

User Re-authentication via Mouse Movements and Recurrent Neural Networks

Paul R. B. Housel¹ ^a and Luis A. Leiva¹ ^b

¹University of Luxembourg, Luxembourg

^aname.surname@alumni.uni.lu, ^bname.surname@uni.lu

Keywords: Mouse Movements; Biometrics; Authentication; Neural Networks

Abstract: Behavioral biometrics can determine whether a user interaction has been performed by a legitimate user or an impersonator. In this regard, user re-authentication based on mouse movements has emerged as a reliable and accessible solution, without being intrusive or requiring any explicit input from the user other than regular interactions. Previous work has reported remarkably good classification performance when predicting impersonated mouse movements, however, it has relied on manual data preprocessing or ad-hoc feature extraction methods. In this paper, we design and contrast different recurrent neural networks that take as input raw mouse movements, represented by discrete sequences of coordinate derivatives (coordinate offsets relative to time), as a mean of user re-authentication that could be used on web platforms. We show that a 2-layer BiGRU model outperforms state-of-the-art approaches while being much simpler and more efficient. Our software and models are publicly available.

1 INTRODUCTION


Our fingerprints, face, and eyes are among the most used biometric schemes as a mean of authentication, since they offer a reliable, secure, and accessible solution [Abdulrahman and Alhayani, 2021]. Although they are considered intrusive, in the sense that they require an explicit user intervention to work, such as approaching the finger or eyes to a special reading device. While browsing the web, however, our mouse movements can be considered as an alternative option [Leiva et al., 2021, Lin et al., 2012]. They can be collected unobtrusively, in the background, while the user is naturally using their computer mouse. Therefore, mouse movements can serve as a low-cost implicit secondary mean of re-authentication. This method, also defined as continuous authentication, seeks to verify the user's identity during ongoing web sessions to ensure that their authorization remains valid, while being unintrusive.


Previous work has shown that mouse movements can disclose a lot of information about ourselves and our behaviors. For example, it is not only possible to determine simple demographics such as gender [Yamauchi et al., 2015] or age [Leiva et al., 2021], but

it goes as far as predicting feelings [Yamauchi and Bowman, 2014]. This information can be deemed sufficient to authenticate a person using mouse movements, and in fact, many researchers have experimented with it for dynamic user profiling [Shen et al., 2012, Oak, 2018, Almalki et al., 2021, Muda et al., 2017, Eberz et al., 2017].

Many researchers have benchmarked their mouse-based authentication methods on the Balabit dataset [Fülöp et al., 2016], which includes, among other information, the timing and mouse coordinates of ten users across several browsing sessions. Some studies have achieved remarkably good accuracy for impersonation detection [Revett et al., 2008, Almalki et al., 2019], however, they rely on manual data preprocessing (e.g. to filter unwanted mouse actions) and computationally expensive feature extraction methods.

To solve this research gap, we study different Recurrent Neural Nets (RNNs) that do not rely on cumbersome feature extraction or data preprocessing, just on the mouse movement coordinates themselves. We show that a 2-layer BiGRU model outperforms state-of-the-art approaches while being much simpler and more efficient. Taken together, our results can inform researchers interested in developing their biometrics solutions at web scale.

^a  <https://orcid.org/0009-0009-8302-4393>

^b  <https://orcid.org/0000-0002-5011-1847>

2 RELATED WORK

Prior work has approached user authentication using mouse movements by either introducing different ways of preprocessing the data or using classic Machine Learning models. While some approaches have shown promise [Pramila et al., 2022, Antal and Egyed-Zsigmond, 2019, Antal and Fejer, 2020], most of them rely on computationally expensive feature extraction methods, including e.g. average angle between two consecutive coordinates, directionality of movement, minimal values of angular velocity, etc. All in all, it seems clear that mouse movements offer an ideal solution for re-authentication on websites and web applications.

