International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 3 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT1953137

User Authentication by Keystroke Dynamics Using Machine Learning Algorithms

Najla Alavi*[1], Kasim K[2]
Computer Science and Engineering, MEA Engineering College, Perinthalmanna, Kerala, India

ABSTRACT

Due to the expanding vulnerabilities in cyber forensics, security alone is not sufficient to forestall a rupture; however, cyber security is additionally required to anticipate future assaults or to distinguish the potential aggressor. Keystroke Dynamics has high use in cyber intelligence. The paper examines the helpfulness of keystroke dynamics to build up the individual personality. Three schemes are proposed for recognizing an individual while typing on keyboard. Lib SVM and binary SVM are proposed and their performance are shown. Lib SVM is showing a better performance when comparing with binary SVM. As the number of samples are increased it shows an increase in the accuracy. Pair wise user coupling technique is proposed. The proposed procedures are approved by utilizing keystroke information. In any case, these systems could similarly well be connected to other examples of pattern identification problems. This system is applicable in highly confidential areas like military.

Keywords : Keystroke Dynamics, Pair wise User Coupling, Cyber Forensics

I. INTRODUCTION

Cyber Security are the techniques of protecting computers, networks and data from unauthorized access or attacks that are aimed for exploitation. The major areas in cyber security are Application Security, Information Security, Disaster Recovery and Network Security. Machine learning is an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Due to the expanding vulnerabilities in cyber forensics, along with providing security to the computer it is also required to anticipate future assaults or to identify the attacker. Keystroke Dynamics (KD) is a well-established behavioural biometrics methodology because of the inconspicuous idea of biometric information gathering, low computational multifaceted nature and no unique equipment required for information gathering. KD is a well investigated explore area in authentication, where the examination issue is a two class issue for example genuine or fake user, yet there is a little research on the capability of KD for an individual identification. Biometrics is a pattern recognition scheme comprising of strategies which can be utilized to restrain the user's access to an application.

The biometrics highlights caught from the users belongs to two classes. They are Physiological and Behavioural. Physiological are the physical qualities of the users. Examples are fingerprint, face recognition etc. Behavioural are the behavioural traits of the users. Examples are typing rhythm, voice etc [1]. The paper focuses on identifying the persons, so the attacker could be identified thus giving more security to the system to anticipate future assaults.

The previous research used KD to identify persons, but they didn't provide more security to the systems.

They mainly focused on students attending online exams. They used different classifiers which didn't give better results. In most cases, it is very necessary to discover that the present user is not the verified user, for instance in case of PC hijacking, where the information on a framework is secured against unapproved access or alteration. Another case could be an online test where the student behind the keyboard is, the one that ought to take the test. Now and again, it could likewise be compelling to confirm the present user as well as possibly to distinguish that person. In the past instance of the online test, in case misrepresentation is identified when the verification module recognizes that the student typing the exam is not the expected student; it could be interesting to recognize this individual.

We face many problems in social media, mail accounts these days i.e., there are people who can use our account by tracing our password. In current systems, if the password is detected by some other person then that person will type this correct password and our account will be opened for him and he can do anything in that account. However, by KD, even if the intruder has typed the correct password, it will detect that it is not the correct user and the account will not be opened. These are applicable in highly confidential areas. Therefore, KD has much importance these days [2].

II. METHODS AND MATERIAL

A. Dataset

The dataset comprises of typing data 64 students who provided at least 100 keystrokes. A subset of data from both hands typing was utilized as a training set, while the remainder of the information was utilized for testing. The dataset was gathered in an uncontrolled environment. From different past investigations, we discovered that gathering experimental data under controlled environment, with a specific task on a particular PC, has critical drawbacks. In such a case, the user will be concentrated more on finishing the undertaking, and their KD will not represent their normal typing behaviour [7].

Here the keystrokes of the users were taken in software named "pygame". The users were asked to type a 5-character password comprising of string and numbers. The features were collected in this software and were stored in a file. "Fig. 1" explains the database creation.

