A. Brömme, C. Busch, A. Dantcheva, C. Rathgeb and A. Uhl (Eds.): BIOSIG 2019,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2019 1

Fast and Accurate Continuous User Authentication by Fusion of Instance-based, Free-text Keystroke Dynamics

Blaine Ayotte, Mahesh K. Banavar, Daqing Hou, Stephanie Schuckers[1]

Abstract: Keystroke dynamics study the way in which users input text via their keyboards, which is unique to each individual, and can form a component of a behavioral biometric system to improve existing account security. Keystroke dynamics systems on free-text data use n-graphs that measure the timing between consecutive keystrokes to distinguish between users. Many algorithms require 500, 1,000, or more keystrokes to achieve EERs of below 10%. In this paper, we propose an instance-based graph comparison algorithm to reduce the number of keystrokes required to authenticate users. Commonly used features such as monographs and digraphs are investigated. Feature importance is determined and used to construct a fused classifier. Detection error tradeoff (DET) curves are produced with different numbers of keystrokes. The fused classifier outperforms the state-of-the-art with EERs of 7.9%, 5.7%, 3.4%, and 2.7% for test samples of 50, 100, 200, and 500 keystrokes.

Keywords: Keystroke dynamics, biometrics, continuous authentication, monograph, digraph

1 Introduction

As more and more private data becomes digitized, protecting private user data has never been more important. Many systems are only protected with some sort of single-sign-on (SSO) security measure such as a password. When compromised, these systems fail to protect users leaving their data vulnerable. Additionally, people tend to choose easily guessable passwords as found in [Bu19] where 60% of passwords in a major data breach were easily guessable. An additional form of verification is required to continuously monitor the user of a device to ensure they are authorized. Keystroke dynamics is a behavioral biometric that offers strong performance distinguishing users based on how they type [TTY13, Al17, AW13, BW12, YC04]. Keystroke dynamics can be used to provide an additional layer of security to supplement an existing system to more robustly detect intruders. This layer does not require any additional hardware as most computers already have a keyboard. For the truly uncontrolled free-text environment, most of the developed algorithms rely on comparing distributions of flight times between key-presses from the reference user and the test user [AW13, Hu17]. Comparing these distributions is shown to be effective; however, it requires large numbers of keystrokes for both training and testing/evaluation. A large number of graphs in the testing sample means authentication will occur less often or after more keystrokes. Currently, many existing algorithms require 500, 1,000, or more

[1] Department of Electrical and Computer Engineering, Clarkson University, Potsdam, NY. {ayottebj, mbanavar, dhou, sschuckers}@clarkson.edu.
This work has been supported in part by the NSF CPS award 1646542, Clarkson Niklas Ignite Fellowship, and material is based upon work supported by the Center for Identification Technology Research (CITeR) and the National Science Foundation under Grants 1650503 and 1314792.

keystrokes to authenticate users. The average sentence is just under 100 characters which means an imposter could type more than 10 sentences before these systems can detect an intruder [Ay19]. To reduce the number of keystrokes needed for authentication, instance-based algorithms are used. Instance-based algorithms compare graph times individually from the test sample to the reference profile which could be a probability density function (pdf) or other similar statistical or instance-based representation [JG90].

Instance-based methods are not foreign to keystroke dynamics, although they are primarily used on fixed-text data [KM09, ZDJ12]. Existing instance-based works for free-text data include [BM15a, MB16, MB17], where a trust model is used to build a trust score after each keystroke. If the trust score dips below a threshold, the user is locked out of the system. Mondal and Bours [BM15b] introduce two new metrics for evaluating performance of a biometric system, ANGA and ANIA. The average number of genuine actions (ANGA) is the average number of genuine keystrokes before the true user is rejected, and the average number of imposter actions (ANIA) is the average number of keystrokes before an imposter is detected. The users were classified into four categories depending on system performance [BM15a]. For 41 out of the 53 users the trust based model never rejected the genuine user with ANIA ranging from 304 to 1,317 keystrokes, but 10 of those users failed to detect 4% of the imposters. The performance for 9 of the users, however, was worse with ANGA of 1,772 and ANIA of 411. Additionally, 2 of the users had ANGA of 8,942 and ANIA of 771 (failing to detect 3% of the imposters), and 1 user had ANGA of 105 with ANIA 187.

