International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Mouse Dynamics as Continuous User Authentication Tool

Anam Khan, Suhail Javed Quraishi, Sarabjeet Singh Bedi

Abstract: ***User Specific traits are a very strong methodto strengthen the security of any system as it makes thesystem connected to a specific individual instead of beingaccessed through some token, key, etc. Behavior-based user authentication with pointing devices, such as touchpads or mice, has been obtaining attention. Mouse Dynamicsis method which is inexpensive and provide uniquecharacteristic to prevent unlocked workstations attacks tolock out unauthorized users from accessing the system. Aperceptive survey with comparison on mouse dynamicsbiometrics study performed till now is the objective of thispaper. We consider here the best results reported in terms of FalseRejection Rate (FRR) &False Acceptance Rate (FAR).***

Keywords: ***Authentication, Biometric, FAR, FRR, Mouse dynamics, Supervised learning, Unsupervised learning.***

I. INTRODUCTION

It is very challenging to detect and defend insider attacks and that they have long been recognized as a serious concern within the cyber-security setting. The problem of insiders who misuse their privileges for malicious purposes is than the greatest threat [1]. Up-to-date study on computer crimes conducted by the US Computer Security Institute (CSI) indicates that the foremost serious losses in enterprises are due to unauthorized access or privilege escalation by insiders [4]. However, if we have a tendency to have some means that by that which we can identify, who gets access to the system, apart from the traditional password mechanism, we can significantly curb the threats delineate by the insiders [3]. One in every of the rising approach to deal with this problem is that the use of mouse dynamics, the analysis of mouse operating behaviour to be used as a behavioural biometric to check a user's identity. In comparison with other different biometric techniques such as voice or fingerprints, mouse dynamics is non-intrusive, and therefore the data is comparatively effortless to collect. Throughout interactions with the system using mouse, insiders could be detected and rejected as a result of their mouse operating styles are significantly completely different from those of legitimate users [8]. Moreover, the users' mouse operating characteristics is associate analyzed continuously throughout the subsequent interaction method to enforce an identity observation. Particularly, there are two different tasks of interest: verification and identification. Verification is termed to checks a user's claimed identity; whereas the identification, that must to establish a user's identity, might fight against insider threats. In spite of the very fact, many previous studies are conducted to evaluate the performance of different classification algorithms [8]-[15], these results are tough to compare with one other. Most of the studies approaches proposed within the literature

1) used completely different datasets;
2) built the classifier using differing sizes of training data;
3) designed different analysis procedures;
4) Adopted different types of threshold on classification results.

So many factors vary from different approaches which might make it not possible to assess the state of the art in mouse dynamics and to measure future progress.

II. BACKGROUND

Table I presents an outline of eight studies from the literature that use classification algorithms to evaluate mouse behavior data.

These studies offer experimental settings and results on completely different datasets. The first column offers a reference to the source study and therefore the remaining columns offers the following information:

Classifier: Classifier is the name for the classification algorithmic rule utilized in the study.

Environment: The experimental conditions under which the experiments were run within the study (*Controlled*: the environment was carefully set up to minimize environmental side-effects; *Uncontrolled*: no constraint on the experimental environment; *N/A*: corresponding details weren't available in the study).

Apparatus: The hardware utilized in the study (*Same*: users used the similar computer and mouse; *Completely different*: users used different computers and mice).

Users: Users are the number of subjects involved in the study.

Task: The type of mouse-operation task used in the study.

Feature Source: The source of features used to train and evaluate the classifiers within the study (*Click*: features were derived from mouse click actions; *Move*: features were extracted from mouse movement actions).

Manuscript published on November 30, 2019.
* Correspondence Author
Anam Khan*, Department of Computer Science & Engineering, Invertis University, Bareilly, UP, INDIA.
Suhail Javed Quraishi*, Department of Computer Science & Engineering, Invertis University, Bareilly, UP, INDIA. suhail.q@invertis.org
Dr. Sarabjeet Singh Bedi, Department of Computer Science & Information Technology, MJP Rohilkhand University, Bareilly, UP, INDIA.

