User Authentication using Keystroke Dynamics

Jyotsna Gaikwad
Student
Department of Information Technology
MET Institute of Engineering College, Nashik, India

Bhagyashree Kulkarni
Student
Department of Information Technology
MET Institute of Engineering College, Nashik, India

Nikita Phadol
Student
Department of Information Technology
MET Institute of Engineering College, Nashik, India

Sneha Sarukte
Student
Department of Information Technology
MET Institute of Engineering College, Nashik, India

Abstract

There is need to secure sensitive data and computer systems from intruders while allowing ease of access for authenticating the user is one of the main problems in computer security. Traditionally, passwords have been the usual method for controlling access to computer systems but this approach has many inherent flaws. Keystroke dynamics is a biometric technique to recognize and an analysis of his/her typing patterns. In the experiment, we measure mean, standard deviation and median values of keystroke features such as latency, duration, digraph and their combinations and compare their performance. The latest trend in authenticating users is by using the potentiality of biometrics. Keystroke dynamics is a behavioral biometrics which captures the typing rhythms of users and then authenticates them based on the dynamics captured. In this paper, a detailed study on the evaluation of keystroke dynamics as a measure of authentication is carried out. This paper gives an insight from the infancy stage to the current work done on this domain which can be used by researchers working on this topic.
Keywords- Biometric, Digraph, Infancy, Keystroke Dynamics, Neurons

I. INTRODUCTION

Nowadays, there is a major threat to misuse the personal as well as official data. Data which is easily accessible from the storage device by an unauthenticated user is a major concern and needs to be eradicated. Increase in the number of software and devices for hacking and cracking causes gains in unauthorized access which results in exploitation of important data. Methods like user ID and password which is mostly used as security is now not reliable and secure due to rapidly increase in hackers and crackers. Also, this method no longer provides consistent safety measures because passwords are prone to shoulder surfing and passwords can also be hacked. To gain secure and efficient access either user must change his password frequently or the user should use a strong password (combination of alphabets, numeric and special symbols). Users do not accept these conditions because they feel that they are restricted or strict and also difficult to be applied. The solution for above said problems is keystroke dynamics. Keystroke Dynamics is a behavioral biometric approach to improve the quality of the computer access rights. It verifies the individual person by its keystroke typing pattern. It is based on the belief that each person has unique typing pattern.

The original technology was derived from the idea of identifying a sender of Morse code using a telegraphy key known as the "first of the sender", where operators could identify that senders transmitting a message by the rhythm, pace and counterpoint rhythm of the signal taps. In the World War II, the Army Signal Core found and identified that an individual persons keying rhythm on a telegraph key were unique.[13] In the early-'80s the National Science Foundation and the National Bureau of Standards in the United States conducted studies establishing that typing patterns contain unique characteristics that can be identified.

In keystroke dynamics measures two variables: first is dwell time, which is the period of time you hold down a particular key and second is flight time, which is the period of time you release those keys. Keyboard dynamics systems can measure one's keyboard input up to 1000 times per second. Keystroke dynamics requires, as most biometrics technologies, a reference template. In this, keystroke dynamic system involves some sessions so the system can build the template by identifying typing rhythm. There are used some characteristics, like frequency of typing error, the frequency of using characters etc. These typing characteristics are extracted from a structured text (login, password, and phrase...) same as from an unstructured (free) text, but results of recognition are now much better for structured text.

II. RELATED WORKS

Keystroke-dynamics research was inspired by telegraph operators by their keying rhythms. This capability which was allegedly quite useful during World War II for identifying radio operators and tracking troop movements was first formally investigated by Bryan and Harter (1897, 1899) as part of a study on skill acquisition in telegraph operators.[2] Many automatic approaches were used for code reorganization in the last several years. Code reorganization is the division of the software into modules, publication

of the API (Application Programming Interface) for the modules, and then requiring that the modules access each other's resources only through the published interfaces.[3] Here, the metrics that measure the quality of modularization of a software system are used. A set of design principles are proposed to capture the concept of modularity and denned metrics centered on these principles. The values of metrics are used to decide whether the software system is properly modularized or not.

A. Existing System

1) Keystroke Dynamics for Login Authentication

- In a UNIX password file got hacked cryptographic technique makes it challenging for a hacker to recover the password from that file.
- In this system, it stores its typing profile in such way if password file or stored information got hacked then a hacker would be unable to recover the detail password or authorized users typing rhythm.

2) Two Broad Categories of Keystroke Dynamics

There are two categories ok keyboard dynamics:

- Login type: In the login type application classifiers are typically presented with short typing samples similar to what would be seen at logins such as user id's, passwords, names and passphrases
- In Session: In this classifiers are typically presented with longer and freer typing samples. e.g. writing emails and word processing(normal typing).

III. PROPOSED SYSTEM

Identification of the person by analyzing their keystroke behavioral patterns is quite challenging due to limited information, fluctuate or changing nature and large variations in information. It's observed that statistical analysis is successful for authentication but it fails to get good results for identification to overcome this issues machine learning based algorithms are followed. There is also used Multi-classifier Fusion (MCF) to get possible improvements on the results for the single classifier.[10] There are three schemes for identifying a person when typing on a keyboard. It shows the performance of each separate technique as well as the performance when combining two or more together. Here shows that the pairwise user coupling in a bottom-up tree structure scheme gives the best performance, both concerning accuracy and time complexity. These techniques are validated by using keystroke data. However, these techniques could equally well be applied to other pattern identification problems. Here, investigated the optimized feature set for person identification by using keystroke dynamics.[4] It was examined the performance of the identification system when a user, unlike his normal behavior, types with only one hand, and system show that performance then is not optimal, as was to be expected.

