International Journal of Electrical and Electronics Engineers
Volume 14, Issue No. 01, Jan-June 2022
ISSN (O) 2321-2055
ISSN (P) 2321 -2045

User Authentication using Keystroke Dynamics

Dr. Rahul Desai[1], Aman Mishra[2]

[1]*Associate Professor, Department of Information Technology, Army Institute of Technology, Pune, India*

[2]*Alumni, Army Institute of Technology, Pune*

Abstract –

Traditional methods of user authentication via user id and password cannot guarantee that the person who had logged in using the login credentials is the authentic user. Anyone who got hold of the login credentials can login to the system which can comprise the system. Other methods such as fingerprint scan or retina scan needs the use of expensive and cumbersome hardware which is not user friendly. Thus, authentication system dealing with behavioral aspect of the user can overcome the above problems and provide a reliable authentication system. Therefore, Authentication using Keystroke Dynamics came into existence. Identifying the authentic user and providing her/him with the session is a matter of prime importance when we deal with critical data. This can be used for user authentication while logging in into the system that manages critical data. Biometric qualities become the most ideal and perfect for verification since they can't be stolen, lost or mimicked. Since biometrics techniques like unique finger impression, iris scanner requires outer and expensive durable goods, Keystroke biometrics is an effective and financial strategy anybody can use to give security dependent on biometrics. Keystroke biometrics is the investigation of the composing conduct so as to distinguish the typist, utilizing highlights extricated during composing. This paper presents the user authentication using keystroke dynamics using machine learning techniques.

Keywords - ***Auto-encoder, keystroke biometrics, authentication, machine learning*** .

I. INTRODUCTIONA AND SCOPE

The interest for present day instruments and techniques to limit access to applications and administrations which contain sensitive information is expanding exponentially every year. Customary techniques, for example, passwords, PIN or tokens ignore the difficulties exhibited in light of the fact that they can be stolen or lost easily, which risk the framework security. Stolen passwords have the potential to make extreme harm to organizations and people of the same, prompting the prerequisite that the security framework must have the option to recognize and forestall fake login. Biometrics dependent on identifying the individual or how the individual acts, present a critical security progression to meet these new challenges [1].

Biometrics, characterized as the physical attributes and conduct attributes that make every one of us remarkable, are a characteristic decision for personality confirmation. Biometric qualities become the most ideal and perfect for verification since they can't be stolen, lost or mimicked. Since biometrics techniques like unique finger impression, iris scanner requires outer and expensive durable goods, Keystroke biometrics is an effective and financial strategy anybody can use to give security dependent on biometrics. Keystroke biometrics is the investigation of the composing conduct so as to distinguish the typist, utilizing highlights extricated during

composing. The highlights generally utilized in keystroke biometrics are straight blends of the timestamps of the keystrokes.

These days anybody can disguise to be another person just by utilizing the substantial user's user-id and password. For this situation the password can't validate its legitimate user. Many electronic authentication frameworks have been proposed to protect business exchanges and to verify data. User ID and passwords, IP address sifting, message digest authentication, and so
forth are the famous ones. But these are inclined to be abused. There exists progressing research into distinguishing the peculiarity of a user by using user connection with a PC as a type of authentication. The most believing strategy has been Keystroke biometrics which alludes to the constant examples or rhythms an individual show while composing on a keyboard input gadget. Contrasted with other biometric mappings, keystroke has the essential preferences that:

1. No outside equipment like scanner or identifier is required. All that is needed is a keyboard.
2. The pattern of users is a truly dependable measurement.
3. It can undoubtedly be concatenated with the existing authentication frameworks.
4. This can additionally be utilized to decide the distinction between a human composing or scripted program, for example, malware.

The keystroke authentication approach has been separated into two. A large portion of the current methodologies center around static confirmation, where a user types explicit pre-enlisted string, during a login procedure, and afterward their keystroke features are breaking down for authentication reason.

