Mohammad Saleem
Bence Kovari

OPTIMIZED jk-NEAREST NEIGHBOR BASED ONLINE SIGNATURE VERIFICATION AND EVALUATION OF MAIN PARAMETERS

Abstract *In this paper, we propose an enhanced jk-nearest neighbor (jk-NN) algorithm for online signature verification. The effect of its main parameters is evaluated and used to build an optimized system. The results show that the jk-NN classifier improves the verification accuracy by 0.73–10% as compared to a traditional one-class k-NN classifier. The algorithm achieved reasonable accuracy for different databases: a 3.93% average error rate when using the SVC2004, 2.6% for the MCYT-100, 1.75% for the SigComp'11, and 6% for the SigComp'15 databases. These results followed a state-of-the-art accuracy evaluation where both forged and genuine signatures were used in the training phase. Another scenario is also presented in this paper by using an optimized jk-NN algorithm that uses specifically chosen parameters and a procedure to pick the optimal value for k using only the signer's reference signatures to build a practical verification system for real-life scenarios where only these signatures are available. By applying the proposed algorithm, the average error rates that were achieved were 8% for SVC2004, 3.26% for MCYT-100, 13% for SigComp'15, and 2.22% for SigComp'11.*

Keywords k-nearest neighbor, online signature verification, classification

Citation Computer Science 22(4) 2021: 539–551

1. Introduction

Biometrics are measurements of the body that are connected to human appearance or behavior that are commonly used in identification and access control methods. The signature is a behavioral biometric that represents a person's specific physical activity; it is one of the most frequently used biometrics in our lives. The two ways in which signatures get captured for automatic processing are known as offline and online approaches. In an offline system, the signer uses a standard paper and pen to write his signature, which is then scanned and processed as an image file. In online systems, special devices such as tablets, digital pens, and PDAs [7] obtain real-time signatures. When used for online signature acquisition, these devices can capture several real-time features that cannot be captured using the static approach; this makes it harder to forge online signatures than offline ones. These features can be categorized as functions and parameters. The function feature is a function of time features, while the parameter feature is a vector of elements. Some features can be extracted directly from the signature data (such as position and pressure), while some features can be derived from other features [13].

There are several public databases that are available, including SVC2004 [32], MCYT-100 [21], SigCom'11 [15], and SUSIG [12]. These databases differ in their numbers of signees, and they also vary in the number of original and forged signatures for each signee. Thus, the usie of different signature databases is necessary for evaluating the performance of online signature-verification systems. Although signatures from the same signee may look very similar, these signatures may have internal variations that are caused by various factors (such as mood, age, or others). Such internal distinctions often make it difficult to discern between original and forged signatures; therefore, discrepancies require careful consideration when developing a signature--verification system (SVS)–minimizing the internal differences through preprocessing methods before extracting the features from a signature. Several preprocessing algorithms can improve the similarity measurements between signatures and reduce the SVS's error. Normalization methods include position normalization (translation), size normalization (scaling), angle normalization, and other preprocessing methods that are described in the literature (such as zero pressure removal, resampling, or filtering) [14, 16, 26].

In addition to data collection, preprocessing, and feature extraction, the verification phase defines the critical steps of online signature-verification systems. There are many methods that are used for similarity measurement and classification, including dynamic time warping (DTW) [20], neural networks [10], support-vector machines (SVM) [9], k-nearest neighbor (k-NN), and many more. During this phase, signatures are divided into training and testing sets; the chosen algorithm uses the training set to train the model. By measuring the error, the SVS can then be evaluated using the test set. In a real-world situation, only the reference signatures of a signee are available; therefore, we want to build an SVS that relies only on using the references (thus, the data of the genuine class must define the boundary between the two categories). A one-class classification algorithm is k-nearest neighbor [4], where the k-nearest neighbor objects identify whether the examined object belongs to the training process's original class. We presented a new optimized formula of the k-NN algorithm [11], which was used to implement a signature-verification system that represents real-life situations and outcomes. The work methodology, the SVS main steps, and the experimental results are investigated extensively in the following sections.

2. Related works

Online signature verification promises comparable verification accuracies to those of human experts. A crucial step in the verification process is the actual evaluation and classification of results; for this, we apply several classification approaches. One-class classification aims to determine the class for a set of objects by using only that data that is available for one class; objects that do not belong to this class are treated as outliers. The one-class methodology was discussed and used in several fields [5, 11, 23]. A study of online signature-verification systems that are based on one-class classifiers is presented in [19].

