Pattern Recognition Letters 80 (2016) 129–136

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Online signature verification based on writer dependent features and classifiers☆

K.S. Manjunatha [a], S. Manjunath [b], D.S. Guru [a,*], M.T. Somashekara [c]

[a] Department of Studies in Computer Science, University of Mysore, Manasagangothri, Mysore 570 006, Karnataka, India
[b] Department of Computer Science, Central University of Kerala, Kasargod 671316, India
[c] Department of Computer Science and Applications, Bangalore University, Bangalore 560056, India

ARTICLE INFO

Article history:
Received 9 September 2015
Available online 29 June 2016

Keywords:
Writer specific features
Writer dependent classifier
Common classifier
Online signature verification

ABSTRACT

In this work, an approach for online signature verification based on writer specific features and classifier is investigated. Existing models for online signatures are generally writer independent, as a common classifier or fusion of classifier is used on a common set of features for all writers during verification. In contrast, our approach is based on the usage writer dependent features as well as writer dependent classifier. The two decisions namely optimal features suitable for a writer and a classifier to be used for authenticating the writer are taken based on the error rate achieved with the training samples. The performance of our model is tested on both MCYT-100 (DB1), a sub corpus of MCYT data set, consisting of signatures of 100 writers, MCYT-330 (DB2) consisting of signatures of all 330 writers and visual subcorpus of SUSIG dataset. Experimental results confirm the effectiveness of writer dependent characteristics for online signature verification. The error rate that we achieved is lower when compared to many existing contemporary works on online signature verification especially when the number of training samples available for each writer is sufficient enough.

1. Introduction

Signature has been the most commonly adapted behavioral biometric trait for human identity establishment in many applications. Depending on the acquisition mode, signature verification can be categorized as offline and online [21]. In an offline mode, verification is done based on the information extracted from the hard copy of the signature image captured from a paper document. In an online mode, signature is captured using special devices such as smart pens, pressure sensitive tablets etc., which can record dynamic features of a writer such as velocity, pressure, acceleration etc., and verification is done considering both static and dynamic features. As these dynamic features are unique for an individual writer and also difficult to forge, online signature verification is more reliable than an offline mode.

Based on the representation schemes and matching techniques, online signature verification methods can be categorized as parametric and function based approaches [34]. A parametric based approach results in more compact representation as the entire signature is represented by means of a few parameters [25,36,38]. During verification, corresponding parameters of a test signature and a reference signature are compared. Parameters are further classified as global and local parameters depending on whether they correspond to the whole signature or to a specific point in the signature [20]. In a function based approach, a signature is represented by means of time functions of various dynamic properties such as pressure, velocity, acceleration etc., and verification is done by comparing the time functions of a test signature and a reference signature [22,33,39,40]. A function based approach generally takes a longer matching time compared to a parametric based approach yet resulted in lower error rate.

In literature we can see the application of various classifiers for online signature such as SVM [15,32], neural networks [1,5], HMM [2,3,12], Parzen window [29,30,43], distance based [4,35], random forest [16] and symbolic classifier [17,32]. Further, fusion based approaches are also proposed. Fusion may be either at the feature level or at the score level. In [28], the effect of different dynamic features such as pen pressure, azimuth and pen altitude on the verification performance is investigated. Rohilla et al. [37] proposed an approach where the various online signature features are categorized and are fused in different combinations for verification. Aguiliar et al. [1,2] proposed an approach where the matching scores obtained from two classifiers trained on different categories of features are fused to obtain a combined score for authenticating a signature. Nanni [29] proposed an approach where the matching

☆ This paper has been recommended for acceptance by Umapada Pal.
* Corresponding author. Tel.: +91 9620228005.
E-mail address: dsg@compsci.uni-mysore.ac.in, dsguruji@yahoo.com (D.S. Guru).

http://dx.doi.org/10.1016/j.patrec.2016.06.016
0167-8655/

score of various single class classifiers are fused using sum rule. In these works, it has been well established that the fusion based approaches result in a considerable improvement in the performance of the system when compared to the performance of an individual classifier. Cordella et al. [9] proposed a multi-expert approach where the decision on a test signature is taken based on combined decision of the individual experts. Zhang et al. [42] proposed a three stage verification system considering global, local and function features. Verification is done in stages considering these categories of features and a test signature is accepted as genuine if it passes through all the three stages. In multi expert approach [6], a signature is segmented into different strokes and each stroke is represented in different domains. Each stroke is authenticated individually and the final decision is taken based on the weighted average of the decisions of individual strokes. Approaches based on ensemble of classifiers also have been attempted [26,30].

As a signature of a writer depends on his/her physical and mental state, the effectiveness of a verification system depends on how best the writer dependent characteristics are considered. Generally, in a signature, writer dependent characteristics include writer dependent threshold, writer dependent features and writer dependent classifiers. Most of the existing works on online signature verification exploit writer dependency at the threshold level where different similarity thresholds are used for different writers [1,2,17,21]. It has been well argued in these works that the usage of writer dependent threshold resulted in lower error rate compared to the usage of a common threshold for all writers.

Few attempts exploiting writer dependency at feature level can be traced where different set of features are used for different writers to effectively preserve the characteristics of the respective writer. In [41], optimal features for a writer are selected using genetic algorithm based on the discriminating power of the feature vector of the writer. But the main drawback of the genetic algorithm is the need for setting up of a number of parameters such as mutation probability, crossover probability, stop condition etc. Guru et al. [18,19] proposed a model based on writer dependent features which are selected based on a score computed for each feature of the respective writer, thereby resulting in selection of different set of features for different writers.

In the existing works, the utilization of writer dependency is limited to the usage of writer dependent thresholds and writer dependent features. Writer dependency has not been still exploited at classifier level especially for online signatures. Eskander et al. [11] proposed a hybrid approach for offline signatures where initially a writer independent classifier is built for each individual and later a writer dependent classifier is designed for each writer when enough number of samples are available. In spite of several approaches, still there is a difference in the way a human expert does verification when compared to a machine. Generally, a human expert looks for a different set of discriminating characteristics for different writers. Hence for a verification system to be effective, it requires considering writer dependent features rather than a common set of features for all writers. Further, the matching strategy adopted by a human expert will also be different for different writers. As the performance of any classifier depends on the nature of training samples, usage of same classifier for all writers is not effective. The reason for variations in the distribution of training signatures for different writer is due to variations in signing from a writer to a writer [24]. Hence, an automatic verification system based on the usage of writer dependent classifier is more effective when compared to the usage of a common classifier.