Manual feature extraction comes with important drawbacks and challenges [Jorgensen and Yu, 2011]. First of all, it is not very scalable [Li et al., 2017], since many different (and sometimes computationally demanding) features must be extracted in almost real-time, in order to not degrade the user’s browsing experience (i.e., users should not wait more than a second to be re-authenticated). Further, not only do these classic models become unnecessarily complex, but also take longer to train [Cai et al., 2018]. Since a user-dependent model is created for every user that accesses an application, it is therefore of utmost importance to keep the authentication model as simple and as performant as possible. Another problem is that the vast majority of previous work has relied on additional information other than the mouse movements themselves, also known as “mouse actions”. For example, whether the user is scrolling, clicking, or using the drag-and-drop functionality. By relying on these additional events, it is possible to segment mouse movements and thus keep only the “interesting” or relevant parts. However these make an authentication system more limited, since a user might just move the mouse without clicking or performing any of those above-mentioned actions.

In this context, previous work has proposed different preprocessing techniques applied to the raw mouse movements (coordinates and optionally their associated timestamps). For example, Tan et al. proposed curve fitting to smooth mouse movements and thus remove noise [Tan et al., 2017]. They used a Linear SVM model for classification and achieved better performance as compared to using raw movements. These results are counter-intuitive, as previous work has shown that subtle details and imperfections in our mouse movements are key to telling humans and machines apart [Leiva et al., 2020].

Qin et al. used Dynamic Time Warping and a segmentation algorithm to allow mouse movements to be

differentiated among themselves [Qin et al., 2020]. These distances served as input to a classification model. A more interesting approach was proposed by Chong et al., who used movement heatmaps as input to a 2D Convolutional Neural Network (2D-CNN) classifier [Chong et al., 2018]. Other authors have followed this approach [Wei et al., 2019, Hu et al., 2019]. The main problem is that generating heatmaps may require as much computational resources as doing manual feature extraction, so they are hardly scalable in practice, as it is not feasible to generate a heatmap image every time a user should be re-authenticated.

More recent solutions do not require manual feature extraction [Antal et al., 2021, Antal and Fejer, 2020, Fu et al., 2020], suggesting that it is possible to handle raw movements with Deep Learning models [Chong et al., 2019, Levi and Hazan, 2020, Hema and Bhanumathi, 2016], however, they rely on explicit segmentation based on mouse actions. As previously stated, these approaches have a limited application as not every user is always using mouse actions while browsing. We therefore compare and contrast RNNs relying on raw mouse movements alone.

3 METHOD

3.1 Dataset

We used the Balabit Mouse Challenge dataset [Fülöp et al., 2016] to compare our method against others, given its popularity among the web biometrics community. It comprises mouse movements collected from ten different users across several sessions. These users were asked to log in with their remote desktop client. A network monitoring device was set between the client and the remote computer that inspects all web traffic, including any mouse interactions, e.g. coordinates and timestamps, event actions (dragging, moving, pressed, released, etc.) and how such actions were initiated (left or right button, scroll, or none). A session is either considered *legal* (the recorded mouse movements belong to the legitimate user) or *illegal* (the recorded mouse movements belong to another user).

3.2 Data splits

The dataset is randomly split into training and test partitions of 90% and 10%, respectively. Since in some of our experiments, the data classes are unbalanced, the proportion of each class is the same in the testing and training sets (stratified data splits). To present representative results, for every experiment,

the compiled model is trained and evaluated 5 times using 5 different random seeds, and the average metrics are computed and presented as the results.

3.3 Data normalization

Since in our work we only consider x, y, t tuples (cf. subsection 3.1), we ignore every other column of the original dataset CSV files. The mouse movement coordinates are then normalized over time, by computing the difference between each consecutive coordinate and dividing it by the corresponding timestamp offset. This has proved effective in working with mouse movements in web-related experiments [Brückner et al., 2020]. We illustrate in Figure 1 what raw and normalized coordinates look like.

3.4 Data augmentation

In order to get more samples for model training, we tried two different ways to augment the legal data. First, by perturbing the coordinates with small random noise [Brückner et al., 2020, Leiva et al., 2021]. However, this method was discarded since it worsened the results. Second, by padding with zeros the sequences that had less than 600 coordinates, until completing that length. Note that padding values can be added either at the beginning or at the end of the sequence (also known as post- and pre-padding, respectively), however previous work noted that pre-padding was preferred for RNN training [Dwarampudi and Reddy, 2019], therefore we adopted pre-padding as our data augmentation method. Further, we considered negative data (mouse movements from sessions that belong to other users) in addition to positive data (legitimate mouse movements) for model training. For each user, illegal mouse movements are obtained by randomly selecting legal mouse movements from other users in the dataset.