Keystroke-dynamics research was inspired by telegraph operators by their keying rhythms. This capability which was allegedly quite useful during World War II for identifying radio operators and tracking troop movements was first formally investigated by Bryan and Harter (1897, 1899) as part of a study on skill acquisition in telegraph operators.[2] Many automatic approaches were used for code reorganization in the last several years. Code reorganization is the division of the software into modules, publication of the API (Application Programming Interface) for the modules, and then requiring that the modules access each other's resources only through the published interfaces.[3] Here, the metrics that measure the quality of modularization of a software system are used. A set of design principles are proposed to capture the concept of modularity and denned metrics centered on these principles. The values of metrics are used to decide whether the software system is properly modularized or not.

A. System Pipeline Architecture

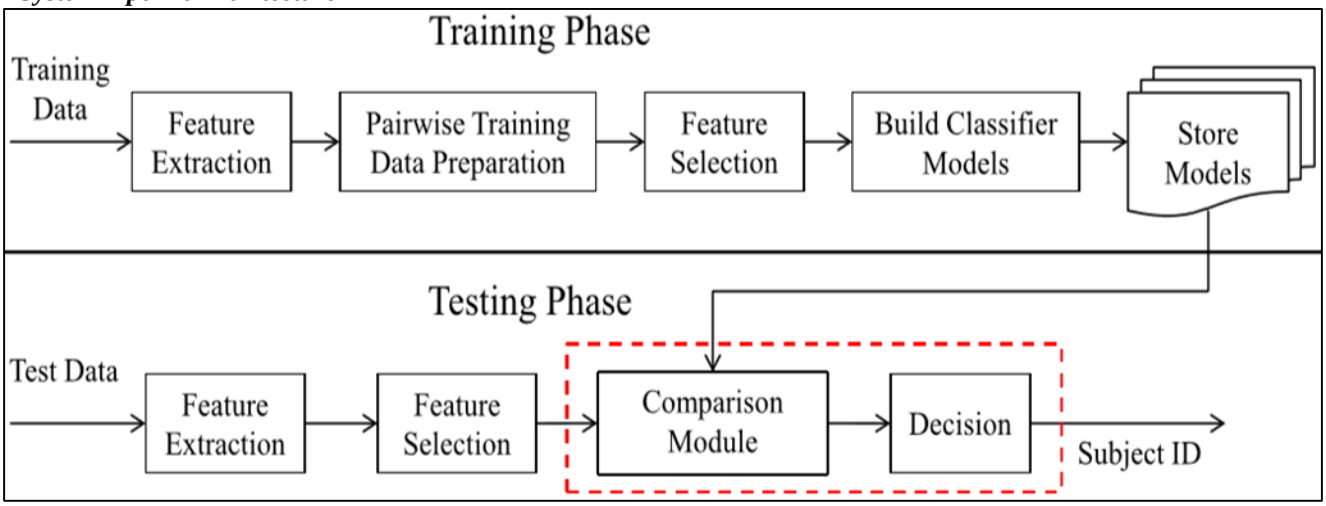

Fig. 1: Architecture Diagram

The architecture diagram involves the process of training phase and testing phase. In training phase, training data is given for feature extraction, feature extracted using pairwise training. Then feature selection is done. System build classifiers and stored this models for further module comparison in the testing phase. In testing phase, features are extracted then features are selected and it uses in comparison module.

B. Machine Learning Algorithms

1) Artificial Neural Network (ANN)

Artificial Neural Networks are based on the neural structure of the brain. The brain basically learns from experience. It is natural proof that some problems that are around the scope of current computers are solvable by small energy efficient packages. This brain modeling also assures a less technical way to develop machine solutions. This new approach to computing also provides a more graceful degradation during system overload than its more traditional counterpart.[12]

These biologically inspired methods of computing are thought to be the next major advancement in the computing industry. Even simple animal brains are capable of functions which are nowadays impossible for computers. Computers do rote things well, like keeping ledgers or performing complex math. But computers have trouble recognizing even simple patterns much less generalizing those patterns of the past into actions of the future artificial neural networks are basically simulated brains. But it's important to note their given software "neurons" basically any programming they want; they can try to set up their rules so their behavior mirrors that of a human brain, but they can also use them to solve problems they could never consider before. Interconnected group nodes, similar to the vast network of neurons in a brain it means ll8 can say that it works same as the human brain. So it can take the output of one neuron to an input of another neuron.

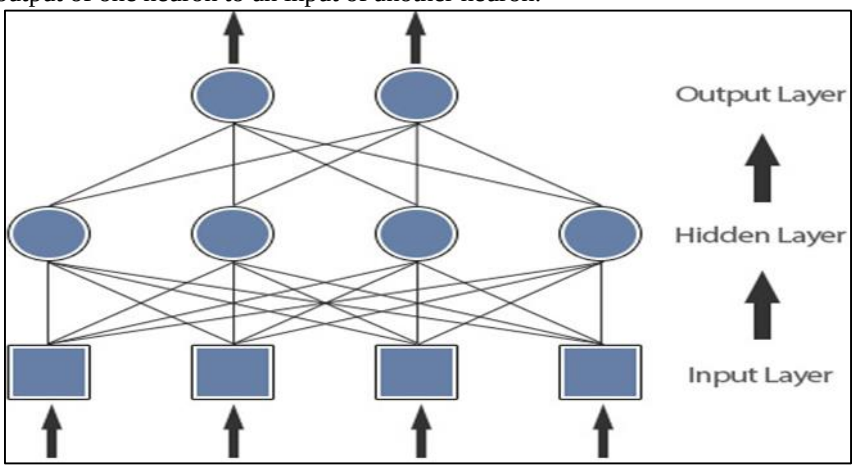

Fig. 2: Artificial Neural Network (ANN)

2) Counter Propagation Artificial Neural Networks (CPANN)

Counter propagation network is a hybrid network. It consists of an out star network and a competitive filter network. It was developed in 1986 by Robert Hecht-Nielsen. It is guaranteed to find the correct weights, unlike regular back propagation networks that can become trapped in local minimums during training. The input layer neurode connects to each neurode in the hidden layer. The hidden layer is a Kohenan network which categorizes the pattern that was input.