Considering these factors, in this work, we investigate an approach for online signature verification utilizing writer dependent characteristics. We exploit writer dependency both at feature level and at classifier level in two different stages. In the first stage, writer dependent features are selected to effectively preserve the characteristics of a particular writer. In the second stage, a classifier suitable for a writer is trained using the selected features. Even though a writer specific model requires a classifier to be trained each time when a new user is enrolled to the system, it is more secured than the writer independent system. Considering the security issues in most of the applications, it is necessary to build a verification system based on writer dependent characteristics. Overall, the major contributions of this work are:

- Exploration of writer dependent features and adaption of writer dependent classifier.
- A quantitative study on the relationships between writer dependent features and writer dependent classifiers on verification performance.

This paper is organized as follows. In Section 2, we discuss different stages of our proposed model. Details of training and testing data, experimental protocol along with the results are given in Section 3. A comparative study of our model with other existing models is reported in Section 4. Detailed critical discussion of the proposed model is presented in Section 5 and finally conclusions and future avenues are drawn in Section 6.

2. Proposed model

The proposed model has three stages; selection of writer dependent features, fixing up of a suitable classifier for a writer followed by signature verification based on the selected features and classifier.

2.1. Writer dependent feature selection

In this work, writer dependent features are selected using the feature selection algorithm proposed by Cai et al. [7]. It is a filter type feature selection algorithm which works on the principle of spectral clustering. Features selected indicate the ability of the feature in preserving the cluster structure. In our work, features for each writer are selected as follows. Given n number of signatures of a writer each characterized by P features, the feature selection algorithm computes the score for each of the P features and selects d features ($d < P$) out of P features with top scores. The steps in the adapted feature selection method are

- Define a graph with n vertices each corresponding to a data point x_i and a weight matrix representing the relationships between each data point and its nearest neighbor using heat-kernel weighting scheme.

$$W_{ij} = e^{-\frac{\|x_i - x_j\|}{\sigma}} \tag{1}$$

- Compute the graph Laplacian $L = D - W$ where W is the weight matrix and D is the diagonal matrix whose elements are the row sum or column sum of the weight matrix.
- Solve the generalized eigen problem $Ly = \lambda Dy$ where $Y = (y_1, y_2, y_3, \ldots, y_K)$ are the eigen vectors of the above eigen problem. Each row of Y is the flat embedding for each data points.
- After flat embedding for the data points are obtained, the contribution of each feature in differentiating each cluster is measured as follows; given y_k, a relevant subset of features is obtained by minimizing the fitting error as

$$\min_{a_k} \|y_k - X^T a_k\|^2 + \beta\,|a_k| \tag{2}$$

Each a_k contains the combination coefficients for different features in approximating y_i. $|a_i|$ is the $L-1$ norm of a_k. If the data set consists of K clusters, then after obtaining K sparse coefficient vectors as discussed, a subset containing non-zero coefficients in a_k

corresponding to a feature is obtained. For every feature J, a score called $MCFS(J)$ is computed as

$$MCFS(J) = \max_k |a_{k,j}| \tag{3}$$

where $a_{k,j}$ is the jth element of a_k

Based on the MCFS score, only d features with top MCFS scores are selected for each writer. In our work, the value of the parameter d is empirically fixed up during experimentation based on the equal error rate (EER). The d features selected results in lowest EER for a particular writer and is decided based on the EER obtained with validation samples. Even though for every writer, d number of features is selected, the indices of the selected d features vary from a writer to a writer thereby resulting in writer dependent features. After selecting the d number of features, the indices of all the d features selected are stored in the knowledgebase for future usage during verification stage.

2.2. Classifier selection

A decision regarding the adaption of writer dependent classifier is arrived as follows. Let there be N number of users and each providing n number of samples. Out of n number of samples, n_t samples are used for training purpose and n_v samples are used for validation. For validation, we need forgery samples also and hence we considered n_f number of random forgery samples during validation process. Let there be Cnumber of classifiers. Given a writer i with n number of samples, P number of features are extracted. Out of the available P features, we select d number of features for each writer. To select d number of writer dependent features, we recommend using the feature selection method discussed in Section 2.1. Hence after selecting d number of features, we have a data matrix of size $n \times d$ for ith writer. Out of $n \times d$ data matrix, $n_t \times d$ is used as training set and trained each of the C classifiers. Using $n_v \times d$ and $n_f \times d$, EER is obtained for each classifier. i.e. for ith writer we have

$$E_C = \{EER^i_{C_1}, EER^i_{C_2}, EER^i_{C_3}, \ldots, EER^i_{C}\} \tag{4}$$

where $EER^i_{C_j}$ refers to EER of jth classifier for ith writer.

The experimentation is carried out for T number of trails by changing the training and validation samples. The training and validation samples are randomly selected without overlapping in each of the T trails. For each trial, a classifier with a minimum error rate is identified.

$$\text{i.e } C^T_{sel} = \min\{E_c\} \tag{5}$$

Let $C_{sel} = \{C^1_{sel}, C^2_{sel}, C^3_{sel}, \ldots, C^T_{sel}\}$ be the set of classifiers identified because of T different trials, where C^k_{sel} is the classifier selected at kth trial. In order to select the best classifier among the C_{sel} list, we rank each classifier based on its frequency for a particular writer as defined in (6).

$$Frequency\,(C_j,\ i) = \frac{\text{No. times } j^{\text{th}} \text{ classifier selected for } i^{\text{th}} \text{ writer}}{\text{Number of trials conducted}} \tag{6}$$

The classifier having the highest frequency say C^i_j shall be the best classifier for the ith writer and is selected for writer i. Similarly for all writers in the database, a classifier is selected using the above mentioned procedure. All parameters selected for a writer i namely the indices of all the d features selected and the classifier selected for a writer i say C^i_j along with all internal parameters of the classifier are stored in the knowledgebase.