Result Threshold: The name of the procedure for selecting a threshold on classification scores to get false-acceptance rates (FAR) and false-rejection rates (FRR) (*Heuristic*: threshold was chosen with a heuristic described within the study; *Equal-error*: threshold was chosen to get equal FARs and FRRs).

Results FAR/FRR: FAR is the percentage of the insiders that aren't detected and FRR is the percentage of the legitimate users that are erroneously detected as insiders.

Table I. Previous works to evaluate mouse behavior data

Source Study	Classifier	Data Collection				Feature Source		Result(%)		
		Environment	*Apparatus*	*Users*	*Task*	*Click*	*Move*	*Threshold*	*FAR*	*FRR*
Pusara et al. (2004)[9]	Decision Trees	uncontrolled	different	18	free		✓	heuristic	0.43	1.75
Gamboa et al. (2007)[10]	Weibull Distribution	uncontrolled	different	50	free		✓	equal-error	6.2	6.2
Ahmed et al. (2007)[11]	Neural Network	uncontrolled	different	22	free	✓	✓	equal-error	2.46	2.46
Kaminsky et al. (2008)[12]	Nearest Neighbor	controlled	same	15	free	✓	✓	equal-error	20.7	20.7
	Support Vector Machine								42.5	42.5
Revett et al. (2008)[13]	Outlier Counting	uncontrolled	N/A	6	fixed	✓	✓	heuristic	3.5	4
Aksari et al. (2009)[14]	Outing Counting	controlled	same	10	fixed		✓	equal-error	5.9	5.9
Bours et al. (2009)[15]	Levenshtein Distance	uncontrolled	different	28	free		✓	equal-error	26.8	26.8
Quraishi et al. (2019) [8]	One-class SVM	uncontrolled	different	23	free	✓	✓	heuristic	10	16

The necessary observation from the above survey is that, despite the surface similarities, the studies contain substantial variations beyond the mixed results. Most of the approaches proposed in the literatureused totally different datasets; designed the classifier using different sizes of training data; designed different analysis procedures and adopted different kinds of thresholds on classification results. Moreover, since these studies don't tend to be replicated, it's tough to pin the dissimilarity on any one thing. As a conclusion, it's unclear whether the results of those evaluations really reflect variations in classifiers, or differences among the computing environments or many other factors. If the higher factors were all controlled, we may able to discover that however well each classifier performs, and the way we can compare completely different classifiers with one another.

III. FEATURE CONSTRUCTION

A. Mouse Dynamics Features

Table 2 condenses the derived features during this study. We are able to characterized mouse behaviour on the basis of four basic properties: time, distance, speed, acceleration and speed. Every property was then analyzed separately, and translated into numerous mouse features. This study distributes these features into two categories: dynamic features and static features. Dynamic features characterize the motion habits of individual mouse actions like movement speed and acceleration. Static features characterize the constituents of mouse actions throughout Graphical User Interface (GUI) interactions like travelled distance and time. For elaborate explanations of the mouse features, users can refer to previous work [7]-[10], [19].

B. Distance-feature Transformation

As a result of high dimensionality and behavioural variability the raw mouse feature can't be used directly by a classifier. Hence, we advanced a procedure to transform raw mouse features to distance-based feature vectors such as the input for classifiers. At first, we produced the reference feature vector for every subject from the training feature vectors. Then the reference feature vector is generated for each subject:

Step 1: Calculate the pairwise distance vector of mouse features between every pair of training vectors. Use dynamic time warp [17] to calculate the distance vector of dynamic features between dynamic elements of feature vectors, and apply Manhattan distance to calculate the distance vector of static features.

Step 2: Concatenate distance vectors of static features and dynamic features along to get a distance vector.