B. Feature Extraction

Each keystroke crude information k is encoded as $k = (A, T^p, T^r)$ where T^p and T^r are the timestamps in millisecond for key press and key release and A is the value of the pressed key. From the crude data are the feature vectors encoded as $FV_i = (A_i, A_{i+1}, d_i, l_i^{rp}, l_i^{rr}, l_i^{pp})$ where A_i and A_{i+1} are the i^{th} and $(i+1)^{th}$ keys encoded with ASCII values, d_i is the duration of the i^{th} pressed key and l_i^{rp}, l_i^{rr} and l_i^{pp} indicates the latencies between the i^{th} and $(i+1)^{th}$ keys. The classifier used here is SVM i.e., Lib SVM and binary SVM are used.

Support Vector Machines

In machine learning, SVM are supervised learning models with associated learning algorithms that analyse data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An addition to performing linear classification, SVMs can efficiently perform a non-linear classification. When data is unlabeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to

groups, and then map new data to these formed groups.

LIB SVM

Lib SVM is integrated software for support vector classification, regression and distribution estimation. It is a toolbox used to make SVM classifications multiple predictions. Lib SVM provides a simple interface where users can easily link it with their own programs. Some feature are different SVM formulations, efficient multi-class classification, Python, R, MATLAB, Perl, Ruby, Weka, Common LISP and PHP interfaces. The algorithm for classification using LIB SVM is:

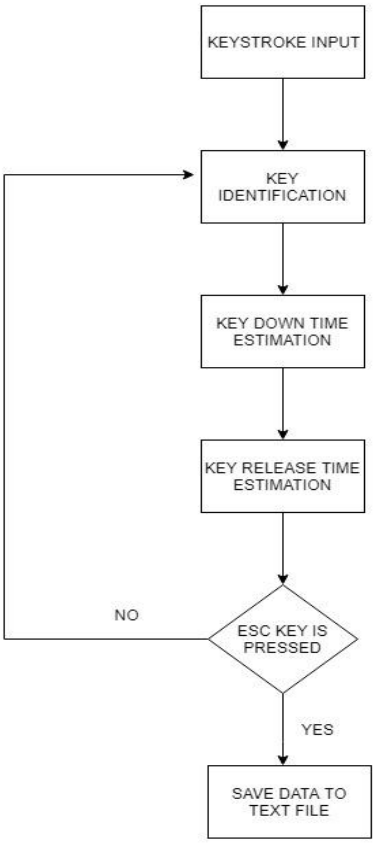

Figure 1. Flowchart of database creation

Algorithm 1 LIB SVM Classifier Algorithm

Input

The keystroke dataset of all persons

Output

The class label prediction

Method

1. Transform data to the format of an SVM package
2. Conduct simple scaling on the data
3. Consider the RBF kernel
4. Use cross-validation to find the best accuracy model
5. Use that best model to train the whole training set
6. The best model is the optimized training model
7. The model predict the label of the input

Binary SVM

Binary SVM is used when your data has exactly two classes. This will not give accurate results, when comparing with two. The algorithm for classification using binary SVM is:

Algorithm 2 Binary SVM Classifier Algorithm

Input

The keystroke dataset of two persons

Output

The class label prediction

Method

1. The keystroke data of 2 persons is labeled as 1 and 2
2. The data is given to SVM classifier training
3. A hyper plane is created for separating the data
4. Procedure is repeated until the optimum hyper plane is created
5. The trained model is then used for testing
6. The model predict the label of the input

C. Model Implementation

We gained from the past research on this dataset that the distance-based classification approaches neglected to accomplish a good identification rate. To achieve better performance, we applied the Pair wise User Coupling (PUC) procedure utilizing the above-mentioned classifiers. The models are implemented using MATLAB. MATLAB is a multi-paradigm numerical computing environment and proprietary programming language developed by Math Works developed by Cleve Moler. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python.