2.3. Signature verification

The decision regarding the acceptance or rejection of a test signature claimed to be of writer i is arrived as follows. Given an unknown sample S_{test} claiming that it belongs to the writer$i$, first, the features selected for a claimed writer i available in the knowledgebase are retrieved. The same features of the test signature S_{test} are compared with the corresponding features of the reference signatures of writer i. The recommended classifier C^i_j along with its fixed internal parameters available in the knowledgebase is used for verification of test signature of the claimed writer i.

2.4. Time complexity of signature verification

The two main stages in any biometric system are enrollment and verification. In this work, the enrollment phase includes selection of suitable features and a classifier for each writer. Indices of all selected features and details of the classifier are stored in the knowledgebase which will be used during verification stage. As enrollment takes place offline, we do not take into account the time spent for enrollment.

During verification, first we need to fetch the indices of all $d$features of the claimed ith writer from the knowledgebase which is basically a searching operation which takes d units of time. Then, we need to compute only those d features with respect to ith writer. The time required to compute these d features varies from a writer to a writer. Let $T_1(F^i_d)$ be the time taken to compute all d features. Then we need to select a classifier C^i_j that has been selected for the ith writer during validation which takes 1 unit of time. Finally, the test signature is given to the classifier which decides whether the given signature is genuine or not. The time complexity of the classifier also varies from a writer to a writer as different classifiers are selected for different writers. Let $T_2(C^i_j)$ be the time complexity of the jth classifier selected for ith writer. So, overall time complexity of the proposed signature verification for ith writer is,

$$\text{Time} = O\left(d + T_1\left(F^i_d\right) + 1 + T_2\left(C^i_j\right)\right) \tag{7}$$

The time complexity of each of the classifier depends on the size of enrollment samples and also on the number of features used. Since all the classifier used in this work are well known classifiers in the field of pattern recognition, for description about complexity of these classifiers the reader can refer [10].

3. Experimentation

Dataset: We conducted experimentations on the MCYT online signature dataset DB1 consisting of 25 genuine and 25 skilled forgery samples of 100 writers and also on the dataset DB2 consisting of same number of genuine and forgery samples of 330 writers. We have considered 100 global features for our experimentation. The details of these 100 features can be found in the work [1]. Also, we have conducted experimentation on publicly available Visual Subcorpus of SUSIG database which consist of signatures by 94 writers [23].

Classifiers: We have considered 6 different classifiers which are either statistical classifiers or neural network based classifier. Statistical classifiers include Naïve Bayesian (NB), nearest neighbor (NN), support vector machine (SVM), principal component analysis (PCA) and linear discriminant analysis (LDA) and from the neural network based category we have considered probabilistic neural network (PNN) classifier.

3.1. Experimental results on MCYT dataset

We trained the system with 05 and 20 genuine signatures per writer. In both the situations, we have considered equal number of random forgery samples for validation purpose. Genuine signatures of other writers are taken as a random forgery for a

Table 1
EER with the usage of a single classifier as common to all writers and also EER obtained by the proposed approach (C_7).

	DB1				DB2			
	Skilled		Random		Skilled		Random	
	5	20	5	20	5	20	5	20
C_1	51.05	6.10	37.98	7.40	31.05	6.00	39.13	6.94
C_2	20.33	1.20	8.90	1.00	18.94	1.03	7.75	0.88
C_3	21.10	1.20	7.90	1.20	19.58	1.01	7.54	0.82
C_4	20.98	13.60	15.93	14.60	20.05	11.15	13.55	11.67
C_5	20.88	3.20	12.60	3.30	20.41	1.82	12.76	0.91
C_6	20.40	1.60	8.93	1.00	19.53	1.64	7.86	1.05
C_7	19.43 (75)	1.10 (50)	7.75 (60)	0.80 (50)	18.41 (25)	0.94 (60)	7.32 (60)	0.67 (65)

writer. Further, the training set is split into training and validation set. Fifty percent of the available training samples are used for validation purpose to fix up the values for d and a classifier. During validation, the parameters are adjusted so that the two error rates false acceptance rate (FAR) and false rejection rate (FRR) are equal i.e., equal error rate (EER). Once the parameters are set, without altering them we have used the same parameters during testing also. It is not possible to carry out the experimentation during testing under varying threshold and identifying EER where FAR will be equal to FRR. Hence, the average of best values FAR and FRR is taken as equal error rate (EER) as recommended in [8]. We conducted verification experiments with both skilled and random forgeries. In case of skilled forgery testing, remaining genuine signatures and all the skilled forgery samples are used for calculating FRR and FAR, respectively. In case of random forgery testing, remaining genuine signatures and one genuine signature of other writers not considered for validation process are used for calculating FRR and FAR, respectively. Depending on the training and testing set used, we have four different categories of testing namely Skilled_05, Skilled_20, Random_05 and Random_20. Details of training and testing samples used in all the four categories of testing for DB1 and DB2 are available in [17].

We conducted experimentation for different number of trials (T) and in each trial, training and testing signatures were randomly selected. It is observed that the number of trials has an effect on the selection of classifiers. The performance of the system improves marginally with the increase in the number of trials. In our work, with T=20, we have achieved the best result. In case of tie between two classifiers, we have prioritized the order of the classifiers based on the ease of implementation depending on the complexity of various classifiers. Hence whenever a tie occurs among the classifiers, the classifier which comes first in the list is preferred. But the list can be altered according to the criteria as decided by the implementer. Initially, we conducted verification experiments using each of the individual classifier common to all writers as in a traditional setup. EER obtained when same classifier is used for all writers is as shown in Table 1.