In this paper, the instance-based algorithms that are used and compared include the Manhattan distance, Mahalanobis distance, and the transformed Mahalanobis distance. Feature importance is calculated for five total features including the monograph feature and four different digraph features. Performance of the algorithms with individual features and a single fused feature are evaluated and compared for different numbers of graphs in the testing sample. Our algorithm achieves EERs of 7.9%, 5.7%, 3.4% and 2.7% with 50, 100, 200, and 500 graphs in the testing sample, respectively. For comparison with the work of Bours and Mondal [BM15a], with a block size of 50 graphs and the point on the DET curve where FRR = 0 and FAR = 0.418, our algorithm will never reject the true user and has an ANIA of 134 keystrokes.

2 Dataset

An uncontrolled free-text dataset from Clarkson University was used in this study [Mu17]. There are 103 users that participated in the study and they contributed a combined 12.9 million keystrokes. For all the keystrokes the times of the corresponding up and down events were recorded (see Fig. 1). To the best of our knowledge, this dataset is the largest available, where an average user has 125,000 keystrokes. The keystrokes of participants were recorded along with their mouse movements regardless of application or context. Users had the option of temporarily disabling the keylogger to protect their private information. Due to the uncontrolled aspect of this dataset, researchers typically achieve lower performance on this dataset compared to others [Hu17]. State-of-the-art algorithms

range in performance from 4% to 15% EERs with 1,000 digraphs in the testing sample [Hu17, Ay19]. Huang et al. found that by filtering out "gibberish" keystrokes before authentication, overall results could be improved [Hu16]. They used "gibberish" to define any non-contextual typing behavior such as when playing video games, using keyboard shortcuts, or during other forms of random typing. In this paper no "gibberish" filtering is done to ensure the data is truly representative of a real-world setting with normal interactions between a user and their keyboard.

3 Feature Selection

We treat the timing information from the keystroke dataset (see Section 2) as a time series. In its raw form, it is non-stationary because the time interval between keystrokes can occur at any interval and is not sampled at a continuous rate [GSG18]. Non-stationary time series data can be very challenging to work with and one of the common approaches to extract stationary data is differencing [Ha94, MJK15]. The concept of differencing in keystroke dynamics goes as far back as the 1980s, when researchers used digraphs defined as the time taken to type two consecutive characters [Ga80]. Features commonly used today are the result of differencing, and consist primarily of monographs and digraphs. A monograph, also referred to as latency or dwell time, is defined as the time a particular key is pressed down to when it is released. There are four different definitions of digraph, referred to as DD, UD, DU, and UU, that we use in our work. D corresponds to a key-down event and U corresponds to a key-up event. The four features are the time from the first key either being pressed or released to the second key being pressed or released. The monograph feature and the four digraphs features can be seen in Fig. 1.

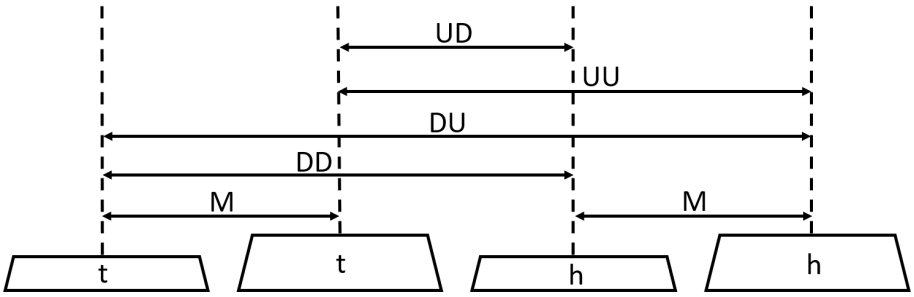

Fig. 1: This schematic shows how monographs and the four different digraphs features can be extracted from two consecutive keystrokes. These digraphs can also be referred to as press-press, press-release, release-press, and release-release (see [Ha17]).

When extracting monographs and digraphs from the raw timing data, only those graphs that fall within a user-defined duration are used so that the typing is representative of normal typing behavior. In this work, the durations are selected based on empirical observations and references in the literature [HHS17] and are as follows: monograph: 0ms to 200ms, DD digraph: 0ms to 500ms, UD digraph: 0ms to 400ms, DU digraph: 0ms to 800ms, and UU digraph: 0ms to 500ms. Note that with some other researcher's definitions of digraph it is possible for some of the digraph times to be negative, for example the UU

digraph when a user presses the “t” key but before releasing it presses and releases the “h” key. This “t+h” graph time is negative because “t” is released after “h”. For this work, however, the previous scenario would result in a positive UU digraph with the reverse order, “h+t”. Tab. 1 shows the rates of occurrences of keystrokes and the corresponding monograph and digraph features. To allow us to compare our results to literature [Hu17, Ay19], we normalize the DD digraph feature to 1 and present the other feature frequencies relative to this. There are almost 50% more keystrokes than there are DD digraphs largely due to “gibberish” typing and breaks between typing sessions [Hu16].

