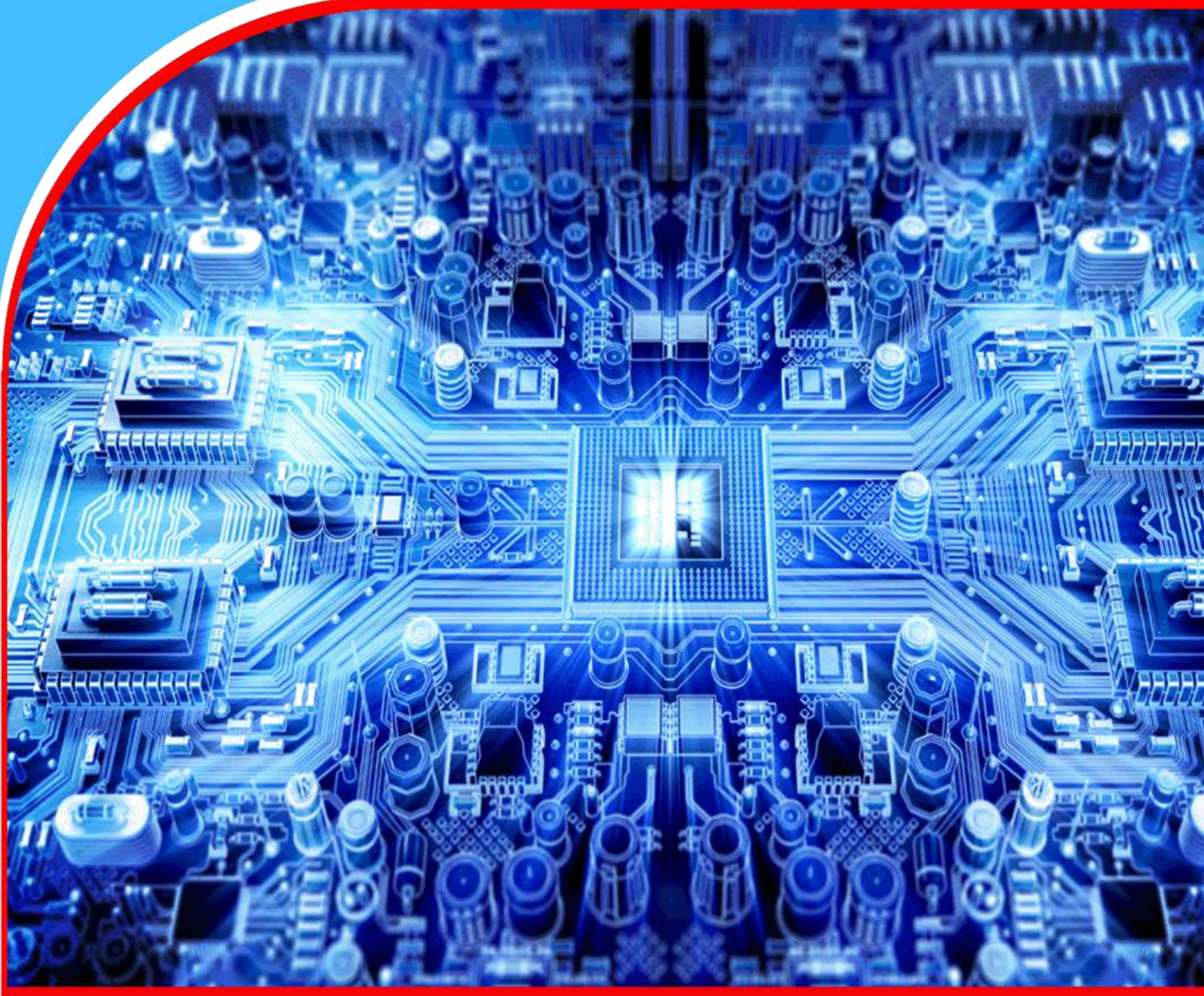
American Journal of
Computing and Engineering
(AJCE)



Authenticating Passwords by Typing Pattern Biometrics.

