Behavioral authentication by cursor tracking

Felix Neutatz
School of Electronic Information and
Electrical Engineering
Shanghai Jiao Tong University
Email: neutatz@gmail.com

***Abstract*—We present an approach to a mostly unsupervised user authentication and identification based on mouse dynamics. Our hypothesis is that one can successfully identify a user on the basis of cursor movements. Our system identifies a user as unauthorized if the behavior within a 10 seconds period deviates sufficiently from the learned behavior of an authorized user. Our results for four users show that we can identify these as unauthorized based on their cursor dynamics with a false positive rate of 0% and a false negative rate of 20% on the authorized user data. Nevertheless, we have to research more thoroughly and use more data to validate our results. We point out that analysing cursor dynamics alone is not yet sufficient for a practical user authentication system.**

Keywords*—*user verification, mouse dynamics, anomaly detection.

I. Introduction

Authentication plays a major role in securing a system today. There are three main ways to tackle the problem of authentication. The user can be identified by knowledge (e.g. passwords, PIN), by possession (e.g. RFID [1], smart cards [2]) or by the user himself/herself (e.g. voice [3], face [4], iris [5] detection and fingerprint [6] recognition). [7] The first two ways have their weaknesses. The concept of possession is not bulletproof. The tokens can be stolen and/or copied. With increasing computation capabilities, passwords can be cracked using brute force attacks in shorter time periods. Especially when quantum computing is understood, it will be trivial to calculate almost any password. [8] Another problem of knowledge is the human factor. Humans forget. So you have to install processes for the case of resetting a password. These can be exploited using social engineering as a hacker used against Central Intelligence Agency (CIA) Director John Brennan. [9] Biometric authentication is a bit more secure, but these methods can also be cracked by e.g. taking a photo of the finger, iris, ... An interesting step ahead is research on gait authentication. [10] So instead of using the properties of a person, the actions and behavior is used to identify a person. The behavior of a person is hard to fake, maybe not at all. So it is interesting to apply this idea on personal computer authentication.

There are many ways to track the behavior of computer users, but in this paper we want to focus on the cursor position over time. This can be seen as a first step to research whether our approach works in general. Mouse tracking on its own is not enough for computer authentication since the user can navigate by keyboard only. So this model will be extended to keyboard input in the future.

Monitoring the behavior of user has one main advantage over the common authentication: The user doesn't need to log off the system. So there is no way that you forget to log off. This is one of the key advantages of behavioral authentication because it is less susceptible to insider attacks. According to the US State of Cybercrime Survey [11], ”Almost one-third (32%) say insider crimes are more costly or damaging than incidents perpetrated by outsiders.”. So this can be critical.

The quality of an authentication method is based on its accuracy, how fast it can recognize the user and how difficult it is to circumvent it. It is extremely hard to simulate a person's behavior, so this criterion is already met for this method. So in this paper we focus on achieving also the other two.

The remainder of the paper is structured as follows: Section II discusses the related work. Section III introduces user authentication via mouse dynamics and describes a mostly unsupervised learning method for modeling user behavior. Section IV summarizes and discusses future research directions.

II. Related work

One of the first papers on the idea of authentication by behavioral patterns was published by Denning in 1985 [12]. Moreover there are already a lot publications about authentication by analysing mouse dynamics. Pusara et al. show that you can differentiate users by their cursor movement with a very low error rate [7]. Zheng et al. extended their work by developing an angle-based feature set which could reduce the recording time to a time span of 20 mouse clicks. [13] Gamboa et al. came up with a 63-dimensional feature vector and showed the error for a different number of strokes (segment between two mouse clicks). [14] However, all these methods are purely supervised learning methods and based on training on data of several users.

In this paper we present a mostly unsupervised user authentication system that builds a model of a users behavior from his/her mouse dynamics without using data from other users. Therefore this approach introduces less privacy issues. Moreover we don't use mouse events in the data. This causes a disadvantage in comparison to the papers described before, because we have less information. Therefore we can exploit less entropy. But this will change in future work.

III. User authentication via mouse dynamics

As introduced in [7], we use an unsupervised method to identify users. We model the behavior of the authorized user and apply this model on the mouse movements of any user. The idea is that if this model reproduces the given data very good, the probability that the data originated from the authorized user is very high. This corresponds to the problem of an anomaly detection task. The norm is the user's model and the anomaly is an unauthorized user.