In Table 1, the labels C_1–C_6 denote the classifiers NB, NN, SVM, PNN, LDA and PCA, respectively. Further to demonstrate the superiority of our approach denoted by C_7, verification experiments were conducted with writer dependent features and classifier. The EER obtained with writer dependent features and classifier for all the four categories of testing are shown in last row of Table 1. In Table 1, the number of features $'d'$ selected in each category of testing for the best EER for our approach is also mentioned within the parenthesis. From Table 1 it is clear that the error rate with a common classifier for all writers is higher when compared to that of the usage of writer dependent classifier.

It is interesting to observe that for some categories of testing, usage of a common classifier (as in traditional setup) for all writers resulted in an EER which is closer to the EER obtained by the proposed model (shown in Tables 2 and 3 for DB1 and DB2, respectively).

Table 2
EER obtained with a common classifier nearest to the EER of the proposed model on DB1.

Test category	Conventional classifier with lowest EER	EER	EER of the proposed model
Skilled_05	$\{C_2\}$	20.33	19.43
Skilled_20	$\{C_2,C_3\}$	1.20	1.10
Random_05	$\{C_3\}$	7.90	7.75
Random_20	$\{C_2,C_6\}$	1.00	0.80

Table 3
EER obtained with a common classifier nearest to the EER of the proposed model on DB2.

Test category	Conventional classifier with lowest EER	EER	EER of the proposed model
Skilled_05	$\{C_2\}$	18.94	18.41
Skilled_20	$\{C_3\}$	1.01	0.94
Random_05	$\{C_3\}$	7.54	7.32
Random_20	$\{C_3\}$	0.82	0.67

However, from Tables 2 and 3, it is also clear that none of the individual classifier gives lowest EER for all four categories of testing leading to confusion in selecting a classifier which works well for all categories of testing. For instance, Table 2 suggests that, it is better to use NN classifier in case of Skilled_05, SVM or NN classifier in case of Skilled_20, SVM classifier in case of Random_05 and NN or PCA classifier in case of Random_20. For DB2 it is NN classifier for Skilled_05 and SVM classifier for Skilled_20, Random_05 and Random_20. But it is not the case with our proposed model as it gives lowest EER for all four different categories of testing. Overall, the proposed model suggests a classifier for a writer which results in lowest EER irrespective of the category of testing.

Even though the proposed model is based on the usage of writer dependent features, number of features (feature dimension) for every writer is kept same in this work. To arrive at the decision regarding the number of features to be selected, we have conducted experimentation under varying number of feature dimension. For each value of the feature dimension, writer dependent classifier is selected as discussed in Section 2.2 and the verification is done using the selected classifier. The EER obtained for varying feature dimensions is shown in Table 4 for DB1 and DB2. We also have conducted verification experiments using all the features for all writers without any feature selection. The last row in Table 4 indicates the EER obtained without feature selection (WFS) but using writer dependent classifier for verification. It is clear from Table 4 that the performance of the model enhances with the combination of writer dependent features and writer dependent classifier.

Table 4
EER of the proposed model under varying feature dimension on DB1 and DB2.

Features	DB1				DB2			
	Skilled		Random		Skilled		Random	
	5	20	5	20	5	20	5	20
5	20.08	8.50	15.50	5.00	20.16	6.87	13.22	5.45
10	21.40	4.20	12.08	4.00	19.54	3.79	10.55	3.33
15	20.55	2.70	11.78	3.60	19.16	2.78	9.39	2.73
20	19.80	2.60	11.05	2.60	18.89	2.24	8.95	1.97
25	19.55	2.10	9.68	2.40	18.41	1.82	8.17	1.45
30	20.18	2.20	9.13	1.80	18.85	1.85	8.28	1.42
35	20.60	2.40	9.90	1.80	19.27	1.76	8.61	1.48
40	20.95	2.70	8.90	2.20	18.49	1.91	7.73	1.27
45	20.05	1.80	9.20	1.70	18.83	2.12	7.55	1.30
50	20.15	1.10	8.50	0.80	19.25	1.21	7.64	1.06
55	19.98	2.90	9.45	2.10	19.07	1.60	7.40	1.06
60	19.80	2.50	7.75	1.70	18.74	0.94	7.32	0.73
65	19.75	2.00	8.38	1.20	18.61	1.24	7.53	0.67
70	19.65	2.30	8.73	1.80	18.79	1.27	7.39	0.85
75	19.43	2.50	9.15	1.90	18.92	1.24	7.48	0.76
WFS	20.53	1.90	8.00	1.40	19.34	1.55	7.52	1.03

Table 6
EER of the proposed model under varying number of features selected for SUSIG dataset.

Features	Skilled	Random
5	11.33	8.67
10	5.53	4.47
15	4.36	4.10
20	4.42	4.20
25	2.71	2.71
30	1.92	1.92
35	2.39	2.34
40	1.92	1.81
45	2.82	2.61
WFS	2.55	2.45

Table 7
EER with the usage of a single classifier as common to all writers on SUSIG dataset.

Classifier	Skilled	Random
C_1	16.06	21.27
C_2	2.70	8.46
C_3	2.81	4.73
C_4	4.45	4.06
C_5	6.49	8.08
C_6	2.34	9.36
C_7	1.92 (30)	1.81 (40)

We further studied the effect of training size and number of features selected on the selection of a classifier. The cardinality of different classifiers for varying features selected is shown in Table 5 for 100 writers of DB1. Cardinality of a classifier is the number of users for which a particular classifier is selected. Analysis on Table 5 indicates that for less training size, NN classifier is suitable for majority of the writers irrespective of number of selected features. With large training size NB classifier is selected for majority of the writer especially when the number of features selected is less. In case of large training size, frequency of selection of SVM classifier is high when the number of features selected is high. The probability of selection of NB classifier decreases with the increase in the number of features selected for small training size. PNN classifier is not sensitive to either increase in training size or increase in the number of features selected.

3.2. Experimental results on SUSIG dataset

The database contains a total of 2000 genuine signatures collected in two sessions and 1000 skilled forgeries which include 500 highly skilled forgeries. We have used 10 genuine signatures of every writer for training purpose and the remaining genuine and all skilled forgeries for testing as in Pirlo et al. [33]. In case of random forgery testing, remaining genuine signatures and one genuine signature of other writers not considered for validation process are used experimentation. We have computed 47 global features characterizing each signature. The details of the computed features are given in appendix.