3.5 Evaluation metrics

We report balanced Accuracy, Area Under the ROC (AUC) score, and Equal Error Rate (EER). Together, these evaluation metrics are the most representative ones for biometric authentication systems, which helps us to compare our results against previous work. Balanced Accuracy is weighted by class distributions, AUC informs about the discriminative power of any classifier, and EER is the location on a ROC curve where the false acceptance rate and false rejection rate are equal.

3.6 Models

To decide upon the design and architecture of our model, we conducted different experiments on the mouse movements of user 7 (chosen at random as a reference user) in the Balabit dataset. These experiments guided us toward the best-performing model for predicting anomalies in the mouse movements of that user. This final model was then evaluated on all users of the Balabit data set.

Critically, given that a mouse position depends on the previous positions, we need to come up with a model that can process sequential data and that has some memory, to remember the dependencies between coordinates at different timestamps. Therefore, it seems natural to experiment with RNN-based architectures [Ackerson et al., 2021]:

1. Vanilla RNN, a neural network designed for handling time-series which can remember short-term dependencies [Sherstinsky, 2020]. It was first introduced as the *Hopfield Net* [Hopfield, 1982].
2. Long Short-Term Memory (LSTM), an extension of RNN which does not have the vanishing gradient problem [Hochreiter and Schmidhuber, 1997].
3. Gated Recurrent Unit (GRU), a version of LSTM with a forget gate and fewer hyperparameters [Cho et al., 2014]. GRU layers are witnessing great performance on small datasets, like Balabit.

For each of these architectures, we consider its bidirectional version. This allows the network to learn relationships between previous and future mouse movements at a certain time. Given the excellent performance of the Bidirectional GRU model, as measured by the AUC score (Figure 2), we chose it for further finetuning:

- One Input layer with 600 input neurons, to process all mouse sequence lengths in the Balabit dataset.
- Two Bidirectional GRU layers with Hyperbolic Tangent activation function and 200 hidden units.
- One Dropout layer with a dropout rate of 0.20.
- One Output layer with one neuron, predicting the probability that the input mouse movements originated from an impostor or not. This layer uses the Sigmoid activation function.
- And the following design parameters:
 - (a) Batch Size of 150 mouse sequences.
 - (b) Adam optimizer with a Learning Rate of 0.005.
 - (c) Binary Cross Entropy loss function.

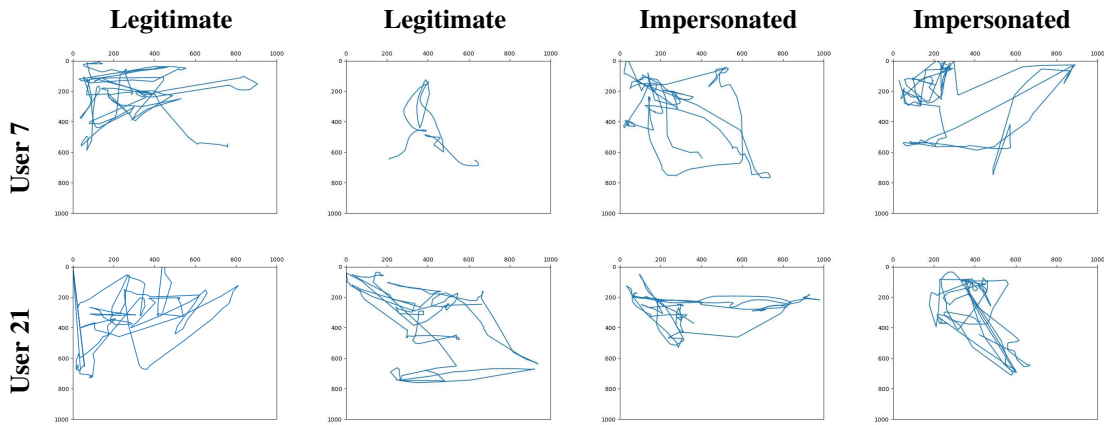


Figure 1: Comparison of legitimate and impersonated mouse movements for two users in the Balabit dataset.

