Acta Univ. Sapientiae, Informatica **12,** 1 (2020) 39–50

DOI: 10.2478/ausi-2020-0003

Mouse dynamics based user recognition using deep learning

Margit ANTAL
Sapientia Hungarian University of Transylvania
Department of Mathematics–Informatics
Tirgu Mures
email: manyi@ms.sapientia.ro

Norbert FEJÉR
Sapientia Hungarian University of Transylvania
Department of Electrical Engineering
Tirgu Mures
email:
fejer.norbert@student.ms.sapientia.ro

Abstract. Behavioural biometrics provides an extra layer of security for user authentication mechanisms. Among behavioural biometrics, mouse dynamics provides a non-intrusive layer of security. In this paper we propose a novel convolutional neural network for extracting the features from the time series of users' mouse movements. The effect of two preprocessing methods on the performance of the proposed architecture were evaluated. Different training types of the model, namely transfer learning and training from scratch, were investigated. Results for both authentication and identification systems are reported. The Balabit public data set was used for performance evaluation, however for transfer learning we used the DFL data set. Comprehensive experimental evaluations suggest that our model performed better than other deep learning models. In addition, transfer learning contributed to the better performance of both identification and authentication systems.

Computing Classification System 1998: I.2.1
Mathematics Subject Classification 2010: 68T10
Key words and phrases: behavioural biometrics, mouse dynamics, deep learning, convolutional neural networks

1 Introduction

Behavioural biometrics provide an invisible layer of security for applications, and continuously authenticates users by analyzing the user's unique interactions with their devices. Mouse dynamics is a kind of behavioural biometrics which analyzes the users' mouse movements and detects intruders.

Most of the previous studies in mouse dynamics used machine learning methods with handcrafted features. In this study we propose deep neural networks that use raw mouse data, thus avoiding the typical feature extraction process.

Mouse data sets usually contain the following data about the mouse pointer: time, (x, y) coordinates and other auxiliary information about the buttons and the type of mouse event. When using handcrafted features in the feature extraction process, one has to use the auxiliary information in order to segment the raw data into meaningful mouse actions such as mouse movements or drag and drop operations. In contrast, our proposed architecture uses the raw data segmented into fixed-size units. Then, we used convolutional filters for extracting relevant features from the raw data. Instead of using the raw coordinates, we used directional velocities $(dx/dt, dy/dt)$, which are not only translation invariant, but produce significantly improved results.

Our contribution can be summarized as follows: (i) We proposed a new one-dimensional convolutional network architecture. (ii) We evaluated the impact on performance of two preprocessing methods for handling short mouse movement sequences. (iii) We evaluated the impact of different model training types. We compared transfer learning to training from scratch. These were performed for biometric identification as well as for biometric authentication. In addition, our research is reproducible: the data sets are publicly available and the results can be replicated with the software available on GitHub[1].

Following this section the most important research results in the field of mouse dynamics biometric are summarized. The third section presents our methods: data preprocessing, the architecture of our convolutional neural network, and the ways in which transfer learning were applied in this study. This is followed by a new section presenting the data sets, performance metrics, measurement protocol, as well as the identification and authentication results. The last section concludes the paper.

[1] https://github.com/norbertFejer/AFE_Project

2 Related works

Several behavioural biometrics are already implemented in operational authentication systems. These methods are most often used to continuously verify the user's identity. On-line courses use keystroke dynamics to continuously verify the identity of the registered users. While keystroke data may contain sensitive personal information, such as names or passwords, mouse dynamics do not contain sensitive data at all. In contrast to physiological biometrics which require the usage of a special sensor by the user, usually behavioural biometric data can be collected without the consent of the user.

One of the first studies regarding the performance of mouse dynamics authentication was written by Gamboa and Fred [8]. They implemented a memory game as a web application and collected the mouse interactions of the game users. Mouse interactions were segmented into so called mouse strokes defined as mouse movements performed between successive clicks. A set of 63 handcrafted features were extracted from these strokes. The feature extraction phase was followed by the learning phase which consisted of the estimation of the probability density functions of each user interaction. The system performance based on a sequence of 10 strokes was 11.8% EER (Equal Error Rate). Unfortunately, this data set is not publicly available.