The verification results of the proposed model under varying *d* are as shown in Table 6. In Table 6, the last row denotes the EER obtained with all the 47 features for all writers without feature selection. To demonstrate the effectiveness of writer dependent classifier selection, we conducted experiments as in a traditional setup by using each of the classifier common to all writers. The EER obtained when a same classifier is used for all writers is shown in Table 7. The last row denotes EER obtained with proposed model. From Tables 6 and 7, it can be observed that the usage of writer dependent features and classifier resulted in enhanced performance when compared to a common set of features and a common classifier for all writers.

4. Comparative study

Comparing the performance of different verification systems is difficult due to the variations in the dataset used, variations in training and testing size. For comparative study we have considered similar models which are validated based on MCYT data corpus (DB1). From Table 8, it is clear that the error rate that we achieved is lowest when compared to all the models especially in case of Skilled_20 and Random_20 (except [26]). The reason for higher error rate in case of Skilled_05 and Random_05 is due to the fact that number of training samples is very less for extracting writer dependent characteristics and also for a fixed number of features, the performance of a classifier degrades due to a

Table 5
Cardinality of the different classifiers for different training size on DB1.

Classifier	Features													
	5 training signatures							20 training signatures						
	10	20	30	40	50	60	70	10	20	30	40	50	60	70
C_1	10	2	1	0	0	0	0	78	46	30	20	16	15	14
C_2	42	55	55	56	61	49	56	9	21	24	21	20	20	21
C_3	21	20	22	27	20	26	28	5	23	35	49	55	50	53
C_4	6	2	3	4	3	3	2	0	0	1	2	1	3	2
C_5	13	14	10	5	11	14	12	1	0	2	3	3	2	2
C_6	8	7	9	8	5	8	2	7	10	8	5	5	10	8

Table 8
EER of different online signature verification approaches on DB1.

Method	Skilled_05	Skilled_20	Random_05	Random_20
Proposed model	19.4	1.1	7.8	0.8
Symbolic classifier [17]	5.8	3.8	1.9	1.7
Linear programing description (LPD) [29]	9.4	5.6	3.6	2.5
Principal component analysis description (PCAD) [29]	7.9	4.2	3.8	1.4
Support vector description (SVD) [29]	8.9	5.4	3.8	1.6
Nearest neighbor description (NND) [29]	12.2	6.3	6.9	2.1
Random ensemble of base (RS) [31]	9.0	–	5.3	–
Random subspace ensemble with resampling of base (RSB) [31]	9.0	–	5.0	–
Base classifier (BASE) [31]	17.0	–	8.3	–
Parzen window classifier (PWC) [29]	9.7	5.2	3.4	1.4
Ensemble of Parzen window classifier [30]	8.4		2.9	
Ensemble of one class classifier based on over completer feature generation [26]	4.5	2.2	1.5	0.5
Mixture of Gaussian description_3(MOGD_3) [29]	8.9	7.3	5.4	4.3
Mixture of Gaussian description_2 (MOGD_2) [29]	8.1	7.0	5.4	4.3
Gaussian model description [29]	7.7	4.4	5.1	1.5
Kholmatov model (KHA) [31]	11.3	–	5.8	–
Fusion methods [31]	7.6	–	2.3	–
Regularized Parzen window classifier RPWC [31]	9.7	–	3.4	–
[27]	–	–	4.2	–

Table 9
Comparative analysis of the verification performance on SUSIG dataset with skilled forgeries.

Approach	FRR	FAR	EER	# TS
Yuen et al. [39]	14.8	2.64	8.72	10
Wang et al. [40]	2.46	2.46	2.46	05
Khalil et al. [22]	3.06	3.06	3.06	05
Pirlo et al. [33] with all domain	3.6	4.15	3.88	10
Pirlo et al. [33] with stable domain	2.15	2.10	2.13	10
Kholmatov and Yanikoglu [23]	3.03	3.03	3.03	05
Rashidi et al. [36]	2.09	2.09	2.09	05
Liu et al. [25]	0.51	0.51	0.51	05
Proposed model	3.83	0	1.92	10

#TS – number of training samples.

limited number of training samples [13]. In this work, objective is to build, a verification model based on writer dependent characteristics. Intra class variations are common in case of signature biometric trait and it needs to be characterized effectively to understand the biometric trait. In order to extract such writer dependent characteristics, definitely one needs more number of samples which is contrast to conventional writer independent models. The same has been demonstrated by the experimental results that the proposed model performs better, when enough number of samples are used for training rather than using less number of samples.

In the models that we have considered for comparative study in Table 8, for some categories of testing respective authors have not quoted the results and hence such entries are filled with (-). In Table 8, except symbolic classifier model [17], remaining are writer independent where same set of features and same classifier are used as common for all writers. Even in model [17], writer dependency has been exploited in the form of writer dependent threshold only. Further, in all the models considered for comparative study, verification is done by means of a same classifier trained with all the 100 global features for all writers. On the contrary, our model works in lower dimension when compared to other models.

Further, to demonstrate the efficacy of the proposed model, the results obtained by our model on SUSIG dataset is compared against the other state of the art models on the same dataset. Table 9 shows the verification performance of different models on SUSIG dataset.

In Table 9, it can be observed that our model performs better than other existing approaches even on SUSIG dataset. EER of our model is lower than the EER of the state of the art models except the model proposed by Liu et al. [25]. The results obtained demonstrated that the proposed model performs better than most of the functional as well as parametric models as given in Table 9.