Rose Nakasi
Safari Yonasi
John Ngubiri

American Journal of Computing and Engineering
ISSN 2520-0449 (Online)
Vol.4, Issue 1, pp 1-12, 2021

www.ajpojournals.org

Authenticating Passwords by Typing Pattern Biometrics

[1]Rose Nakasi
Makerere University, Kampala, Uganda
g.nakasirose@gmail.com
[2]Safari Yonasi
Mbarara University of Science and Technology, Mbarara, Uganda
safij86@gmail.com
[3]John Ngubiri
Makerere University, Kampala, Uganda
ngubiri@cis.mak.ac.ug

Abstract

Passwords are a common measure used in Authentication systems to make sure that the users are who they say they are. The complexity of these Passwords is relied on while ensuring security. However, the role of complexity is limited. Users are forced to write down complex passwords since easy ones are easily guessed. This study aimed at evaluating the uniqueness of typing patterns of password holders so as to strengthen the authentication process beyond matching the string of characters. Using our own dataset, this research experimentally showed that k Nearest Neighbor algorithm using Euclidean distance as the metric, produces sufficient results to distinguish samples and detect whether they are from the same authentic user or from an impostor based on a threshold that was computed. Results obtained indicated that typing patterns are distinct even on simple guessable passwords and that typing pattern biometrics strengthens the authentication process. This research extends work in typing pattern analysis using k Nearest Neighbor machine learning approach to auto detect the password pattern of the authentic and non-authentic users. It also provides an investigation and assessment to the effect of using different k values of the KNN algorithm. Further to this field is the methodology for calculating an optimal threshold value with higher accuracy levels that acted as a basis for rejection or acceptance of a typing sample. Additionally is an introduction of a new feature metric of a combined dataset which is a concatenation of both the dwell and latency timings. A comparison of performance for independent and a combined dataset of the feature metrics was also evaluated.

1. Introduction

Authentication is one of the most crucial areas in computer security, and the use of traditional text-based passwords has been well studied. However, this type of authentication mechanism has drawbacks [22]. Accessing today's web-based services requires users to provide some credentials as authentication measures [15]. Unfortunately, in terms of usability, text-based password authentication is quite problematic [22]. A good password needs to be "easy to remember and hard to guess" at the same time, as suggested by Wiedenbeck et al. [23]. However, for passwords that are easy to remember, they are generally short making them vulnerable to different cyber-attacks [22] and all sorts of security threats such as password guessing attacks, data access via cookies and sniffing. Furthermore, due to the openness of public network, recent advancements in technologies such as Internet of Things whose benefits are realized for instance cloud services, are also vulnerable to a wide range of attacks [24].

These Pass-word threats have caused a big financial and information loss [4] [15] as well as loss of trust. Research recommends complex passwords as well as passwords that expire to ad-dress password based threats [9]. Distinguishing an authentic user from an impostor with the same password is still a challenge [2]. Some systems use security questions which may also be forgotten by authentic users. This study Proposes user characteristics that go beyond experience.

2. Related Work

2.1. Typing Pattern Metrics

Gaines *et al* [13] first investigated the possibility of using typing timings for authentication. When entering data via a keyboard, different timing features are extracted and recorded through a time stamp value. These include; Key press duration (dwell time) and latency (time between two key presses). There are different kinds of latencies like press-to-press (PP) (digraph) [6], Release-to-Release (RR) and Release-to Press (PR) (flight time) [18].Most of the litrature analyses dwell and flight separately [7] [18] but not in combination.

2.2. Typing pattern verification approaches

Typing patterns are either static or dynamic [18]. Static typing pattern [11] [8]. Dynamic patterns involves continuous analysis of the user's typing behavior over a period of time [7] [18]. This research study focused on static patterns.

3. The Experimental Set Up

Typing pattern data was collected from an experiment that enabled participants to enter their username and pass-word. This study was experimented on a standard QWERTY keyboard layout of a Samsung laptop with 2 GB of RAM and 1.9 GHz under the Windows 7 operating system and ran under a Mozilla Firebox browser. It was further conducted under a supervised environment where Users were given the same password and they provided samples from the same platform. Implementation of a background client side Java Script program with a jQuery API was done to capture the each user's dwell and latency durations [18]. Data was recorded in a database created using MySQL database programming language. It could directly be sent to a csv file for easier retrieval and further analysis.