The subsequent one is called as free-content elements which does not have a pre-decided solid. It adjusts to the composing design. For increasingly secure applications, free-content ought to be utilized to constantly verify a user. Timing data for keystroke biometrics are made by recording the time when each key is pressed and the time when it is discharged. From the planning data, one can extract various features, for example, latencies. The features that are normally utilized are hold time, flight time and digraph which are consequently alluded to as the regular features. Hold time is characterized as the time between the key is pressed and its discharge [2]. The time between when a key is released and when the next key is pushed is known as flight time [3]. The digraph is characterized as the time between key presses of two resulting keys.

We look at Auto-encoder with Support Vector Machine

(SVM) and Principal Component Analysis (PCA) as an oddity detection strategy along with other different models such as logistic regression, Random Forest Classifier, Decision Tree Classifier etc.

II. LITERATURE SURVEY

In the rapidly growing biometrics sector, an affordable and useful biometric like keystroke dynamics has not risen at the same rate. Due to a lack of data collecting and benchmarking standards, comparing the work of different researchers is difficult.

International Journal of Electrical and Electronics Engineers
Volume 14, Issue No. 01, Jan-June 2022
ISSN (O) 2321-2055
ISSN (P) 2321 -2045

Acceptance of standards like these should speed up the exchange of information among academics, allowing us to compare algorithms more efficiently. This would undoubtedly eliminate redundancy and the multiplicity of attempts. The majority of previous systems depend on static verification, in which a user inputs a pre-determined string, such as a password, and then their keystroke attributes are retrieved and evaluated for authentication purposes.

In most studies, the features retrieved from keystroke dynamics patterns are timing features. Timing characteristics extracted:

1. Key Hold: Key delay between the same key pressed and the same key released.
2. Down-Down Time: The time in between two consecutive presses of a password.
3. Up-Up Time: The time between release of 2 successive keys in a password.
4. Up-Down Time: The time gap between the current key released and the next key of the password pressed.
5. Down-Up Time: The time gap between the current key pressed and the next key of the password released.

Once the features from the password have been extracted and templates for the original user created, classification of users into imposter or original user is performed based on the resemblance and dissimilarities among the templates.

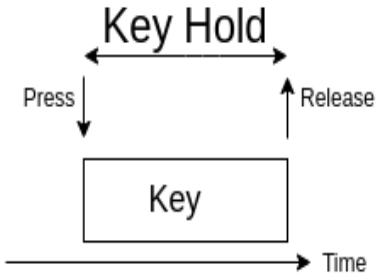

Fig 1: Key Hold Time

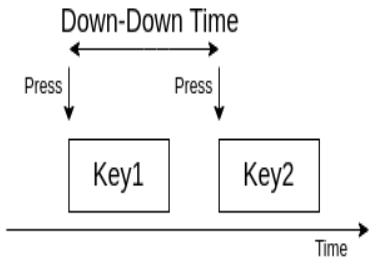

Fig 2: Down-Down Time

International Journal of Electrical and Electronics Engineers
Volume 14, Issue No. 01, Jan-June 2022
ISSN (O) 2321-2055
ISSN (P) 2321 -2045