A. Data set

For the authorized user we recorded the mouse position for 1h. To test the model for unauthorized users we recorded 4 different persons for 10 seconds. This is a very marginal data set. Extending this data set or applying the method on another data set will be also part of future work. The recoding keeps track of the x- and y-coordinate of the cursor every 30 milliseconds. In comparison to Pusara [7] we can leverage about 3 times more data, because our monitoring frequency is higher. Our model is based on equidistant data points in time. Therefore we focus on this to ensure this constraint. This requires a lot of unnecessary computation. Therefore another task for future work would be to find a model which can incorporate non-equidistant data points. The coordinates are normalized by screen size.

B. Anomaly detection

To model the user behavior of an authorized user we use linear regression. The idea is to fit a model which is capable of reconstructing the training data. For example that means predicting the cursor position at time t given that you know at which position the mouse was 1, ..., k time steps ago. In this way we can calculate the coefficients accordingly and find the best model which excels in predicting and reconstructing sequential data. This approach was introduced by Dunning [15]. This is an unsupervised problem because we can only train on records of class "authorized user". So there is no value in the label of the observations which forces us to work without labels. Reformulating this problem to be a sequence reconstruction problem makes it supervised again.

To train a model we use linear regression. In the general case the model of linear regression is described in the following way:

$$y = a^T x \tag{1}$$

In our case linear regression solves the following problem:

$$y_t = a_0 + a_1 * x_{t-1*0.03s} + ... + a_k * x_{t-k*0.03s} \tag{2}$$

such that the sum of squared residuals is minimized:

$$\arg\min_a \sum (y - a^T * x)^2 \tag{3}$$

In this case linear regression solves a constrained optimization problem. The target values are within the interval [0,1] because otherwise this would mean it would be possible for the cursor to leave the screen. To solve this problem we set the predicted value to 0 if it is smaller than 0 and 1 if it is bigger than 1. In future work we will elaborate what the best optimization strategy is to solve such a problem.

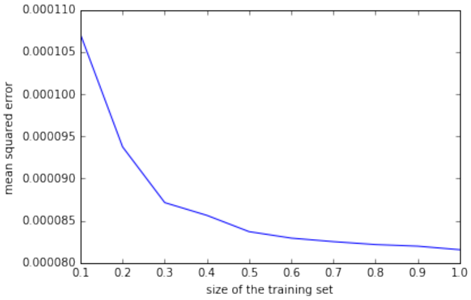

Fig. 1. Error by training set size

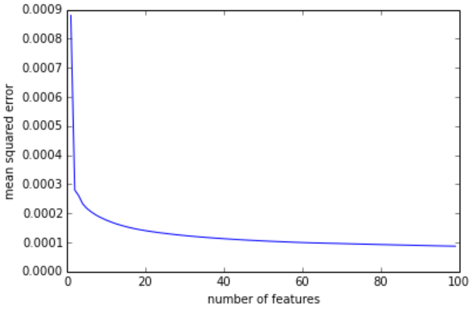

Fig. 2. Error by number of features

1) Training: For the training of the linear regression model for the authorized user we first analyse a recording of one hour. 40 % of the data is used as the validation set. To understand how much improvement we gain by training on more data we run an experiment using k = 100 (figure 1) which shows that the accuracy is starting to converge using the whole validation set (36 min). This means that it suffices to use 1h as training data.

2) Feature selection: The challenge of feature selection in this case is the trade-off between the information gain and how fast we can classify a sample. This means if we could decide that the model should predict the next mouse position by only knowing the previous cursor position. So theoretically we would only need 30 ms to be able to classify whether it is an authorized user. But the downside is that the model is not very good, because it is hard to learn only by one previous step. One the other hand we could decide to give the model the last 1000 steps. The model would have a high accuracy but that would also mean that we have to wait for 30 seconds until we are able to classify. So the question to answer is how many steps do we let the model look back into the past. We run an experiment to find a good middle way. Figure 2 shows the mean squared error on a test set using k features or looking k times steps back, respectively. It turns out that the error flattens more and more. It only decreases minimally after k=100. So we decide to use 100 time steps in our experiments.

3) Model evaluation: As test set we use a two hour recording of the authorized user. Table I shows that training and validation error are almost the same which is a clear sign that the model doesn't suffer from overfitting. The mean squared test error is twice as high as the validation error, but is still very low. These results show that the model is good in predicting cursor dynamics.