Manual selection of the features used in this study was done from the csv file. A vector matrix of the dwell and latency timing features of the participants was done and saved into different new csv files for authentic users. These samples became a basis of template creation for each participant. Data from one user considered as authentic was labeled a "1" to represent authentic and that from other users taken as impostor labeled a "0" to represent non-authentic. The acquired template data for each user considered authentic was divided into the training

feature set, the positive test feature set and the positive validation feature set. If one user was taken as authentic and the rest as non-authentic, then the general format that was adopted for the split criteria in this study required 50% of each authentic users dataset to be used as the training set, 20% to be used as the positive test dataset and the remaining 30% to be considered as the validation positive feature set. Only 10% of each of the non-authentic user data was used as the negative feature data set in this study.

Analysis of the data was performed by a statistical machine learning approach which is not sufficient enough to provide accurate results [5]. This was done by fitting a K-Nearest Neighbor algorithm with a Euclidean distance metric. The Nearest Neighbor algorithm which is a simple machine learning algorithm was used [16][17][19]. The choice of value of k was a major point of consideration in this research as compared to previous literature. Testing was done using 30 positive validation feature data sets and 30 negative feature data sets. Testing was carried out in order to determine the values of the True Positives (TP), and False positives (FP). This is done with respect to a decision rule on basis of a user dependent threshold value to either reject a test sample as being non-authentic or accept a test sample as being authentic.

Performance evaluation was through determining the sensitivity, false positive rate the accuracy score and the positive predictive values. A receiver operating characteristic (ROC) curve was also used to visualize the results by use of the area under the curve measure. A comparison of results of the AUC of a ROC curve for the performance of the algorithm using different values of k neighbors was done.

4. Results

4.1. Data collection code

A data collection code was generated and built using JavaScript scripting language that employs JQuery API that handles keyboard events of key press and key release durations. The timing duration captured was collected in milliseconds. All the users signed in as authentic users but during analysis every user acted as an impostor (non-authentic) to the other with the guidance of a user interface that was developed using PHP and JavaScript programming languages.

4.2. Participants

The experiment considered four users who were given the same character based password. This was because the respondents were not computer experts. Details of the participants are shown in the Table 10:

4.3. Password

The password was ”tiebwoansk” and more than eight (8) characters as recommended by many systems for a standard strong password [14] and the choice was to have characters spread over the keyboard to avoid them being skewed on one part of the keyboard. Each participant provided 100 samples for both the dwell and the latency.

4.4. Interaction interfaces

The participants interacted with the program through a Mozilla Firefox browser home interface. When an authentic user could be registered for the first time, he/she could click on signup now and would be prompted with a field form to enter his full name, username and password and confirm password. The password field only allowed between 8 and 15 characters in length to create a stronger password.

After successful registration, a user was prompted to the login screen for the two metrics fields using only the username and password that was provided at registration phase. At the login phase a user was required to enter a password 100 times. Viewing the contents of the database was through an administration interface and it was password restricted therefore participants were restricted from viewing the details of collected information.

5. Feature Extraction

The feature of interest in this research that were extracted from the experiment included the key press duration for each key (dwell time) and the key duration between two consecutive keys (latency time) [18]. Given key up and key down durations for two keys k and e, T1, T2, T3 and T4 respectively, then:

$$\text{dwell, } D = T2 - T1 \quad (1)$$

$$\text{Latency, } L = T3 - T2 \quad (2)$$

If given N dwell feature timings, then N-1 is latency timings. With each user providing 100 samples, the total number of timing samples collected from the four users was a vector matrix of (4*100) producing 400 samples. Two csv files containing the username column, a yes column to mean it is an authentic input and the metric column for either dwell or latency feature timing samples were generated. Since the files contained some data that was not considered in the analysis, manual cropping of the dwell and latency timing samples, the features of interest was done to pre-process the files in a proper way meant for splitting and analysis as required in this research.