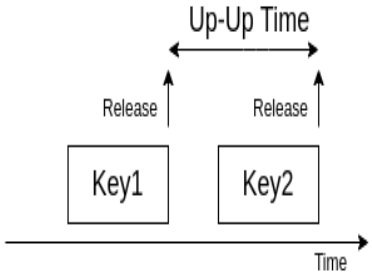

Fig 3: Up-Up Time

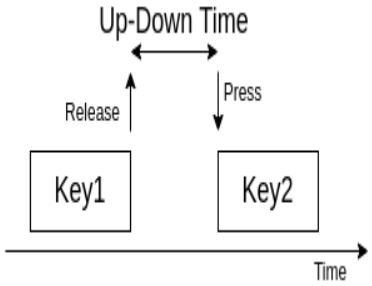

Fig 4: Up-Down Time

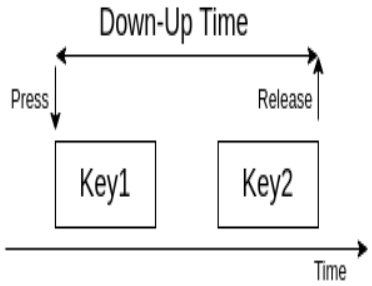

Fig 5: Down-Up Time

The least complex statistical strategy comprises of processing the mean of the features and their standard deviations in the template. These might then be used to calculate distances, such as absolute distance, weighted absolute distance and Euclidean distance. Joyce and Gupta were among the first researchers to use absolute distances to verify their findings. Using only absolute distances, a false acknowledgement rate (FAR) of 0.25 percent and a false reject rate (FRR) of 16.36 percent were achieved. With a 95% accuracy rate, Guven and Sogukpinar [4] used vector examination to classify and categorize clients. The features which is non-linear in nature because keystroke is dependent on the subject's (user's) behavior. As a result, utilizing linear, statistical approaches may not yield excellent results.

Artificial neural networks are nonlinear quantifiable data modelling tools that are inspired by and related to the natural interconnection of neurons. There are two methods for allocating (or learning) the loads: supervised learning and unsupervised learning. Backpropagation is one of the most well-known supervised learning algorithms. Using an artificial neural system, Obaidat and Macchiarolo proposed a method for grouping

between character timings. Three different neural system models were tested during the examination stage: backpropagation, sum-of-products, and half and half sum-of-products. Half breed sum-of-products were found to perform better than other architectures in tests, with an ID rate of 97.8%.

For ordering clients, Yong [5] recommended employing weightless neural networks. They discretized the data into linear and non-linear intervals after scaling it. They also discovered that nonlinear intervals produced better results than linear intervals. Neural networks are capable of dealing with a wide range of characteristics. They can, however, be slow during the preparation and application stages. Because of its black box manner of activity, it's difficult to decide which highlights are important for grouping in neural networks. This could be a problem for continuous keystroke verification, as results are frequently required in a timely manner.

Identifying Patterns is the act of utilizing patterns or protests and ordering them into various classes dependent on specific algorithms and similarities in data points. Various machine learning algorithms for example, the nearest neighbor and grouping to substantially more mind boggling algorithms for example, data mining, Fishers linear discriminant (FLD), Bayes classifier, SVM and chart hypothesis. Yu and Cho [6] improved the performance of keystroke identification by utilizing a 3 stage way. An error rate of 0.81% was accomplished with The SVM novelty detector. A strategy was proposed by Giot et al [7] to distinguish PC users and rate of identification came out to be 95% by utilizing a SVM. One of the greatest advantages of utilizing such algorithms is that they furnish a certainty esteem related with the choice made. Probabilistic learning algorithms can likewise decrease the issue of mistake engendering by disregarding yields with low certainty esteems. Also, unaided learning systems can distinguish patterns in the data naturally.

Keystroke features selection utilizing a system which is combination of other algorithms, SVM and stochastic optimization algorithms, for example, Azevedo [8] created Genetic algorithm (GA) and particle swarm optimization (PSO), these are features selection and features extraction algorithms. For features selection using developmental algorithms like GA, the SVM classifier gave a minimum error of 5.18%, FAR of 0.43% and FRR of 4.75%. With individual and global acceleration of 1.5 in PSO, the minimum error was 2.21% with a FAR of 0.41% and FRR of 2.07%. The benefit of utilizing genetic algorithms is that they can without much of a stretch handle huge databases. It additionally gives multiple solutions and can deal with multidimensional, non-continuous, non-differential, and non-parametrical issues.

III. PROPOSED SYSTEM

The problem with the existing work and implementations related to keystroke authentication based on static text is that the statistic chosen and the model built are not very accessible and compatible with each other. Therefore, we propose an easier and a much simpler model and metric to achieve the desired classification which has better interpretability as shown.