The first publicly available mouse data set was published in 2007 by Ahmed and Traore [1], although this data set does not include raw data, but segmented and processed data. The data set contains general computer usage mouse data of 22 users, that is, users performed their daily work on their computers. Raw mouse data was segmented into three types of action: PC - point and click: mouse movement ending in a mouse click; MM - general mouse movement; DD - drag and drop. Histogram-based features were extracted from sequences of consecutive mouse actions. They reported on their data set of 22 users 2.46% EER using 2000 mouse actions for user authentication. The authors extended their data set to 48 users and published a new study on continuous authentication based on this extended data set [2].

Shen et al. published three papers in the topic of user authentication based on mouse dynamics [10], [11], [12]. Two data sets were also collected, one for static (57 subjects) and one for continuous user authentication (28 subjects) through mouse dynamics. Several machine learning and anomaly detectors were tested. Authentication performance having low equal error rates (below 1% EER) were obtained by using a large amount of mouse movement data (e.g. 30 minutes).

Zheng et al. also investigated the user authentication problem in their studies [13], [14]. They proposed some novel features such as angle based metrics. They obtained 1.3% EER using a sequence of 20 mouse actions. Unfortunately, their data sets containing general mouse usage data are also private.

Another study was conducted by Feher et al. [6]. They also collected their own dataset containing data from 25 subjects. Their best performance was 8.53% EER using a sequence of 30 mouse actions. All these studies were based on classical machine learning algorithms using some handcrafted feature sets.

The first study to use deep neural networks for mouse dynamics was published by Chong et al. [5]. They investigated one and two-dimensional convolutional neural networks (CNN) for mouse dynamics. While 1D-CNN network was trained by using the mouse movement trajectory's time series, the 2D-CNN network was trained using images of mouse movement trajectories. Despite the loss of time information in the case of 2D-CNN, this model outperformed both 1D-CNN and SVM models using handcrafted features. They extended their study [4] by considering Long Short-Term Memory (LSTM) and hybrid CNN-LSTM networks as well. Among these models the 2D-CNN model performed best resulting in a 0.96 average AUC (Area Under the Curve) for the Balabit data set.

3 Methods

3.1 Data preprocessing

A mouse dynamics data set consists of several log files containing mouse events with the following information: the x and y coordinates, the timestamp and the type of event. Based on the type of event we distinguish mouse move, mouse click, drag and drop and scroll actions. Usually a sequence of mouse movement events is ended in a mouse click, but there are mouse movement sequences without the ending click. A drag and drop operation performed by a user results in a sequence of drag mouse events. All mouse events contain the x and y coordinates of the mouse pointer with the exception of the mouse scroll event. Therefore, scroll events were not considered.

Mouse events were segmented into sequences. A sequence was ended when the time difference between two consecutive mouse events exceeded a threshold. These sequences were segmented into fixed sized blocks. When the length of the sequence is not a multiple of the block size we end up in a few shorter sequences. These shorter sequences can be dropped or can be concatenated to obtain full length blocks. Both cases were evaluated in our measurements.

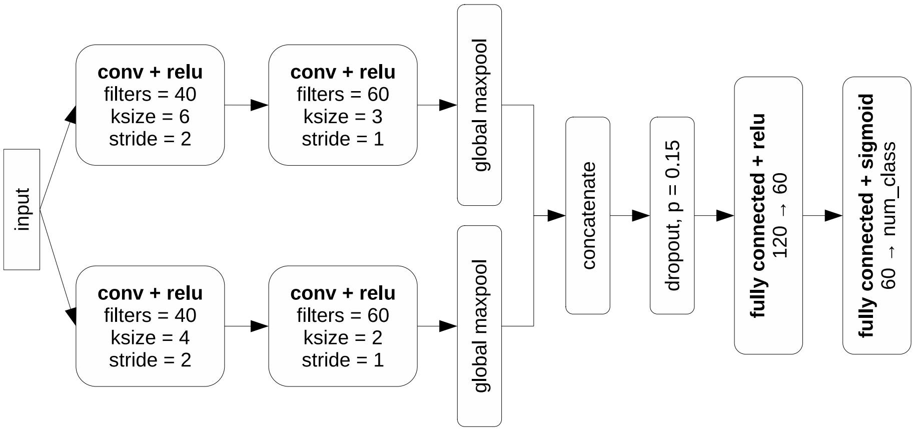

Figure 1: 1D-CNN architecture.

In order to obtain translation invariant mouse position sequences we decided to use speed values $(dx/dt, dy/dt)$ instead of absolute position coordinates (x, y).

3.2 1D-CNN

One dimensional convolutional neural networks (1D-CNNs) are used for time series modeling. As mouse movement sequences $x(t), y(t)$ are one dimensional time series, 1D-CNN models are well suited for modeling this type of signal. Our 1D-CNN architecture can be seen in Figure 1.