5.1. Data selection

In this study, the data collected was from a supervised experiment since each data sample came along with a specific username for each participant. Selection consideration was made in such a way that if one user is taken as authentic, then the rest of the users become impostor to the authentic. This resulted in a binary problem where a user was either authentic or non-authentic but not both.

5.2. Splitting of data

Splitting of the dwell and latency datasets become easy with use of the labels that were manually appended on the datasets in the csv files. However, since data in a csv file is hard to manipulate this study employed a Python Pandas data frame for easy transformation of the data into a Python NumPy array for easy splitting. Splitting basing on proportions set in this study was done automatically using python array manipulation and in order to achieve the intended research objectives. The choice of the dataset from each user and the splitting criteria was based on the fact that good results were desired from the machine learning approach used. This therefore meant that the choice of a training dataset proportion had be made carefully because it is the one that would act as a reference template from which every testing sample would be compared to. Since each user provided 100 samples of the same password, 50% of each authentic user's data was used for training, 20% was used as a positive test for calculation of a threshold value and 30% was used for validation and the negative validation test set was comprised of 10% from each non-authentic user creating a total of 30 negative test samples. The output were different feature vector of datasets containing training data, positive testing data and validation test data from both the authentic user and non-authentic user.

6. Analysis

After successful splitting of the datasets, analysis is done by use of statistical machine learning techniques using a k- nearest neighbor algorithm and performed using Euclidean distance metric. Implementation of the algorithm was done using Python and Scikit-learn platforms. Scikit-learn is an open source machine learning library for the Python programming language [20]. Stage one involved calculation of a matching score between the positive test datasets and the training dataset using Euclidean distance metric. A distance metric was therefore used to produce an output of a matching distance score or value. Euclidean distance metric was used in this study because of its simplicity. Given two feature vectors one for training as;

$$Xtrain = x,\ x2, .., xn \quad (3)$$

and another for testing as;

$$Xtest = p1, p2, .., pn \quad (4)$$

where all are discrete real valued features, then the Euclidean distance d is a measure defined as:

$$d(xtrain,\ xtest) = \sqrt{(\sum_{i=1}^{d}(Xitrain - Xitest)^2)} \quad (5)$$

Xi_{train} is the training feature vector for an authentic user and xi_{test} is the test feature vector for an authentic user. This is performed in both cases of dwell and the latency timing datasets for k = 1 and k = 3.

Table 1: User's Threshold value, k=1

User	Dwell-threshold	Latency-threshold
User01-authentic	0.510181	0.073207
User02-authentic	0.056076	0.278897
User03-authentic	0.033362	0.206375
User04-authentic	0.042351	0.273973

6.1. Determination of optimal threshold

It is of great importance to choose an appropriate threshold as it helps in assessing the quality of the algorithm in terms of precision and accuracy [3]. Usually biometric data especially the behavioral biometrics are affected by noise. It is therefore discouraged to use a maximum and a minimum value for a threshold since there is likely to be a lot of rejections for true values in case of a minimum and a lot of acceptance of false values in case of a maximum since these boundary values are usually affected by noise [10]. It is imperative to choose an optimal threshold like the one recommended in [1] which is considered less computationally intensive and effective [1].

Using a feature vector of the distance scores, a threshold value was calculated that would serve as a reference for evaluation criteria. One of the major innovations in this study was the determination of variable threshold values for each user to achieve the intended hypothesis of better performance in terms of having more of the true positive values and less of the false positive values.

Given a matching distance value vector calculated from each authentic user dataset:

$$d = d1, d2, ..., dn \quad (6)$$

The threshold value considered in this research was defined as:

$$Threshold, t = \frac{(\max(di) + \min(di))}{2} \quad (7)$$

The output were the threshold values for different authentic users as illustrated in Table 1 given k = 1: From Table 1, we observe different threshold values for each user considered authentic implying that different users have different typing pattern threshold values given different k values.