Training is done in two stages. Firstly, the hidden layer separate and label the patterns and the weights are then fixed for that layer. Then the output layer is trained. Each pattern input needs a unique node in the hidden layer, which is too large to work on real-world problems. CPANN consist of two-layer: kohenan first layer that cluster input. Out star second layer to provide the output values to each cluster.

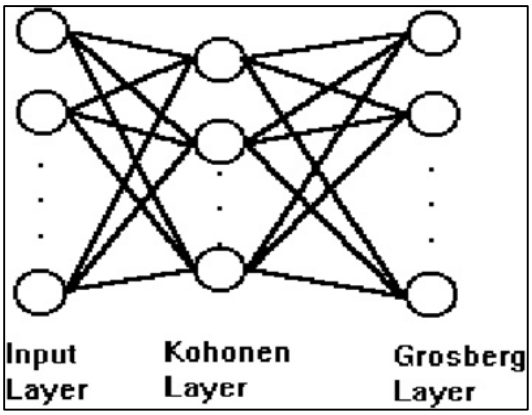

Fig. 3: Counter Propagation Artificial Neural Networks (CPANN)[5]

3) Support Vector Machine (SVM)

A Support Vector Machine (SVM) is used for classification and regression. SVM is used learning algorithms that analyze data for classification and regression analysis. SVMs are more commonly used in classification problems. SVMs are based on the idea of finding a hyperplane that best divides a data set into two classes, SVM is used for text classification tasks such as category assignment, detecting spam and sentiment analysis, image recognition, handwritten digit recognition, such as postal automation services, performing particularly in aspect-based recognition and color-based classification. In this algorithm choose the optimal solution from feasible solution. [1]

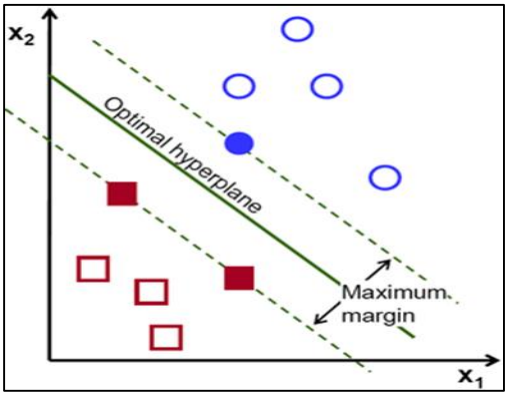

Fig. 4: Support Vector Machine (SVM)[1]

C. Schemes to be used

Three different identification schemes (i.e. S1, S2, and S3) were developed for the comparison and decision module:

1) Scheme 1

In scheme S1 it will randomly arrange the set of users into pairs and for each pair (user i, user j) we will determine if the data fits better to the profile of user i or user j. The user whose profile fits best to the data will proceed to the next round of the scheme. Figure 3.4 illustrates the insight of the Algorithm 1 for Scheme1 (S1) where the total number of users is 20 and the required rank is 1, i.e. N = 20 and r = 1. In this example figure, the pairs are created in increasing order but in actual analysis there selected these pairs randomly in each round of the loop. After each iteration, the number of users reduces the number of users satisfied the required rank based on the maximization of the average score i.e. Si > Sj for m number of keystrokes where,

$$S^i = \frac{1}{m}\sum_{p=1}^{m} sc_p \quad \text{and} \quad S^j = 1 - S^i$$

f the number of users in a round is even (2n), then this number will be halved (n). In case of an odd number of users (2n + 1), will one user continue without comparison, so at the end of the round n + 1 users are left. The number of comparisons T1 for this scheme, when starting with N users and stopping at rank r is T1(N; r) = N □ r.