TABLE I. AUTHORIZED USER BEHAVIOR MODEL - ACCURACY (MEAN SQUARED ERROR)

Metric	Training error (36 min)	Validation error (24 min)	Test Error (120 min)
MSE	0.00007866	0.00007826	0.00016176
Distance normed MSE	0.0221	0.0183	0.0299
Smoothed MSE	3.23e-07	3.42e-07	4.02e-07
Distance smoothed MSE	0.179	0.168	0.160

4) Error metrics: We need a good error metric to be able to compare the error of an authorized user and a non-authorized user. First, we try mean squared error (showed in equation 4). The problem of the mean squared error is that it doesn't take into account the amount of "no movement". This means if there is no movement at all it is very easy to classify and therefore the mean squared error will be small. So the amount of "no movement" matters. To avoid this effect we delete all samples which have a past of no movement (all previous k time steps doesn't change in position). We apply this procedure also on the training data which increases the accuracy and speeds up training.

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - a^T * x_i)^2 \tag{4}$$

But there is still an issue with that metric. It is harder for the model to classify when you move the mouse fast. To incorporate this into the error metric we can divide by the distance which the cursor was moved instead of the number of samples (equation 5).

$$\frac{1}{distance(y)}\sum_{i=1}^{n}(y_i - a^T * x_i)^2 \tag{5}$$

But this metric is still not sufficient. There are still larger error peaks analysing the data of authorized users. These originate from rapid concept drifts. This means there is no way to read the mind of the user. For example, the user wants to close an editor and moves the cursor to the exit button, but on the way to the button he realizes he made a typographical error. So he moves the cursor to the mistake. This example shows clearly that sometimes it is impossible to predict the next step.

One approach to circumvent this problem is again to skip all records which are impossible to predict. But it is hard to say what impossible means. One way would be the following. We assume our model of the authorized user behavior is close to perfect whenever it is possible. This means whenever we get a higher error on an authorized user data set it is most probably a record which is impossible to predict. In order to find the right boundary we run an experiment. Figure 3 shows the smaller the threshold the smaller the resulting error (but the number of skipped records will be high). So if we set the threshold too low the error of unauthorized samples will be also low and we cannot differentiate anymore. Moreover the lower the threshold the more records are skipped and we have less data to classify on. So if the error of any record is higher than the threshold it is set to 0.

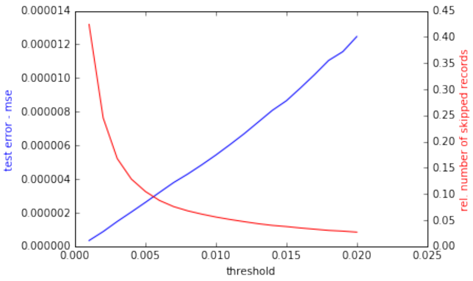

Fig. 3. Error by threshold

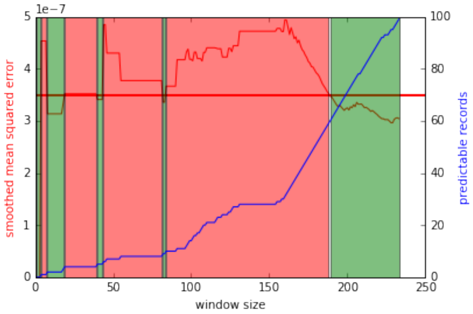

Fig. 4. Error by window size on authorized sequence

It turns out that the model for the authorized user also works very well for unauthorized users. So there is no profound difference in the mean squared error. But if we set the threshold to be very small (t = 0.001), the difference is greater. The impact on the authorized data can be seen in table I. Smoothing the error by a threshold reduces the variance and therefore makes the test error and the validation error more alike.

C. Classification window

To find the right window for classifying whether a user is authorized or not, we run some experiments. Figure 4 shows the smoothed mean squared error by window size and the resulting classification for a sequence of an authorized user. Moreover the blue line shows the number of records which were classified as predictable. The straight red line is the decision boundary. When the error is greater than this threshold the sequence is classified as unauthorized. The first 100 records are used as features. This means classification can start after 3 seconds. The figure shows that the chart converges to the right classification after about 190 time steps. We have to add the time steps of building the features. In total, we need in this case 290 time steps to classify correctly. This corresponds to 9.7 seconds. Also other examples show similar characteristics. Therefore 10 seconds seems to be a good estimate for a prototype. We will have to research in more detail whether we can reduce the window size because the longer the window the longer the potential unauthorized user can use the system.