5. Discussion

In this work, an online signature verification model is proposed based on the application of writer dependent features as well as writer dependent classifiers. Error rate that we achieved is lowest when compared to the other models for online signatures reported in Table 8 when number of training samples available for each writer is sufficiently large (Skilled_20 and Random_20). This is due to the fact that a writer dependent system requires sufficient number of samples for extracting the characteristics of an individual writer. From Table 8, it can be observed that the performance of the model is poor when the number of training samples available for each writer is less (Skilled_05 and Random_05). The decision regarding the features and also classifier suitable for a writer is arrived at based on the validation samples. In practice, obtaining a large number of samples for training purpose is not feasible. The reasons for recommending few training samples by the researchers are (i) the writers may be reluctant to give more number of samples during enrollment, and (ii) it may be difficult to store all training samples as it may require more storage space. However, in the former case, it can be generated synthetically as mention in the work Galbally et al. [14]. In the latter case, in the current trend, storage is not a big issue. In many applications such as banking where security is a major issue, obtaining enough number of training samples is not at all difficult as the customer has to give his/her signature during every transaction. Hence, our model can be deployed once enough number of signatures are captured for each writer.

In most of the work in the literature, EER for Skilled_05, Skilled_20, Random_05 and Random_20 are provided. However the performance of any model depends on the training size and hence it requires to study for a specific model what should be the training size required to achieve the best performance. In this respect, we have conducted experimentation under varying size of training samples with fixed feature dimension for both DB1 and DB2. Tables 10 and 11 show the EER obtained under varying training size for different values of number of features selected in DB1 and DB2. In Tables 10 and 11, the results are shown for features selected from 50 to 75 in step of 5 for varying training samples from 5 to 15. With respect to each value of the number of

Table 10
EER for varying training size and features selected on DB1.

Training	→ Features											
	50		55		60		65		70		75	
	Skilled	Random	Skilled	Random	Skilled	Random	Skilled	Random	Skilled	Random	Skilled	Random
5	20.15	8.50	19.98	9.45	19.80	7.75	19.75	8.38	19.65	8.73	19.43	9.15
6	13.05	7.58	13.26	6.73	12.89	6.34	12.31	6.55	12.97	6.79	13.13	6.68
7	10.72	5.69	10.69	5.22	10.19	5.58	9.75	6.08	9.61	5.22	9.50	5.03
8	9.76	5.12	9.53	5.61	8.94	5.44	8.17	5.03	8.64	4.94	8.06	4.76
9	7.56	4.28	7.97	4.81	7.00	4.47	8.37	4.62	7.25	4.44	7.94	4.47
10	6.60	3.40	6.33	2.93	5.83	3.23	6.43	3.26	6.23	3.06	6.10	3.23
11	3.21	2.21	3.18	2.29	2.64	1.78	2.50	1.50	2.25	1.71	2.28	1.43
12	2.15	1.50	1.38	0.84	1.42	0.53	1.53	1.04	1.81	1.23	1.23	0.61
13	1.29	0.83	0.92	0.58	1.12	0.46	0.96	0.63	1.12	0.42	0.91	0.71
14	1.32	0.86	1.41	1.14	1.09	0.86	1.54	1.27	1.13	0.82	1.09	1.00
15	0.95	0.90	1.55	0.95	1.25	0.60	0.70	0.45	0.90	0.70	1.05	0.75

Table 11
EER for varying training size and features selected on DB2.

Training	→ Features											
	50		55		60		65		70		75	
	Skilled	Random	Skilled	Random	Skilled	Random	Skilled	Random	Skilled	Random	Skilled	Random
5	19.25	7.64	19.07	7.40	18.74	7.32	18.61	7.53	18.79	7.39	18.92	7.48
6	12.92	6.31	13.41	6.21	12.70	6.39	12.83	6.05	12.97	6.20	13.21	6.04
7	10.56	4.99	10.26	5.13	10.19	4.87	10.55	5.08	10.07	5.17	10.37	5.25
8	8.98	4.66	8.71	4.94	8.85	4.65	8.75	4.55	9.01	4.50	8.71	4.45
9	7.83	4.25	7.88	4.42	7.41	4.36	7.63	4.15	7.25	4.15	7.58	4.24
10	6.64	3.44	6.57	3.40	6.58	3.62	6.62	3.43	6.48	3.22	6.77	3.37
11	3.92	2.70	3.82	2.83	3.08	2.04	3.10	2.06	3.45	2.30	3.12	2.15
12	2.49	1.40	2.03	0.95	1.92	0.98	2.00	0.74	2.11	1.04	2.06	1.01
13	1.44	0.86	1.42	0.87	1.24	0.81	1.55	0.74	1.65	0.84	1.34	0.74
14	1.79	1.23	1.75	1.42	1.64	1.15	1.47	0.96	1.25	0.82	1.43	0.91
15	1.57	1.27	1.54	1.08	1.58	1.47	1.24	1.11	1.51	0.92	1.71	1.29

features selected, the first column indicates the EER obtained with skilled forgery and the second column indicates the EER with random forgeries. From Tables 10 and 11, it is clear that error rate of a verification model, not only depends on the feature dimension and classifier adapted but also on the size of the training samples.

Further, in Table 8, the lowest EER for skilled forgery category is 3.8 [17] and for random forgery category, it is 1.4 [29] for 20 training samples. Most of the authors have quoted their result for DB1. However, in [17], the EER for DB2 is also quoted and in case of DB2, the best average result for skilled and random forgery category is 4.7 and 1.67 respectively with writer dependent threshold. However in case of the proposed model, even with 12 training samples, we achieve an EER of 1.23 and 0.53 for DB1 (Table 10) and 1.92 and 0.74 for DB2 (Table 11) with skilled and random forgeries respectively which is very much less compared to the state of the art with 20 training samples as given in Table 8. This indicates the superiority of the proposed model with respect to usage of training samples in obtaining low EER.

Overall, the proposed model can be treated as a generalized model which can be applied on any category of online signature features i.e., parametric or functional. Based on the type of the features, corresponding pool of classifiers need to be considered. Example, for global features (i.e., parametric in nature) one can think of Bayesian classifier, nearest neighbor classifier, neural networks, SVM etc., and for local features (i.e., functional features), classifiers such as DTW, HMM can be used. The model has not been tuned suitable for any specific category of features and classifiers. Depending on the type of features, only modification in the proposed model may be with respect to the decision on writer dependent feature selection. In case of parametric features, feature selection is necessary and in case of functional it may be eliminated as entire signature will be used for verification process. However, in this work we have concentrated only on parametric or global features for experimental purpose.