A tower model was used with different kernel sizes, which helped the network to learn input sequences on different time scales. We used the sigmoid activation function and a dropout layer with 0.15 probability to avoid overfitting. The network was trained in Keras [9] using the Adam optimizer (learning rate: 0.002, decay: 0.0001, loss function: binary cross-entropy). 16 epochs were used for training and a batch size of 32.

3.3 Transfer learning

Transfer learning is defined as reusing knowledge from previously learned tasks for the learning of a new task. This method is very popular in computer vision because it allows us to build accurate machine learning models faster. One may use a pre-trained model (a model trained on a large benchmark data set) instead of starting the learning process from scratch. In computer vision it is a common practice to use well-proven models from the published literature. This means that both the architecture and the parameters of the model are reused. In this study we used transfer learning in a slightly different way. As a first step we developed our own model architecture. Thereafter we trained our model on a large data set and saved the model. This pre-trained model was reused for all the measurements performed on another data set. In conclusion, we transferred only the representation learning that is the knowledge of extracting the features.

4 Experiments

4.1 Data sets

In this study we used two public data sets: the Balabit Mouse Challenge data set [7] and the DFL data set [3].

The Balabit Mouse Dynamics Challenge data set contains timing and positioning information of mouse pointers. As the authors of the data set state, it can be used for evaluating the performance of user authentication and identification systems based on mouse dynamics. The data set contains mouse dynamics data of 10 users, and is divided into training and test sessions where the training sessions are much longer than the test sessions.

The DFL data set contains mouse dynamics data of 21 users (15 male and 6 female). The raw data format is similar to the Balabit data set therefore it contains timing and positioning information of mouse pointers. A data collector application was installed on the users' computers which logged their mouse dynamics data, therefore the acquisition of the data was uncontrolled. The sessions of this data set are not divided into training and test sessions. The details of the data set are available at: https://ms.sapientia.ro/~manyi/DFL.html.

Table 1 shows the quantity of data available for training using the two types of settings presented in the 3.1 section. The second column of the table shows the number of blocks available for each user of the data set when we drop the short sequences, and the third column contains the number of blocks in the case of concatenating the shorter sequences into full-size blocks.

4.2 Performance metrics

Accuracy is defined as the proportion of correctly predicted labels among the total number of testing samples. Although this is the most intuitive metric

User	Drop	Concatenate
7	2457	3119
9	2408	3081
12	459	1800
15	385	1098
16	871	1716
20	1269	1928
21	449	894
23	345	889
29	324	933
35	217	695

Table 1: Number of blocks for each user of the Balabit data set. Each block contains 128 mouse events.

when measuring the performance of a classifier, it is not always the best choice, e.g. when the data set is highly imbalanced. A commonly used metric when measuring the performance of biometric systems is the Receiver Operating Characteristics (ROC) curve. This curve plots the true positive ratio (TPR) against the false positive ratio (FPR), and the area under the curve (ROC AUC) is often used to compare the performances of different biometric systems.

4.3 Measurement protocol

As the acquisition of the DFL data set was uncontrolled, we decided to use this data set only for representation learning, which means that this data set was used to initialize the weights of our models (e.g. convolution kernels).

We evaluated both identification and authentication biometric systems. While the identification is a multi-class classification problem, authentication is a binary classification problem.

As described in section 3.1 mouse dynamics data was segmented into fixed sized blocks. There are big differences between users in terms of data volume. The user having the most data has ten times as much data as the user with the least data. Based on the amount of data used for the measurement, two types of measurements were made: (i) measurement using 300 blocks from each user – 300; (ii) measurement using all blocks of data form each user – ALL. While the first type is a class-balanced measurement, the second is a class-imbalanced measurement.

From the point of view of training the models, we distinguish three cases: (i) models trained from scratch using the training data from the Balabit data set – PLAIN models; (ii) models using the transfer learning - the models were pre-trained on the DFL data set – $\mathsf{TRANSFER1}$ models; (iii) models initialised with transfer learning, then updating the weights using the training data from the Balabit data set –$\mathsf{TRANSFER2}$. This case is similar to the PLAIN one. While in the first case we start with random weights, here we adjust the weights obtained from the $\mathsf{TRANSFER1}$ model.

In the case of the identification measurements, we trained a single classifier using the training data (balanced - using the same number of blocks from each user or imbalanced using all the available data from each user), then we used the same number of test data from each user for computing the evaluation metrics.