Note: Previous studies have shown no standardized mechanism for the choice of threshold value though some studies usually recommend use of a minimum value [12]. However this was tried in this study and it produced few true positive values implying a lock out for more of the authentic users and a consideration of a high threshold would mean allowing more of the non-authentic users to access a security system which is a very undesirable attribute in security.

6.2. Validation decision rule

Validation was done by use of a decision rule that is based on the threshold value obtained in section attached against the matching distance scores. The decision criteria adopted in this research for validation, v is based on the following decision:

$$Validation, v = \{0 \text{ if matching distance score, } d > \text{ threshold, } t; \quad (8)$$

$$1 \text{ if matching distance score, } d \quad \text{threshold, } t$$

6.3. Testing

We present the results for the values of the True Positives (TP), and False positives (FP) values as shown in Table 6 which shows that True Positives which are the desired outcomes are more in all the cases of the dwell and latency, and are in the range of (15-30) out of the total 30 samples from the authentic users as compared to the overall results of the False Positives from the non-authentic users in the range(0-9).This therefore implies that the algorithm based on the threshold used is able to allow more authentic users to access the system and allow only a few of the non-authentic users.

More so, in some instances for user 03 and user 04 the dwell could not perform better because of few TP values and in other instances for user 01, the latency produced few TP values as compared to other users. The study further investigated a combined approach for both the dwell and latency datasets. This is shown in Table 4 showing the number of TP and FP and the user's respective thresholds as shown in Table 2.

Table 2: TP, FP and threshold for each user considered authentic (K=1)

User	Threshold Value	TP	FP
User01	0.223180	29/30	0/30
User02	0.293624	29/30	4/30
User03	0.214827	27/30	4/30
User04	0.292900	29/30	7/30

Results from Table 2 show that combining the dwell and latency timing together produces better results. The number of TP are in the range of (27-29) out of the 30 as compared to the (0-9) before the combination.

6.4. Authentication Evaluation by Typing Patterns

This was done to establish sensitivity, the false positive rate, positive predictive value and the accuracy score for the algorithm. It also includes examination of the results for area under the curve with visualization by use of the ROC curve.

Results from the calculated TPR in both cases of the dwell and latency timing scenarios for k = 1 considering all situations where each user was authentic show that the there was a high True Positive Rate in both cases though better results are evidenced with the latency timing (50%-100%) than with the dwell (43%-97%). However, a very low FPR for the case of dwell is evidenced for user 01(7%), user 02(13%), user 03(0%) and user 04(13%) and likewise for latency user 01(0%), user 02(3%), user 03(13%) and user 04(23%). Better results overall in this scenario are evidenced with the dwell (0%-13%) and poor results are evidenced with latency (0%-23%). Results from the PPV scores indicate better results in both cases of the dwell and the latency where it is (0.8-1.0) for the dwell and (0.77-1.0) for the latency While results from the algorithm accuracy also indicate the good performance for the algorithm used in both cases of the timing scenarios though it worked better in latency scenario (75%-87%) than in dwell scenario (70%-95%) over all users. The combined approach for dwell and latency resulted in the following evaluation results as shown in Table 5 for k = 1.

From the results summarized in Table 5, better results were obtained in terms of sensitivity for all users in range of (90%-97%) than that evidenced for dwell (43%-97%) and latency (50%-100%). This therefore implies that combining dwell and latency datasets together into one authenticates and distinguishes user samples better compared to each dwell and each latency. Furthermore the combined approach achieves a low FPR (0%-23%) with a higher PPV (0.81-0.88) and higher Accuracy (0.88-0.98) which are desirable results for authentication by typing patterns. This accuracy is presented by the Area under Curve (AUC) of a receiver operating characteristic (ROC) curve [21]. AUROC curve has a range of 0 to 1.0. If the algorithm yields 1.0, then it is a perfect algorithm for prediction, 0.5 indicates random performance and any performance below 0.5 indicates that the algorithm is poor in making the prediction as presented by authors in [19].

Table 3 and Table 4 illustrate the dwell and latency performance results of the different evaluation measures for each user considered authentic.