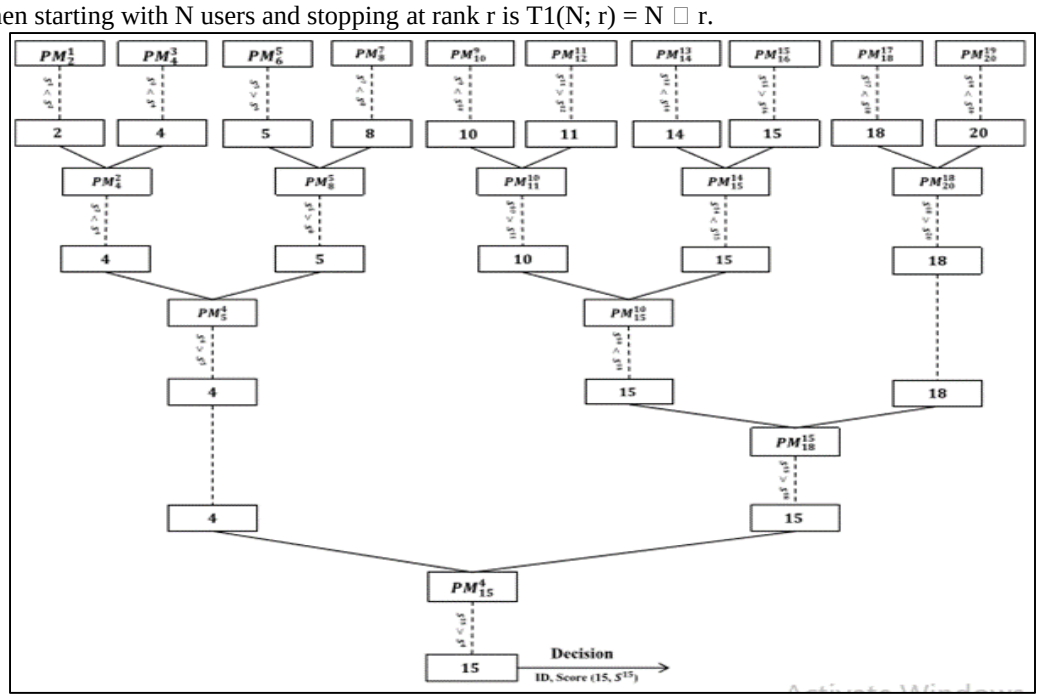

Fig. 5: Graphical Representation of Scheme 1

– Algorithm for Scheme 1

Algorithm 1 Algorithm for Scheme 1

Input: $PM_j^i \leftarrow$ The pairwise classifier model for any given classifier, where $i \in I = \{1, 2, 3 \ldots N\}$, N is the number of users and $j \in J_i = I - \{i\}$; $\Gamma \leftarrow$ The set of test keystrokes, where $|\Gamma| = m$, *i.e.* m is the number of performed keystrokes; $r \leftarrow$ The required rank.

Output: $User_{id,\ score}$

```
while |I| > r do
    k = ⌊|I|/2⌋; T = I; SC = ∅
    while |I| ≥ k do
        i = random{T}; T = T − {i}; j = random{T}; T = T − {j}
        Calculate score sc (i.e. sc_p = P(γ_p|H_i) where γ_p is the feature vector after feature selection of the performed
        keystroke p and H_i is the hypothesis for the user i) for all test keystrokes Γ from the selected pair PM_j^i with
        given classifier(s).
        So, S^i = (1/m) Σ_{p=1}^{m} sc_p; S^j = 1 − S^i
        if S^i > S^j then
            I = I − {j}; SC = SC ∪ {(i, S^i)}
        else
            I = I − {i}; SC = SC ∪ {(j, S^j)}
        end
    end
end
User_{id, score} = SC
```

2) Scheme 2

In scheme S2, for each user i, randomly choose k other users and determine the mean score for user i when comparing the test data in the k pairwise comparisons with the randomly chosen other users. The user has the highest total score is selected as an identified user.

Figure 3.4 shows an example of a graphical representation of Algorithm 2 for Scheme 2 (S2) where k = 6 and r = 1. In comparison to the ith subject, first randomly choose k pairs from the set PMi and calculate the classification scores for the test set □ for each of these k pairs. This will give a total of k _ m score values for any given classifier. Let _p denote the score value of any given classifier where p = 1; 2; : : :m, then denote by scq = _ the score vector obtain qth pair where q = 1; 2; : : : k. Then obtain the resultant score for the ith user by

$$S^i = \frac{1}{m \square k} \sum_{q=1}^{k} \sum_{p=1}^{m} sc_p^q$$

This procedure repeats for all users (i.e. i = 1; 2;::: N) and selects the user with the highest score Si as the identified user. The number of comparisons T2 for this scheme is independent of r, but depends on N and k, in particular, we have T2(N; k) = N _ k.

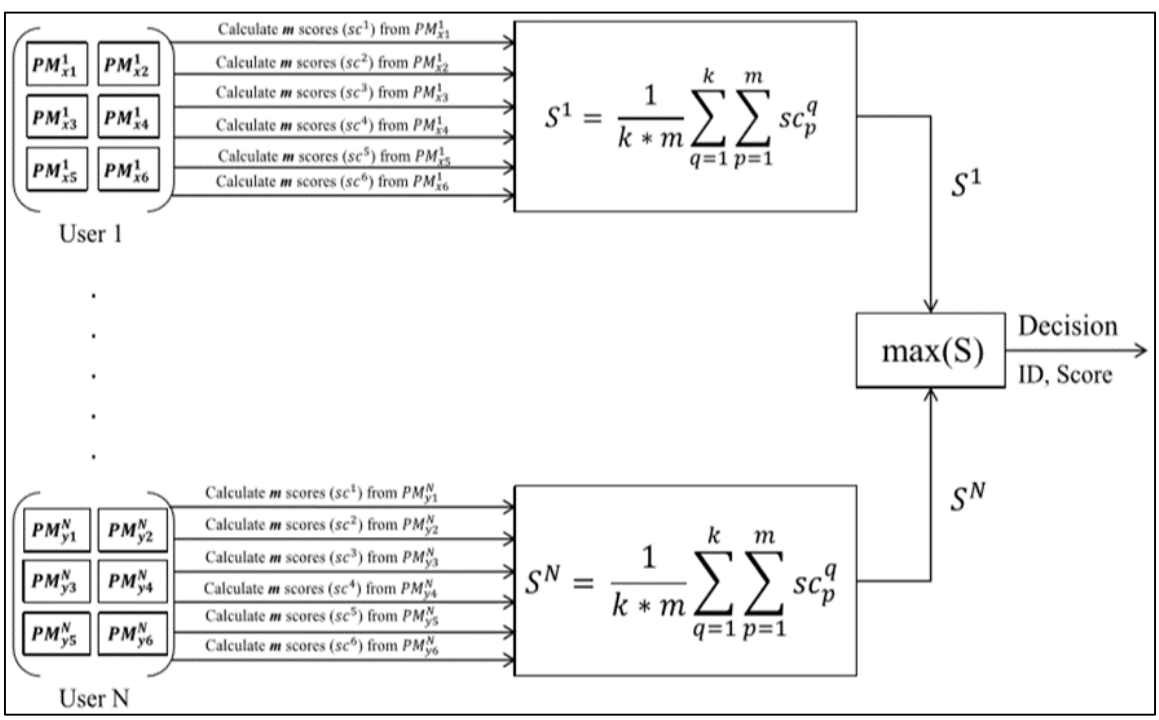

Fig. 6: Graphical representation of scheme 2

- Algorithm for scheme 2

Algorithm 2 Algorithm for Scheme 2

Input: $PM_j^i \leftarrow$ The pairwise classifier model for any given classifier, where $i \in I = \{1, 2, 3 \ldots N\}$, N is the number of users and $j \in J_i = I - \{i\}$; $\Gamma \leftarrow$ The set of test keystrokes, where $|\Gamma| = m$, *i.e.* m is the number of performed keystrokes; $k \leftarrow$ The number of comparison and $k < N$; $r \leftarrow$ The required rank.

Output: $User_{id,\ score}$

$SC = \emptyset$; $T = I$
while $|I| \neq 0$ **do**
  $i = \{I\}$; $I = I - \{i\}$; $J = T - \{i\}$; $S^i = 0$
  while $k > 0$ **do**
    $j = random\{J\}$; $J = J - \{j\}$
    Calculate score sc (*i.e.* $sc_p = P(\gamma_p|H_i)$ where γ_p is the feature vector after feature selection of the performed keystroke p and H_i is the hypothesis for the user i) for all test keystrokes Γ from the selected pair PM_j^i with given classifier(s).
    So, $S^i = S^i + \frac{1}{m \times k} \sum_{p=1}^{m} sc_p$; $k = k - 1$
  end
  $SC = SC \cup \{(i, S^i)\}$
end
$X = \{i | (i, S^i) \in SC\}$
$X_1 \subseteq X$ where $|X_1| = r$
$User_{id,\ score} = \{(i, S^i) | \min\{S^i | i \in X_1\} > \max\{S^j | j \in X - X_1\}\}$

3) Scheme 3

Scheme S3 is based on applying scheme S2 twice. First scheme S2 is used to reduce the set of potential users from the original N users to only c users. In the second step the remaining c users are compared in a full comparison, i.e. we apply scheme S2 on N = c users, and use k = c□1 to compare a user with all c □ 1 other remaining users.

Algorithm 3 shows the algorithm for Scheme 3 (S3). Let_ = [_1; _2; : : : _c] be the set of the Rank-c users after applying S2 with r = c, where _ _ f1; 2; : : :Ng. Now it repeats S2 with the set of _ users with a fixed value of k, i.e. k = c □ 1. This means that here consider all impostor users in the reduced set of c users. S3 is not an independent scheme, it is a combination of S2 with an additional correction made after getting the Rank-c users from S2. This scheme can be considered as using S2 for a re-ranking process after the initial S2 scheme. For S3 the number of comparisons T3 depends on N, k, and c. To be precise we have T3 (N; k; c) = T2 (N; k) + c _ (c □ 1).

D. Multi Classifier Fusion (MCF)

In case of MCF, we denote by (c1; c2; c3) = (_p; _p; _p), where _p, _p and _p are the obtained score values from CPANN, SVM and DT classifiers respectively, where p = 1; 2; :::;m. Then the resultant score value scp is a weighted sum of the three separate scores (see Algorithms 1 and 3 for scp). In particular,

$$sc_p = \left(\sum_{q=1}^{3} w_q c_p^q\right) / \left(\sum_{q=1}^{3} w_q\right)$$

Where wp denote the weights for the weighted fusion technique. We used the following different MCF settings:
ALL MCF: In this setting we included all three classifiers, i.e.w1= w2= w3= 1;
SVM-DT: In this setting we excluded CPANN,
i.e. w1 =0;
CPANN-DT: We excluded SVM in this setting,
i.e. w2 =0;
SVM-CPANN: In this setting we excluded DT,
i.e. w3 =0.

Algorithm 3 Algorithm for Scheme 3

Input: $PM_j^i \leftarrow$ The pairwise classifier model for any given classifier, where $i \in I = \{1, 2, 3 \ldots N\}$, N is the number of users and $j \in J_i = I - \{i\}$; $\Gamma \leftarrow$ The set of test keystrokes, where $|\Gamma| = m$, *i.e.* m is the number of performed keystrokes; $k \leftarrow$ The number of comparison for Algorithm 2 and $k < N$; $c \leftarrow$ The rank correction; $r \leftarrow$ The required rank and $r < c$.