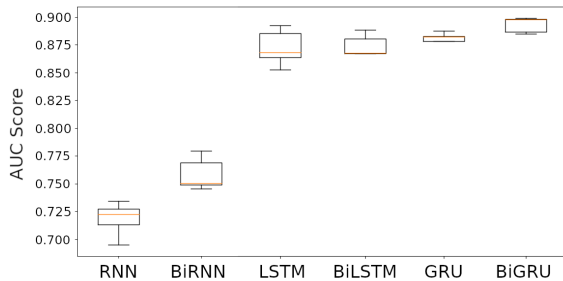


Figure 2: Model comparisons, according to AUC scores.

4 EXPERIMENTS

After finetuning our model, we can evaluate how it performs on the set of all users. While doing so, we also need to specify one last parameter that was not taken into account yet, the amount and distribution of legal and illegal data. We are not only going to ask ourselves how much positive and negative data benefits the most for training the model, but also how the proportions of these two influence the final model performance.

Another problem we need to address is the lack of data. When collecting the mouse movements of a new user, we may assume that only the first session, when the user is signing up into his account, is a legal session that can be trusted. This implies that we can only collect data of new users during their first sessions, limiting the amount of mouse data that can be collected for each user later on. In this series of experiments we want to find out if training with larger amounts of negative (illegal mouse movements) data is a solution when dealing with small amounts of positive (legal mouse movements) training data. Below, we defined four experiments, for which we evaluate different data distributions and how they impact the final performance of an authentication system.

- (A) In this first experiment, we will, for every user, take into account all available legal and illegal data.
- (B) To balance out the proportion between legal and illegal data, the negative training data needs to be augmented. For a given user, legal data from any other random user is taken as illegal data for the given user. This illegal data is added to the dataset until both legal and illegal data are equally balanced. In this experiment, there is as much illegal as legal data for every user.
- (C) In this third experiment, we investigate an augmentation technique to slightly increase the amount of legal mouse movement data. Since the input layer has 600 neurons, each session in the dataset is split into chunks of 600 timestamps. For some of them, the remaining amount of timestamps is inferior to 600. As explained in subsection 3.3, we pad these sessions with zero dummy values such that they have a final length of 600 mouse movements. In this experience, the legal data is augmented by adding padded sequences, there is the same amount of negative data as in experiment B.
- (D1) In this experiment, the amount of negative data is augmented as in experiment B. By taking legal data from other random users, we obtain twice as much negative data as positive data. Furthermore, we add the padded sequences as legal data.
- (D2) Here the distribution of the experiment is the same as in D1 but without taking into account the padded sequences.

Since each experiment deals with unbalanced data, class weights are set to prevent the classifier from being biased toward the illegal or legal class. We

User	EER ↓	Accuracy ↑	AUC Score ↑
7	0.3048	0.55	0.8286
9	0.2500	0.725	0.8875
12	0.3403	0.5875	0.7597
15	0.4245	0.5279	0.6107
16	0.2677	0.6085	0.8338
20	0.5294	0.4941	0.5000
21	0.6125	0.5000	0.4458
23	0.5850	0.5138	0.5000
29	0.4556	0.5000	0.5593
35	0.1242	0.8046	0.9425
All users	0.3894	0.5811	0.6868

Table 1: Results of Experiment A.

report Balanced Accuracy (weighted by class distribution) and weighted AUC Score. For each experiment, the models are trained for over 400 epochs with early stopping, monitoring the validation loss: the training is stopped if this metric is not improving over 40 consecutive training epochs and the optimal model weights are retained.