Step 3: For every training feature vector, calculate the arithmetic mean distance between this vector and therefore the remaining training vectors, and discover the reference feature vector with minimum mean distance.

Then, given the reference feature vector for each subject, we have a tendency to work out the feature-distance vector between a brand new mouse feature vector and therefore the reference vector, to represent a new-fangled mouse behavioural sample.

In this paper, we have a tendency to use all mouse features as shown in Table II to generate the distance vector. For individual mouse movement, there are 2 distance-related features, 5 speed related features, 1 time-related feature, and 5 acceleration-related features, that are taken along and transformed to create a 13*8=104dimensional feature distance vector, characterizing every behavioral sample.

Table II. Different Mouse Features

Mouse Features	Definition
Traveled distance	The distance between two adjacent positions of mouse click actions.
Movement offset	The distance between the practical mouse trajectory and the ideal mouse trajectory.
Movement elapsed time	The time interval between starting point and ending point of mouse movements.
x-speed	The movement speed in abscissa direction.
y-speed	The movement speed in ordinate direction.
x-speed against distance	The movement speed compared to traveled distance in abscissa direction.
y-speed against distance	The movement speed compared to traveled distance in ordinate direction.
Average speed against distance	Average movement speed compared to accumulatively traveled distance.
x-acceleration	The movement acceleration in abscissa direction.
y-acceleration	The movement acceleration in ordinate direction.
x-acceleration against distance	The movement acceleration in compared to traveled distance in abscissa direction.
y-acceleration against distance	The movement acceleration in compared to traveled distance in ordinate direction.
Acceleration against distance	Average movement acceleration compared to accumulatively traveled distance.

IV. ALGORITHMS USED FOR MOUSE DYNAMICS

A. Euclidean

This basic classification algorithm [18] models for each mouse feature-distance vector as a point in p-dimensional space, wherever p is that the number of features within the vector. Throughout training, the mean value of training vectors for separately class is calculated (for each component classifier). Throughout testing, the score of every class is calculated as the Euclidean distance between the test vector and also the mean vector of that class. The resulting label is that the class with the minimum distance. In our estimation, we also constructed the Euclidean classifier with "normalized distance". The distinction is that within the testing phase, the score is calculated by "normalizing" the distance divided by the product of the norms of the test vector and mean vector for each class.

B. Manhattan

This algorithm [18] signifies the Euclidean classifier except that the distance measure is Manhattan distance. The distinction is within the testing phase, the score of separately class is calculated as the Manhattan distance between the mean vector and the test vector of that class. In our estimation, we also constructed the normed Manhattan classifier. The score of each class is calculated by normalizing the Manhattan distance using the same divisor such as the Euclidean (normed) classifier.

C. Mahalanobis

This algorithm [18] signifies the Euclidean and Manhattan classifiers however with a lot of complex distance measure. Throughout training, both the mean vector and covariance matrix of training vectors for every class are calculated. Throughout testing, the score of separately class is evaluated as the Mahalanobis distance between the mean vector and also the test vector. In our estimations, we also built the normed Mahalanobis classifier that's similar to the Euclidean (normed) classifier, but using the Mahalanobis distance.

D. Nearest Neighbor

A nearest-neighbor classification algorithm assumes that new mouse data from the user will resemble one or more of those in the training data [12]. Throughout training, the algorithm estimates the covariance matrix using training vectors for every class (for each component classifier).

Throughout testing, the algorithm calculates Mahalanobis distances, and also the distance of every class is calculated from the new vector to the training vectors for that class. The resulting label is the class with the littlest distance. In our analysis, we also assessed the Nearest neighbor classifier using Manhattan distance.

E. Outlier Counting

This algorithm was given by Revett and Jahankhani [13]. Throughout training, the algorithm calculates the mean and standard deviation of every component of training vectors for every class. Throughout testing, every component of the test vector is checked to understand whether it lies within ±1.5 standard deviations of the mean value for every class. The score of every class is that be a count of how many components in test vector exceed the range. The resulting label is that the class with the littlest count.