The whole model can be divided four distinct steps. These are listed as follows:

1. The individual register their name and password with the database. Then the user has to type his username and train the machine for six times.

International Journal of Electrical and Electronics Engineers
Volume 14, Issue No. 01, Jan-June 2022
ISSN (O) 2321-2055
ISSN (P) 2321 -2045

2. When people push and release keys, features are extracted. More specifically the delay between the key-down and key-up time.
3. The algorithm is applied and the threshold is generated based on the variations that the user has done while typing the 6 training set.
4. To acquire the user's score, calculate the Euclidean distance between the training and test samples.
5. The user's score is compared to the threshold before a decision is made.

To analyze the suggested system, a dataset is developed. A software application validates the entered data at the time of registration and the credentials are implemented to acquire and extract features from samples. The user has to simply type his username and passwords that they can comfortably type and the rhythm of which they can easily remember. KD, DDKL, UUKL, UDKL, and DUL are calculated using the stamps of each key press and release. For our model we take the key delay between the key up of the current stroke and the key up of the next stroke. These become the attributes of our data set and determine the class labels of our machine learning algorithm. Typically, these key delays are stored in a comma separated value fashion.

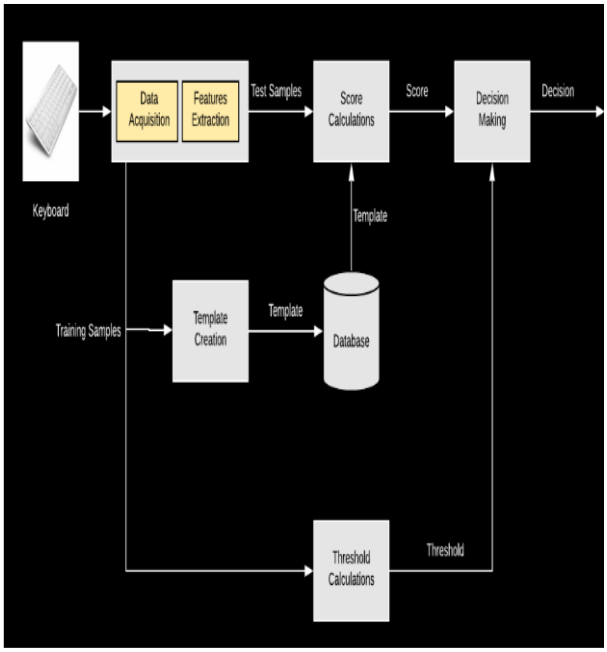

Fig 6: Proposed Model

The threshold calculation is what makes the model adaptive and different than other existing models and algorithms. The window for error is the space in which he is permitted to cause any errors. This is decided by a method called Leave One-Out-Method (LOOM). This method is explained below in some detail in steps:
1. Divide the training space of (n) samples into one sample that will be used as a test sample and (n-1) samples that will be used to produce the training sample.
2. Calculate the distance between the test sample and the mean vector of the (n-1) training samples using a distance measure (Euclidean in our model).
3. Repeat step 2 for (n) times to generate (n) different feature vector thresholds.

International Journal of Electrical and Electronics Engineers
Volume 14, Issue No. 01, Jan-June 2022
ISSN (O) 2321-2055
ISSN (P) 2321 -2045