6. Conclusion and future work

In this work, a new approach for online signature verification has been proposed exploiting writer dependency both at feature level as well as classifier level. The efficacy of the model has been tested considering 6 different classifiers which are extensively used in the field of signature verification. The experimental results suggest that the proposed model which is based on the usage of writer dependent features and classifier is better than the existing models especially when the number of training samples available for a writer is more. This paper is expected to open up a new issue for further study on writer dependent classifier selection. In this work, verification is done considering only the top ranked classifier for each writer. As a future work, the model can be extended by considering the combination of classifiers based on their ranking. Further, investigation of an approach for writer dependent feature dimension and study on different feature selection approaches for selection of writer dependent features will be promising future avenues.

Acknowledgment

We acknowledge the support rendered by Prof. Anil K Jain and also his help in getting MCYT dataset through Julian Fierrez Aguilar, Biometric Research Lab-AVTS, Madrid, Spain when the author D.S. Guru was doing his Postdoctoral work at PRIP Lab, Michigan State University. Authors also thank Dr. Julian Fierrez Aguilar for his support.

The authors do appreciate the effort put in by reviewers in giving us critical comments and suggestions which were useful and improved substantially the quality of the presentation of the content in the paper. First and second authors acknowledge the support rendered by UGC, under faculty improvement programme and startup grant F.30-99/2015(BSR), respectively.

Appendix. List of 47 global features computed for SUSIG dataset.

Feature #	Feature description	Feature #	Feature description
1	Number of sample points (S)	25	std(Y velocity)
2	count(PenDown samples)	26	Median(X velocity)
3	count(PenUp samples)	27	Median(Y velocity)
4	$\frac{\text{count(PenDown samples)}}{\text{count(PenUp samples)}}$	28	Mode(X velocity)
5	Signature height (H): $\max(Y)-\min(Y)$	29	Mode(Y velocity)
6	Signature width (W) : $\max(X)-\min(X)$	30	corr($X - Y$ velocity)
7	Width to height ratio : W/H	31	$\frac{\text{Time of maximum } X \text{ velocity}}{\text{count(PenDown samples)}}$
8	Sample points to width ratio : S/W	32	$\frac{\text{Time of maximum } Y \text{ velocity}}{\text{count(PenDown samples)}}$
9	max(pressure)	33	Large eigen value (λ_L)
10	Sample point at max(pressure)	34	Small eigen value (λ_S)
11	Mean(pressure)	35	Total signing duration: $\sum_{i=1}^{S}\sqrt{(velocity\, x_i)^2+(velocity\, y_i)^2}$
12	var(pressure)	36	Mean(X-acceleration)
13	max(pressure) − min(pressure)	37	Mean(Y-acceleration)
14	avg(X velocity)	38	corr($X-Y$ acceleration)
15	avg(Y velocity)	39	var(X acceleration)
16	max(X velocity)	40	var(Y acceleration)
17	max(Y velocity)	41	std(X acceleration)
18	$\frac{\text{count}(S)\text{with}-\text{ve } X \text{ or } Y \text{ velocity}}{\text{count(PenDown samples)}}$	42	std(Y acceleration)
19	$\frac{\text{count}(S)\text{with}+\text{ve } X \text{ or } Y \text{ Vvelocity}}{\text{count(PenDown samples)}}$	43	Strokes count: count(PenUp samples)
20	count(S) with positive X velocity	44	count(local maxima in X-direction)
21	count(S) with positive Y velocity	45	count(local maxima in Y-direction)
22	var(X velocity)	46	$\frac{((\max(Y)-\min(Y))*(\max(X)-\min(X)))}{\Delta_x*\Delta_y}$
23	var(Y velocity)	47	$\frac{((\max(X)-\min(X))*\Delta_y)}{((\max(Y)-\min(Y))*\Delta_x)}$
24	std(X velocity)		

References

[1] J.F. Aguiliar, L. Nanni, J.L. Penalba, J.O. Garcia, D. Maltoni, An on-line signature verification system based on fusion of local and global information, in: Proceedings of the Fifth International Conference (AVBPA), 3546, 2005, pp. 523–532.

[2] J.F. Aguiliar, S. Krawczyk, J.O. Garcia, A.K. Jain, Fusion of local and regional approaches for on-line signature verification, in: Proceedings of the International Workshop on Biometric Recognition Systems (IWBRS), 3781, 2005, pp. 188–196.

[3] J.F. Aguiliar, J.O. Garcia, D. Ramos, J.G. Rodriguez, HMM-based on-line signature verification: feature extraction and signature modeling, Pattern Recognit. Lett. 28 (16) (2007) 2325–2334.

[4] R.S.A. Araujo, G.D.C. Cavalcanti, E.C.D.B.C. Filho, On-line verification for signatures of different sizes, in: Proceedings of the Tenth International Workshop on Frontiers in Handwriting Recognition (IWFHR), 2006, pp. 539–544.

[5] H. Baltzakis, N. Papamarkos, A new signature verification technique based on a two-stage neural network classifier, Eng. Appl. Artif. Intell. 14 (1) (2001) 95–103.

[6] L. Bovino, S. Impedovo, G. Pirlo, L. Sarcinella, Multi-expert verification of hand-written signatures, in: Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003), 2003, pp. 932–936.

[7] D. Cai, C. Zhang, X. He, Unsupervised feature selection for multi-cluster data, in: Proceeding of the Sixteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010, pp. 333–342.

[8] J. Coetzer, B.M. Herbst, J.A. Preez, Off-line signature verification using discrete radon transform and a hidden Markov model, EURASIP J. Appl. Signal Process. 4 (2004) 559–571.

[9] L.P. Cordella, P. Foggia, C. Sansone, F. Tortorella, M. Vento, Reliability parameters to improve combination strategies in multi-expert systems, Pattern Anal. Appl. 2 (1999) 205–214.

[10] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley-Interscience, 2000.

[11] G.S. Eskander, R. Sabourin, E. Granger, Hybrid writer-independent-writer-dependent offline signature verification system, IET 2 (12) (2013) 169–181.