F. Neural Network (Standard)

This algorithm was given by Ahmed et al. [11]. Throughout training, a network is built for each class (for every component classifier), with p input nodes, 1 output node, and (2p+1) hidden nodes. Throughout testing, the test vector is run through the networks of all classes, and also the output of the networks was recorded. Denote {s1, s2,…,sk} to be the output of the networks for k networks; intuitively, if si is maximum value, the test vector is classified to the ith class.

G. Support Vector Machine

A one-class SVM generalizes the idea of mapping the data into a high dimensional feature space, and treating the origin as the only example from other classes [12]. For each class we built a SVM using training vectors with the RBF kernel function. During testing, the test vector was projected into the same space and the decision function was calculated to generate a score for each class. The resulting label is the class with the largest score. In our estimation, we also implemented the one-class SVM using a linear kernel function.

H. K-means

The assumption behind k-means clustering is that points from a similar subject are going to be close to one other, whereas points from totally different subjects are going to be so much apart [18]. Throughout training, the center of every of k clusters is calculated because the mean of all training data from that class (for each component classifier). Throughout testing, the distance between the test vector and every cluster is recorded because the classification score. The resulting label is that the class with the littlest distance.

V. CONCLUSION

Lots of classification algorithms have been given for recognizing users' identities using mouse dynamics. Although, it had not been possible to conduct a sound comparison of classifiers using the results in the literature because of inconsistent datasets, experimental conditions and methodologies. In this study, we implemented and evaluated few multi-class classifiers from the mouse dynamics and pattern-recognition literature for a user identification task by using a dataset collected from a controlled environment, and that we established which classifiers have top performances with lowest false negative identification rates on the dataset. We also provided an objective and repeatable calculation methodology that may be shared throughout the research community to evaluate new classifiers and assess progress.

REFERENCES

1. S. J. Quraishi, and S. S. Bedi, "Keystroke Dynamics Biometrics, A tool for User Authentication–Review," in Proc. IEEE 2018 International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 2018, pp. 248-254. doi: 10.1109/SYSMART .2018.8746932.
2. CENELEC, "European Standard EN 50133-1: Alarm systems. Access control systems for use in security applications," Part 1: System requirements, Standard Number EN 501331:1996/A1:2002, Technical Body CLC/TC 79, European Committee for Electrotechnical Standardization (CENELEC), 2002.
3. S. J. Quraishi, and S. S. Bedi, "On keystrokes as continuous user biometric authentication," International Journal of Engineering and Advanced Technology Volume 8, Issue 6, August 2019, Pages 4149-4153. DOI: 10.35940/ijeat.F9301.088619.
4. R. Richardson, 2010/2011 CSI computer crime and security survey, 2011.
5. D. M. J. Tax, "One-class classification: concept learning in the absence of counter-examples," Ph.D. dissertation, Delft Univ. Technology, Delft, The Netherlands, Jun. 2001.
6. Z. Jorgensen and T. Yu, "On mouse dynamics as a behavioral biometric for authentication," in Proc. 6th ACM Symp. Information, Computer and Communication Security, Hong Kong, pp. 476-482, 2011.
7. C. Shen, Z. M. Cai, X. H. Guan, and J. L. Wang, "On the Effectiveness and Applicability of Mouse Dynamics Biometric for Static Authentication: A Benchmark Study," in Proc. IAPR/IEEE Int. Conf. Biometrics, New Delhi, India, March, 2012.
8. S. J. Quraishi, and S. S. Bedi, "On Mouse Dynamics as Continuous User Authentication," International Journal of Scientific & Technology Research, Volume 8, Issue 10, October 2019.
9. M. Pusara and C. E. Brodley, "User reauthentication via mouse movements," in Proc. 2004 ACM Workshop on Visualization and Data Mining for Computer Security. Washington DC, USA, pp. 1-8, 2004.
10. H. Gamboa, A. L. N. Fred, and A. K. Jain, "Web biometrics: user verification via web interaction," in Proc. Biometrics Symp., Baltimore, MD, pp. 1-6, 2007.
11. A. A. E. Ahmed and I. Traore, "A new biometric technology based on mouse dynamics," IEEE Trans. Depend. Secure Comput., vol. 4, no. 3, pp. 165-179, Jul-Sep. 2007.
12. R. Kaminsky, M. Enevand, and E. Andersen, "Identifying game players with mouse biometrics," Tech. rep. University of Washington., Aug. 2008.
13. K. Revett, H. Jahankhani, S. T. de Magalhes, and H. M. D. Santos, "A survey of user authentication based on mouse dynamics," in Proc. 4th Int. Conf. Global E-Security, London, pp. 210-219, 2008.
14. Y. Aksari and H. Artuner, "Active authentication by mouse movements," in Proc. 24th Int. Symp. Computer and Information Science, Guzelyurt, pp. 571-574, 2009.
15. P. Bours and C. J. Fullu, "A login system using mouse dynamics," in Proc. 5th Int. Conf. Intelligent Information Hiding and Multimedia Signal Processing, Kyoto, Japan, pp. 1072-1077, 2009.
16. C. Shen, Z. M. Cai, X. H. Guan, H. L. Sha, and J. Z. Du, "Feature analysis of mouse dynamics in identity authentication and monitoring," in Proc. IEEE Int. Conf. Communication (ICC), Dresden, Germany, pp. 1-5, 2009.
17. D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," Advance in Knowledge Discovery in Database: Papers from the 1994 AAAI Workshop, pp. 359-37, Jul. 1994.