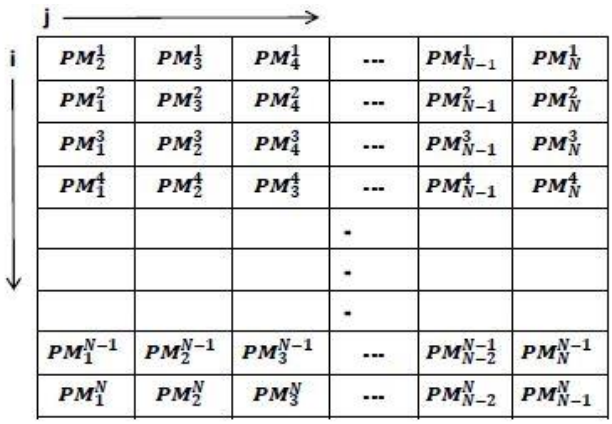

Figure 2. Pair wise training models for multi-class PUC

Pairwise Training Data Preparation

All the data to be trained will be paired i.e., the users are paired randomly to achieve good results. "Fig. 2" shows a pictorial representation of a pair wise training models for multi-class PUC. The training dataset, to construct any given pair wise model for any given classifier, was made. We train the classifier for each training pair and store the classifier(s) models to be utilized in comparison module. Three different identification schemes are introduced for identification i.e., Scheme1, Scheme2 and Scheme3. In scheme S1 we will arbitrarily organize the set of users into pairs and for each pair (user i, user j) we will decide whether the data fits better to the profile of user i or user j. The user whose profile fits best to the data will continue to the following round of the scheme. In equations 1 and 2 'm' is the number of keystrokes.

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

In scheme S2 we will, for each user i, arbitrarily pick k different users and decide the mean score for user i when comparing the test data in k pairwise correlations with the arbitrarily picked different users. The user having the most astounding absolute score is chosen as the identified user.

$$S^i = \frac{1}{mxk} \sum \sum_{p=1}^{m} sc_p^q \quad (2)$$

Scheme S3 depends on applying scheme S2 twice. First scheme S2 is utilized to decrease the set of potential users from the original N users to just c users. In the second step the rest of the c users thought about in a full comparison, i.e. we apply scheme S2 on the c users. This implies we consider all impostor users in the c users.

"Fig. 3" shows the block diagram for our system. A keystroke input will be given to the system, then the feature is extracted i.e., the key is detected and the key up time and down time will be calculated. The pairwise data will be prepared and that will be send to the SVM models. The best model will be saved and that will be send for testing. Then we will get the identified user.

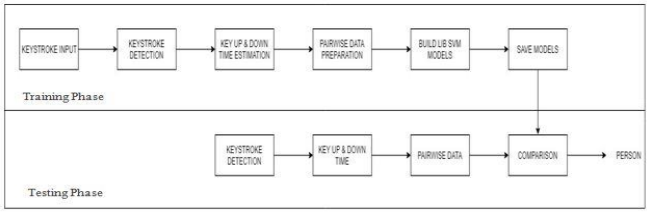

Figure 3. Block diagram for the system design

III. RESULTS AND DISCUSSION

We can clearly observe from our analysis that for the dataset the keystroke span of a given key is the most stable element for individual identification. Pairwise User Coupling (PUC) is the most vigorous identification scheme. Also, this method has a lower computational complexity. LIB SVM is found to be the most powerful classifier in combination with PUC. LIB SVM gave accurate results; whereas binary SVM did not identify the users correctly. This model is mainly used for attacker identification. There is high recognition precision in case of ordinary two hands typing contrasted with single hand typing. It is self-evident that typing with just a single hand impacts the typing behaviour, yet in spite of this, we can even now recognize users. Typing with right hand showed a better performance is because most of the people are right-handed and henceforth typing with right hand resembles the natural typing behaviour better than with left hand. The previous research did not mainly focus on account protection.