Besides being used for multi-class classification, the nearest neighbor is one of the available methods that are used to solve the one-class classification problem [3, 18]. Here, signatures are labeled according to the class of their nearest neighbors. It is also useful to consider more than one neighbor. Hence, the approach is generally referred to as k-nearest neighbor (k-NN) classification, where the k-nearest neighbors are used to determine the questioned sample class. Khan et al. [11] presented a theoretical analysis of k-NN one-class classification algorithm variants; their results showed that, by optimizing the parameters, jk-nearest neighbor could achieve good results (especially with low values of j and high values of k). Their evaluation tested several datasets but not signature datasets. Pippin [22] suggested a technique for the online authentication of signatures; it extracts global signature features and compares these features to stored signature models by using the k-NN classification. Yang et al. [31] suggested a new writer-dependent online-verification technique for signature verification based on relief and using the k-nearest neighbor for classification. They used the SVC2004 database in their work and achieved an average error rate of 5.312%. Nanni [19] used k-NN for a one-class online signature-verification system and achieved 12.2% and 6.3% error rates when using 5 and 20 skilled signatures, respectively, for the training set on the MCYT-100 database. The k-NN has also been used for offline signature-verification systems [8,27,28]. Azmi et al. [2] used k-NN and Freeman chain code (FCC) in their work and achieved a 9.85% AER on the MCYT-100 database. Abdelrahaman et al. [1] also achieved 80% accuracy by using k-NN for their offline SVS. The k-NN has also been applied in related fields such as text recognition [29], iris recognition [30], and emotion detection [25]. This work is an extension of the performance evaluation of the k-NN [24] work and presents a novel optimized k-NN online signature-verification system.

3. Contribution

This paper proposes and evaluates an online signature-verification approach using a generalized k-NN algorithm (the jk-NN). k-nearest neighbors (k-NN) is a non-parametric classification approach that was proposed by Thomas Cover [4]. In the one-class k-NN algorithm, the k-nearest neighbors of the first-nearest neighbor classify the tested object. Each distance d between the tested signature (S)-nearest neighbor (S_{nn}) and its (S_{knn})-nearest neighbor is measured, and the average of the latter (D_{avg}) is calculated. If the distance between the tested signature and its nearest neighbor is less than or equal to the threshold θ D_{avg}, then S is classified as genuine; otherwise, it is considered to be forged:

$$d\,(S, S_{nn}) < \theta \frac{1}{K} \sum_{k=1}^{K} d(S_{nn}, S_{knn}) \tag{1}$$

In the one-class k-NN approach, only the distance between the tested signature and its first-nearest neighbor verifies the signature to be genuine or forged by comparing the average distances between the first-nearest neighbor and its k-nearest neighbors. While in the jk-NN method, the j-nearest neighbors are used for the classification by comparing each Jth neighbor with its k-nearest neighbors (see Figure 1). We can say that the k-NN and jk-NN are the same if $J = 1$. Each neighbor $j \in J$ is tested using the k-NN algorithm. If the majority of the J-nearest neighbors of S is accepted, then the S is classified as genuine; otherwise, it is considered to be forged.

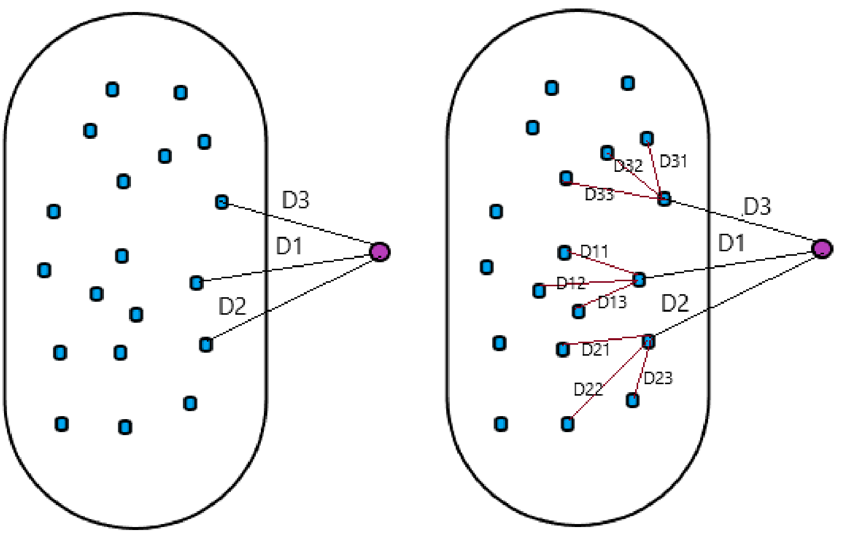


Figure 1. Algorithms k-NN (left) and jk-NN (right)