Keystrokes	M	DD	UD	DU	UU
1.48	1.36	1.00	0.95	1.08	0.98

Tab. 1: Frequency of occurrence of different features relative to the DD digraph. There are more keystrokes than monographs and digraphs, likely due to the uncontrolled nature of the dataset.

To determine which features are relevant for distinguishing between users, the mean decrease in impurity (MDI), or Gini importance, is used. MDI is calculated from the random forest classifier where it is defined as the total decrease in node impurity (weighted by the probability of reaching that node) averaged over all the trees in the ensemble [Br01]. The probability of reaching the node is approximated by the proportion of samples reaching that node. The scikit-learn implementation of random forests and MDI are used in this paper [Pe11]. The feature analysis will provide information about which features are worth extracting from the data and which are most informative of user behavior. The feature importance is used to construct the fused classifier.

4 Algorithms

To achieve fast authentication, the algorithms used will be instance-based to leverage as much information from every keystroke as possible. These instance-based algorithms compare the individual test graph times against the corresponding training graphs pdf. The algorithms investigated in this paper are the Mahalanobis distance, Manhattan distance, and the transformed Mahalanobis distance [Ma36, Bl19]. The Manhattan and Mahalanobis distances find the mean and variances from the distributions of graphs and use them to compute distance scores. However, some of the graph distributions are non-Gaussian, and therefore, cannot be described with only a mean and variance, causing the algorithms to perform poorly (see Fig. 2).

To compute the transformed Mahalanobis distance, the first step is to transform the non-Gaussian graph distribution to a normal distribution through the use of the cumulative distribution functions (cdfs) [BEA09]. In our implementation, the graph distributions are transformed into the normal distribution using a scipy implementation [Jo01]. Following this, the Mahalanobis distances are calculated on these transformed distributions. This method improves distance score representations for non-Gaussian data.

To ensure adequate data, only users with at least 30,000 DD digraphs are kept. This leaves 52 users in the dataset meeting the criteria. The training pdfs are only constructed when

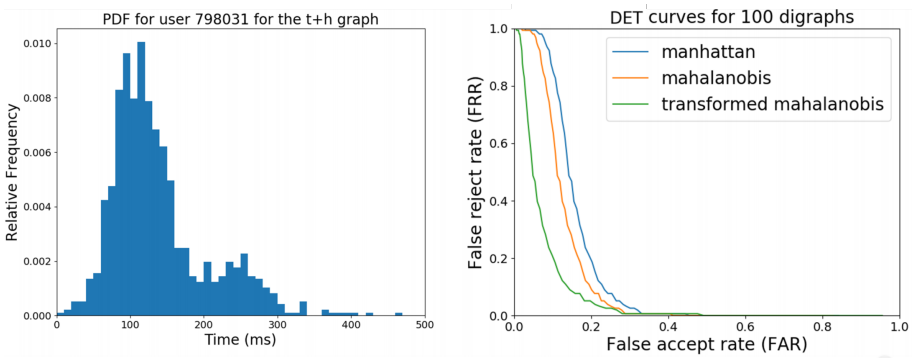

Fig. 2: (Left) A non-Gaussian pdf for the DD digraph feature (digraph t+h for user 798031). (Right) The DET curves for Manhattan, Mahalanobis, and transformed Mahalanobis distances for 100 test digraphs. Some graphs exhibit non-Gaussian behavior and as result, the transformed Mahalanobis distance outperforms the other distance metrics.

there are 30 or more occurrences of that graph in the training sample. The training sample consists of 29,500 graphs and the testing sample is varied from 50, 100, 200, and 500 graphs. The monograph and digraph features are calculated from the raw timing data for each of the users, only keeping the features if they fall within the specified time ranges as discussed in Section 3. To generalize our results across the entire dataset, Monte Carlo cross-validation is used with 30 random subsets of the data. Each subset contains a different random sampling of training and testing graphs from the set of all the available graphs. Our results are then averaged across Monte Carlo iterations to ensure the results are representative of the entire dataset.