To determine the best classifier and improve the accuracy of the model, the 10-fold cross-validation method and 20-fold cross-validation method is used for the training set. We can see that the accuracy of LIB SVM is roughly 80 above, but for binary SVM the accuracy is low.

A. Analysis of Results

The identification outcomes obtained from various investigations are described here. The investigations centre around the proposed algorithms and analysis of user's typing hand.

B. Analysis of the Proposed Algorithms

Table I below shows the identification accuracy obtained for LIB SVM and binary SVM with using different number of keystrokes. We can see that LIB SVM performs better than binary SVM and as the number of keystroke increases, accuracy also increases for LIB SVM but doesn't increase much for binary SVM. In the table, it is shown that for 10 number of keystrokes, the accuracy for LIB SVM is 75.2% and for binary SVM, the accuracy is 10.5%. Then when the keystroke was increased to 15, the accuracy also increased highly for LIB SVM to 77% but did not increase much for binary SVM. For 20 number of keystrokes, accuracy was 82% for LIB SVM and for binary SVM 13%. From this, it is clear that LIBSVM performs better.

TABLE I
COMPARISON TABLE OF THE CLASSIFIERS W.R.TO KEYSTROKES

Accuracy(%)		
Keystrokes	**LIB SVM**	**Binary SVM**
m = 10	75.2	10.5
m = 15	77	12
m = 20	82	13

C. Analysis of user's Typing Hand

Here the identification accuracy for different handedness of the samples are being analysed. The different typing hand samples are:

- Both Hands: Here the samples indicate normal typing when using both hands. Since we use mainly both our hands, accuracy is better.
- Right Hand: Here the analyses were the samples that are typed with only right hand. This too gave a better result as it was with right hand.
- Left Hand: Here the analyses were the samples that are typed with only left hand. Comparing with the other typing, this was not that better result, since left hand writing or typing is not common for all.

Accuracy is calculated with the help of cross validation using confusion matrix. Confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data. There are True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). TP is the number of instances correctly predicted as required. FP is the number of instances incorrectly predicted as required. TN is number of instances correctly predicted as not required and FN is the number of instances incorrectly predicted as not required. Cross validation is one of a model validation technique for assessing how the results perform. It is mainly used when one wants to estimate how accurately a predictive model performs. One round of cross validation involves partitioning a sample of data into subsets performing the analysis on one subset and validating the analysis on other subset. Multiple rounds of cross validation are performed using different partitions and the results are combined.

D. Performance of Proposed Classifiers

LIB SVM and binary SVM are the classifiers used for user authentication by KD. Cross validation was applied to improve the accuracy of the model. Both these algorithms are efficient algorithms but LIB SVM is a better classifier comparing with other classifiers, it shows an accuracy of around 83%. Binary SVM gave a low performance.

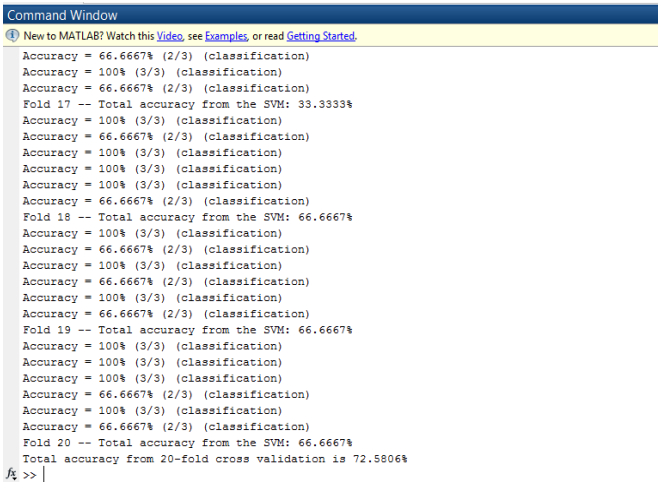

Figure 4. Accuracy at fold 20 when 10 keystrokes are taken- LIB SVM

"Fig. 4" shows the accuracy at fold 20 when 10 keystrokes are taken for LIB SVM. It has an accuracy of about 72%. "Fig. 5" and "fig. 6" shows the identified fourth and second person respectively. "Fig. 7" shows accuracy at fold 40 when 20 keystrokes are taken for LIB SVM. In LIB SVM, as the number of keystroke increases, accuracy also increases. "Fig. 8" shows the accuracy for binary SVM when 10 keystrokes are taken.