The jk-NN algorithm can be formalized as the following:

$$\frac{\sum_{j=1}^{J}\left[d\left(S_j, S_{jnn}\right) < \theta\frac{1}{K}\sum_{k=1}^{K} d(S_{jnn}, S_{jknn})\right]}{J} > 0.5 \tag{2}$$

In the following subsections, we start by discussing the main steps of the verification system, followed by the proposed algorithm and the automatic calibration of the main parameters of the algorithm. The contribution and the methodology of the proposed work and its experimental results are explained in the further sections.

3.1. Data acquisition

In this work, we used four different databases to evaluate the accuracy of the proposed algorithms. The Signature Verification Competition 2004 database (SVC2004) [32], the Spanish Ministry of Science and Technology database (MCYT-100) [21], the Dutch subsets of the Signature Verification Competition 2011 database (SigComp'11) [15], and the German database of the Signature Verification Competition 2015 (SigComp'15) [17]. Table 1 shows a comparison of these databases.

Table 1
Databases utilized

Database	Language	Signers/Signatures	Sampling frequency
SVC2004	English and Chinese	40/1600	100 Hz
MCYT-100	Spanish	100/5000	100 Hz
SigComp'11	Dutch	64/1905	200 Hz
SigComp'15	German	30/750	75 Hz

3.2. Preprocessing

The Z-normalization algorithm [6] was applied to enhance the accuracy of the similarity measurements. The z-normalization method ensures that all input vector elements are translated into an output vector with a mean of approximately 0, while the standard deviation is close to 1. The formula behind the transformation described as follows:

$$\hat{x}(i) = \frac{x(i) - \mu}{\sigma},\ where\ i \in N \tag{3}$$

3.3. Feature selection

Several features are from online signatures. This work features a combination of three components: horizontal position (X), vertical position (Y), and pressure (P). This XYP combination calculates the similarity between each pair of signatures.

3.4. Verification and evaluation

The jk-NN classifier that was mentioned at the beginning of this section is implemented and used for the classification after applying the preprocessing algorithm to the acquired signatures and extracting the required features. Each signature is compared to its j-nearest neighbors and their k-nearest neighbors and then classified as genuine or forged based on the computed thresholds. The test set evaluates the verifier's output. There are two forms of errors observed; a false acceptance implies that a forged signature is classified as genuine. Thus, in a false rejection where an authentic signature is identified as forged, the reverse occurs. The evaluation considers both the false acceptance rate (FAR – Type-II error rate) and the false rejection rate (FRR – Type-I error rate). The average error rate (AER) evaluates the method for both of these errors.

4. Experimental results

The experiment starts by using various reference numbers and thresholds to evaluate and analyze the verification accuracy's performance. After that, we examine several j and k values and the effect of adjusting the number of nearest neighbors upon the results, and we compare the k-NN and jk-NN algorithms. We evaluate the proposed practical online signature verification using jk-NN and its main parameters.

4.1. Evaluation of parameters

For the jk-NN online signature verifier, the j values, k values, numbers of references used, and threshold selection (value of θ) are the significant factors that determine the performance. In the following subsections, each of these factors is discussed and evaluated.

4.1.1. Number of references and threshold scale

Our experiments show greater accuracy when using more than ten original signatures as references—particularly between 13 and 15 (see Table 2). A greater number of reference signatures can provide a better representation of the intra-class variations.