5 Results

To determine which features are useful for distinguishing between users, we use the MDI taken from a random forest classifier. MDI is calculated for the five features with 50 random subsets of each user's data and averaged together. The feature importance is determined using different numbers of testing graphs: 50, 100, 200, and 500. The number of graphs used is the same for each feature. It is important to note that not all features occur with the same frequency (see Tab. 1).

Tab. 2 shows the feature importance for four different test sample sizes. The variance of the importances never increases much above 0.01, suggesting feature selection does not vary too much by user. The most important feature is the UU digraph followed by the DD and DU digraphs. The importance of the UU digraph is surprising, although the high importance of the DD digraph is supported by previous works. The feature importance does not change noticeably depending on the number of testing graphs, except for the monograph feature. As the number of test graphs increases, the importance of the monograph feature decreases. While this implies that with more information available the monograph feature becomes less informative, it is important to keep in mind that the monograph fea-

Testing Graphs	Feature Importance				
	M	DD	UD	DU	UU
50	0.144 ± 0.014	0.244 ± 0.010	0.113 ± 0.007	0.210 ± 0.011	0.289 ± 0.010
100	0.113 ± 0.008	0.257 ± 0.010	0.109 ± 0.008	0.216 ± 0.010	0.304 ± 0.007
200	0.109 ± 0.011	0.270 ± 0.011	0.100 ± 0.007	0.200 ± 0.009	0.320 ± 0.010
500	0.083 ± 0.008	0.253 ± 0.011	0.117 ± 0.011	0.229 ± 0.013	0.319 ± 0.011

Tab. 2: Feature importance at 50, 100, 200, and 500 testing graphs. The importances are averaged across all users and are reported along with the variance. For each testing graph size, the cells highlighted in blue and red denote the feature with the highest and lowest importance respectively.

ture occurs at a higher rate than the digraph features. For fast authentication with fewer keystrokes, monographs are a very useful feature.

There is no feature that provides no information and no feature that dominates all others. This indicates that a fused classifier with all five features is likely to perform best overall. A fused classifier was constructed using the feature importance as weights, which are selected based on the size of the testing pool as indicated in Tab. 2. The detection error tradeoff (DET) curves of classifiers using the individual features and the fused features are shown in Fig. 3. In all cases, the fused classifier performs the best, followed by UU, DD, and the DU digraphs, consistent with our findings from the feature importance (see Tab. 2). The fused classifier sees the most improvement over the next best feature (UU digraph) when the number of graphs used is smaller. As the number of graphs increase, the fused classifier still outperforms the other features but by a smaller margin. Previous work on this dataset with pdf-based algorithms achieved EERs of 3.6%, 7.6%, 10.3%, and 15.7% respectively with 1,000 digraphs [Ay19, Hu17, GP05, ÇU16]. The previous KDE state-of-the-art algorithm [Hu17] is overlayed on the DET curves for all test sample sizes except for 50 graphs as their was not enough data to match pdfs. Previous state-of-the-art achieves EERs of 3.6% with 1,000 graphs in the testing sample [Ay19] while our algorithm achieves an EER of 3.4% with 200 digraphs in the testing sample (see Tab. 3). With 100 graphs we are able to achieve an EER of 5.7% which is better than the recent state-of-the-art requiring 1,000 graphs [Hu17]. At 500 graphs our algorithm achieve an even better EER of 2.7%. Our fused classifier can authenticate users with 5-10 times less data and still achieve the same accuracy, a significant improvement over existing methods.

Features	EER at # of test digraphs			
	50	100	200	500
M	26.3%	23.7%	21.1%	19.1%
DD	16.7%	12.7%	10.6%	7.4%
UD	27.6%	23.7%	20.5%	16.7%
DU	19.0%	15.2%	12.9%	9.7%
UU	14.8%	10.5%	7.6%	5.9%
Fused	7.9%	5.7%	3.4%	2.7%

Tab. 3: EERs for different number of testing graphs for the five features individually and fused.

