The Warping Window Size Effects the Accuracy of Person Identification based on Keystroke Dynamics

Zakarya Farou and Krisztian Buza

Telekom Innovation Labortories, Department of Data Science and Engineering
Faculty of Informatics, ELTE - Eötvös Loránd University in Budapest
{fzfxih,buza}@inf.elte.hu
http://t-labs.elte.hu/

Abstract: 1-nearest neighbor (1NN) with Dynamic Time Warping (DTW) distance is a popular time series classification technique. In the last decades, research on DTW aimed to improve its classification accuracy, memory usage, and efficiency. According to a recent study, the appropriate selection of the Warping Window Size (WWS) is crucial for the accuracy of 1NN-DTW. In this work, we consider person identification based on keystroke dynamics as a time series classification task, and we examine whether WWS is crucial in this case. We performed experiments on a real-world dataset containing more than 400 typing sessions obtained from 12 users. In the case of this dataset, each user typed the same fixed text. We found that the aforementioned hypothesis was correct, i.e., the classification accuracy indeed depends on the WWS. Furthermore, according to our observations, in our case, the optimal WWS is less than 3% which is substantially different from 10% that has been used as a default value in various works.

Keywords: keystroke dynamics, person identification, dynamic time warping, DTW, warping window size, nearest neighbor.

1 Introduction

Biometric authentication systems are based on user traits and characteristics. There are two major forms of biometrics: those based on physiological attributes and those based on behavioral characteristics. The physiological type includes biometrics based on stable body traits, such as the face, iris, fingerprint, and the hand, and are considered to be more robust and secure. However, they are also considered to be more vulnerable to intrusions, expensive and require special equipment [17]. On the other hand, behavioral biometrics include learned movements such as handwritten signatures, keyboard dynamics (typing pattern), mouse movements, gait, and speech. Collecting these biometrics is less obtrusive and they do not require extra hardware [20].

Recently, keystroke dynamics has gained popularity [2, 5]. Keystroke dynamics may be used for continuous user authentication or to enhance existing authentication methods, such as password-based authentication or the credit card number-based access to payment services. Keystroke dynamics is appealing for many reasons: it is non-invasive (users type on the computer keyboard anyway), it does not require extra hardware and is available after the authentication step at the start of the computer session [11].

Various methods have been used for user authentication based on keystroke dynamics, such as Bayesian classifiers, neural networks, SVM, Markov chains and dimensionality reduction [13, 19], see also [4] for a recent review.

2 Background

Measuring the similarity between a pair of time series is an important task which is usually applied in many data mining applications such as anomaly detection [7], clustering [1, 9], and classification tasks [10, 12]. In a naive solution, the Euclidean distance could be used to calculate the similarity between time series, however, the Dynamic Time Warping (DTW) distance allows for shifts and elongations, therefore it is more appropriate in various applications [1, 3, 6]. We will first begin with a review of DTW and its recent extensions.

2.1 Measuring The Similarity of Time Series

Given two time series T_1 of length n, and T_2 of length m

$$T_1 = (\ T_1[1]\ ,\ T_1[2],\ \dots\ ,\ T_1[i]\ ,\ \dots\ ,\ T_1[n]\ )$$

$$T_2 = (\ T_2[1]\ ,\ T_2[2],\ \dots\ ,\ T_2[j]\ ,\ \dots\ ,\ T_2[m]\ ),$$

their similarity can be computed in various ways.

The simplest way to define the similarity S is to treat these sequences as vectors and to compute the Euclidean distance between them directly. However,

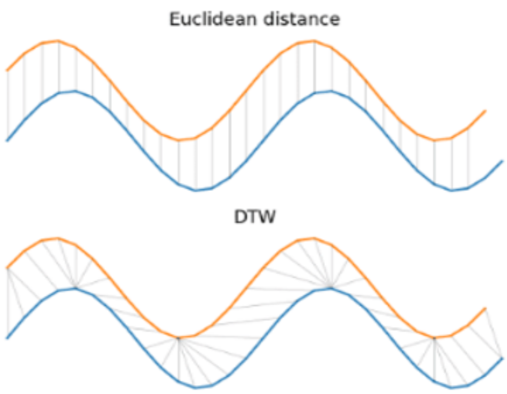


Figure 1: Comparison of two time series: Euclidean distance (top) compares the values at the same positions, whereas DTW (bottom) allows for shifts and elongations.

this requires both time series to be of the same length, i.e., $m = n$, but this is not always the case in real-world applications [15]. That's why the similarity should be computed between sequences of various lengths ($m \neq n$). To this end, DTW is an appropriate choice that allows for non-linear matching of two time series with different lengths [15], see Fig. 1.