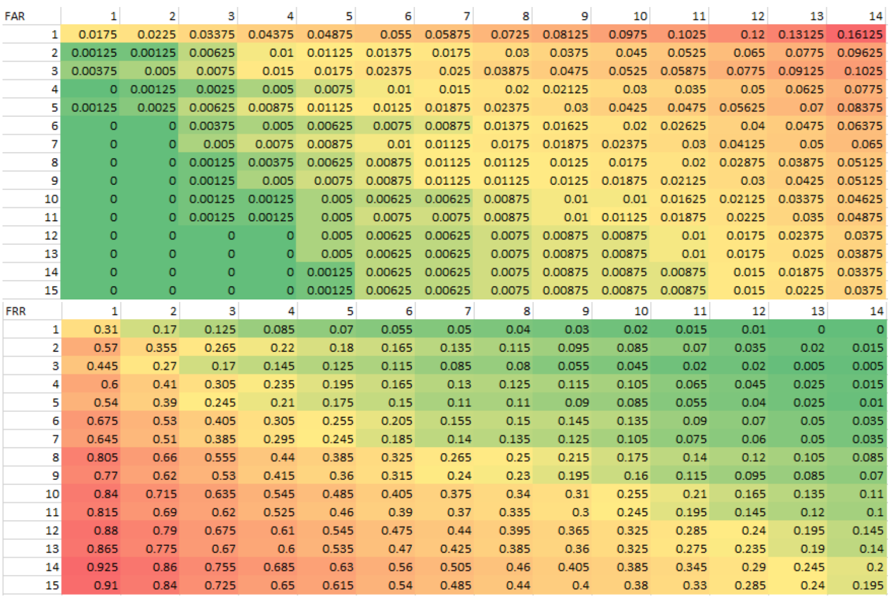
To accept a test signature, the average distance between it and the j-nearest signatures should be less than the average of the average distance between each j signature and its k-NN signatures or a predetermined scale of that threshold (e.g., 1.2 or 1.4). As shown in Table 2, the coefficients (θ) that were used in the top-ten results of each database were between 1.1 and 1.8. The tested threshold coefficients were between 0.8 and 1.8, but using values that were below 1 showed less accurate results. We can say that the optimal value of θ was between 1.2 and 1.6. A combination of 15 references and a scaling threshold of $\theta = 1.5$ will perform very well in a jk-NN-based online signature-verification system.

Table 2
Best results of experiments

SVC2004			MCYT			SigComp'11			SigComp'15		
References	Threshold	Min AER%	References	Threshold	Min AER%	References	Threshold	Min AER%	References	Threshold	Min AER%
15	1.2	3.938	13	1.6	2.605	15	1.5	1.752	13	1.5	6.000
15	1.4	4.063	13	1.7	2.678	13	1.5	1.841	13	1.3	7.000
15	1.5	4.063	13	1.8	2.690	14	1.5	1.917	13	1.2	7.167
15	1.6	4.125	12	1.8	2.691	13	1.8	1.929	13	1.4	7.167
15	1.3	4.188	12	1.6	2.717	13	1.6	1.978	11	1.3	7.250
15	1.1	4.438	14	1.6	2.722	13	1.7	1.978	11	1.5	7.500
15	1.7	4.438	13	1.5	2.738	15	1.6	2.007	10	1.4	7.667
14	1.5	4.729	12	1.5	2.742	11	1.6	2.036	12	1.3	7.667
14	1.2	4.750	14	1.5	2.742	11	1.7	2.057	12	1.4	7.778
14	1.4	4.750	15	1.7	2.760	13	1.4	2.064	12	1.5	7.778

4.1.2. The values j and k

The j and k values have different effects on the FAR and FRR. The FRR improves with small values of j and larger values of k. This effect happens because of the more distant neighbors that are included by increasing k (which eases the chosen threshold and accepts more signatures, thereby decreasing the rejection rate).



FAR	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0.0175	0.0225	0.03375	0.04375	0.04875	0.055	0.05875	0.0725	0.08125	0.0975	0.1025	0.12	0.13125	0.16125
2	0.00125	0.00125	0.00625	0.01	0.01125	0.01375	0.0175	0.03	0.0375	0.045	0.0525	0.065	0.0775	0.09625
3	0.00375	0.005	0.0075	0.015	0.0175	0.02375	0.025	0.03875	0.0475	0.0525	0.05875	0.0775	0.09125	0.1025
4	0	0.00125	0.0025	0.005	0.0075	0.01	0.015	0.02	0.02125	0.03	0.035	0.05	0.0625	0.0775
5	0.00125	0.0025	0.00625	0.00875	0.01125	0.0125	0.01875	0.02375	0.03	0.0425	0.0475	0.05625	0.07	0.08375
6	0	0	0.00375	0.005	0.00625	0.0075	0.00875	0.01375	0.01625	0.02	0.02625	0.04	0.0475	0.06375
7	0	0	0.005	0.0075	0.00875	0.01	0.01125	0.0175	0.01875	0.02375	0.03	0.04125	0.05	0.065
8	0	0	0.00125	0.00375	0.00625	0.00875	0.01125	0.01125	0.0125	0.0175	0.02	0.02875	0.03875	0.05125
9	0	0	0.00125	0.005	0.0075	0.00875	0.01125	0.01125	0.0125	0.01875	0.02125	0.03	0.0425	0.05125
10	0	0	0.00125	0.00125	0.005	0.00625	0.00625	0.00875	0.01	0.01	0.01625	0.02125	0.03375	0.04625
11	0	0	0.00125	0.00125	0.005	0.0075	0.0075	0.00875	0.01	0.01125	0.01875	0.0225	0.035	0.04875
12	0	0	0	0	0.005	0.00625	0.00625	0.0075	0.00875	0.00875	0.01	0.0175	0.02375	0.0375
13	0	0	0	0	0.005	0.00625	0.00625	0.0075	0.00875	0.00875	0.01	0.0175	0.025	0.03875
14	0	0	0	0	0.00125	0.00625	0.00625	0.0075	0.00875	0.00875	0.00875	0.015	0.01875	0.03375
15	0	0	0	0	0.00125	0.00625	0.00625	0.0075	0.00875	0.00875	0.00875	0.015	0.0225	0.0375