Table 3: Evaluation dwell results for each user authentic (k=1)

Metric	User01	User02	User03	User03
TPR	0.97	0.8	0.43	0.53
FPR	0.07	0.13	0	0.13
PPV	0.94	0.86	1.0	0.80
ACC	0.95	0.83	0.72	0.70

Table 4: Evaluation latency results for each user authentic (k=1)

Metric	User01	User02	User03	User04
TPR	0.50	1.00	0.90	0.97
FPR	0.00	0.30	0.13	0.23
PPV	1.00	0.87	0.81	
ACC	0.75	0.85	0.88	0.87

Table 5: Combined evaluation results for each user authentic (k=1)

Metric	User01	User02	User03	User04
TPR	0.97	0.97	0.9	0.97
FPR	0.00	0.13	0.13	0.23
PPV	1.00	0.88	0.87	0.81
ACC	0.98	0.92	0.88	0.87

Table 6: Evaluation latency results for each user authentic (k=1)

Feature	User01	User02	User03	User04
Dwell-true positives	29/30	24/30	13/30	16/30
Dwell-false positives	2/30	4/30	0/30	3/30
Latency-true positives	15/30	30/30	27/30	29/30
Latency-False positives	0/30	9/30	4/30	7/30

Table 7: Dwell and Latency AUC (k = 1) for all users

User	Dwell AUC	Latency AUC
User 01	0.98	1.00
User 02	0.93	0.98
User 03	0.98	0.93
User 04	0.89	0.97

Table 8: Dwell and Latency AUC (k=1) for all users

User	Dwell AUC	Latency AUC
User 01	0.51	0.54
User 02	0.60	0.55
User 03	0.52	0.60
User 04	0.51	0.52

Table 9: Combined Dwell and Latency AUCwhenk=1 for all users

User	Combined AUC
User 01	1.00
User 02	0.99
User 03	0.94
User 04	0.97

Table 10: Participant's details

User	Description
User01-authentic	female aged 16
User02-authentic	male aged 17
User03-authentic	male aged 21
User04-authentic	male aged 23

6.5. Area under the Curve (AUC)

Performance of the K-NN was done on basis of the AUC and represented graphically on a ROC curve. Tests were performed on both the dwell timing and the latency timing where the AUC for K-NN algorithm for each of the timing metric scenarios was obtained. This was performed for every user with consideration that in each case one user was treated as authentic and the others taken as the non-authentic.

ROC curves for different users using k = 1 and k = 3 were obtained. AUROCC for different authentic users when k = 1 and k = 3 are presented in Figures 1, 2, 3, 4, 5, 6, 7, 8.

The details of the ROC performance were summarized in the Table 8 and Table 9 to clarify detail of the AUROCC:

Results from the AUC indicate good performance with k = 1 in each case of the dwell (0.89-0.98) and latency for all the users (0.93-1.00) while the AUC performance with k = 3 is poor in each scenario of the dwell (0.51-0.60) and latency (0.52-0.60). This implies that a consideration of k = 1 produces better results in terms of distinguishing users compared to a consideration of k = 3. ROC performance was further investigated using a combination of dwell and latency and results tabulated as shown in Table 9. From the results of the AURROC performance of the combined dwell and latency, excellent performance is evidenced for all the users considered authentic and it is in the range of (0.94-1.0). This therefore implies that consideration of the combined approach produces better results as compared to consideration of independent dwell and latency Metrics.

7. Conclusion

This research has demonstrated that there is a consistent and efficient mechanism for distinguishing user's passwords based on their typing patterns. Through the implementation of the different methodological stages of data collection, feature extraction, analysis and evaluation, we were able to show that typing pattern biometrics can reliably authenticate both the authentic and non-authentic users. Using different feature metrics of the dwell and latency timing, a k Nearest Neighbor algorithm with the Euclidean distance metric showed some variations where it was observed that in certain cases of different users considered authentic, the dwell performed better than the latency and that was for user 01 and user 02 cases while in other cases considering user 03 and user 04, the latency performance was better compared to the dwell. We also compared the performance of both the dwell and latency using k = 1 and k = 3, we observed that poor performance was realized.

