




A Dempster–Shafer theory based classifier combination for online Signature recognition and verification systems

Rajib Ghosh¹ · Pradeep Kumar²  · Partha Pratim Roy²

Received: 20 September 2017 / Accepted: 3 November 2018 / Published online: 11 November 2018
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract

With the advancement in technology, the society demands a robust method for person authentication. Traditional authentication methods are based on the person's knowledge such as PIN, passwords, and tokens etc. However, such methods are prone to steal and forgotten risks. Therefore, an efficient method for person identification and verification is required. In this paper, we present a novel biometric approach for online handwritten signature recognition and verification using Dempster–Shafer theory (DST). DST has been used effectively for combination of different information sources which provide incomplete, and complementary knowledge. Initially, signature identification and verification processes have been carried out using two different classifiers, namely, Hidden Markov Model (HMM) and Support Vector Machine (SVM). Next, the performance in terms of accuracy and the reliability of the system has been increased using DST by combining the probabilistic outputs of SVM and HMM classifiers. The feasibility of the approach has been tested on MCYT DB1 and SVC2004 biometric public databases for Latin script and a new online signature dataset for Devanagari script. To our knowledge there exist no dataset on online signature available in Devanagari script. Experimental results shows that the present approach is efficient in recognition and verification of signatures and outstrips existing work in this regard till date.

Keywords Online signature · Latin script · Indic script · SVM · HMM · Dempster–Shafer theory

1 Introduction

Signature is one of the popular biometric method as it represents the behavioral properties of a person instead of physical properties, such as fingerprint, iris, and face etc. Signature posses static (e.g. shape) and dynamic features (e.g. elevation and pressure) which make them unique for each person. Therefore, signatures are widely accepted by the public for person authentication and are successfully used in applications including Banking, offices, mobiles and retail industry [39]. The online mode of signature acquisition is

considered more robust in comparison to its offline mode because of the associated dynamic features like speed, pressure, elevation, and azimuth signals in addition to order of strokes [21, 31]. It is easier for skilled forgers to imitate the shape of the genuine signature, however, they fail often in producing the dynamic properties of the original one.

The signature biometric problem is usually divided into two sub-problems, namely, signature recognition and signature verification. Signature recognition systems require an individual to supply his/her signature sample that serve as a basis of their identity [23, 33]. The purpose is to identify the test signature in the database using feature-matching, whereas, in signature verification, features of the test signature are compared with the features of a set of reference signatures whose identity is claimed [13, 18]. The decision of accepting or rejecting a user is generally governed on the basis of some threshold. Guru et al. [10] have proposed an online signature recognition and verification methodology using interval-valued symbolic features. The authors have proposed parameter-based features using global analysis of signatures and feature-dependent threshold for achieving lower equal error rate. Best verification results were

✉ Pradeep Kumar
pradeep.iitr7@gmail.com

Rajib Ghosh
grajib1@gmail.com

Partha Pratim Roy
proy.fcs@iitr.ac.in

¹ Department of Computer Science and Engineering, NIT Patna, Patna, India

² Department of Computer Science and Engineering, IIT Roorkee, Roorkee, India

achieved using writer-dependent threshold when tested with a distance-based classification model.

In literature, there exist a number of fusion techniques for improving the performance of the system, including data level, feature level and decision level fusion [2, 15, 40, 46, 48, 51]. In data level, fusion information from multiple sources are combined before feature extraction. Feature level fusion combines the features extracted from multiple data sources which is used for pattern recognition, however, this approach requires features to be commensurate. Finally, the decision level fusion, combine the results processed by different classifiers using some combination techniques or rule such as Bayesian method, weighted fusion, probability fusion, and Dempster–Shafer Theory (DST). DST is considered both robust and accurate tool for classifier fusion and indecisive data. An evidence based combination is performed using this theory. This theory involves combining multiple source evidences and arriving at a degree of belief that takes into account all the available evidences. The theory is specifically effective in combining information from multiple sources involving incomplete, biased and conflict knowledge which allows one to implement this theory on the combination of various classifiers. On the other hand, other classifier fusion strategies are not evidence based. They simply rely on either fusion of weightages of results produced by multiple classifiers or fusion of probabilistic outputs provided by different classifiers, whereas combination using DST is achieved by converting the probabilistic outputs of each classifier into a mass function and calculating the conjunction of all these mass functions. Kessentini et al. [14] have used DST to improve the performance in terms of accuracy and reliability of the handwritten text recognition systems by combining the output of multiple HMM classifiers. In this paper, we present a robust and efficient signature recognition and verification system using DST by combining the decisions of two different classifiers. The main contributions of the paper are as follows.