2.2 Review of DTW

DTW [18] is an edit distance. Given various edit steps (elongation, replacement of value) and their costs, the DTW distance of two time series is the minimal cost of transforming one of the time series into the other one. Technically, DTW distance is calculated by filling an n by m matrix, called DTW cost matrix (denoted as σ), where n and m refer to the length of time series T_1 and T_2 respectively, see Alg. 1 and Fig. 2.

In order to understand DTW in more detail, we define an n by m matrix D, in which each entry $d_{i,j}$ corresponds to the squared distance between two values of the time series: $d_{i,j} = (T_1[i] - T_2[j])^2$. To find the best match between the two time series, we retrieve a path through the matrix, called *warping path*, that minimizes the total cumulative distance. The warping path is a sequence of neighboring entries[1] of D so that it begins in the top left (first row, first column) corner of D, and it ends in the bottom right corner (last row, last column). In particular, the optimal warping path W^* is the path that minimizes the cost. The DTW distance is

[1]With neighbors of $d_{i,j}$ we mean: $d_{i-1,j-1}$, $d_{i-1,j}$, $d_{i-1,j+1}$, $d_{i,j-1}$, $d_{i,j+1}$, $d_{i+1,j-1}$, $d_{i+1,j}$, $d_{i+1,j+1}$.

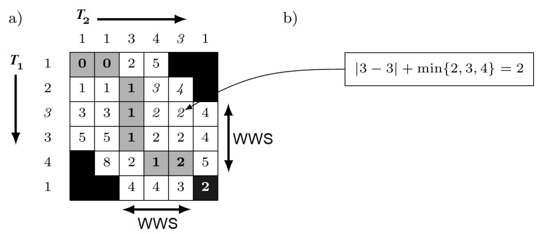


Figure 2: Example of the calculation of the DTW-matrix. a) The DTW-matrix. The time series T_1 and T_2 are shown o the left and top of the matrix. The marked entries of the matrix correspond to the mapping between both time series. b) The calculation of the value of a cell.

the sum of entries of D along that path:

$$DTW(T_1, T_2) = \min_{W \in \mathbb{P}} \sum_{w_k \in W} D_{w_k}, \tag{1}$$

where $\mathbb{P}$ is the set of all warping paths.

This warping path can be found using dynamic programming to evaluate the following recurrence:

$$\sigma[i,j] = d_{i,j} + min \begin{cases} \sigma[i-1, j-1] \\ \sigma[i-1, j] \\ \sigma[i, j-1] \end{cases} \tag{2}$$

Note that $\sigma(i,j)$ is the cumulative distance along the optimal warping path.

In most applications, it may not be reasonable to allow for arbitrarily large shifts and elongations, therefore, the calculations are usually restricted to the entries around the diagonal of the matrix, i.e.,

$$\sigma_{i,j} \text{ is calculated } \Leftrightarrow |i-j| \leq w,$$

Algorithm 1: DTW Distance

Input: T_1: array [1..n], T_2: array [1..m]
Output: distance
σ = array[0..n][0..m]
for i = *1, ..., n* **do**
  for j = *1, ..., m* **do**
    σ[i,j] = ∞
σ[0,0] = 0
for i = *1, ..., n* **do**
  for j = *1, ..., m* **do**
    cost = $(T_1[i] - T_2[j])^2$
    σ[i,j] = cost +
      min(σ[i-1, j], σ[i , j-1], σ[i-1, j-1])
return σ[n,m]

where w is the *Warping Window Size* (WWS) which controls the maximum amount of warping. In the cases, where some of the terms $\sigma_{i,j-1}$, $\sigma_{i-1,j}$, $\sigma_{i-1,j-1}$ are undefined (i.e., they have not been calculated), they are excluded from Eq. (2). WWS is often set to 10% of the length of the time series.

The effect of WWS on the classification performance has been studied, for example, in [14] and [8]. Dau et al. [8] claimed that obtaining the best performance from DTW requires setting WWS carefully, i.e., using the appropriate WWS can produce significant improvements in classification accuracy. In the following section, we perform an experiment to examine whether this hypothesis is true or not in case of typing dynamics data.