4. The average of these (n) thresholds yield a single threshold that represents the entire measure of all thresholds.

5. The specific thresholds for the other three distance metrics are calculated using the same steps.

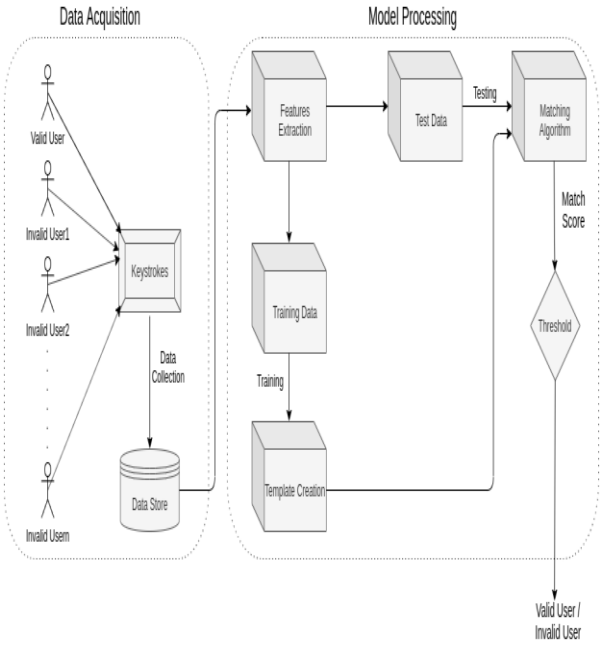

Fig 7: Model Architecture

Our dataset generation is unique for each individual user. Each original user has his own dataset, which will contain his password written many times by different users including himself (valid user) as well as other users who are not valid(imposters). This will enable our model to differentiate the typing pattern of original user from others. Dataset consists of 2 features flight time and hold time. If length of password is n, then dimensions of the dataset will be 2n-1. We can use capital letters and all types of special characters. Our dataset therefore consists of 17 features with their timing information. These 17 features consist of 9 dwell times of password length and 8 flight time between consecutive pairs of letters in the password. It also consists of 250 rows which have the valid password written by different individuals randomly including valid and invalid users.

A model processing environment is developed so that models like as auto-encoders and classifiers can be used to discover anomalies. For model validation, we extract features from the dataset by partitioning it into two parts: training and testing. The selection of hyper-parameters for various models was accomplished through a series of tries and mistakes, as well as an evaluation of their performance on a tiny development system, before reaching a conclusion.

International Journal of Electrical and Electronics Engineers
Volume 14, Issue No. 01, Jan-June 2022
ISSN (O) 2321-2055
ISSN (P) 2321 -2045

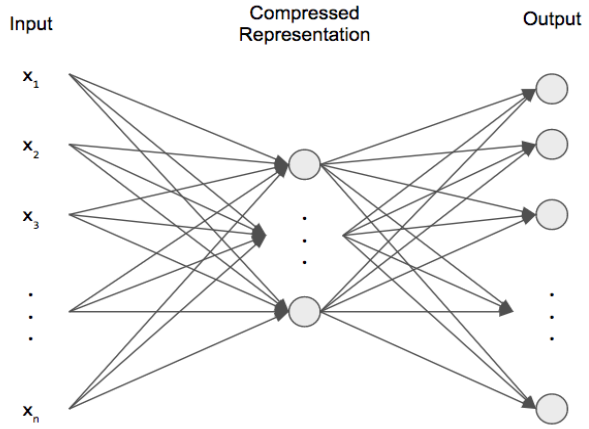

Fig 8: Simple Encoder

Features are extracted from input layer. They are then encoded into a representation of input, which the Auto-encoder learns during training. Classifier is then prepared using the encoded sample, either utilizing just the encoded representation or it is combined with different features, for example, hold time, flight time, digraph, tri-graph etc. Auto-encoders comprises of two encoder layers with 17 input neurons in the first layer and 16 neurons in the second layer. The compressed data consists of 8 neurons. This architecture is specific for a password of length 9, Password can be different but its length should be strictly 9. For passwords of different lengths, this architecture will be different with different number of layers and different number of neurons in it. This implies, with every original user this architecture and dataset will differ. Diverse optimizers like Adam, Stochastic Gradient Descent, were tried so as to find an appropriate strategy for training an Auto-encoder. Adadelta and RMSprop, however we arrived at the conclusion that Adam is best for our architecture. ReLU was utilized as the activation function as it helped in a slight improvement of the reconstruction error. Loss is determined by optimizing the mean squared error of the reconstructed input. A few trials were done utilizing the hyper-parameters featured in Table 1 and mean values were determined to represent the final result.

