappert, C. C, "Keystroke biometric studies with short numeric input on smartphones," in *Proc. of Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on*, pp. 1-6, 2016.

[15] Ali, M. L., Tappert, C. C., Qiu, M., & Monaco, J. V, "Authentication and identification methods

used in keystroke biometric systems," in *Proc. of High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conferen on Embedded Software and Systems (ICESS), 2015 IEEE 17th International Conference on*, pp. 1424-1429, 2015. Article (CrossRef Link)

[16] Bakelman, N., Monaco, J. V., Cha, S. H., & Tappert, C. C, "Keystroke biometric studies on password and numeric keypad input," in *Proc. of Intelligence and Security Informatics Conference (EISIC), 2013 European*, pp. 204-207, 2013. Article (CrossRef Link)

[17] Ali, M. L., Monaco, J. V., Tappert, C. C., & Qiu, M, "Keystroke biometric systems for user authentication," *Journal of Signal Processing Systems*, 86(2-3), 175-190, 2017. Article (CrossRef Link)

[18] Killourhy, K. S., & Maxion, R. A, "Comparing anomaly-detection algorithms for keystroke dynamics," in *Proc. of Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on*, pp. 125-134, 2009. Article (CrossRef Link)

[19] Obaidat, M. S., & Sadoun, B, "Verification of computer users using keystroke dynamics," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 27(2), 261-269, 1997. Article (CrossRef Link)

[20] Ramzi, S., Salem, A., Zaidan, D., & Swidan, A, "A Survey of behavioral authentication using keystroke dynamics: Touch screens and mobile devices," *ournal of Social Sciences (COES&RJ-JSS)*, 5(1), 29-41, 2016. Article (CrossRef Link)

[21] Monaco, V, Datasets, 2018. Retrieved from http://www.vmonaco.com/keystroke-datasets.

[22] Zhao, Z., Xu, S., Kang, B. H., Kabir, M. M. J., Liu, Y., & Wasinger, R, "Investigation and improvement of multi-layer perceptron neural networks for credit scoring," *Expert Systems with Applications*, 42(7), 3508-3516, 2015. Article (CrossRef Link)

[23] Zaidan, D., Salem, A., Swidan, A., & Saifan, R, "Factors affecting keystroke dynamics for verification data collecting and analysis," in *Proc. of Information Technology (ICIT), 2017 8th International Conference on*, pp. 392-398, 2017. Article (CrossRef Link)

[24] Britz, D, "Implementing a Neural Network from Scratch in Python – An Introduction," 2015. Article (CrossRef Link)

[25] Mohammad, R. M., Thabtah, F., & McCluskey, L, "Predicting phishing websites based on self-structuring neural network," *Neural Computing and Applications*, 25(2), 443-458, 2014. Article (CrossRef Link)

[26] Jha, V, "Random Forest – Supervised classification machine learning algorithm," 2017. Article (CrossRef Link)

[27] Mrion, N, "17: Bagging," 2018. Article (CrossRef Link)

[28] Jain, R, "Decision Tree Learning. It begins here," 2017. Article (CrossRef Link)

[29] Tanwani, A. K., Afridi, J., Shafiq, M. Z., & Farooq, M, "Guidelines to select machine learning scheme for classification of biomedical datasets," in *Proc. of Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, pp. 128-139, 2009. Article (CrossRef Link)

[30] Popescu, M. C., Balas, V. E., Perescu-Popescu, L., & Mastorakis, N, "Multilayer perceptron and neural networks," *WSEAS Transactions on Circuits and Systems*, 8(7), 579-588, 2009.

[31] Sleit, A., AlMobaideen, W., Baarah, A. H., & Abusitta, A. H, "An efficient pattern matching algorithm," *Journal of Applied Sciences*, 7(18), 269-2695, 2007

[32] Sleit, A., Saadeh, H., Al-Dhamari, I., & Tareef, A, "An enhanced sub image matching algorithm for binary images," in *Proc. of American conference on Applied mathematics*, pp. 565-569, 2010.

[33] Sleit, A., Al-Akhras, M., Juma, I., & Alian, M, "Applying ordinal association rules for cleansing data with missing values," *Journal of American Science*, 5(3), 52-62, 2009.