1. Our first contribution is combining the decisions of two classifiers, namely, HMM and SVM based on the DST approach for improving the accuracy of the authentication system.
2. To the best of our knowledge, no benchmark dataset exists on online signature in Indic scripts. We have developed a dataset of online signature for Devanagari script. This dataset can be used for further research on authentication purpose in Indic scripts.¹

The rest of the paper is organized as follows. Section 2 narrates the relevant and contextual works. In Sect. 3, we

present the details of the proposed system followed by the preprocessing and feature extraction techniques. Section 4 deals with the combination of classifiers using DST for improvement in recognition and verification performance. The experiment results of recognition and verification are discussed in Sect. 5. Finally, in Sect. 6 we conclude with future possibilities of this work.

2 Related work

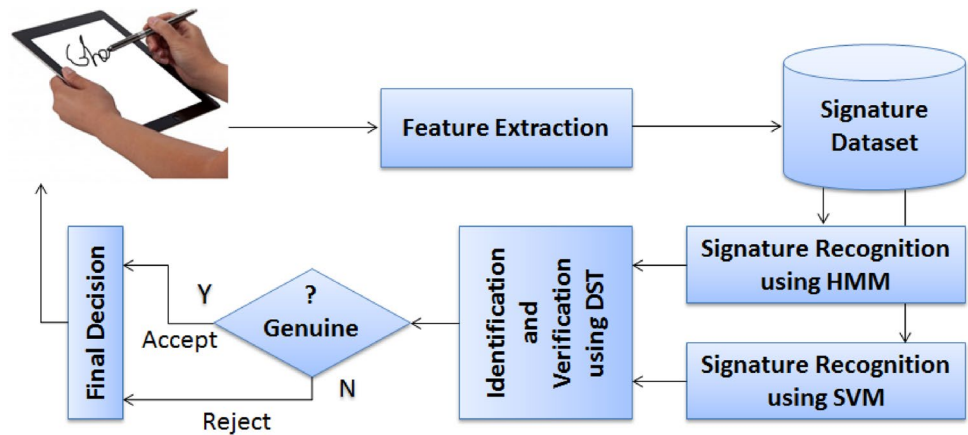
Online signature recognition and verification is a promising research area, many relevant studies are available in the literature in non-Indic scripts such as English [6], Chinese [22], etc. In this section, we discuss the works of various authors in the field of signature recognition and verification. The details are as follows.

Parizeau et al. [27] have proposed a comparative analysis for signature verification using three different algorithms, namely regional correlation, Dynamic Time Warping (DTW) [34, 37], and skeleton tree matching. The authors have used position, velocity and acceleration as features. The analysis was done on three different scripts in terms of verification error rates, execution time, and number and sensitivity of algorithm parameters where regional correlation based algorithm was observed fastest than the other two algorithms. In [20], the authors have used wavelet transform for handwritten signature verification using the back-propagation neural network. The authors extracted x and y velocity, pressure, angle and angular velocity from the signature and applied Daubechies-6 wavelet transform. The authors reported False Rejection Rate (FRR) of 0.0% and False Acceptance Rate (FAR) of 0.1% by testing in a dataset of 41 Chinese and 7 Latin script writers. In [41], the authors have performed a logarithmic spectrum analysis on signatures for developing a verification system. The mean value corresponding to the extracted coefficients using scatter matrices were used as reference templates for similarity matching. FRR and FAR of 1.4% and 2.8% were recorded based on the experiments performed on 27 registered users.

Garcia et al. [25] have proposed a signature verification system by characterizing a signature as a time function using HMM classifier. Each signature was represented by 5 time sequence based dynamic properties, namely x , y coordinate position, pressure, azimuth, and altitude. The authors have considered a signer-dependent threshold where an ERR of 0.35% has been considered. Generally function-based systems show improved performance in comparison to parametric-based systems but have high matching/comparison procedures. The authors in [8] have shown that both approaches are equally competitive. Lee et al. [19] have used dynamic programming to perform boundary segmentation of signatures using geometric extrema. The signature verification

¹ <https://sites.google.com/site/iitrcsepradeep7/>.

Fig. 1 The block diagram of the proposed framework of signature identification and verification



was performed using a back-propagation neural network by integrating global features and dynamic programming results.

Rua et al. [31] have performed the analysis and suitability of feature set, role of dynamic information order, and the importance of inclination angles and pressure in online signatures. They have considered two different HMM models for the study, namely user-adapted universal background models and user-specific HMM. The authors have reported that US-HMM system performed better if they include lowest order dynamics in the feature set, otherwise, UA-UBM performed better with likelihood ratio. In [3], the authors have analyzed the power of velocity and pressure signals in online signature for building a verification system. They have partitioned the signature trajectories to represent areas with high and low speed of signature and pen pressure. Weights have been assigned to the partitions that were later used in classification phase. Neuro-fuzzy system has been used to perform the signature verification task where it has been shown that areas with high signature velocity and low pen pressure were important. A two-stage normalization approach for DTW has been proposed in [9]. The approach detects simple forgeries in the first stage whereas deals with skilled forgers in the second stage. An average EERs of 1% and 3% have been recorded for random and skilled forgeries, respectively. Xinghua et al. [42] proposed two methods of feature extraction of each signature—one is based on full factorial experiment design and another is based on optimal orthogonal experiment design. In another recent study [43], a signature alignment method based on Gaussian Mixture Model has been proposed to obtain the best matching. Das et al. [1] has proposed a DST based method to perform natural scene labelling. Authors have performed superpixel level semantic segmentation considering three various levels as neighbors for semantic contexts. They have used the DST of uncertainty to analyze confusion among various classes. In [49], the authors have discussed guidelines for improving the generalization ability of classifiers by adjusting uncertainty