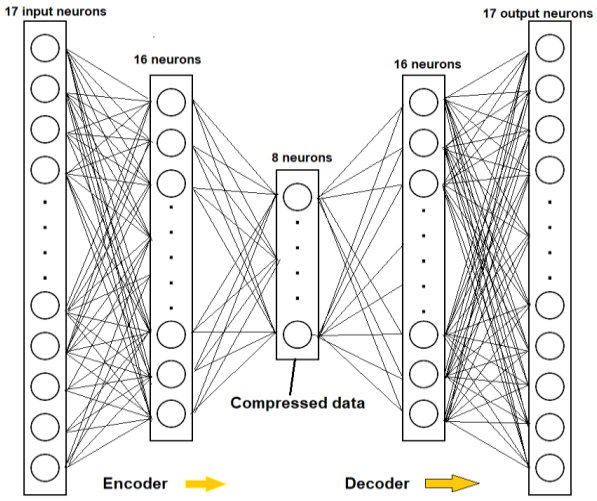

Fig 9: Our Architecture of Auto-encoder

International Journal of Electrical and Electronics Engineers
Volume 14, Issue No. 01, Jan-June 2022
ISSN (O) 2321-2055
ISSN (P) 2321 -2045

Parameters	Values
Activation Function	ReLU
Optimizer	Adam Optimizer
Epochs	150
Learning Rate	0.001
Decay rate	0.001 / 50
Loss	MSE (Mean Squared Error)

Table 1: Network Parameters for Auto-encoder

We have tried different tuning parameters and come to the conclusion, as shown in Table 2 for best result.

Parameters	Values
Kernel	RBF
Gamma	0.05

Table 2: Tuning Parameters for SVC

IV. RESULTS

Initially we had 60 data entries in our data-set, so we applied various models by tuning different hyper-parameters of the various models and compared there training and testing accuracies. The comparison is shown in Table 3.

After applying such models, we increased our dataset upto 250 entries. Several experiments were performed in this dataset. First try, resulted in over-fitting of the data and 36% testing accuracy. Next try, number of epochs was reduced to 150.Here the testing accuracy turn out to be 63%. Then, we again reduced the epochs to 100, here the accuracy was 54%. Then we increased the epocs to 200, here the accuracy was 36%. So we found that 150 epochs were optimum for training. With gamma as 0.1, the same data and 150 epochs accuracy was 71%.

All of the above variations are done in the model using Auto-encoder, PCA and SVC. So we took the final variation and applied it on different models, the testing and training accuracies of various models with the increased data.

Model	Training Accuracy	Testing Accuracy
Logistic Regression	86.00%	50.00%
Decision Tree Classifier	100.00%	40.00%
Random Forest	100.00%	60.00%

International Journal of Electrical and Electronics Engineers
Volume 14, Issue No. 01, Jan-June 2022
ISSN (O) 2321-2055
ISSN (P) 2321 -2045

Classifier		
SVC	94.00%	40.00%
SVC + PCA	84.00%	70.00%
Auto-Encoder + SVC	84.00%	50.00%
Auto-Encoder + SVC + PCA	88.00%	70.00%

Table 3: Training & testing Accuracies

Model	**Training Accuracy**	**Testing Accuracy**
Logistic Regression	91.00%	100.00%
Decision Tree Classifier	98.50%	96.07%
Random Forest Classifier	99.50%	100.00%
SVC	100.00%	98.04%
SVC + PCA	72.00%	47.06%
Auto-Encoder + SVC	100.00%	98.04%
Auto-Encoder + SVC + PCA	91.00%	100.00%

Table 4: Training & testing Accuracies

Three data sets where used for evaluation of the comparative efficiency of our system; the first one is by Yu Zhong (2012); the second one being CMU by Kevin S. Killourhy (2009) and the fourth one being our model.