5 RESULTS

The results in experiment A show that the lack of negative training data significantly affects model performance. In all cases, it can be noted that users 15 and 35 have most of the time the worst results. On the other hand, the experiments D1 and D2 have the most promising results. It shows us that training the model with more negative data is a feasible solution in case we lack positive data. As such, we obtained promising authentication with simply 80 seconds of legal mouse movements. Furthermore, the unbalancing of the data does not affect model performance. Since both of these experiments have very similar results, we perform an investigation on possible overfitting.

By comparing the evaluation accuracy and loss with the number of epochs over the training period, we identified the presence of overfitting in all experiments except in D2. We noticed that padding mouse sequences do not add value to the model and can even diminish performance. Because bidirectional GRUs learn from a sequence of data and attempt to identify relationships between future and past coordinates, including zero values with pre-padding in the input prevents them from forming this relationship.

User	EER ↓	Accuracy ↑	AUC Score ↑
7	0.1117	0.9349	0.9394
9	0.0525	0.9355	0.9850
12	0.2036	0.8762	0.8688
15	0.1241	0.8828	0.9517
16	0.0615	0.9462	0.9820
20	0.1000	0.9562	0.9797
21	0.1175	0.9097	0.9533
23	0.1000	0.9125	0.9719
29	0.0562	0.9829	0.9850
35	0.0824	0.9412	0.9806
All users	0.1010	0.9278	0.9597

Table 2: Results of Experiment B.

User	EER ↓	Accuracy ↑	AUC Score ↑
7	0.0609	0.9435	0.9750
9	0.0846	0.9212	0.9647
12	0.1829	0.8857	0.8844
15	0.1091	0.9212	0.978
16	0.1572	0.8847	0.9221
20	0.0680	0.9714	0.9752
21	0.0235	0.9714	0.9961
23	0.1684	0.8789	0.903
29	0.0737	0.9526	0.9839
35	0.1462	0.8683	0.9586
All users	0.1074	0.9199	0.9541

Table 3: Results of Experiment C.

User	EER ↓	Accuracy ↑	AUC Score ↑
7	0.0478	0.9696	0.9803
9	0.0360	0.9613	0.9882
12	0.0914	0.92	0.9621
15	0.1303	0.8818	0.9501
16	0.0546	0.9559	0.9700
20	0.0286	0.9739	0.9936
21	0.0340	0.9746	0.9803
23	0.0474	0.9632	0.9787
29	0.0263	0.9763	0.9898
35	0.0195	0.9379	0.9944
All users	0.0516	0.9515	0.9788

Table 4: Results of Experiment D1 (with padding).

6 DISCUSSION

Our experiments provide evidence about the usefulness and effectiveness of mouse movements as an online user re-authentication method; i.e. after the user has logged in to the application. In a nutshell, our model can predict whether mouse movement data come from a legitimate user or an impersonator, with excellent performance (95.3% Accuracy and 98.6% AUC), establishing new state-of-the-art results. Our experiments show that training our model with a 2:1

User	EER ↓	Accuracy ↑	AUC Score ↑
7	0.0	0.9905	1.0
9	0.0444	0.9839	0.9810
12	0.0834	0.9316	0.9563
15	0.1000	0.8983	0.9684
16	0.0269	0.9615	0.9966
20	0.0000	0.9438	1.0000
21	0.0585	0.9581	0.9819
23	0.0625	0.9156	0.9848
29	0.0000	1.000	1.000
35	0.0588	0.9559	0.9962
All users	0.0434	0.9539	0.9865

Table 5: Results of Experiment D2 (without padding).

negative:positive data ratio further improves performance. Our experiments also show that sequence pre-padding slightly decreases classification performance, so it should be avoided.

We now have to ask ourselves how this model compares against previous work. As discussed in section 2, we identified only one state-of-the-art model that could be compared to our work. Antal et al. [Antal and Fejer, 2020] trained a 1D-CNN model on the Balabit dataset and achieved 93% Accuracy and 98% AUC. Their model did not use ad-hoc feature engineering but required explicit segmentation of mouse actions. Our model achieved better performance using a simpler architecture that requires no data preprocessing. Other competitive approaches are reported in Table 6.