In the case of the authentication measurements, we trained a separate model to each user using the same number of positive and negative data. In the first case (300), we took 300 positive blocks of data from a given user, then the same number of negative data was selected from the remaining users. The only user not having 300 blocks of data is user35 (see Table 1). In order to increase the number of training examples we used data augmentation. We added a random noise drawn from a uniform distribution in the range $[-\epsilon, \epsilon]$ to each signal (we used $\epsilon = 0.2$). Data augmentation was performed independently on $x(t)$ and $y(t)$ signals. In the second case (ALL), we considered all the positive data available from a given user, then the same number of negative data was selected from the remaining users.

Regardless of the measurement type we always separated 70 blocks of data from each user for evaluating the model. Therefore, all types of training were evaluated using the same amount of test data.

We used a single pre-trained model for transfer learning. This model was trained on the DFL data set. Therefore, we transferred the learned data representation from one data set to another.

All the evaluations were performed in Python 3.6.8 (Anaconda distribution) using Keras [9].

4.4 Results

4.4.1 Biometric identification

The effect of using a class-balanced subset (300 blocks/class) for evaluation compared to using all the available data is shown in Table 2. We evaluated three types of models: PLAIN, TRANSFER1 and TRANSFER2. First of all, it can be seen that using all data resulted in lower performances than using a class-balanced subset of the available data. Secondly, we see that using transfer learning with frozen weights (TRANSFER1 - data representation was learned using another data set), resulted in much poorer identification rate than training the model from scratch. Thirdly, as we expected, the pre-trained model with updated weights (TRANSFER2) resulted in the best identification accuracies.

Number of blocks	PLAIN	TRANSFER1	TRANSFER2
300	0.63	0.50	0.66
ALL	0.55	0.34	0.62

Table 2: Identification results in terms of accuracy. Class-balanced subset vs. all data.

The results shown in the Table 2 were obtained using full sized mouse events blocks by dropping the shorter mouse event sequences (see subsection 3.1). The measurements were repeated for the other case where the training data included concatenations of shorter series. Table 3 shows the comparative results for the two cases.

Preprocessing	PLAIN	TRANSFER1	TRANSFER2
Drop	0.55	0.34	0.62
Concatenate	0.57	0.37	0.61

Table 3: Identification results in terms of accuracy using all data. Preprocessing type: concatenate vs. drop.

4.4.2 Biometric authentication

Tables 4 and 5 show the results of different authentication measurements in terms of accuracy and AUC respectively. Each performance is reported using the average performance value and in parenthesis the standard deviation. We can observe that there is no significant difference between PLAIN and TRANSFER2 results. This suggests that transfer learning does not significantly improve system performance. We can also notice that using a pre-trained model without updating the weights for the new data set (TRANSFER1) results in lower performance than training the model from scratch (PLAIN).

We should also notice that using all the available positive data for training (ALL) the models resulted in better performances for all types of training (see Figure 2). Not only are the average AUC values higher but the standard deviations are much more lower. This means that there are negligible differences in performance between users.

Number of blocks	PLAIN	TRANSFER1	TRANSFER2
300	0.86 (0.10)	0.80 (0.11)	0.87 (0.10)
ALL	0.93 (0.04)	0.79 (0.10)	0.93 (0.04)

Table 4: Authentication results in terms of accuracy. 300 vs. all data.

Number of blocks	PLAIN	TRANSFER1	TRANSFER2
300	0.92 (0.09)	0.86 (0.10)	0.93 (0.09)
ALL	0.98 (0.02)	0.87 (0.11)	0.98 (0.01)

Table 5: Authentication results in terms of AUC. 300 vs. all data.

We compared our best results with other results obtained on the Balabit data set using approximately the same size of mouse sequences for predicting the authenticity of the users. The comparison is shown in Table 6. It can be seen that our model has brought a significant improvement compared to Chong et al.'s [4] 1D-CNN model, moreover it is better than their optimized 2D-CNN model performance.

Paper	Model type	Average AUC
Chong, 2018 [5]	SVM	0.87
Chong, 2019 [4]	1D-CNN	0.90
Chong, 2019 [4]	2D-CNN	0.96
This	1D-CNN	0.98

Table 6: Comparison of authentication systems' performances on the Balabit data set.