3 Experiments

The goal of our experiments is to examine the effect of warping window size on the accuracy of person identification.

3.1 Person Authentication Dataset

We used the dataset associated with Task 1 of the Person Identification Challenge. Typing dynamics of 12 users were recorded in 458 sessions with a JavaScript (JS) application. In each typing session, the users were asked to type some sentences and the keyboard events keyup, keydown and keypress were captured by the JS application. For each of the aforementioned events, the time (in milliseconds) was also recorded, see [5] and the webpage of the challenge[2] for details.

We are given the true identity of the users (coded by integer numbers from 1 to 12) for 5 typing patterns per user. We use this data as training data.

Additionally, hypothetical user identities for the rest of the typing patterns are given. This data is used as test data.

Our task is to decide if the hypothetical identities match the true identities of the users who typed those patterns.

The submission system at the webpage of the challenge is used to evaluate our predictions for the test data.

3.2 Preprocessing and Feature Extraction

In order to experimentally examine the role of the warping window size, we use 1-nearest neighbor with DTW with various warping window sizes $w \in \{0, 1, 2 \ldots, l_{max}\}$ where l_{max} is the length of the longest typing pattern sequence. Before running the DTW distance between each pair of keystroke time series, the data must be pre-proceeded, then we classify users' test data.

[2] `biointelligence.hu/typing-challenge/`

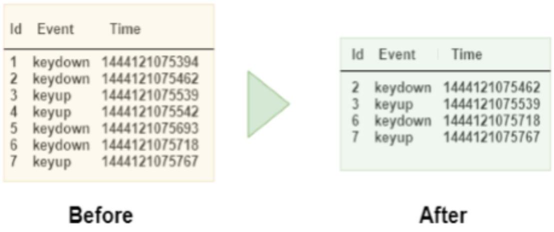
```
TYPING PATTERN 1
keyup 9 9 0 false 1444121074805
keydown 16 16 0 true 1444121075394
keydown 84 84 0 true 1444121075462
keypress 0 84 84 true 1444121075462
keyup 16 16 0 false 1444121075539
keyup 84 84 0 false 1444121075542
keydown 72 72 0 false 1444121075693
keypress 0 104 104 false 1444121075693
keydown 65 65 0 false 1444121075718
keypress 0 97 97 false 1444121075719
keyup 72 72 0 false 1444121075767
keyup 65 65 0 false 1444121075809
keydown 84 84 0 false 1444121075873
keypress 0 116 116 false 1444121075874
keyup 84 84 0 false 1444121075938
keydown 16 16 0 true 1444121076082
keydown 49 49 0 true 1444121076607
keypress 0 39 39 true 1444121076607
keyup 49 49 0 true 1444121076670
keyup 16 16 0 false 1444121076700
```

Figure 3: Example of user typing pattern raw data, see `http://biointelligence.hu/typing-challenge/task1/index.php` for the description of the raw data.

Pre-processing of Raw Typing Patterns In order to extract down-down and down-up durations, the raw data must be preprocessed. In particular, we performed the following preprocessing steps:

- select time and relevant event types, i.e., keydown and keyup.
- Sometimes the first event of the typing pattern is a keyup event (see Fig. 3) which is caused by clicking the shift key to change the cursor between registration form fields, so we removed it.
- For simplicity, we removed sequential successive keydown or successive keyup events, and we kept the latest keydown and the first keyup (see Fig. 4).

Extraction of Features Alg. 2 and Alg. 3 show the approach for extracting down-down (between keystroke) and down-up (keystroke duration) metrics.



Before

Id	Event	Time
1	keydown	1444121075394
2	keydown	1444121075462
3	keyup	1444121075539
4	keyup	1444121075542
5	keydown	1444121075693
6	keydown	1444121075718
7	keyup	1444121075767

After

Id	Event	Time
2	keydown	1444121075462
3	keyup	1444121075539
6	keydown	1444121075718
7	keyup	1444121075767

Figure 4: Elimination of successive keydown and keyup events

Algorithm 2: Extraction of the times between keystrokes

Input: typingP : List []
Output: ksDuration: List []
counter = 0
typing = []
betweenKS = []
while *not end of list* **do**
  if *typingP[counter]== 'keydown'* **then**
    typing.append(typingP[counter])
  counter = counter + 1
counter = 0
while *not end of list* **do**
  betweenKS[counter] =
    typing[counter + 1] - typing[counter]
  counter = counter + 1
return betweenKS

3.3 Experimental Results

To test the effect of the warping window size on the classification accuracy, we performed empirical experiments on the datasets containing (i) keystroke durations and (ii) between-keystroke-times. Additionally, we experimented with a combination of the two aforementioned options.

We vary the warping window size from 0 (Euclidean) to l_{max} (no constraint/full calculation) and record the accuracy for each warping window size. The results are shown in Fig. 5, Fig. 6 and Fig. 7.

3.4 Discussion

A study made by Ratanamahatana et al. [16] where a similar experiment has been performed with 6 datasets retrieved from "UEA & UCR Time Series Classification Repository" ended by the following conclusions: "Wider warping constraints do not always improve the accuracy, as commonly believed [21]." More often, the accuracy peaks very early at much smaller window size, as shown in Table 1.

Algorithm 3: Extraction of the duration of keystrokes

Input: typingP : List []
Output: ksDuration: List []
counter = 0
ksDuration = []
while *not end of list* **do**
  ksDuration[counter] =
    typingP[counter + 1] - typingP[counter]
  counter = counter + 2
return ksDuration

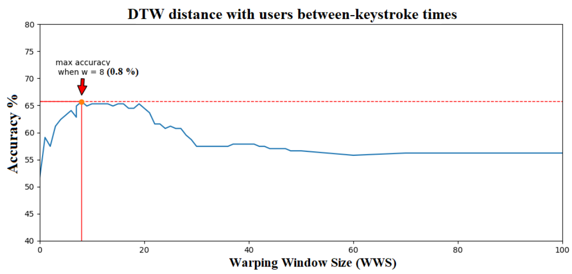


Figure 5: The classification accuracy as function of warping window size using between-keystroke-times

4 Conclusion

Based on the result of Ratanamahatana et al.[16] and our own experiments on the data associated with the typing pattern-based person identification challenge, we can conclude that working with a small warping window (approximately 3% in our case) could be beneficial not only to reduce the execution time and memory usage but also to increase the accuracy of the classification. We also noticed that working with a lager warping window size could reduce the accuracy of the classification in some cases. That's why it is better to use a low warping window size.

Our results confirm the hypothesis of Dau et al. [8]

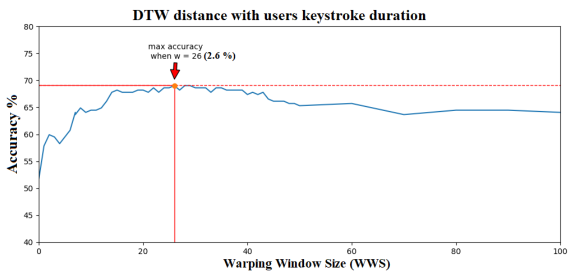


Figure 6: The classification accuracy as function of the warping window size using keystroke duration

Table 1: The warping window size that yields maximum classification accuracy for each dataset.

Dataset	Accuracy (%)	WWS (%)
Face	96.43	3
Gun	99.00	3
Trace	100.00	1
Leaf	96.38	10
Control chart	99.67	8
Two-pattern	100	3
Typing Pattern – keystroke duration	69.10	2.6
Typing Pattern – between-keystroke-times	65.70	0.8
Typing Pattern – combined	69.67	2.1

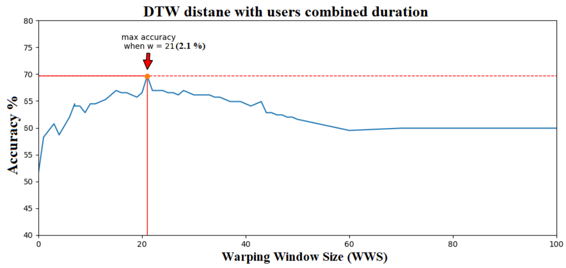


Figure 7: The classification accuracy as function of the warping window size using the minimum of the two DTW-distances (DTW-distance based on between-keystroke-times and DTW-distance based on keystroke duration)

for the case of person identification based on typing patterns. According to the aforementioned hypothesis, ”obtaining the best performance from DTW requires setting its only parameter, the maximum amount of warping (w) and thus can produce significant improvements in classification accuracy”.

Acknowledgments

This work was supported by the project no. 20460-3/2018/ FEKUTSTRAT within the Institutional Excellence Program in Higher Education of the Hungarian Ministry of Human Capacities.

References

[1] Aach, J., Church, G. M.: Aligning gene expression time series with time warping algorithms. Bioinformatics **17(6)** (2001) 495–508

[2] Antal, M., Szabó, L. Zs., László, I.: Keystroke dynamics on android platform. Procedia Technology **19** (2015) 820–826

[3] Bar-Joseph, Z., Gerber, G., Gifford, D., Jaakkola, T. , Simon, I.: A new approach to analyzing gene expression time series data. In Proceedings of the 6th Annual International Conference on Research in Computational Molecular Biology (2002) 39–48

[4] Bhatt, S., Santhanam, T.: Keystroke dynamics for biometric authentication–A survey. In Proceedings of the International Conference on Pattern Recognition, Informatics and Mobile Engineering. IEEE (2013) 17–23

[5] Buza, K.: Person Identification Based on Keystroke Dynamics: Demo and Open Challenge. In CAiSE Forum (2016) 161–168

[6] Caiani, E. G., Porta, A., Baselli, G., Turiel, M., Muzzupappa, S., Pieruzzi, F., Cerutti, S.: Warped-average template technique to track on a cycle-by-cycle basis the cardiac filling phases on left ventricular volume. In Computers in Cardiology 1998. IEEE **25** (Cat. No. 98CH36292) (1998) 73–76

[7] Dasgupta, D., Forrest, S.: Novelty detection in time series data using ideas from immunology. In Proceedings of the international conference on intelligent systems (1996) 82–87

[8] Dau, H. A., Silva, D. F., Petitjean, F., Forestier, G., Bagnall, A., Mueen, A., Keogh, E.: Optimizing dynamic time warping's window width for time series data mining applications. Data Mining and Knowledge Discovery **32(4)** (2018) 1074–1120

[9] Debrégeas, A., Hébrail, G.: Interactive Interpretation of Kohonen Maps Applied to Curves. In KDD (1998) 179–183

[10] Diez, J. J. R., González, C. J. A.: Applying boosting to similarity literals for time series classification. In International workshop on multiple classifier systems. Springer, Berlin, Heidelberg (2000) 210–219

[11] Gunetti, D., Picardi, C.: Keystroke analysis of free text. ACM Transactions on Information and System Security (TISSEC) **8(3)** (2005) 312–347

[12] Kadous, M. W.: Learning Comprehensible Descriptions of Multivariate Time Series. In ICML **454** (1999) 463

[13] Kalina, J., Schlenker, A.: Dimensionality reduction

methods for biomedical data. Lékař a technika – Clinician and Technology, **48(1)** (2018) 29-35

[14] Kurbalija, V., Radovanović, M., Geler, Z., Ivanović, M.: The influence of global constraints on similarity measures for time-series databases. Knowledge-Based Systems **56** (2014) 49–67

[15] Rabiner, L. R., Juang, B. H., Rutledge, J. C.: Fundamentals of speech recognition. Englewood Cliffs: PTR Prentice Hall **14** (1993)

[16] Ratanamahatana, A., Keogh, E.: Everything you know about dynamic time warping is wrong. In 3rd Workshop on Mining Temporal and Sequential Data, in conjunction with 10th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD-2004), Seattle, WA **1** (2004) 1–11

[17] Revett, K., Gorunescu, F., Gorunescu, M., Ene, M., Magalhães, P. S. T., Santos, H. D. D.: A machine learning approach to keystroke dynamics based user authentication. International Journal of Electronic Security and Digital Forensics**1(1)** 55–70

[18] Sakoe, H., Chiba, S., Waibel, A., Lee, K. F.: Dynamic programming algorithm optimization for spoken word recognition. Readings in speech recognition **159** (1990) 224

[19] Stefan, D., Shu, X., Yao, D.D.: Robustness of keystroke-dynamics based biometrics against synthetic forgeries. Computers & security **31(1)** (2012) 109–121

[20] Sultana, M., Paul, P. P., Gavrilova, M.: A concept of social behavioral biometrics: motivation, current developments, and future trends. In 2014 International Conference on Cyberworlds. IEEE (2014) 271–278

[21] Zhu, Y., Shasha, D.: Warping indexes with envelope transforms for query by humming. In Proceedings of the 2003 ACM SIGMOD international conference on Management of data. ACM (2003) 181–192