FRR	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0.31	0.17	0.125	0.085	0.07	0.055	0.05	0.04	0.03	0.02	0.015	0.01	0	0
2	0.57	0.355	0.265	0.22	0.18	0.165	0.135	0.115	0.095	0.085	0.07	0.035	0.02	0.015
3	0.445	0.27	0.17	0.145	0.125	0.115	0.085	0.08	0.055	0.045	0.02	0.02	0.005	0.005
4	0.6	0.41	0.305	0.235	0.195	0.165	0.13	0.125	0.115	0.105	0.065	0.045	0.025	0.015
5	0.54	0.39	0.245	0.21	0.175	0.15	0.11	0.11	0.09	0.085	0.055	0.04	0.025	0.01
6	0.675	0.53	0.405	0.305	0.255	0.205	0.155	0.15	0.145	0.135	0.09	0.07	0.05	0.035
7	0.645	0.51	0.385	0.295	0.245	0.185	0.14	0.135	0.125	0.105	0.075	0.06	0.05	0.035
8	0.805	0.66	0.555	0.44	0.385	0.325	0.265	0.25	0.215	0.175	0.14	0.12	0.105	0.085
9	0.77	0.62	0.53	0.415	0.36	0.315	0.24	0.23	0.195	0.16	0.115	0.095	0.085	0.07
10	0.84	0.715	0.635	0.545	0.485	0.405	0.375	0.34	0.31	0.255	0.21	0.165	0.135	0.11
11	0.815	0.69	0.62	0.525	0.46	0.39	0.37	0.335	0.3	0.245	0.195	0.145	0.12	0.1
12	0.88	0.79	0.675	0.61	0.545	0.475	0.44	0.395	0.365	0.325	0.285	0.24	0.195	0.145
13	0.865	0.775	0.67	0.6	0.535	0.47	0.425	0.385	0.36	0.325	0.275	0.235	0.19	0.14
14	0.925	0.86	0.755	0.685	0.63	0.56	0.505	0.46	0.405	0.385	0.345	0.29	0.245	0.2
15	0.91	0.84	0.725	0.65	0.615	0.54	0.485	0.44	0.4	0.38	0.33	0.285	0.24	0.195

Figure 2. Effect of J (up to down) and K (left to right) on FAR (up) and FRR (bottom)

Also, this has the exact opposite effect on the FAR for the same reason; more signatures are accepted in this situation, leading to the acceptance of some forged signatures, which leads to a higher FAR. Thus, the FAR is smaller with smaller values of k and greater values of j (see Figure 2).

These two effects require consideration when choosing the values of j and k since the AER is affected by both the FAR and FRR. In our approach, the parameters' optimal values will be those values that tolerate both error types.

4.1.3. A jk-NN algorithm performance

A jk-NN classifier was used for verification instead of k-NN for the proposed algorithm. Although k-NN provides good accuracy, our results showed that using jk-NN can improve the verification system's accuracy as compared to the k-NN algorithm. A comparison between the two algorithms is shown in Figure 3. In the SVC2004 database, the increase in accuracy was 2.02%; this increase was 0.59% for the MCYT-100, 0.73% for the SigComp'11, and 10% for the SigComp'15 databases.