[12] M.M. Fahmy, Online handwritten signature verification system based on DWT features extraction and neural network classification, Ain Shams Eng. J. 1 (2010) 59–70.

[13] K. Fakunaga, R.R. Hayes, Effects of sample size in classifier design, IEEE Trans. Pattern Anal. Mach. Des. 11 (8) (1989) 873–885.

[14] J. Galbally, J. Fierrez, M.M. Diaz, J.O. Garcia, Synthetic Generation of Online Signature Based on Spectral Analysis, 7306, SPIE, 2009.

[15] C. Gruber, T. Gruber, S. Krinninger, B. Sick, Online signature verification with support vector machines based on LCSS kernel functions, IEEE Trans. Syst. Man Cybern. Part B Cybern. 40 (4) (2010) 1088–1100.

[16] D.S. Guru, H.N. Prakash, Symbolic representation of on-line signatures, in: Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA), 2, 2007, pp. 313–317.

[17] D.S. Guru, H.N. Prakash, Online signature verification and recognition: an approach based on symbolic representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (6) (2009) 1059–1073.

[18] D.S. Guru, K.S. Manjunatha, S. Manjunath, Online signature verification based on recursive subset training, in: Proceedings of the International Conference on Mining Intelligence and Knowledge Exploration (MIKE), 8284, 2013, pp. 350–361.

[19] D.S. Guru, K.S. Manjunatha, S. Manjunath, User dependent features in online signature verification, in: Proceedings of the First International Conference on Multimedia Processing, Communication and Computing Applications (ICMCCA), 213, 2013, pp. 229–240.

[20] D. Impedovo, G. Pirlo, Automatic signature verification: the state of the art, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 38 (5) (2008) 609–635.

[21] A.K. Jain, F.D. Griess, S.D. Connell, On-line signature verification, Pattern Recognit. 35 (2002) 2963–2972.

[22] M.I. Khalil, M. Moustafa, H.M. Abbas, Enhanced DTW based on-line signature verification, in: Proceedings of the Sixteenth IEEE International Conference on Image Processing, 2009, pp. 2713–2716.

[23] A. Kholmatov, B. Yanikoglu, SUSIG: an on-line signature database, associated protocols and benchmark results, Pattern Anal. Appl. 12 (2009) 227–236.

[24] H. Lei, V. Govindaraju, A comparative study on the consistency of features in on-line signature verification, Pattern Recognit. Lett. 26 (15) (2005) 2483–2489.

[25] Y. Liu, Z. Yang, L. Yang, Online signature verification based on DCT and sparse, Representation 45 (11) (2015) 2498–2511.

[26] A. Lumini, L. Nanni, Ensemble of on-line signature matchers based on Over-Complete feature generation, Expert Syst. Appl. 36 (2009) 5291–5296.

[27] E. Maiorana, P. Campisi, D.L. Rocca, G. Scarano, Use of polynomial classifiers for on-line signature recognition, in: Proceedings of the IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS), 2012, pp. 265–270.

[28] D. Muramatsu, T. Matsumoto, Effectiveness of pen pressure, azimuth, and altitude features for online signature verification, in: Proceedings of the International Conference on Biometrics, ICB, 4642, 2007, pp. 503–512.

[29] L. Nanni, Experimental comparison of one-class classifiers for on-line signature verification, Neurocomputing 69 (7–9) (2006) 869–873.

[30] L. Nanni, A. Lumini, Ensemble of Parzen window classifiers for on-line signature verification, Neurocomputing 68 (2005) 217–224.

[31] L. Nanni, A. Lumini, Advanced methods for two-class problem formulation for on-line signature verification, Neurocomputing 69 (7–9) (2006) 854–857.

[32] M. Parodi, J.C. Gómez, Legendre polynomials based feature extraction for online signature verification. Consistency analysis of feature combinations, Pattern Recognit. 47 (1) (2014) 128–140.

[33] G. Pirlo, V. Cuccovillo, M.D. Cabrera, D. Impedovo, P. Mignone, Multidomain verification of Dynamic signatures using local stability analysis, IEEE Trans. Hum. Mach. Syst. 45 (6) (2015) 805–810.

[34] R. Plamondon, G. Lorette, Automatic signature verification and writer identification: the state of the art, Pattern Recognit. 2 (2) (1989) 107–131.

[35] S. Rashidi, A. Fallah, F. Towhidkhah, Feature extraction based DCT on dynamic signature verification, Sci. Iran. 19 (6) (2012) 1810–1819.

[36] S. Rashidi, A. Fallah, F. Towhidkhah, Authentication based on pole-zero models of signature velocity, J. Med. Signals Sens. 3 (4) (2013) 195–208.

[37] S. Rohilla, A. Sharma, R.K. Singla, Online signature verification at sub-trajectory level, Advanced Computing, Networking and Informatics-2, 28, Springer International Publishing Switzerland, 2014, pp. 369–374.

[38] B. Yanikoglu, A. Kholmatov, Online signature verification using Fourier descriptors, EURASIP J. Adv. Signal Process. 2009 (2009) 1–13.

[39] C.T. Yuen, W.L. Lim, C.S. Tan, B.M. Goi, X. Wang, J.H. Ho, Probabilistic model for dynamic signature verification system, J. Appl. Sci. Eng. Technol. 3 (2011) 1318–1322.

[40] K. Wang, Y. Wang, Z. Zhang, On-line signature verification using graph representation, in: Proceedings of the Sixth International Conference on Image and Graphics, 2011, pp. 944–947.

[41] W.S. Wijesoma, Selecting optimal personalized features for online signature verification using GA, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 4, 2000, pp. 2740–2745.

[42] K. Zhang, E. Nyssen, H. Sahli, Multi-stage online signature verification system, J. Pattern Anal. Appl. 5 (3) (2001) 288–295.

[43] J. Zou, Z. Wang, Application of HMM to online signature verification based on segment differences, in: Proceedings of the Eighth Chinese Conference, CCBR, 2013, pp. 425–432.