6.1 Limitations and Future Work

The main limitation of our mouse-based authentication method is that it requires mouse movements to work, so it cannot be used on mobile devices, where only a limited number of interactions (e.g. taps or scrolls) is available. Previous work has proposed to use mobile touch interactions for biometric authentication [Jorgensen and Yu, 2011, Yazji et al., 2009] but the achieved classification performance was not ready for production use. For a mobile scenario, it may make more sense to rely on browser fingerprinting techniques that could uniquely profile each user based on hardware and browser settings, but this can only prove something the user *has* (the mobile phone), not what the user *is* (how they move their mouse).

Previous work has shown that a mouse-based biometric system can be susceptible to replay attacks, where the attacker captures or imitates the victim’s mouse movements [Tan et al., 2019, Lee et al., 2019, Lee et al., 2016]. To address this, and thus avoid bypassing the biometric system, additional measures should be considered. For example, not allowing the

same sequence of mouse movements to be considered for analysis or using full timestamps as an additional (automatic) feature for classification, so that the biometric system can compare the actual time against the submitted mouse movements’ time.

For future work, it would be interesting to combine mouse movements with other input modalities. For example, previous work has experimented with keyboard presses [Zheng et al., 2022, Handa et al., 2019, Fridman et al., 2015, Traore et al., 2012, Roth et al., 2014, Thomas and Mathew, 2022], however, the reported results are no better than ours. Other authors have proposed to combine mouse and eye movements [Rose et al., 2017, Liu et al., 2020] but their proposed classifiers degrade with an increasing number of users. For example, from 93% of F1-score with 5 users to 37% with 32 users [Rose et al., 2017]. Together with the fact that eye-tracking devices are expensive, these approaches are rendered impractical.

Finally, as hinted in INTRODUCTION section, previous work has shown that mouse movements encode sensitive information about the user [Leiva et al., 2021], therefore privacy issues may emerge if a classifier like ours is deployed without informing the user or requiring their explicit consent. Overall, we believe it is important to reflect on the tradeoffs between privacy and technological innovation, and the impact that unethical practices may have on users.

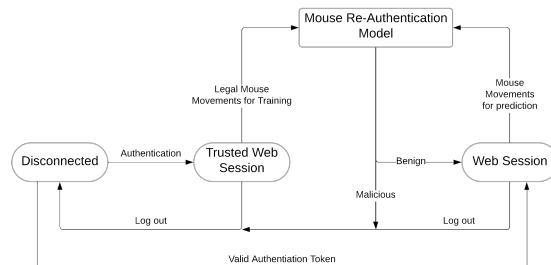


Figure 3: System diagram of user re-authentication based on mouse movements. Legal mouse movements are collected on trusted sessions, established with primary authentication methods. Authenticated sessions are then monitored via mouse movements to verify impersonation.

7 CONCLUSION

We have presented a new approach for user re-authentication using raw mouse movements as sole input. Our approach, a 2-layer Bidirectional GRU, is much simpler than any other model proposed in previous work and can be trained for any user with just 80 seconds of mouse movement data which can

Ref.	Features	Eval. metrics (%)			Notes
		EER ↓	Acc. ↑	AUC ↑	
[Tan et al., 2017]	Smooth coords.	0.18	–	86.0	SVM
[Chong et al., 2018]	Heatmaps	0.10	–	93.0	2D-CNN
[Antal and Fejer, 2020]	Coord. offsets	–	93.0	98.0	1D-CNN, explicit segmentation
Ours	Coord. offsets	0.04	95.3	98.6	BiGRU, no segmentation

Table 6: State-of-the-art results on the Balabit dataset using automatic feature engineering approaches. Cells with ‘–’ denote a result not reported in the respective paper. The best result is denoted in bold font.

be collected on trusted sessions established after a primary user authentication (Figure 3). Critically, no manual preprocessing and no feature extraction methods are needed, thereby making our classifier suitable for practical real-time applications. Our software is publicly available at <https://github.com/jetlime/Mouse-Movements-Re-authentication>.

ACKNOWLEDGEMENTS

This work is supported by the Horizon 2020 FET program of the European Union through the ERA-NET Cofund funding (BANANA, grant CHIST-ERA-20-BCI-001) and Horizon Europe’s European Innovation Council through the Pathfinder program (SYM-BIOTIK, grant 101071147).