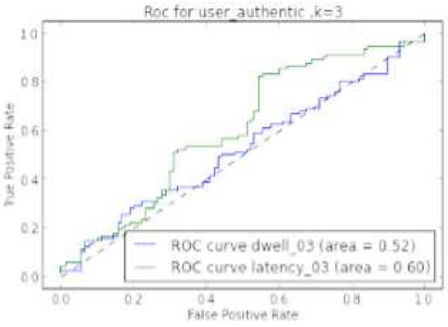


Figure 1: AUROCC user 01 auth, k=3

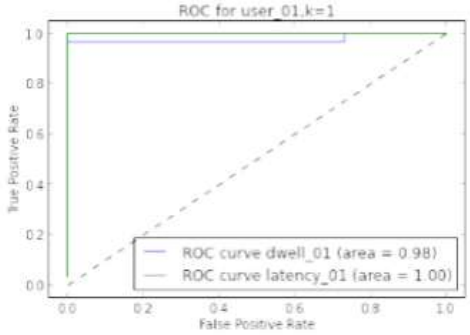


Figure 2: AUROCC user 01 auth, k=1

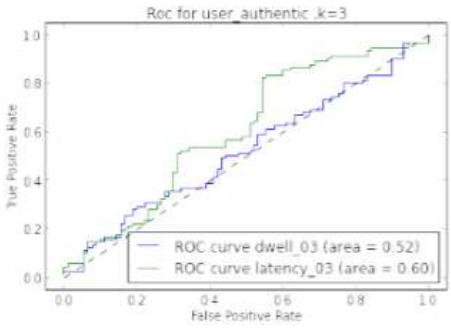


Figure 3: AUROCC user 02 auth, k=3

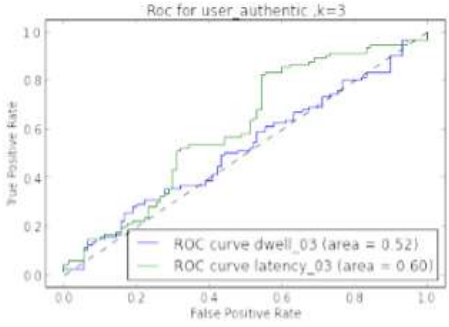


Figure 4: AUROCC User 02, auth, k= 3

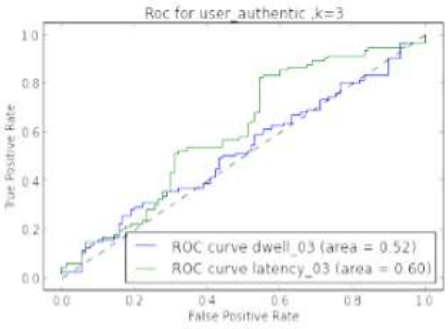


Figure 5: AUROCC user 03, auth k=1

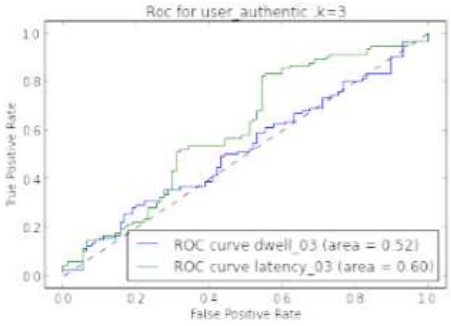


Figure 6: AUROCC user 03 auth, k=3

8. Future work

In this study, a small number of participants was used and this therefore prevents us from generalizing the validity of the approach. This can be solved by considering a big number of participants to investigate the different variations that can possibly occur. Additionally this study only considered one password for all the users, however in the real application, we can't have all authentic users providing the same password. So research on the different user passwords should be considered. In this study we only used one Samsung QWERTY keyboard

platform for user input and in future investigating whether different keyboard platforms affect the performance of the typing pattern biometrics can be an interesting piece of work. The k Nearest Neighbor algorithm was used in this study, studying different machine learning approaches for typing pattern data analysis and making a comparison for the best performing algorithm can be investigated. This work can further be extended for consideration of other keyboard keys apart from the character based keys that were considered in this work. Keys like the numerical, the shift, the backspace, the caps lock and alpha numeric key characters can be produce interesting results when investigated.