[34] Hammo, B., Sleit, A., & El-Haj, M, "Enhancing retrieval effectiveness of diacritisized Arabic passages using stemmer and thesaurus," in *Proc. of The 19th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS2008)*, 2008.

[35] Li, Z., Tang, J., & Mei, T, "Deep collaborative embedding for social image understanding," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 9, pp. 2070-2083, 2019. Article (CrossRef Link)

[36] Li, Z., Tang, J., & He, X, "Robust structured nonnegative matrix factorization for image representation," *IEEE transactions on neural networks and learning systems*, 29(5), 1947-1960, 2018. Article (CrossRef Link).

Asma Salem is currently a PhD student in Computer Science Department, in the University of Jordan. Asma worked as full time Senior Administrator at NITC (National Information Technology Center) for ten years. Working as Senior Infrastructure Microsoft System Engineer, administrating Web & Data Base environment, concerns with all Servers & systems Security, Availability & Business Continuity. Asma received her Bachelor of computer engineering in form Balqa applied University in 2008 with excellent grades (3.71/4"And completing Master in Computer Engineering and Networks, 2016. University of Jordan. Her research interests are in the areas of Information Security and Privacy in-Behavioral Authentication using Keystroke Dynamics for Touch screen devices.

Dr. Sleit holds B.Sc, M.Sc. and Ph.D," Computer Science. He received his Ph.D," 1995 from Wayne State University, Michigan. Azzam Sleit is currently working as a Research Professor at KINDI Center for Computing Research, Qatar University. He is a Former Minister of Information and Communications Technology, Jordan (2013-2015"He is also a tenured Professor of Computer Science, King Abdulla II School for Information Technology, University of Jordan, where he functioned as the Dean (2015-2016) and the Assistant President/Director of the Computer Center (2007-2009"Before joining the University of Jordan in 2005, Dr. Sleit was the Chief Information Officer at Hamad Medical/ Ministry of Public Health, Qatar. Dr. Sleit has over twenty five years of experience and leadership working at all levels of government, private and international sectors. Before joining Hamad Medical, Dr. Sleit was the Vice President of Strategic Group & Director of Professional Services of Triada, USA, where he introduced the NGram Technology and Associative Memory Structures. The application of NGram technology helped Ford Motor Company to identify patterns of auto-parts failure and State of Michigan to recognize patterns of child abuse. Dr. Sleit served as the Midwest Regional Manager of Professional Services with Information Builders, USA. From 1993 to 1996, Dr. Sleit was in charge of MetSource, a strategic unit of Metlife responsible for providing outsourced health insurance services for large companies such as AT&T, ABB.

Dr. Ahmad Sharieh is a full professor of Computer and Information Sciences. He has BSc," Mathematics from the University of Jordan (UJ), BSc," Computer Sciences from The University of Tennessee, MSc," Computer Science from Western Kentucky University, and a PhD in Computer and Information Sciences from Florida State University. He held several administration and academic positions: Chairman of Computer Science Department at UJ, Assistant Dean of Research Deanship, Chairman of Central Tender Committee, and Director of University Development Affairs. He worked as Dean of IT School at UJ and Dean (Executive President) of Sur University College/ Sultanate of Oman. He published more than 80 articles in journals and in conferences, and authored and prepared 18 books. He gained grant research projects from UJ and Europe. He developed several software systems. He is on the editorial board of several journals and conferences and a referee of several others. His research areas are in: Distributing Systems, Parallel Processing, Pattern Recognition, Modeling and Simulation, Algorithms, and Cloud Computing.

Dr. Sleit has participated in numerous research activities related to Cloud Computing, Imaging Databases, Data Mining, Health and Management Information Systems and Software Engineering. He authored more than one hundred refereed research papers published in reputable journals and conferences. Since 1987, Dr. Sleit has taught Computer Science courses at various universities in the United States and Middle East maintaining high teaching standards. Professor, Computer Science Department, University of Jordan, Amman-Jordan. Received his Ph.D. Degree in Programming Languages and Compiler Design from Higher Institute for Mechanical and Electrical Engineering/ Bulgaria, 1981. His research interest theoretical and applied computer science.