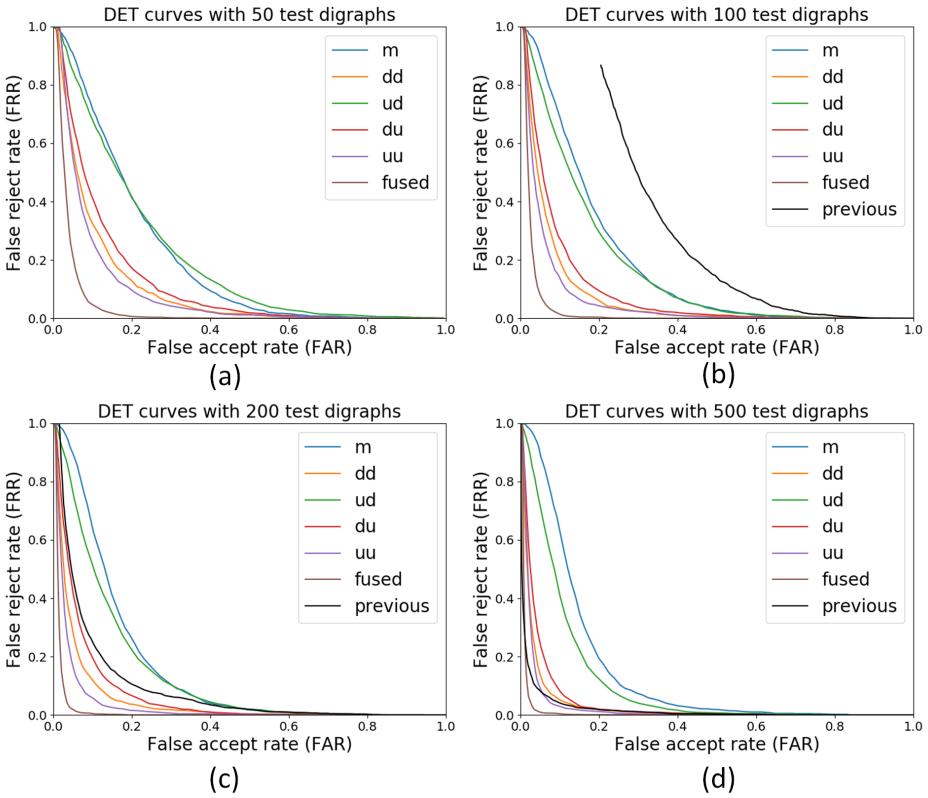

Fig. 3: DET curves for the five features individually and the fused case with 50, 100, 200, and 500 graphs in the testing sample. The "previous" curve denotes previous state-of-the-art work done by Ayotte et al. [Ay19].

It is important to recognize that the EER is a single number that represents algorithm performance at a particular operating point. To get a more complete picture of the performance of the algorithm, DET curves are presented in Fig. 3 for 50, 100, 200, and 500 testing graphs. DET curves plot FRR versus FAR and allow operators of the system to balance tradeoffs between security and convenience [Ma97]. An example of specific points on the DET curve is shown in Fig. 4. Point 'a' favors security over convenience with an FAR = 0.029 and FRR = 0.490, point 'b' is the equal error rate with FAR = FRR = 0.079, and point 'c' favors convenience over security with an FAR = 0.418 and an FRR = 0. From [BM15b], the ANGA and ANIA, in terms of graphs, for points 'a', 'b', and 'c' are shown in Tab. 4. To convert graphs to keystrokes see Tab. 1. Point 'a' clearly favors security as it has the lowest ANIA which detects intruders faster on average, however, it also has the lowest ANGA and will often reject the true user. Point 'c' on the other hand favors convenience never rejecting the true user, while taking the longest to detect an imposter. Point 'b' weights security and convenience equally and falls in between points 'a' and 'c'. The

DET curve allows our system to be highly flexible and lets users decide where to operate depending on application and the desired convenience and security.

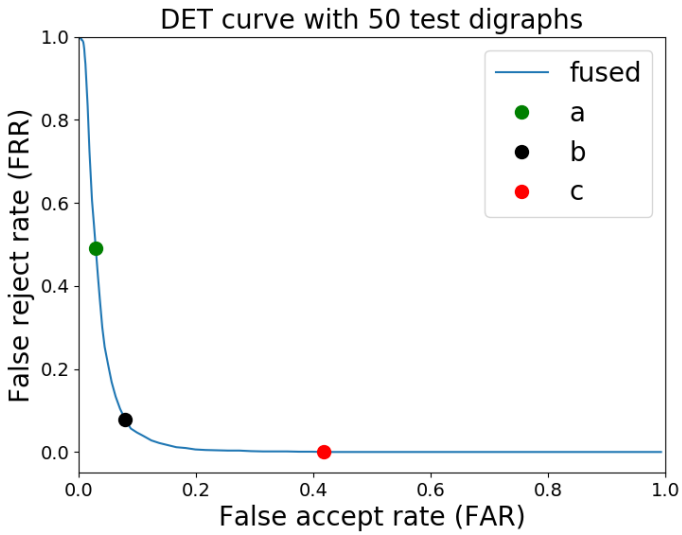

Fig. 4: DET curve for our fused classifier with 50 graphs in the testing sample. Points ‘a’, ‘b’, and ‘c’ provide specific examples of where our system could be operated depending on the desired convenience or security. The ANGA and ANIA for the three points are found in Tab. 4.