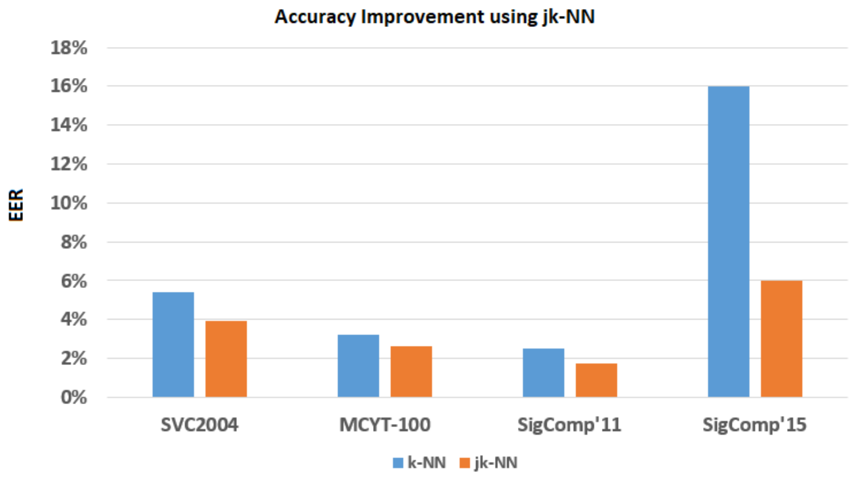


Figure 3. Accuracy improvement of jk-NN algorithm compared to k-NN algorithm

Figure 3 shows that the jk-NN algorithm achieved good performance for the different databases: it had a 3.93% error rate when using the SVC2004 database, 2.6% for MCYT-100, 1.75% for SigComp'11, and 6% for SigComp'15. This approach would not always be feasible in practice, so we suggest an improved jk-NN in the next section using the previous evaluation of the parameters and optimizing the k value while only using available references.

4.1.4. Improved jk-NN

In the previous subsections, we discussed the minimal AER that was achieved in each experiment. We obtain a minimal AER by selecting the best values of j and k, which cannot be reliably predetermined under actual circumstances. Nevertheless, we can use the result for our SVS after examining the effect of the reference number and the best threshold scale (which can provide good results in most situations).

In this section, we introduce an algorithm that is based on the minimum value of the FRR that is reached within the training set to choose the best value of k (calculated under real-life circumstances where a certain number of signature references are available and used for this purpose). Using the previous evaluation of the algorithm parameters, values $J = 5$ and $\theta = 1.5$ are used with 15 reference signatures in the proposed algorithm. The idea is to divide the references (R) into two groups: the first group (R_t) is used to calculate the threshold, while the other is used for testing. The FRR is calculated in each iteration using different values of k from the (K_s) group of values. The best value is assigned for k (K_{opt}) to the k that provided the minimum $FRR(K)$ (the FRR using the k-NN) among all of the K_s values. The SVS can use these values in the verification phase. The minimum value of (FRR) will not always provide the optimal value of (k) since references R and R_t are not the same and will provide different results. Still, it will indicate one of the best values of k that can produce a very accurate result. The new formula of the algorithm is presented in the following equations:

$$\frac{j \in J^* : \left[d\left(S_j, S_{jnn}\right) < \theta \frac{1}{K_{opt}} \sum_{k=1}^{K_{opt-1}} d\left(S_{jnn}, S_{jknn}\right)\right]}{5} > 0.5 \tag{4}$$

where

$$J^* \in \{1, 2, 3, 4, 5\} \tag{5}$$

and K_{opt} is the value of k that provides:

$$\min_{\forall K \in K_s} FRR(K) \tag{6}$$

From the previous conclusions, we chose R_t to be 10, $J = 5$, and $\theta = 1.5$. The proposed algorithm chooses the best k for each database and applies the SVS. The achieved accuracies are shown in Table 3.

Table 3

Min. false acceptance rate and average error rate of proposed algorithm

Database	min FRR	AER
SVC2004	5.83%	8.00%
MCYT100	2.67%	3.26%
SigComp'15	0.00%	13.00%
SigComp'11	2.60%	2.22%

These results show that jk-NN can provide promising practical results.

5. Discussion and conclusion

This work proposed an enhanced algorithm for jk-NN that was based on online signature-verification systems. We began by evaluating the impact of the proposed algorithm's main parameters, then presented and evaluated the verification method. A comparison was also presented between the use of the k-NN and jk-NN algorithms and showed that jk-NN enhanced accuracy when using the same verification system for both methods. The accuracy increased by 2.02% for the SVC2004, by 0.59% for the MCYT-100, by 0.73% for the SigComp'11, and by 10% for the SigComp'15 databases.