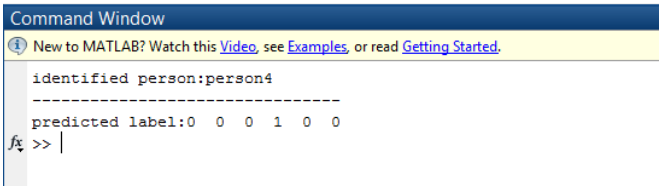

Figure 5. Fourth person identified

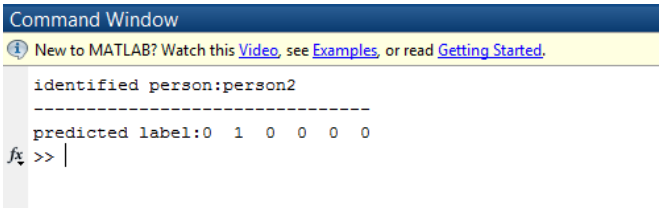

Figure 6. Second person identified

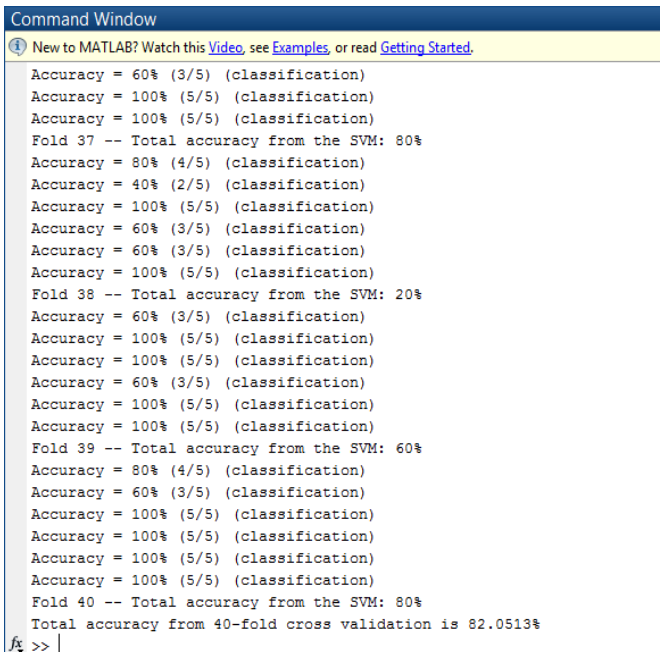

Figure 7. Accuracy at fold 40 when 20 keystrokes are taken- LIB SVM

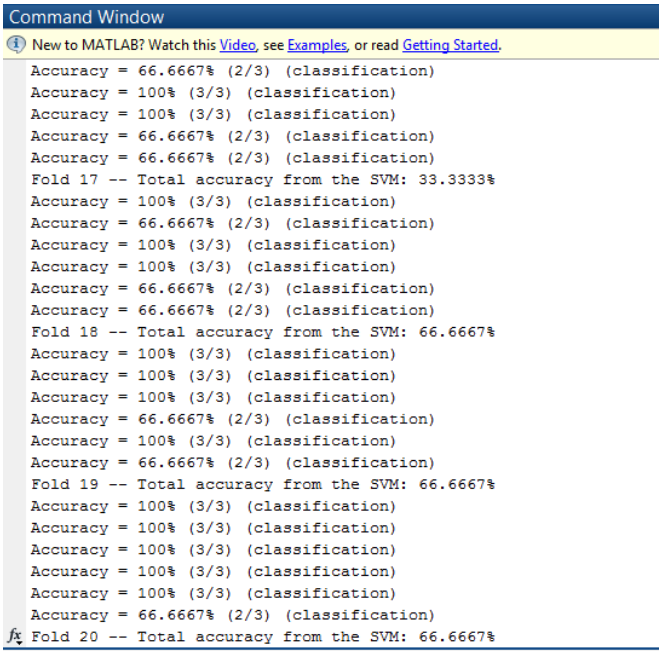

Figure 8. Accuracy when 10 keystrokes are taken – Binary SVM

E. Model Comparison

LIB SVM gives a best result when compared with other algorithms. "Fig. 9" shows a graph comparing with binary SVM with different number of keystrokes.

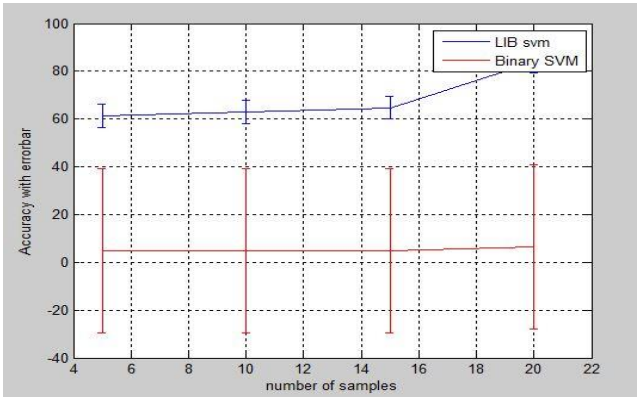

Figure 9. Graph showing comparison between LIB SVM and binary SVM

IV. CONCLUSION

The paper mainly focuses on identifying an individual based on their typing manner. The typing rhythm is unique for each person. The data was collected from some students in software named pygame. Three identification schemes were introduced to identify a person. Compared with the previous researches, the prediction accuracy of the proposed algorithm is around 85%. The target of utilizing typing behaviour to recognize an individual is to utilize it as a tool for cyber-forensics. The main objective of this paper is that it provides user authentication and attacker identification using KD and it can prevent future attacks.

After the analysis, we reached at a conclusion that the performance of binary SVM is comparatively very poor as it involves only two classes. LIB SVM and binary SVM were used to classify the algorithm. In the future, the paper could be extended to perform an experiment on real world cyber forensics.