DET Curve Point from Fig. 4	ANGA	ANIA
‘a’	102	51
‘b’	633	54
‘c’	-	86

Tab. 4: Average number of Genuine Actions (ANGA) and Average Number of Imposter Actions (ANIA) in terms of graphs for three different points ‘a’, ‘b’, and ‘c’ on the DET curve in Fig. 4. Where to operate on the DET curve can be chosen depending on the desired convenience and security.

6 Conclusions and Future Work

In this paper, we present an instance-based algorithm for free-text keystroke dynamics to increase the speed of authentication. Previous free-text research used pdf-matching, which requires a significant number of keystrokes in order to be effective, during which time significant damage could be done. The instance-based method is very effective with fewer numbers of testing graphs, enabling faster and more frequent authentication. The most common features used in keystroke dynamics were investigated and their relative importances determined. These importances were used to construct a fused classifier, which achieved EERs of 7.9%, 5.7%, 3.4% and 2.7% with 50, 100, 200, and 500 graphs in the testing sample, respectively. Existing state-of-the-art algorithms only achieve EERs of

3.6%, 7.6%, 10.3%, and 15.7% with 1,000 digraphs in the testing sample [Hu17, Ay19]. Our algorithm was able to achieve comparable performance with only 100-200 graphs, a significant improvement over the state-of-the-art. Operating at point 'c', in Fig. 4, with a test sample size of 50 graphs or 78 keystrokes, our algorithm achieves the same ANGA as [BM15a] but with a lower ANIA of 86 graphs or 134 keystrokes (converted using Tab. 1).

Future work will include exploring other instance-based metrics and methods of fusing monograph and digraph features. The number of graphs used in the profile and the testing sample will be further investigated to reduce them as much as possible. Using our algorithm on multiple datasets will more robustly validate our results. Other possible research includes classification using principal component analysis (PCA) or support vector machines (SVM) and determining if the order of the typed characters impacts graph authentication performance.

References

[Al17] Ali, M. L.; Monaco, J. V.; Tappert, C. C.; Qiu, M.: Keystroke biometric systems for user authentication. Journal of Signal Processing Systems, 86(2-3):175–190, 2017.

[AW13] Alsultan, A.; Warwick, K.: Keystroke dynamics authentication: a survey of free-text methods. International Journal of Computer Science Issues (IJCSI), 10(4):1, 2013.

[Ay19] Ayotte, B.; Huang, J.; Banavar, M. K.; Hou, D.; Schuckers, S.: Fast Continuous User Authentication using Distance Metric Fusion of Free-text Keystroke Data. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2019.

[BEA09] Beasley, M. T.; Erickson, S.; Allison, D. B.: Rank-based inverse normal transformations are increasingly used, but are they merited? Behavior genetics, 39(5):580, 2009.

[Bl19] Black, P. E.: Manhattan distance. Available online at: https://www.nist.gov/dads/HTML/manhattanDistance.html. Last Accessed: 2019-06-15, 2019.

[BM15a] Bours, Patrick; Mondal, Soumik: Continuous authentication with keystroke dynamics. Norwegian Information Security Laboratory NISlab, pp. 41–58, 2015.

[BM15b] Bours, Patrick; Mondal, Soumik: Performance evaluation of continuous authentication systems. IET Biometrics, 4(4):220–226, 2015.

[Br01] Breiman, L.: Random forests. Machine learning, 45(1):5–32, 2001.

[Bu19] Burnett, M.: , 10,000 Top Passwords. https://xato.net/10-000-top-passwords-6d6380716fe0, 2019. Accessed: 2019-4-15.

[BW12] Banerjee, S. P.; Woodard, D. L.: Biometric authentication and identification using keystroke dynamics: A survey. Journal of Pattern Recognition Research, 7(1):116–139, 2012.