18. R. O. Duda, P. E. Hart and D. G. Stork, Pattern classification, John Wiley & Sons, second edition, 2001.
19. S. J. Quraishi, S. S. Bedi, and M. A. Chandra, "Continuous User Authentication via Mouse Dynamics," in Proc. IEEE 2019 International Conference on Advances in Computing, Communication & Automation (ICACCA 2019), Bareilly, India, 2018 (under publication).

AUTHORS PROFILE

Anam Khanis aM.Tech. scholar in CSEat Invertis University, Bareilly. She has received a B.Tech. Degree in Computer Science & Engineering from Invertis University, Bareilly. She is working on biometric systems and going topublishher work on authentication.

Suhail Javed Quraishi, pursued Bachelor of Engineering in Computer Science and Engineering (CSE) from MJP Rohilkhand University, Bareilly in 2003 and Master of Technology in CSE from Aligarh Muslim University in year 2010. He is currently working as an Assistant Professor in the Department of Computer Science and Engineering, Invertis University, Bareilly since 2010. He is a member of many computer societies like IEEE, CSI, IEI& IAENG. He has published more than 12 research papers in reputed international journals and conferences including IEEE. He is also the coordinator of NPTEL and IIRS-ISRO at Invertis University. His main research work focuses on Biometrics, System Security and IOT based system design. He has 12 years of teaching experience and 3 years of Industry Experience.

Dr. Sarabjeet Singh Bediis an Associate Professor in the Department of Computer Science and Information Technology, MJP Rohilkhand University, Bareilly. He pursued B.E. (CSE) with distinction, M.E. (CSE) Gold Medalist from NIT-TTR, Chandigarh and Ph.D. (IT) from Indian Institute of Information Technology, Gwalior. He has an experience of 23 years in Academic, Research and Administration. He has supervised 06 Ph.D. scholars. His research areas are Digital Image Watermarking, Network, Information Management and Security. He has published 58 research papers in reputed International journals and proceedings with 03 chapters in edited books. He introduced online Examination form system for 6.0 Lakh students and designed online admission system for approx. 1.8 Lakh students for MJP Rohilkhand University. He is currently holding many administrative positions in the university.