REFERENCES

- Abdulrahman, S. A. and Alhayani, B. (2021). A comprehensive survey on the biometric systems based on physiological and behavioural characteristics. *Materials Today: Proceedings*.
- Ackerson, J. M., Dave, R., and Seliya, N. (2021). Applications of recurrent neural network for biometric authentication & anomaly detection. *Information*, 12(7).
- Almalki, S., Assery, N., and Roy, K. (2021). An empirical evaluation of online continuous authentication and anomaly detection using mouse clickstream data analysis. *Appl. Sci.*, 11(13).
- Almalki, S., Chatterjee, P., and Roy, K. (2019). Continuous authentication using mouse clickstream data analysis. In *Proc. SpaCCS*.
- Antal, M. and Egyed-Zsigmond, E. (2019). Intrusion detection using mouse dynamics. *IET Biometrics*, 8(5).
- Antal, M. and Fejer, N. (2020). Mouse dynamics based user recognition using deep learning. *Acta Univ. Sapientiae, Informatica*, 12(1).
- Antal, M., Fejér, N., and Buza, K. (2021). Sapi-Mouse: Mouse dynamics-based user authentication using deep feature learning. In *Proc. SACI*.
- Brückner, L., Arapakis, I., and Leiva, L. A. (2020). Query abandonment prediction with recurrent neural models of mouse cursor movements. In *Proc. CIKM*.
- Cai, J., Luo, J., Wang, S., and Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, 300.
- Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259.
- Chong, P., Elovici, Y., and Binder, A. (2019). User authentication based on mouse dynamics using deep neural networks: A comprehensive study. *IEEE Trans. Inf. Forensics Secur.*, 15.
- Chong, P., Tan, Y. X. M., Guarnizo, J., Elovici, Y., and Binder, A. (2018). Mouse authentication without the temporal aspect – what does a 2D-CNN learn? In *Proc. SPW*.
- Dwarampudi, M. and Reddy, N. V. S. (2019). Effects of padding on LSTMs and CNNs. *CoRR*, abs/1903.07288.
- Eberz, S., Rasmussen, K. B., Lenders, V., and Martinovic, I. (2017). Evaluating behavioral biometrics for continuous authentication: Challenges and metrics. In *Proc. ASIACCS*.
- Fridman, L., Stolerman, A., Acharya, S., Brennan, P., Juola, P., Greenstadt, R., and Kamd, M. (2015). Multi-modal decision fusion for continuous authentication. *Comput. Electr. Eng.*, 41.
- Fu, S., Qin, D., Qiao, D., and Amariuca, G. T. (2020). RUMBA-mouse: Rapid user mouse-behavior authentication using a CNN-RNN approach. In *Proc. CNS*.
- Fülöp, A., Kovács, L., Kurics, T., and Windhager-Pokol, E. (2016). Balabit mouse dynamics challenge data set. <https://github.com/balabit/>.
- Handa, J., Singh, S., and Saraswat, S. (2019). A comparative study of mouse and keystroke based authentication. In *Proc. Confluence*.
- Hema, D. and Bhanumathi, S. (2016). Mouse behaviour based multi-factor authentication using neural networks. In *Proc. ICCPCT*.
- Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput.*, 9(8).
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA*, 79(8).
- Hu, T., Niu, W., Zhang, X., Liu, X., Lu, J., and Liu, Y. (2019). An insider threat detection approach based on mouse dynamics and deep learning. *Secur. Commun. Netw.*, 2019.