[ÇU16] Çeker, H.; Upadhyaya, S.: User authentication with keystroke dynamics in long-text data. In: 2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS). IEEE, pp. 1–6, 2016.

[Ga80] Gaines, S. R.; Lisowski, W.; Press, J. S.; Shapiro, N.: Authentication by keystroke timing: Some preliminary results. Technical report, Rand Corp Santa Monica CA, 1980.

[GP05] Gunetti, D.; Picardi, C.: Keystroke Analysis of Free Text. ACM Trans. Inf. Syst. Secur., 8(3):312–347, August 2005.

[GSG18] Gurdal, M. Mustafa; Sogukpinar, Ibrahim; Gebze: A Novel Method to Improve Performance of the Keystroke Dynamics based User Authentication. 2018.

[Ha94] Hamilton, J.: Time Series Analysis. Princeton: Princeton University Press, 1994.

[Ha17] Harilal, Athul; Toffalini, Flavio; Castellanos, John; Guarnizo, Juan; Homoliak, Ivan; Ochoa, Martín: Twos: A dataset of malicious insider threat behavior based on a gamified competition. In: Proceedings of the 2017 International Workshop on Managing Insider Security Threats. ACM, pp. 45–56, 2017.

[HHS17] Huang, J.; Hou, D.; Schuckers, S.: A practical evaluation of free-text keystroke dynamics. In: Identity, Security and Behavior Analysis (ISBA), 2017 IEEE International Conference on. IEEE, pp. 1–8, 2017.

[Hu16] Huang, J.; Hou, D.; Schuckers, S.; Upadhyaya, S.: Effects of text filtering on authentication performance of keystroke biometrics. In: WIFS. pp. 1–6, 2016.

[Hu17] Huang, J.; Hou, D.; Schuckers, S.; Law, T.; Sherwin, A.: Benchmarking keystroke authentication algorithms. In: Information Forensics and Security (WIFS), 2017 IEEE Workshop on. IEEE, pp. 1–6, 2017.

[JG90] Joyce, Rick; Gupta, Gopal: Identity authentication based on keystroke latencies. Communications of the ACM, 33(2):168–176, 1990.

[Jo01] Jones, E.; Oliphant, T.; Peterson, P. et al.: , SciPy: Open source scientific tools for Python, 2001.

[KM09] Killourhy, K. S.; Maxion, R. A.: Comparing anomaly-detection algorithms for keystroke dynamics. In: 2009 IEEE/IFIP International Conference on Dependable Systems & Networks. IEEE, pp. 125–134, 2009.

[Ma36] Mahalanobis, P. C.: On the generalized distance in statistics. National Institute of Science of India, 1936.

[Ma97] Martin, Alvin; Doddington, George; Kamm, Terri; Ordowski, Mark; Przybocki, Mark: The DET curve in assessment of detection task performance. Technical report, National Inst of Standards and Technology Gaithersburg MD, 1997.

[MB16] Mondal, Soumik; Bours, Patrick: Combining keystroke and mouse dynamics for continuous user authentication and identification. In: 2016 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA). IEEE, pp. 1–8, 2016.

[MB17] Mondal, Soumik; Bours, Patrick: Person identification by keystroke dynamics using pairwise user coupling. IEEE Transactions on Information Forensics and Security, 12(6):1319–1329, 2017.

[MJK15] Montgomery, D. C.; Jennings, C. L.; Kulahci, M.: Introduction to time series analysis and forecasting. John Wiley & Sons, 2015.

[Mu17] Murphy, C.; Huang, J.; Hou, D.; Schuckers, S.: Shared dataset on natural human-computer interaction to support continuous authentication research. In: 2017 IEEE International Joint Conference on Biometrics, IJCB 2017, Denver, CO, USA, October 1-4, 2017. pp. 525–530, 2017.

[Pe11] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[TTY13] Teh, P. S.; Teoh, A. B. J.; Yue, S.: A survey of keystroke dynamics biometrics. The Scientific World Journal, 2013.

[YC04] Yu, E.; Cho, S.: Keystroke dynamics identity verificationits problems and practical solutions. Computers & Security, 23(5):428–440, 2004.

[ZDJ12] Zhong, Yu; Deng, Yunbin; Jain, Anil K: Keystroke dynamics for user authentication. In: 2012 IEEE computer society conference on computer vision and pattern recognition workshops. IEEE, pp. 117–123, 2012.