5 Conclusions

In this study we proposed a novel 1D-CNN model for user authentication based on mouse dynamics. The advantage of our model over the classical machine learning model is that there is no longer need for ad-hoc features; the model is

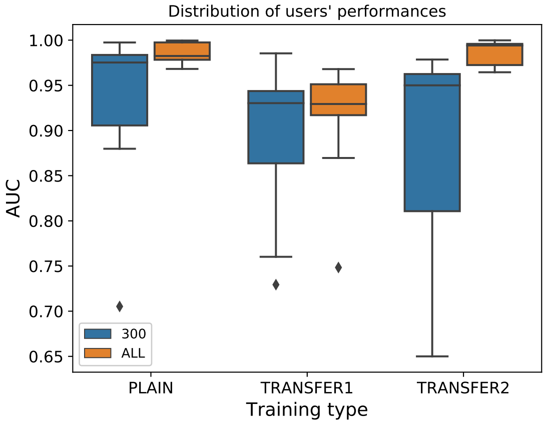

Figure 2: Authentication results for the Balabit dataset. Training data: 300 vs. all. Training methods: PLAIN, TRANSFER1, TRANSFER2. Each box shows the distribution of users's performances (AUC) using the given training data and method.

able to learn the features from raw data. However, we also demonstrated that transfer learning or learning the data representation on an independent large data set could improve the performance of the authentication system. The results show that our 1D-CNN model performs better than the other CNN models proposed for the same task.

Acknowledgements

The work of Norbert Fejér was supported by Accenture Industrial Solutions.

References

[1] A. A. E. Ahmed, I. Traore, A new biometric technology based on mouse dynamics, *IEEE Transactions on Dependable and Secure Computing* **4,** 3 (2007) 165–179. ⇒41

[2] A. A. E. Ahmed, I. Traore, Dynamic sample size detection in continuous authentication using sequential sampling, In *Proceedings of the 27th Annual Computer Security Applications Conference* ACSAC '11, pp. 169–176, New York, NY, USA, 2011. ACM. ⇒41

[3] M. Antal, L. Dénes-Fazakas, User verification based on mouse dynamics: a comparison of public data sets, In *2019 23th International Symposium on Applied Computational Intelligence and Informatics*, pp. 143–147, May 2019. ⇒44

[4] P. Chong, Y. Elovici, A. Binder, User authentication based on mouse dynamics using deep neural networks: A comprehensive study, *IEEE Transactions on Information Forensics and Security*, **15** (2020) 1086–1101. ⇒42, 48

[5] P. Chong, Y. X. M. Tan, J. Guarnizo, Y. Elovici, A. Binder, Mouse authentication without the temporal aspect – what does a 2d-cnn learn? In *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 15–21, May 2018. ⇒42, 48

[6] C. Feher, Y. Elovici, R. Moskovitch, L. Rokach, A. Schclar. User identity verification via mouse dynamics. *Inf. Sci.* **201** (2012) 19–362. ⇒42

[7] Á. Fülöp, L. Kovács, T. Kurics, E. Windhager-Pokol, Balabit mouse dynamics challenge data set, 2016. ⇒44

[8] H. Gamboa, A. Fred. A behavioral biometric system based on human-computer interaction. In *Proc. SPIE 5404, Biometric Technology for Human Identification, (25 August 2004)*, **5404**, pp. 381–392, 2004. ⇒41

[9] KERAS. Keras, 2016. ⇒43, 46

[10] C. Shen, Z. Cai, X. Guan, Continuous authentication for mouse dynamics: A pattern-growth approach, In *Proceedings of the 2012 42Nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, DSN '12, pp. 1–12, Washington, DC, USA, 2012. IEEE Computer Society. ⇒41

[11] C. Shen, Z. Cai, X. Guan, Y. Du, R. A. Maxion, User authentication through mouse dynamics, *IEEE Transactions on Information Forensics and Security*, **8,** 1 (2013) 16–30. ⇒41

[12] C. Shen, Z. Cai, X. Guan, R. A. Maxion, Performance evaluation of anomaly-detection algorithms for mouse dynamics, *Computers & Security* **45** (2014) 156–171. ⇒41

[13] N. Zheng, A. Paloski, H. Wang, An efficient user verification system via mouse movements, In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS '11, pp. 139–150, New York, NY, USA, 2011. ACM. ⇒42

[14] N. Zheng, A. Paloski, H. Wang, An efficient user verification system using angle-based mouse movement biometrics, *ACM Trans. Inf. Syst. Secur.*, **18,** 3 (2016) 11:1–11:27. ⇒42

Received: January 25, 2020 • Revised: February 16, 2020