Killourhy and Maxion used 14 keystroke dynamics anomaly detector to authenticate users, 11 were previously proposed, and 3 where classic recognition patterns using various distance statistic models. (Euclidean, Manhattan, and Mahalanobis distance measures). Their dataset composed of 51 users, who typed the password for 400 times along 8 sessions, i.e., 50 times per session, out of which 200 samples were taken for training the model, and the rest where used for testing the model built. The features from each sample included DDKL, UDKL and KD and achieved and EER of 9.6%. Yu zhong evaluated a keystroke authentication based on a new distance metric, i.e., by combining Mahalanobis distance and Manhattan distance on the keystroke dynamics dataset. They used Nearest Neighbor classifier with their new distance metric to authenticate the user to achieve an EER of 8.4%. Our model used two distance statistics Manhattan distance and Euclidean distance. The model build consists of two distance measures:

1. Manhattan distances: This method compares the average of all the training data to a threshold.

i) This model is less adaptive since the threshold is fixed regardless of the user, and the model is not properly created according to the user's typing pattern.

ii) As per our results, this model is more flexible when compared to the other model. The major reason behind that is while training the model, the user types the same password six times and hence the training model build using Euclidean distances is very small and precise, this means that during authentication, the user must type with the same pattern without even milliseconds of variation in the pattern which is quite inhuman.

2. Euclidean distances: This distance measure is quite adaptive compared to the other model that is the threshold completely depends upon the training data and is not fixed to some constant value.

V. CONCLUSION

The advantage of auto-encoders is the minimal feature engineering that must be done, as well as the much reduced training time. One of the advantages of keystroke dynamics is that it lowers the impact on the client/as experience. The system can be made more robust by supplementing the keystroke authentication method with other risk indicators such as mouse movement, use of shift or caps lock for capital letters, typing pattern when the user is injured or ill, and so on. Combining keystroke authentication with One-Time Password (OTP), Two-Factor Authentication, or Security Questions will strengthen the system's security and eliminate the risk of the original user being hurt in an accident. Furthermore, a time-varying sampling strategy, such as filtering throughout time, may be another way to improve the system's performance, as past research reveals that there is a learning impact in the data of subjects. Hyper-parameters can be investigated further in future study, as well as the effect of various hyper-parameters on feature selection and accuracy for Auto-encoders. The addition of other data, such as mouse pointer behavior data and (where available) touchscreen events in touchscreen devices, can be used to further this research, allowing the relationship between these interactions and keystrokes to be explored.

REFERENCES:

[1]. Yu Zhong, Yunbin Deng, Anil K. Jain. "Keystroke dynamics for user authentication", 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012

[2]. Bours, Patrick, and Vathsala Komanpally. "Performance of keystroke dynamics when allowing typing corrections", 2nd International Workshop on Biometrics and Forensics, 2014.

[3]. Alaa Darabseh, Akbar Siami Namin. "Keystroke Active Authentications Based on Most Frequently Used Words", Proceedings of the 2015 ACM International Workshop on International Workshop on Security and Privacy Analytics - IWSPA '15, 2015

[4]. Guven, Aykut & Sogukpinar, Ibrahim. (2003). Understanding users' keystroke patterns for computer access security. Computers & Security. 22. 695-706. 10.1016/S0167-4048(03)00010-5.

[5]. S. Yong, W. K. Lai, and G. Goghill, “Weightless neural networks for typing biometrics authentication”, in International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, pp. 284-293, Springer,2004.

[6]. E. Yu and S. Cho, Keystroke dynamics identity verification’s problems and practical solutions, Computers Security, vol. 23, no. 5, pp. 428-440, 2004.

[7]. R. Giot, M. El-Abed, and C. Rosenberger, Greyc keystroke: a benchmark for keystroke dynamics biometric systems, in 2009 IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems, pp. 1-6, IEEE, 2009.

[8]. G. L. Azevedo, G. D. Cavalcanti, and E. C. Carvalho Filho, Hybrid solution for the feature selection in personal identification problems through keystroke dynamics, in 2007 International Joint Conference on Neural Networks, pp. 1947-1952, IEEE, 2007.