- Jorgensen, Z. and Yu, T. (2011). On mouse dynamics as a behavioral biometric for authentication. In *Proc. ASI-ACCS*.
- Lee, H., Lee, Y., Lee, K., and Yim, K. (2016). Security assessment on the mouse data using mouse loggers. In *Proc. BWCCA*.
- Lee, K., Esposito, C., and Lee, S.-Y. (2019). Vulnerability analysis challenges of the mouse data based on machine learning for image-based user authentication. *IEEE Access*, 7.
- Leiva, L. A., Arapakis, I., and Iordanou, C. (2021). My mouse, my rules: Privacy issues of behavioral user profiling via mouse tracking. In *Proc. CHIIR*.
- Leiva, L. A., Diaz, M., Ferrer, M. A., and Plamondon, R. (2020). Human or machine? it is not what you write, but how you write it. In *Proc. ICPR*.
- Levi, M. and Hazan, I. (2020). Deep learning based sequential mining for user authentication in web applications. In *Proc. ETAA*.
- Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., and Liu, H. (2017). Feature selection: A data perspective. *ACM Comput. Surv.*, 50(6).
- Lin, C.-C., Chang, C.-C., and Liang, D. (2012). A new non-intrusive authentication approach for data protection based on mouse dynamics. In *Proc. ISBAST*.
- Liu, Y., Jiang, Y., and Devenere, J. (2020). Using deep learning for fusion of eye and mouse movement based user authentication. In *Proc. IJCB*.
- Muda, R., Hamid, N. A., Satar, S. D. M., Mohamad, M., Mahadi, N. A., and Ghazali, F. (2017). Mouse movement behavioral biometric for static user authentication. *Advanced Science Letters*, 23(6).
- Oak, R. (2018). A literature survey on authentication using behavioural biometric techniques. In *Proc. ICICC*.
- Pramila, R. M., Misbahuddin, M., and Shukla, S. (2022). A survey on adaptive authentication using machine learning techniques. In *Proc. IDSCS*.
- Qin, D., Fu, S., Amariuca, G., Qiao, D., and Guan, Y. (2020). MAUSPAD: Mouse-based authentication using segmentation-based, progress-adjusted DTW. In *Proc. TrustCom*.
- Revett, K., Jahankhani, H., de Magalhães, S. T., and Santos, H. M. D. (2008). A survey of user authentication based on mouse dynamics. In *Proc. ICGeS*.
- Rose, J., Liu, Y., and Awad, A. (2017). Biometric authentication using mouse and eye movement data. In *Proc. SPW*.
- Roth, J., Liu, X., and Metaxas, D. (2014). On continuous user authentication via typing behavior. *IEEE Trans. Image Process.*, 23(10).
- Shen, C., Cai, Z., Guan, X., and Wang, J. (2012). On the effectiveness and applicability of mouse dynamics biometric for static authentication: A benchmark study. In *Proc. ICB*.
- Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404.
- Tan, Y. X. M., Binder, A., and Roy, A. (2017). Insights from curve fitting models in mouse dynamics authentication systems. In *Proc. AINS*.
- Tan, Y. X. M., Iacovazzi, A., Homoliak, I., Elovici, Y., and Binder, A. (2019). Adversarial attacks on remote user authentication using behavioural mouse dynamics. In *Proc. IJCNN*.
- Thomas, P. A. and Mathew, K. P. (2022). An efficient optimized mouse and keystroke dynamics framework for continuous non-intrusive user authentication. *Wirel. Pers. Commun.*, 124.
- Traore, I., Woungang, I., Obaidat, M. S., Nakkabi, Y., and Lai, I. (2012). Combining mouse and keystroke dynamics biometrics for risk-based authentication in web environments. In *Proc. ICDH*.
- Wei, A., Zhao, Y., and Cai, Z. (2019). A deep learning approach to web bot detection using mouse behavioral biometrics. In *Proc. CCB*.
- Yamauchi, T. and Bowman, C. (2014). Mining cursor motions to find the gender, experience, and feelings of computer users. In *Proc. ICDMW*.
- Yamauchi, T., Seo, J. H., Jett, N., Parks, G., and Bowman, C. (2015). Gender differences in mouse and cursor movements. *Int. J. Hum. Comput. Interact.*, 31(12).
- Yazji, S., Chen, X., Dick, R. P., and Scheuermann, P. (2009). Implicit user re-authentication for mobile devices. In *Proc. UIC*.
- Zheng, Y., Yu, L., Haque, S., Zhang, P., and Elmaghraby, A. S. (2022). Improving cybersecurity through deep learning on keystroke and mouse dynamics. In *Proc. SPIE Defense + Commercial Sensing*.