Output: $User_{id,\ score}$

$T \leftarrow$ The set of Rank - c users after applying Algorithm 2
$SC = \emptyset$; $I = T$
while $|I| \neq 0$ **do**
  $i = \{I\}$; $I = I - \{i\}$; $J = T - \{i\}$; $t = |J|$; $S^i = 0$
  while $t > 0$ **do**
    $j = \{J\}$;
    Calculate score sc (*i.e.* $sc_p = P(\gamma_p|H_i)$ where γ_p is the feature vector after feature selection of the performed keystroke p and H_i is the hypothesis for the user i) for all test keystrokes Γ from the selected pair PM_j^i with given classifier(s).
    So, $S^i = S^i + \frac{1}{m \times I} \sum_{p=1}^{m} sc_p$; $t = t - 1$
  end
  $SC = SC \cup \{(i, S^i)\}$
end
$X = \{i | (i, S^i) \in SC\}$
$X_1 \subseteq X$ where $|X_1| = r$
$User_{id,\ score} = \{(i, S^i) | \min\{S^i | i \in X_1\} > \max\{S^j | j \in X - X_1\}\}$

IV. Output & Results

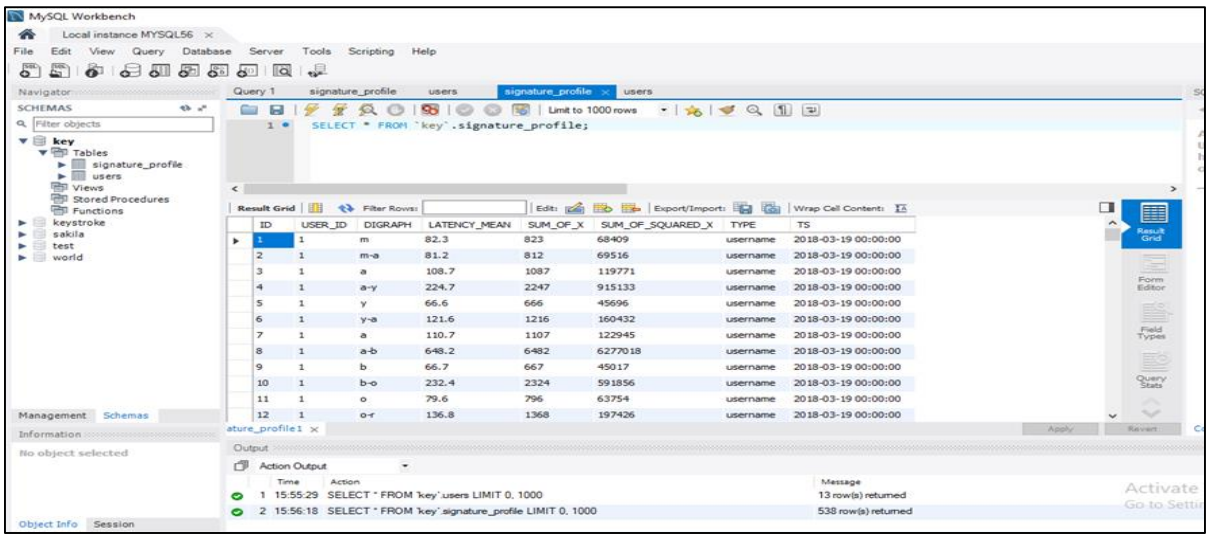

Fig. 7: Data Set

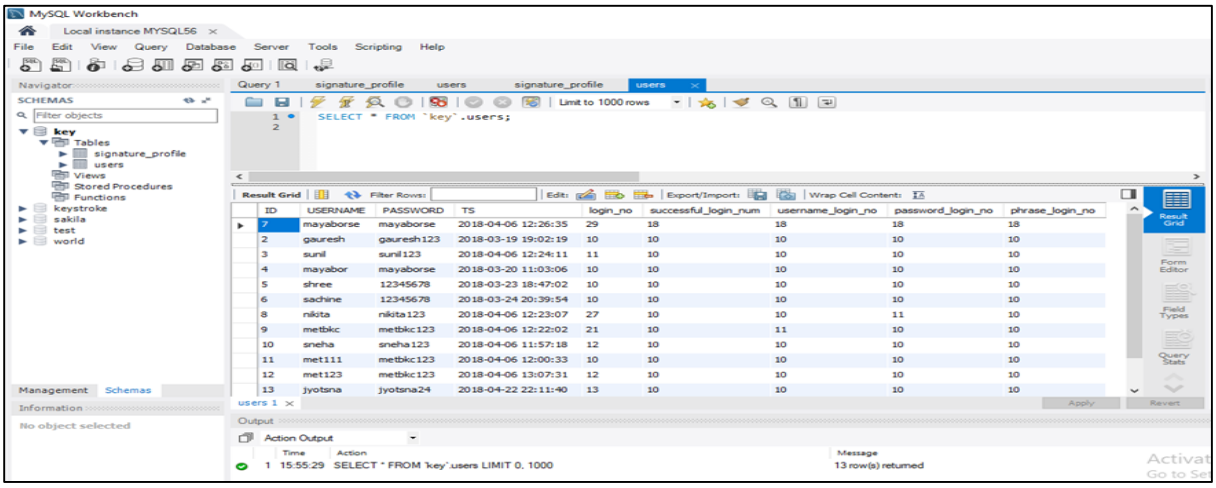

Fig. 8: Data Set

This is the Data sets which is generated by the user's login sessions and store for further verification. So with this huge data sets we can train the machine to keep data of the user such as typing latency, timestamp which is calculated during sessions taken before. By referring those data user validation can be done.

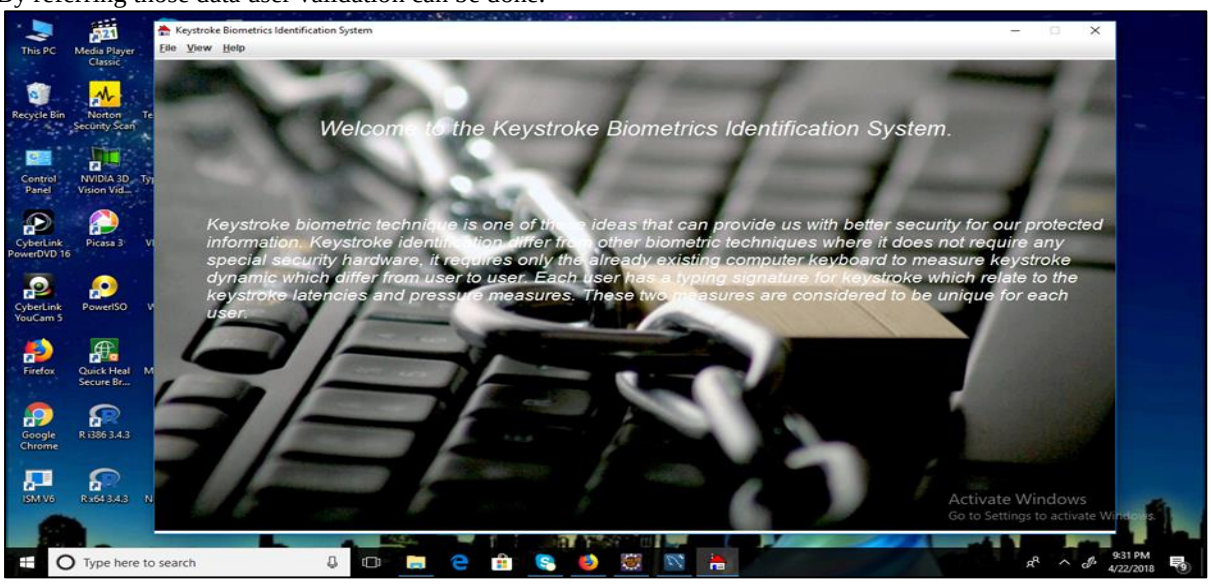

Fig. 9: Home Page

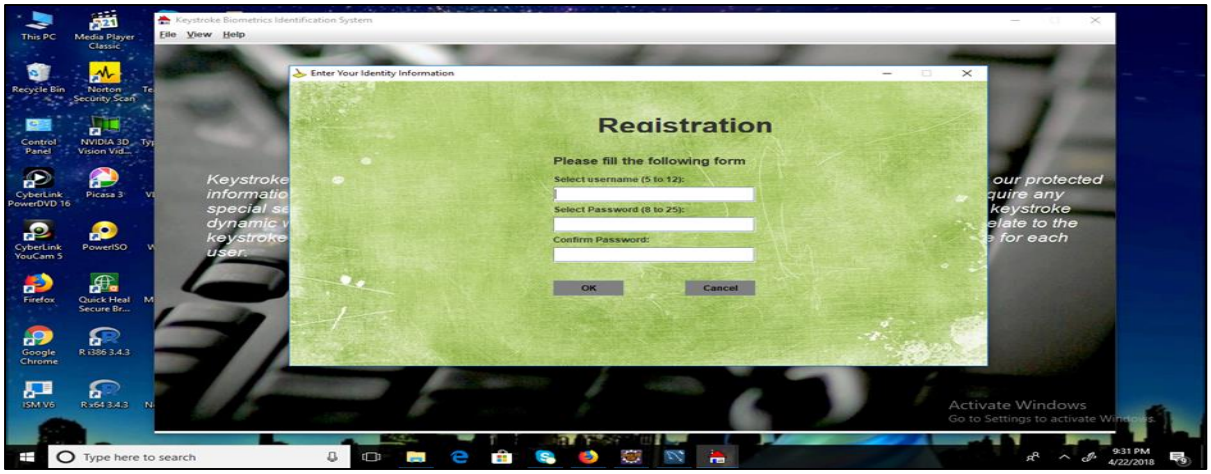

Fig. 10: Registration Page

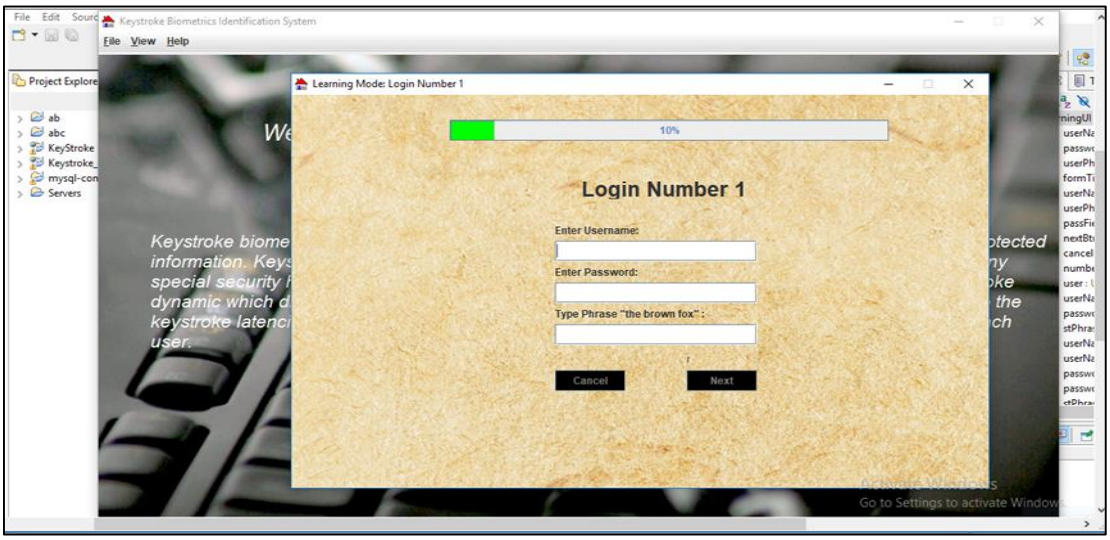

Fig. 11: Learning Mode

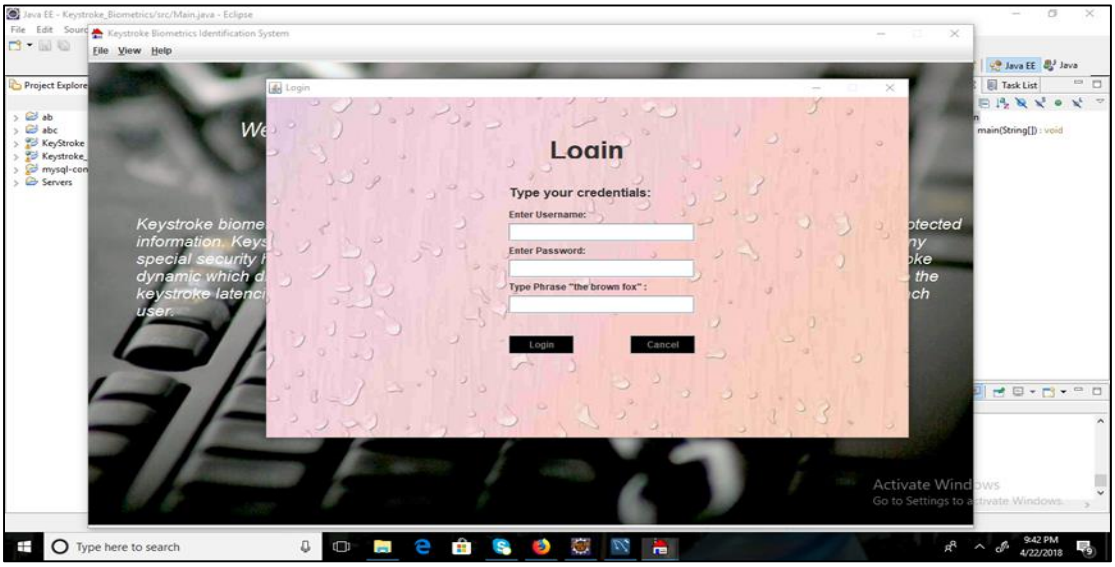

Fig. 12: Login Page

V. FUTURE ENHANCEMENT

The project can be expanded by using an authentication login during login into the system which can automatically ask for authentication using id and password if there is suspicious activities are perform by user.

VI. CONCLUSION

The system focused on identifying a person based on the person's typing behavior. Combined features of maximum pressure with latency which gives an effective way to verify authorized user. Combined ANN, CPANN, SVM, DT for increasing the accuracy of authentication. Extreme different typing patterns among examinees. In this system, the evolution of keystroke dynamics as a measure of authentication is carried out. Keystroke dynamics are one of behavioral biometrics and look at the way a person types at a keyboard. This involves several sessions of a person using a keystroke dynamic system so that the system can construct or build the reference template by detecting one's typing rhythms.

REFERENCES

Journal Papers

[1] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," Data Mining Knowl. Discovery, vol. 2, no. 2, pp. 121–167, Jun. 1998.
[2] R. V. Yampolskiy and V. Govindaraju, "Behavioural biometrics: A survey and classification," Int. J. Biometrics, vol. 1, no. 1, pp. 81–113, 2008.
[3] S. P. Banerjee and D. L. Woodard, "Biometric authentication and identification using keystroke dynamics: A survey," J. Pattern Recognit. Res., vol. 7, no. 1, pp. 116–139, 2012.
[4] P. S. Dowland and S. M. Furnell, "A long-term trial of keystroke profiling using digraph, trigraph and keyword latencies," in Proc. Secur. Protection Inf. Process. Syst., vol. 147. 2004, pp. 275–289.