The values j, k, the reference count, and the threshold are the main parameters of jk-NN. We found that 15 references (a threshold coefficient of 1.5) would provide promising results based on the proposed evaluation phase's experimental results. The case was different for the j and k values since they behaved differently under different circumstances. For j, the best results centered around a value of 5; for k, the optimal value requires careful consideration to achieve optimal results. Using these details, we have proposed an online jk-NN signature-verification method that uses the preferred parameter values and calculates the optimal k value for each signer. This method is realistic and usable in those real-life scenarios in which only the references of the signers are available. The accuracy of the proposed approach is encouraging; the achieved AER was 8% for SVC2004, 3.26% for MCYT-100, 13% for SigComp'15, and 2.22% for SigComp'11.

The optimized algorithm can be used in further research along with other techniques in the verification step, which will result in more-accurate and competitive verification systems for realistic applications.

Acknowledgements

The work presented in this paper has been carried out in the frame of Project No. 2019-1.1.1-PIACI-KFI-2019-00263, which was implemented with the support provided from the National Research, Development and Innovation Fund of Hungary and financed under the 2019-1.1. funding scheme.

References

[1] Abdelrahaman A.A.A., Abdallah M.E.A.: K-nearest neighbor classifier for signature verification system. In: *2013 International Conference on Computing, Electrical and Electronic Engineering (ICCEEE)*, pp. 58–62, IEEE, 2013.

[2] Azmi A.N., Nasien D., Omar F.S.: Biometric signature verification system based on freeman chain code and k-nearest neighbor, *Multimedia Tools and Applications*, vol. 76(14), pp. 15341–15355, 2017.

[3] Cabral G.G., Oliveira A.L.I., Cahú C.B.G.: Combining nearest neighbor data description and structural risk minimization for one-class classification, *Neural Computing and Applications*, vol. 18(2), pp. 175–183, 2009.

[4] Cover T., Hart P.: Nearest neighbor pattern classification, *IEEE Transactions on Information Theory*, vol. 13(1), pp. 21–27, 1967.

[5] Gao L., Zhang L., Liu C., Wu S.: Handling imbalanced medical image data: A deep-learning-based one-class classification approach, *Artificial Intelligence in Medicine*, vol. 108, 101935, 2020.

[6] Goldin D.Q., Kanellakis P.C.: On similarity queries for time-series data: constraint specification and implementation. In: *International Conference on Principles and Practice of Constraint Programming*, pp. 137–153, Springer, 1995.

[7] Guru D.S., Prakash H.N.: Online Signature Verification and Recognition: An Approach Based on Symbolic Representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31(6), pp. 1059–1073, 2008.

[8] Harfiya L.N., Widodo A.W., Wihandika R.C.: Offline signature verification based on pyramid histogram of oriented gradient features. In: *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*, pp. 23–28, IEEE, 2017.

[9] Isha I., Pooja P., Varsha V.: Offline Signature Verification based on Euclidean distance using Support Vector Machine, *International Journal of Advanced Engineering, Management and Science*, vol. 2(8), 2016.

[10] Jain A., Singh S.K., Singh K.P.: Handwritten signature verification using shallow convolutional neural network, *Multimedia Tools and Applications*, vol. 79, pp. 19993–20018, 2020.

[11] Khan S.S., Ahmad A.: Relationship between Variants of One-Class Nearest Neighbors and Creating Their Accurate Ensembles, *IEEE Transactions on Knowledge and Data Engineering*, vol. 30(9), pp. 1796–1809, 2018.

[12] Kholmatov A., Yanikoglu B.: SUSIG: an on-line signature database, associated protocols and benchmark results, *Pattern Analysis and Applications*, vol. 12(3), pp. 227–236, 2009.

[13] Komiya Y., Ohishi T., Matsumoto T.: A Pen Input On-Line Signature Verifier Integrating Position, Pressure and Inclination Trajectories, *IEICE Transactions on Information and Systems*, vol. 84(7), pp. 833–838, 2001.

[14] Liu Y., Yang Z., Yang L.: Online Signature Verification Based on DCT and Sparse Representation, *IEEE Transactions on Cybernetics*, vol. 45(11), pp. 2498–2511, 2014.

[15] Liwicki M., Malik M.I., van den Heuvel C.E., Chen X., Berger C., Stoel R., Blumenstein M., Found B.: Signature Verification Competition for Online and Offline Skilled Forgeries (SigComp2011). In: *2011 International Conference on Document Analysis and Recognition*, pp. 1480–1484, IEEE, 2011.

[16] Malallah F.L., Ahmad S.M.S., Adnan W.A.W., Arigbabu O.A., Iranmanesh V., Yussof S.: Online Handwritten Signature Recognition by Length Normalization Using Up-Sampling and Down-Sampling, *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, vol. 4(1), pp. 302–313, 2015.