References

[1] S. Perkins E. Wolfart A. Walker, R. Fisher. Adaptive thresholding, 2003.

[2] Bellovin. Limitations of the kerberos authentication system. ACM SIGCOM Computer Communication, 1, 1990.

[3] Applied Biosystems. Data analysis on the abi prism7700 sequence detection system: Setting baselines and thresholds. Report 4370923 Revision A, Apple Computer, Inc, 2002.

[4] V. Brennen. Kerberos infrastructure how to. Technical report by kerberos consortium. Report, 2004.

[5] P S Dowland. A preliminary investigation of user authentication using continuous keystroke analysis. IFIP 8th Annual Working Conference on Information Security Management and Small Systems Security, 2001.

[6] C. Xiao E. Lau, X. Liu and X. Yu. Enhanced user authentication through keystroke biometrics. Computer and Network Security, 6(857), 2004.

[7] J. Hu K. Xi, Y. Tang. Correlation keystroke verification scheme for user access control in cloud computing environment. Computer, 54(10):16321644, 2011.

[8] M. Kim. A survey of kerberos v and public key kerberos security. Report, 2009.

[9] S. Lisa. Your password isn't safe: 90 are vulnerable to hacking. Technical report by deloitte's Canadian technology, media telecommunications arm. Report, 2013.

[10] B McCord. Dna typing and threshold setting: Setting instrument parameters and thresholds. Report, International Forensic Research Institute, 2002.

[11] B. Cukic N. Bartlow. Keystroke dynamics-based credential hardening systems. Springer, London, UK, 2009.

[12] Mtenzi Omary. Machine learning approach to identifying the dataset threshold for the performance estimators in supervised learning. IJI, 3(3), 2010.

[13] W.LisowskiR. Gaines. Authentication by key stroke timing:. Some primary results rand report. Report, R-2560-NSF, Rand Corporation, 1980.

[14] SANS. Password protection policy, 2014.

[15] S. Sanyal. Multifactor authentication and security. Fourth ACM Conference on Computer and Communications Security, 10, 2013.

[16] Z.Xiaorong S. Yeqin, T.Zhongqun. Security analysis of kerberos 5 protocol. Computer Knowledge and Technology. Intelligent data analysis, IOS Pres, 6(6):1319–1320, 2010.

[17] Z.Xiaorong S.Yeqin, T.Zhongqun. Security analysis of kerberos 5 protocol. Computer Knowledge and Technology. Intelligent data analysis, IOS Pres, 6(6):1319–1320, 2010.

[18] Teh. A survey of keystroke dynamics biometric. Scientific World, 2013.

[19] G. D. Tambakis Y. S. Boutalis, I. T. Andreadis. A fast fuzzy k-nearest neighbor algorithm for pattern classification. Intelligent data analysis, IOS Pres, 4, 2009.

[20] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. Journal of Machine Learning Research. 12: 2825—2830, 2011

[21] Cook J, Ramadas V. When to consult precision recall curves. The Stata Journal. 2020;20(1):131-148. doi:10.1177/1536867X20909693

[22] Yıldırım, M., Mackie, I. Encouraging users to improve password security and memorability. Int. J. Inf. Secur. 18, 741–759 (2019). https://doi.org/10.1007/s10207-019-00429-y

[23] Wiedenbeck, S., Waters, J., Birget, J.C., Brodskiy, A., Memon, N.: Authentication using graphical passwords: effects of tolerance and image choice. In: Proceedings of the 2005 Symposium on Usable Privacy and Security, SOUPS '05, pp. 1–12. ACM, New York, NY, USA (2005)

[24] Chao Yang, Junwei Zhang, Jingjing Guo, Yu Zheng, Li Yang, Jianfeng Ma, "Fingerprint Protected Password Authentication Protocol", Security and Communication Networks, vol. 2019, ArticleID 1694702, 12 pages, 2019. https://doi.org/10.1155/2019/1694702