[5] J. V. Monaco, N. Bakelman, S, -H. Cha, and C. C. Tappert, "Developing a keystroke biometric system for continual authentication of computer users," in Proc. Eur. Intell. Secur. Inform. Conf. (EISIC), Aug. 2012, pp. 210-216.
[6] R. A. Bradley and M.E. Terry, "Rank anlysis of incomplete block designs: I. The method of paired comparisons," Biometrika, vol. 39, nos. 3-4, pp. 324- 345, 1952.
[7] P. Bours and S. Mondal, "Recent advances in user authentication using keystroke dynamics biometrics," in Continuous Authentication With Keystroke Dynamics. Greece: Science Gate Publishing, 2015, pp. 14-58.
[8] S. Bhatt and T. Santhanam, "Keystroke dynamics for biometric authentication—A survey," in Proc. Int. Conf. Pattern Recognit., Inform. Mobile Eng. (PRIME), Feb. 2013, pp. 17–23.
[9] H. Locklear et al., "Continuous authentication with cognition-centric text production and revision features," in Proc. Int. Joint Conf. Biometrics (IJCB), Sep./Oct. 2014, pp. 1–8.
[10] Y. Zhong, Y. Deng, and A. K. Jain, "Keystroke dynamics for user authentication," in Proc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2012, pp. 117–123.
[11] D. Umphress and G. Williams, "Identity verification through keyboard characteristics," Int. J. Man-Mach. Stud., vol. 23, no. 3, pp. 263–273, Sep. 1985.

Books

[12] C. M. Bishop, Neural Networks for Pattern Recognition. London, U.K.: Oxford Univ. Press, 1995.

Theses

[13] S. Mondal, "Continuous user authentication and identification: Combination of security & forensics," Ph.D. dissertation, Faculty Comput. Sci. Media Technol., Norwegian Univ. Sci. Technol., Trondheim, Norway, Feb. 2016.

Proceedings Papers

[14] Soumik Mondal and Patrick Bours, Person Identification by Keystroke Dynamics Using Pairwise User Coupling, vol. 12, no. 6, June 2017.