[17] Malik M.I., Ahmed S., Marcelli A., Pal U., Blumenstein M., Alewijns L., Liwicki M.: ICDAR2015 competition on signature verification and writer identification for on-and off-line skilled forgeries (SigWIcomp2015). In: *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1186–1190, IEEE, 2015.

[18] Manevitz L.M., Yousef M.: One-Class SVMs for Document Classification, *Journal of Machine Learning Research*, vol. 2, pp. 139–154, 2001.

[19] Nanni L.: Experimental Comparison of One-Class Classifiers for Online Signature Verification, *Neurocomputing*, vol. 69(7–9), pp. 869–873, 2006.

[20] Okawa M.: Online signature verification using single-template matching with time-series averaging and gradient boosting, *Pattern Recognition*, vol. 102, pp. 107–227, 2020.

[21] Ortega-Garcia J., Fierrez-Aguilar J., Simon D., Gonzalez J., Faundez-Zanuy, M., Espinosa V., Satue A., Hernaez I., Igarza J., Vivaracho C., Escudero D., Moro Q.-I.: MCYT baseline corpus: a bimodal biometric database, *IEE Proceedings – Vision, Image and Signal Processing*, vol. 150(6), pp. 395–401, 2003. https://digital-library.theiet.org/content/journals/10.1049/ip-vis_20031078.

[22] Pippin C.E.: Dynamic Signature Verification using Local and Global Features. Technical report, Georgia Institute of Technology, 2004.

[23] Rodríguez-Ruiz J., Mata-Sánchez J.I., Monroy R., Loyola-González O., López-Cuevas A.: A one-class classification approach for bot detection on Twitter, *Computers & Security*, vol. 91, 101715, 2020.

[24] Saleem M., Kovari B.: Performance Evaluation of the JK-nearest Neighbor Online Signature Verification Parameters. In: *2021 4th International Conference on Data Storage and Data Engineering*, pp. 1–5, 2021.

[25] Saxena A., Tripathi K., Khanna A., Gupta D., Sundaram S.: Emotion Detection through EEG Signals using FFT and Machine Learning Techniques. In: *International Conference on Innovative Computing and Communications*, pp. 543–550, Springer, 2020.

[26] Tolosana R., Vera-Rodriguez R., Ortega-Garcia J., Fierrez J.: Preprocessing and Feature Selection for Improved Sensor Interoperability in Online Biometric Signature Verification, *IEEE Access*, vol. 3, pp. 478–489, 2015.

[27] Vargas-Bonilla J.F., Ferrer M.A., Travieso C.M., Alonso J.B.: Off-Line Signature Verification Based on High Pressure Polar Distribution. In: *Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition, ICFHR 2008*, pp. 373–378, 2008.

[28] Vickram P., Sri Krishna A., Swapna D.: Offline Signature Verification Using Support Local Binary Pattern, *International Journal of Artificial Intelligence and Applications (IJAIA)*, vol. 7(6), pp. 85–94, 2016.

[29] Wahyono, Jo K.: LED Dot matrix text recognition method in natural scene, *Neurocomputing*, vol. 151(Part 3), pp. 1033–1041, 2015.

[30] Winston J.J., Hemanth D.J.: Moments-Based Feature Vector Extraction for Iris Recognition. In: *International Conference on Innovative Computing and Communications*, pp. 255–263, Springer, 2020.

[31] Yang L., Cheng Y., Wang X., Liu Q.: Online handwritten signature verification using feature weighting algorithm relief, *Soft Computing*, vol. 22(23), pp. 7811–7823, 2018.

[32] Yeung D.Y., Chang H., Xiong Y., George S., Kashi R., Matsumoto T., Rigoll G.: SVC2004: First International Signature Verification Competition. In: D. Zhang, A.K. Jain (eds.), *Biometric Authentication. ICBA 2004. Lecture Notes in Computer Science*, vol. 3072, pp. 16–22, Springer, Berlin, Heidelberg, 2004.

Affiliations

Mohammad Saleem
Budapest University of Technology and Economics, 1111 Budapest, Muegyetem rkp. 3., Hungary, msaleem@aut.bme.hu

Bence Kovari
Budapest University of Technology and Economics, 1111 Budapest, Muegyetem rkp. 3., Hungary, kovari@aut.bme.hu

Received: 29.01.2021
Revised: 16.05.2021
Accepted: 26.06.2021