D. Evaluation

To decide whether a data sequence is authorized or not, we need a decision boundary. We choose it by the lowest mean squared error which is seen for unauthorized users. We recorded 4 users for 10 seconds. This data serves as the unauthorized data. Using this approach we achieve a false accept rate (FAR) of 0%, but a very high false reject rate (FRR) of 20% on the test set. It turns out that the distance normed error is not suitable to compare the sequences because it doesn't seem to correlate with being authorized as well as the mean squared error based on time.

IV. Conclusion and future work

We presented a mostly unsupervised approach to identify users by only training on the data of authorized user's mouse dynamics. The results show that mouse dynamics contain a good amount of information, but it does not yet suffice for practical applications. The false reject rate of 20% is too high. Since most of the other publications also included mouse events (i.e. [13], [7]) into their model. Our results cannot be compared directly.

Future work has to focus on adding more features like mouse and keyboard events and how to incorporate them into the overall pipeline. This means we have to find a way to merge continuous cursor data with discrete events. The model should be also able to handle non-equidistant data points.

Moreover we have to gather more user data to find a better estimate how to configure the threshold for impossible records and to set up the right decision boundary. Another way to improve the current method is to find an algorithm which better fits the mouse dynamics. One promising candidate could be Long Short-Term Memory (LSTM) Recurrent Neural Networks [16].

The source code for this project can be found on Github: https://github.com/FelixNeutatz/BehavioralAuthentication

Acknowledgment

Thank you, to all my fellow students who helped me to gather the data on which this work is based. Moreover I like to thank Professor Zhang Aixin for making it possible to put so much time and effort in this interesting topic.

References

[1] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for rfid systems using the aes algorithm," in *Cryptographic Hardware and Embedded Systems-CHES 2004*. Springer, 2004, pp. 357–370.

[2] V. Deo, R. B. Seidensticker, and D. R. Simon, "Authentication system and method for smart card transactions," Feb. 24 1998, uS Patent 5,721,781.

[3] P. R. Kennedy, T. G. Hall, and W. C. Yip, "Radio telecommunication device and method of authenticating a user with a voice authentication token," Jul. 4 2000, uS Patent 6,084,967.

[4] C. Mallauran, J.-L. Dugelay, F. Perronnin, and C. Garcia, "Online face detection and user authentication," in *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, 2005, pp. 219–220.

[5] S. C. Chong, A. B. J. Teoh, and D. C. L. Ngo, "Iris authentication using privatized advanced correlation filter," in *Advances in Biometrics*. Springer, 2005, pp. 382–388.

[6] P. Gupta, S. Ravi, A. Raghunathan, and N. K. Jha, "Efficient fingerprint-based user authentication for embedded systems," in *Design Automation Conference, 2005. Proceedings. 42nd*. IEEE, 2005, pp. 244–247.

[7] M. Pusara and C. E. Brodley, "User re-authentication via mouse movements," in *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*. ACM, 2004, pp. 1–8.

[8] A. Steane, "Quantum computing," *Reports on Progress in Physics*, vol. 61, no. 2, p. 117, 1998.

[9] K. Zetter, "Teen Who Hacked CIA Directors Email Tells How He Did It," http://www.wired.com/2015/10/hacker-who-broke-into-cia-director-john-brennan-email-tells-how-he-did-it/, 2015, [Online; accessed 11-November-2015].

[10] D. Gafurov, K. Helkala, and T. Søndrol, "Biometric gait authentication using accelerometer sensor," *Journal of computers*, vol. 1, no. 7, pp. 51–59, 2006.

[11] N. T. A. C. US Secret Service, U. S. of America, C. D. of the Software Engineering Institute, U. S. of America, C. Magazine, and U. S. of America, "Us cybercrime: Rising risks, reduced readiness key findings from the 2014 us state of cybercrime survey," 2014.

[12] D. E. Denning and P. G. Neumann, "Requirements and model for idesa real-time intrusion detection expert system," *Document A005, SRI International*, vol. 333, 1985.

[13] N. Zheng, A. Paloski, and H. Wang, "An efficient user verification system via mouse movements," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 139–150.

[14] H. Gamboa and A. L. Fred, "An identity authentication system based on human computer interaction behaviour." in *PRIS*, 2003, pp. 46–55.

[15] T. Dunning and E. Friedman, *Practical Machine Learning: A New Look at Anomaly Detection.* " O'Reilly Media, Inc.", 2014.

[16] I. Sutskever, O. Vinyals, and Q. V. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112. [Online]. Available: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf