Innovations in Systems and Software Engineering
https://doi.org/10.1007/s11334-022-00464-4

S.I. : COUPLING DATA AND SOFTWARE ENGINEERING TOWARDS SMART SYSTEMS

Impact of augmentation methods in online signature verification

Dawid Najda[1] · Khalid Saeed[1,2]

Received: 19 June 2022 / Accepted: 3 July 2022

Abstract

The aim of this paper is to investigate the impact of selected data augmentation techniques on the learning performance of neural networks for dynamic signature verification. The paper investigates selected data augmentation techniques in deep learning for verification purpose of dynamic signature. Two neural networks were used as classifiers: MLP and LSTM-FCN. Investigation of five selected augmentation methods and experiments were performed on the open source signature database SVC2004. The authors tested both classifiers without augmentation and then with data augmentation for three extensions of the learning set and three sizes of the user database. They presented the results of the experiments in tabular form for each augmentation method. The results were compared with the existing dynamic signature verification methods and given in the paper.

Keywords Signature · Online signature · Biometrics · Verification · Augmentation

1 Introduction

Signature is a commonly used behavioral biometric feature to verify a person's identity. Depending on the signature type, the data can be represented as a time series or an image. A distinction is made between a handwritten signature and a machine written signature. Handwriting system verification can be categorized in two different types: online and offline. Offline signature is represented by digitalized images mostly taken from a document where the signature is present and processed by the system. To obtain the online signature data system must use special hardware for example digitalized tablet or pen [1].

An advantage of the signature verification over other verification systems based on biometric traits is that the signature data can be enrolment when the user is conscious and desires to write, in the other hand systems based on face, for example, can be enrolment without human awareness [2].

Manual verification of identity in the case of dynamic and handwritten signatures is very difficult due to the ease of forging the original signature and the requirement of expertise. In order to facilitate the signature verification process, automatic signature verification approaches were proposed. These systems are mainly focused on biometric solutions and artificial intelligence. Neural networks can be used for signature identification or verification purposes.

In this article, the authors present a methodology for signature verification. The authors develop an automated signature verification application using previously created modules to load data from the database, preprocess the data, extract the signature characteristics, divide the input data into learning, validation and test sets, and select the appropriate classifier and estimation of the results. The neural network learning process needs a lot of data. The authors decided to use augmentation methods based on [3–6] for online signature data and then compare results with the other online signature verification systems. The authors create new augmentation methods modified from the existing ones. The authors modify noise addition and interpolation methods.

The paper contains a few sections: in the next chapter the authors describe other approaches and algorithms for online signature verification. Second section shows the neu-

✉ Dawid Najda
d.najda@pb.edu.pl

Khalid Saeed
k.saeed@pb.edu.pl

1 Faculty of Computer Science, Białystok University of Technology, Białystok, Poland

2 Department of Computer Science and Electronics, Universidad de La Costa - CUC, Barranquilla, Colombia

ral networks and augmentation methods are presented. In this one, the authors presented the architecture of chosen neural networks and five selected augmentation methods. The penultimate chapter provides information on the experiments performed, in particular the different augmentation methods, performance results and the comparison with other different approaches. Last chapter contains the conclusions, information about used hardware in the learning process and future work.

2 State of the art

The most popular algorithms for signature verification systems are hidden Markov models [7], dynamic time warping [8] or neural networks [9]. DTW method gave the best results in determining whether a signature is genuine or forged. DTW approach is the top method which is using in any competition for signature verification [8, 10, 11].

The main problem with signature verification is associated with the intra-class variability of the signature. The signature enrolment relies on practiced and repetitive motion, which causes short-term signature as input. The signature trait can evolve so signature data can lose important properties for verification purposes [12].

Christian Gruber, Sebastian Krinninger, and Thiemo Gruber created a new method for online signature verification based on SVM using LCSS kernel [13]. Using the LCSS kernel function their system determines the resemblances within two time series. The results are even better than results in systems based on DTW [14].

Dynamic time warping is more effective when resolving problems with a small amount of data. HMM model and its derivation with Gaussian model (GMM) can be considered as a soft variant of DTW. In some cases when enough signature data is available it can outperform the DTW approach [15, 16].

Suresh Sundaram and Abhishek Sharma created a new model approach using DTW and GMM [17]. First of all, the authors extract statistical properties for a given signature. Then, the extracted data is warped and analyzed. Finally, the author's fusion DTW score and warped data for better verification results [18, 19].

Lianwen Jin, Weixin Yang and Songxuan Lai proposed to create a recurrent neural network in sequential modeling. RNN system improved the performance of dynamic signature verification. The authors proposed a novel descriptor LNPS (length-normalized path signature) and use it due signature verification problem [20].

Zapata Gabriel posed the problem of small databases within the signature verification systems. The author also states for signatures per user limitation. Gabriel does nine classification methods based on GMM and evaluates them.

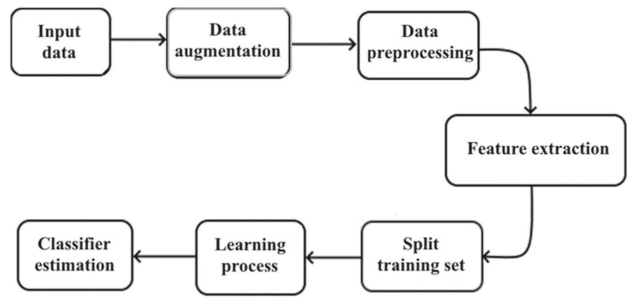

Fig. 1 Extended a block diagram of proposed system

The author tests it using three experiments and a small database. In conclusion of performed experiments, the author says that the method's performance degraded faster when training sets include less than half of the samples [21].

3 Proposed methodology

The authors decide to present used database, neural networks architecture and used augmentation algorithms. Proposed approach is presented as block diagram in Fig. 1. The highlighted one is the block with data augmentation step.

3.1 Database

The authors used the SVC 2004 dynamic signature database [22]. Each genuine or forged signature is stored in a text file. The filename has the following format "UXSY.txt", where (1) stands for the signing user and (2) stands for one signature of user X.

$$X \in \{1, 2, \ldots, 40\} \tag{1}$$

$$Y \in \{1, 2, \ldots, 40\} \tag{2}$$

The first twenty signatures are genuine, while the next twenty signatures are signatures identified as qualified forgeries provided by other users. The SVC 2004 database contains 40 users with 40 signatures each. In summary, the entire SVC 2004 database contains 1600 signatures.

Every single file contains presented properties:

- X coordinate—position along the X-axis
- Y coordinate—position along the Y-axis
- Pressure—corrected pressure condition
- Interval—sample measurement
- Pen state—state when pen is pressed to table
- Azimuth angle—horizontal angle
- Elevation angle—vertical angle

3.2 Data augmentation

Before data preprocessing the authors system extend input data with augmentation methods. The authors choose five augmentation methods based on the state of the art and each method is invoked with $\times$ 0, $\times$ 10, $\times$ 20, $\times$ 40 times for each signature:

1. Interpolation [23] with the authors modifications
2. Noise addition [24] to time series with the authors modifications
3. Signal scaling [3]
4. Signal rotation [3]
5. Warping time series [3, 25]

The authors describe the modified augmentation methods due to limited pages in the paper.

For the interpolation method the authors use sinc interpolation. The sinc interpolation method is computationally complex due to the large number of calculations, as a separate sinc function must be considered for each sample in signal (3).

The authors system takes a vector of interpolation points and duplicates it by rows as many times as there are samples in the array. Next, it takes a column vector of sample indices in the array and duplicates it by columns as many times as there are interpolated points. The two matrices are subtracted from each other, which corresponds to shifting the sinc function. Next, it performs a matrix multiplication of the vector of sample values in the array by the sinc values for the previously computed matrix, this operation corresponds to scaling the sinc function by the sample values and the sum of all sinc functions (3).

$$\sin c(x) = \frac{\sin(\pi x)}{\pi x} \tag{3}$$

For the noise addition augmentation method, the authors use Gaussian noise (4) and the SNR relation.

$$F_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{4}$$

Extending the data just by adding noise to the signal can distort the features, so the authors combine it with other algorithms. In this method, a low-pass filter was applied before adding noise to the dynamic signature data, and after adding noise combined with the filter, the data was averaged using the Locally Weighted Scatterplot Smoothing (LOWESS) algorithm [25].

The remaining algorithms have no changes at all, so the details can be seen in [3, 25].

3.3 Data preprocessing

Data processing layer consists of normalization step and sends the information to the feature extraction layer. For the normalization purpose the authors used given formula (5).

$$x_{\text{norm}} = \frac{x - x_{\min}}{X_{\text{len}}} \tag{5}$$

where x_{norm}—normalized sample, x—input sample, $x_{\min}$—minimum value in the signature signal, X_{len}—length of the signature signal.

3.4 Feature extraction

Given signatures from the database enables the creation of new feature metrics such as signature duration, pen lead velocity and acceleration, coordinates of discrete points drawn from the signature line, or means and standard deviations of individual signal components. All used features in this paper are presented below.

- Coordinate x (from db)
- Coordinate y (from db)
- Pressure pr (from db)
- Velocity $\text{vel} = \sqrt{v_x^2(t) + v_y^2(t)}$
- Azimuth angle $\gamma = \arctan\left(\frac{v_{yi}}{v_{xi}}\right)$
- Speed magnitude $\text{tam} = \sqrt{v_{\text{vel}}^2 + \left(\text{vel}^2 * v_\alpha^2\right)}$
- Velocity changing $\log cr = \log \frac{|\text{vel}|}{|v_\alpha^2|}$

The authors remove some features given from SVC 2004 database and recalculate the data and extract new characteristics. After feature extraction the next layer is the splitting layer.

3.5 Split training set

In the first step in this layer the authors create two folders with sets of training and test data signatures. Created three datasets each with five, ten, and thirty-five users, respectively. Each user in the training dataset had fifteen true signatures and fifteen advanced forgeries, while the test dataset had ten signatures per user with half of them being genuine, also added the signatures of other users not in the learning dataset.

The authors conducted experiments on three sets of input data; the names adopted, respectively, for each set are small (5 users), medium (10 users) and large (35 users). The comparison of the sets is shown in Table 1, where (TR) stands for training data and (TEST) stands for test data. The number of signatures of other users indicates simple forgery, in this case, it does not distinguish whether the signature is genuine

Table 1 Training set splitting

	Small		Medium		Large	
Users	5		10		35	
	(TR)	(TEST)	(TR)	(TEST)	(TR)	(TEST)
Genuine signatures	15	5	15	5	15	5
Forged signatures	15	5	15	5	15	5
Other users signatures	0	5	0	10	0	15
All signatures	150	55	300	110	1050	365

or forged because it will be verified as a forged signature in the system.

The sum of all signatures in the training set was calculated by formula (6), while the sum of all signatures in the test set is defined by formula (7).

$$\mathrm{SUMA}_{\mathrm{TR}} = X * \left(P_{\mathrm{TR}f} + P_{\mathrm{TR}t}\right) \tag{6}$$

$$\mathrm{SUMA}_{\mathrm{TEST}} = X * \left(P_{\mathrm{TEST}f} + P_{\mathrm{TEST}t}\right) + Y \tag{7}$$

where $P_{\mathrm{TR}f}$ stands for the number of forged signatures for training set or test set, respectively, in case of $P_{\mathrm{TEST}f}$ while $P_{\mathrm{TEST}t}$ stands for the number of genuine signatures for test set, the same for the training set is expressed by $P_{\mathrm{TR}t}$, where X is the total amount of users in the database and Y is the amount of signatures of the other users. After preparing the training test the system launched a learning process.

3.6 Neural network architecture

The authors created two neural networks based on [26] presented in Figs. 2 and 3. Selected LSTM-FCN architecture [26] consists of two networks FCN (Fully Convolutional Networks) and LSTM (Long Short-Term Memory). LSTM is a variation of the RNN. In the proposed model, FCN is augmented with an LSTM block and then a dropout layer, as shown in Fig. 2. FCNs are neural networks which contains only convolutional layers and in addition the batch normalization, dropout, or max-pooling layers.

The authors created a multilayer perceptron neural network based on [26] and implemented it in the system consisting of Dense and Dropout layers, where the output is the Dense layer with the same amount of neurons as all the classes.

The output layer is activated by the softmax function, while the other layers are activated by the ReLU function.

In the neural networks shown above the input layer is a tensor where LEN is the number of signatures, TS is the number of samples taken over time for one signature, and CH is the number of features taken for a given sample. The output layer consists of $N + 1$ neurons, where N stands for the number of classes (users), while $N + 1$ stands for the number of classes (users) + forgery class.

4 Experiments

The authors have done around one thousand experiments combining all augmentation methods. They decide to estimate the results with AER metric (8) consisting of FRR (9) and FAR (10) metrics.

$$\mathrm{AER} = \frac{\mathrm{FAR} + \mathrm{FRR}}{2} \tag{8}$$

$$\mathrm{FRR} = \frac{\mathrm{FR}}{\mathrm{FR} + \mathrm{TA}} \tag{9}$$

$$\mathrm{FAR} = \frac{\mathrm{FA}}{\mathrm{FA} + \mathrm{TR}} \tag{10}$$

FAR and FRR metrics include:

- FA (Falsely Accepted)—the number of forged examples accepted as genuine
- TR (Truly Rejected)—the number of forged examples rejected as false
- FR (Falsely Rejected)—the number of genuine examples accepted as forged
- TA (Truly Accepted)—the number of genuine examples accepted as genuine

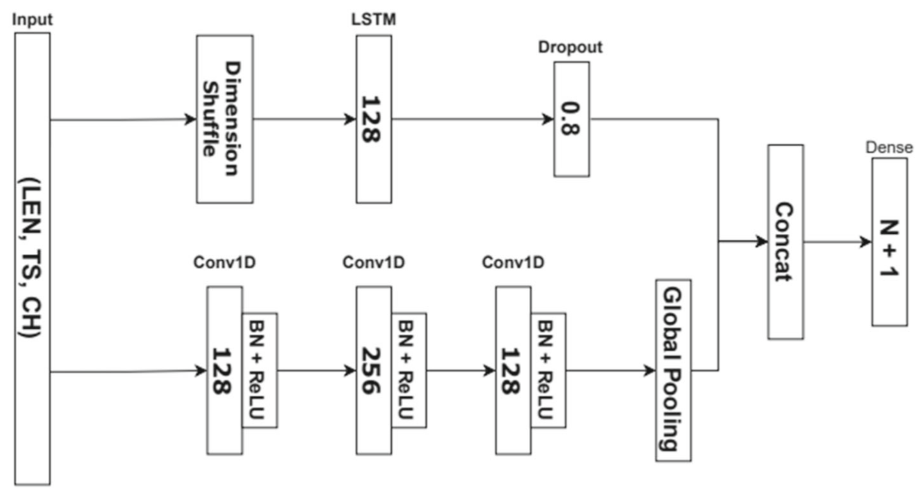

Fig. 2 LSTM–FCN architecture [26]

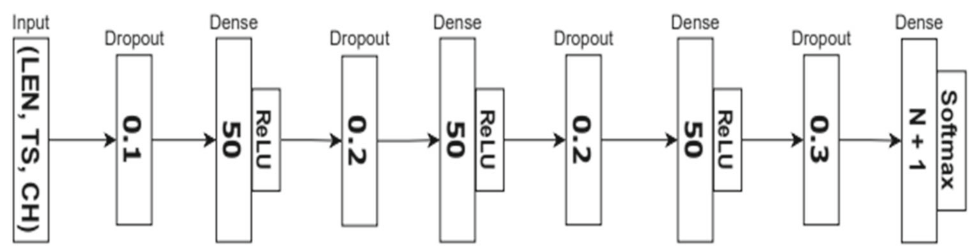

Fig. 3 MLP architecture [26]

Table 2 AER for non-augmented input data

	MLP	LSTM-FCN
Small	20.1	23.2
Medium	18.3	18.2
Large	14.1	**12.3**

Bold value indicates the best result

Table 3 The best results for interpolation augmentation

	MLP	LSTM-FCN
Small	15.3 (× 40)	16.8 (× 40)
Medium	13.5 (× 40)	14.3 (× 40)
Large	10.1 (× 20)	**7.2 (× 40)**

Bold value indicates the best result

The authors pick the best results for each data augmentation method for all created sets (small, medium, large). The signature augmentation was performed additionally for 0 ×, 10 ×, 20 ×, 40 × for each signature.

For the best visibility the authors decide to not show all results. All brackets in cells stand for a count of augmentation each signature for the best results.

All results are shown in Tables 2, 3, 4, 5, 6 and 7 for each standalone augmentation. Table 8 shows overview for the best results for each augmentation method. The best result for data augmentation in this case is reached by noise addition method it is equal AER = 6.2.

Table 4 The best results for noise addition augmentation

	MLP	LSTM-FCN
Small	16.0 (× 40)	13.1 (× 40)
Medium	13.4 (× 40)	8.3 (× 40)
Large	12.3 (× 40)	**6.2 (× 40)**

Bold value indicates the best result

Table 5 The best results for signal scaling augmentation

	MLP	LSTM-FCN
Small	20.3 (× 40)	19.8 (× 40)
Medium	17.5 (× 40)	15.4 (× 40)
Large	13.1 (× 20)	**7.4 (× 20)**

Bold value indicates the best result

Table 6 The best results for signal rotating augmentation

	MLP	LSTM-FCN
Small	19.3 (× 40)	19.1 (× 40)
Medium	17.0 (× 20)	17.3 (× 20)
Large	11.6 (× 10)	**11.2 (× 10)**

Bold value indicates the best result

In the last experiment the authors combine the best augmentation methods. The results are shown in Table 9. The references for used methods:

Table 7 The best results for signal time warping augmentation

	MLP	LSTM-FCN
Small	15.3 (× 40)	19.3 (× 40)
Medium	13.5 (× 40)	16.7 (× 40)
Large	**9.2** (× **40**)	9.8 (× 40)

Bold value indicates the best result

Table 8 Overview of the augmentation methods results

	MLP	LSTM-FCN
Interpolation (1)	10.1 (× 20)	7.2 (× 40)
Noise addition (2)	12.3 (× 40)	**6.2** (× **40**)
Scaling (3)	13.1 (× 20)	7.4 (× 20)
Rotation (4)	11.6 (× 10)	11.2 (× 10)
Time warping (5)	9.2 (× 40)	9.8 (× 40)

Bold value indicates the best result

Table 9 The best results for combined augmentation methods with large set

	MLP	LSTM-FCN
(1) + (2) + (3)	7.6 (× 40)	**2.90** (× **40**)
(1) + (4) + (5)	12.3 (× 40)	7.0 (× 40)
(1) + (2)	13.1 (× 40)	5.9 (× 40)

Bold value indicates the best result

- Interpolation (1)
- Noise addition (2)
- Scaling (3)
- Rotation (4)
- Time warping (5)

The best results for signature verification used augmentation methods obtained with LSTM-FCN neural network with combined interpolation, noise addition and signal scaling methods it is equal to AER = 2.90. The authors mention that for × 40 multiplication for each signature (1) + (2) + (3) sum of all signatures it is equal to 127 050 from 1050 for large training set.

The result of proposed methodology in comparison with other methods for Task 2 SVC2004 is shown in Table 10.

5 Conclusions

The methodology presented in this paper was implemented with Python 3.8. The experiments were done on Intel Core i9-7960X, GeForce 3080 SUPER and 64 GB DDR4 RAM. They were tested more than 900 times. In this paper the experiment is based on database from "The First International Signature Verification Competition SVC 2004" [22], in the future the authors planned to do the next experiment with database from the newest competition for signature verification is called "SVC-onGoing: Signature verification competition" [35].

Table 10 Methods of online signature verification for SVC 2004

Method	AER
Gruber et al. [27]	6.84
Barkoula et al. [28]	5.33
Yahyatabar et al. [29]	4.58
Liu et al. [30]	2.98
Proposed method	**2.90**
Song et al. [31]	2.89
Sharma et al. [32]	2.53
Jia et al. [33]	2.39

Bold value indicates the best result

Proposed methodology and the authors' experiments proved that right chosen augmentation techniques can increase the accuracy of signature verification systems. The differences between the best result with and without augmentation are ~ 9.4 and hence, using the augmentation methods we can decrease the errors of the signature verification systems by four times. Two of the five augmentation methods modified by authors have the best results compared to other augmentation methods used in this paper. Noise addition with modification have AER = 6.2 and the Interpolation method have AER = 7.2 [34, 35].

Acknowledgements This work was supported by grant WZ/WI-IIT/4/2020 from Bialystok University of Technology and funded with resources for research by the Ministry of Science and Higher Education in Poland.

Data availability The dataset analyzed during the current study are available from the following public domain resource: https://cse.hkust.edu.hk/svc2004/Task2.zip and belongs to SVC 2004 [22]. Data is available to the research community.

Declarations

Conflict of interest All authors declare that they have no conflicts of interest.

References

1. Zareen FJ, Jabin S (2013) A comparative study of the recent trends in biometric signature verification. In 2013 sixth international conference on contemporary computing (IC3), August 2013, pp 354–358
2. Lei H, Govindaraju V (2005) A comparative study on the consistency of features in on-line signature verification. Pattern Recogn Lett 26(15):2483–2489
3. Iwana BK, Uchida S (2020) An empirical survey of data augmentation for time series classification with neural networks. arXiv, 2020
4. Um TT, Pfister FMJ, Pichler D, Endo S, Lang M, Hirche S (2017) Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks. In: ACM ICMI, 2017, pp 216–220
5. Le Guennec A, Malinowski S, Tavenard R (201) Data augmentation for time series classification using convolutional neural networks. In: IWAATD, 2016
6. Sawicki A, Zieliński SK (2020) Augmentation of segmented motion capture data for improving generalization of deep neural networks. In: CISIM. Springer, pp 278–290
7. , Jahan MV, Farimani SA (2018) An hmm for online signature verification based on velocity and hand movement directions. In: 2018 6th Iranian joint congress on fuzzy and intelligent systems (CFIS). IEEE, pp 205–209
8. Malik MI, Ahmed S, Marcelli A, Pal U, Blumenstein M, Alewijns L, Liwicki M (2015) Icdar2015 competition on signature verification and writer identification for on-and off-line neural computing and applications skilled forgeries. In: 2015 13th international conference on document analysis and recognition (ICDAR). IEEE, pp 1186–1190
9. Lai S, Jin L, Yang W (2017) Online signature verification using recurrent neural network and length-normalized path signature descriptor. In: 2017 14th IAPR international conference on document analysis and recognition (ICDAR) , vol 1. IEEE, pp 400–405
10. Bibi K, Naz S, Rehman A (2020) Biometric signature authentication using machine learning techniques: current trends, challenges and opportunities. Multimed Tools Appl 79(1):289–340
11. Liwicki M, Malik MI, Van Den Heuvel CE, Chen X, Berger C, Stoel R, Blumenstein M, Found B (2011) Signature verification competition for online and offline skilled forgeries. In: 2011 international conference on document analysis and recognition. IEEE, pp 1480–1484
12. Ureche O, Plamondon R (2000) Digital payment systems for internet commerce: the state of 1201" the art. World Wide Web 3(1):1–11
13. Gruber C, Gruber T, Krinninger S, Sick B (2010) Online signature verification with support vector machines based on LCSS kernel functions. IEEE Trans Syst Man Cybern Part B (Cybern) 40(4):1088–1100
14. Gruber C, Gruber T, Sick B (2005) Online signature verification with new time series kernels for support vector machines. Springer, Berlin, pp 500–508
15. Van BL, Garcia-Salicetti S, Dorizzi B (2007) On using the viterbi path along with HMM likelihood information for online signature verification. IEEE Trans Syst Man Cybern Part B 37(5):1237–1247
16. Fierrez J, Ortega-Garcia J, Ramos D, Gonzalez-Rodriguez J (2007) Hmm-based on-line signature verification: feature extraction and signature modeling. Pattern Recogn Lett 28(16):2325–2334
17. Sundaram S, Sharma A (2017) A novel online signature verification system based on gmm features in a dtw framework. IEEE Trans Inf Forensics Secur 12(3):705–718
18. Faundez-Zanuy M (2007) On-line signature recognition based on vq-dtw. Pattern Recogn 40(3):981–992
19. Kar B, Dutta PK, Basu TK, VielHauer C, Dittmann J (2006) DTW based verification scheme of biometric signatures. In 2006 IEEE international conference on industrial technology, December 2006, pp 381–386
20. Songxuan L, Lianwen J, Weixin Y (2017) Online signature verification using recurrent neural network and length-normalized path signature descriptor. In: 2017 14th IAPR international conference on document analysis and recognition (ICDAR)
21. Zapata G, Arias-Londoño JD, Vargas-Bonilla J, Orozco JR (2016) Online signature verification using gaussian mixture models and small-sample learning strategies. Rev Fac Ing 2016:06
22. Yeung D-Y, Chang H, Xiong Y, George S, Kashi R, Matsumoto T, Rigoll G (2004) SVC2004: first international signature verification competition, 2004
23. Lin A, Wang L (2007) Style-preserving english handwriting synthesis. Pattern Recogn 40(7):2097–2109
24. Galbally J, Fierrez J, Martinez-Diaz M, Ortega-Garcia J (2009) Improving the enrollment in dynamic signature verification with synthetic samples. In: 10th international conference on document analysis and recognition, 2009
25. Dahea W, Fadewar HS (2018) Multimodal biometric system: a review. Int J Eng Technol 4:25–31
26. Wang Z, Yan W, Oates T (2017) Time series classification from scratch with deep neural networks: a strong baseline. In: International joint conference on neural networks, pp 1578–1585
27. Gruber C, Gruber T, Krinninger S, Sick B (2010) Online signature verification with support vector machines based on LCSS kernel functions. IEEE Trans Syst Man Cybern Part B Cybern 40:1088–1100
28. Barkoula K, Economou G, Fotopoulos S (2013) Online signature verification based on signatures turning angle representation using longest common subsequence matching. Int J Doc Anal Recogn (IJDAR) 16:261–272
29. Yahyatabar ME, Baleghi Y, Karami MR (2013) Online signature verification: a Persian-language specific approach. In: 21st Iranian Conference on electrical engineering (ICEE), 2013, pp 1–6
30. Liu Y, Yang Z, Yang L (2017) Online signature verification based on DCT and sparse representation. IEEE Trans Cybern 45:2498–2511
31. Song X, Xia X, Luan F (2017) Online signature verification based on stable features extracted dynamically. IEEE Trans Syst Man Cybern Syst 47:2663–2676
32. Sharma A, Sundaram S (2018) On the exploration of information from the DTW cost matrix for online signature verification. IEEE Trans Cybern 48:611–624
33. Jia Y, Huang L, Chen H (2018) A two-stage method for online signature verification using shape contexts and function features. In: 2019, extended version of paper published in PRCV 2018: Chinese conference on pattern recognition and computer vision, Guangzhou, China, 23–26 November 2018
34. Karim F, Majumdar S, Darabi H, Chen S (2018) LSTM fully convolutional networks for time series classification. IEEE Access 1662–1669
35. Tolosana R, Vera-Rodriguez R, González C, Fierrez J, Morales A, Ortega-Garcia J, Ruiz-Garcia J, Romero-Tapiador S, Rengifo S, Caruana M, Jiang J, Lai S, Jin L, Zhu Y, Galbally J, Diaz M, Ferrer M, Gomez-Barrero M, Hodashinsky I, Jabin S (2022) SVC-onGoing: signature verification competition. Pattern Recogn 127:108609