V. REFERENCES

[1] Bhatt, Shanthi, and T. Santhanam. "Keystroke dynamics for biometric authentication—A survey." *2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*. IEEE, 2013.

[2] Araújo, Lívia CF, et al. "User authentication through typing biometrics features." *IEEE transactions on signal processing*53.2 (2005): 851-855.

[3] Bours P., Mondal S. "Continuous authentication with keystroke dynamics. "Norwegian Information Security Laboratory NISlab (2015): 41-58

[4] Monaco, John V., et al. "One-handed keystroke biometric identification competition." *2015 International Conference on Biometrics (ICB)*. IEEE, 2015.

[5] Obaidat, Mohammad S., and David T. Macchiarolo. "An online neural network system for computer access security." *IEEE Transactions on Industrial electronics* 40.2 (1993): 235-242.

[6] Tappert, Charles C., Mary Villani, and Sung-Hyuk Cha. "Keystroke biometric identification and authentication on long-text input." *Behavioral biometrics for human identification:*

Intelligent applications. IGI global, 2010. 342-367.

[7] Ahmed, Ahmed A., and Issa Traore. "Biometric recognition based on free-text keystroke dynamics." *IEEE transactions on cybernetics* 44.4 (2013): 458-472

Cite this article as :

Najla Alavi, Kasim K, "User Authentication by Keystroke Dynamics Using Machine Learning Algorithms ", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 3, pp. 400-407, May-June 2019.
Available at doi : https://doi.org/10.32628/CSEIT1953137
Journal URL : http://ijsrcseit.com/CSEIT1953137