based on the problem complexity. A non-naive Bayesian classifier approach has been proposed in [47] to improve the performance of Bayesian classifiers by removing the independence assumption and including the joint probability density function. The authors in [50] have proposed a deep-learning approach to train multilayer feed-forward neural networks using restricted Boltzmann machine (RBM). The weights were not updated iteratively which results in quick learning and better generalization.

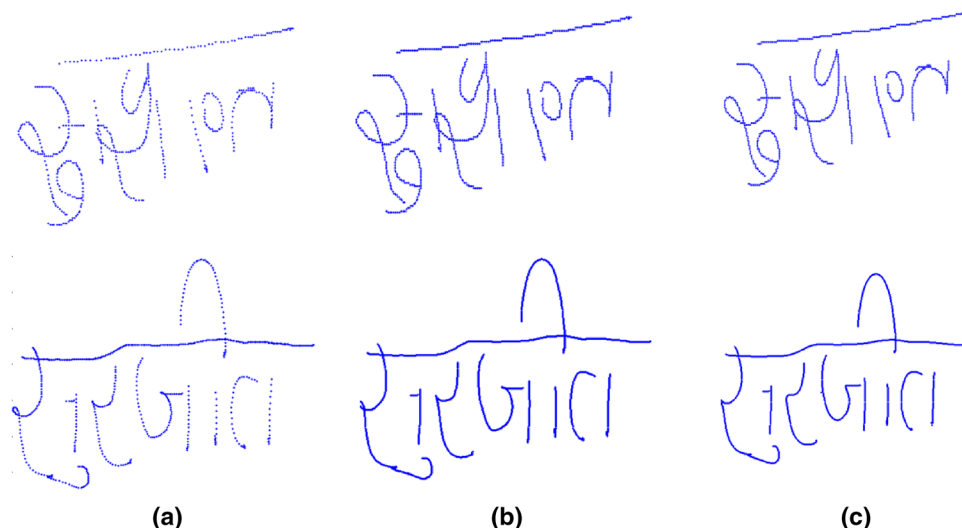
3 Proposed methodology

In our framework, the online signatures are recorded using graphic tablet like Wacom tablet touchpad device with digital stylus [11]. The recorded signatures are preprocessed using smoothing to interpolate missing points and to remove various noises such as stains, unexpected marks, etc. After preprocessing, different features are extracted from the signature trajectories which helps in better recognition. The featured data are fed into two different classifier, i.e., HMM and SVM for recognition purpose. Next, the recognition scores for each test trajectory has been computed from these two classifiers and DST has been applied for score combination. Finally, signature recognition and verification results are evaluated based on the outcome obtained from the DST. For recognition purpose, genuine signatures are used whereas for verification purpose forged signatures are used. A flow diagram of the proposed approach is shown in Fig. 1.

3.1 Preprocessing

In online handwriting recognition systems, handwritten texts are collected using various graphical tablets and light pen. Here, a sensor records the movement of pen-nib $x(i)$, $y(i)$ and switching of pen-up and pen-down. Online signature collection process provides various information like temporal data of plotted points, direction of pen-nib movement, starting

Fig. 2 Examples of preprocessing steps for two different signatures in Devanagari script (column-wise): **a** input raw signature; **b** after interpolating; **c** after smoothing the trajectory of the stroke



point, stopping point and temporal sequence of strokes are obtained. Raw online signature captured by the hardware undergo various steps of preprocessing before extraction of features. This is done to reduce negative effect during recognition process carried out later. Some of these factors involved in this regard, namely speed, accuracy, overwritten samples etc.

Preprocessing phase consists of *interpolation*, *smoothing*, *resampling*, *headline–baseline calculation* and *size normalization* [12]. More details about different preprocessing steps may be found in [12]. Figure 2 shows a Devanagari online signature image before and after applying different preprocessing steps.

3.2 Feature extraction

This phase plays an integral role in recognition and verification of signatures. In this phase, features or characteristics discriminating different signatures are extracted [35]. In the proposed system, the *Writing Direction*, *Slope*, *Curvature*, *Curliness* and *Linearity* features in varying numbers are extracted in the proximity of each plotted point of the entire signature sample [12, 28, 36, 38]. As the proposed system focuses on recognition and verification of online handwritten signatures and during capturing of online data, information such as current position, direction of pen movement and temporal information of plotted points are obtained, so we are focusing on the aforesaid features. The details are as follows.

- (a) **Writing direction:** The writing direction of a point $D(x_i, y_i)$ is computed with the help of its two immediate neighbor points on either side i.e. $C(x_{i-1}, y_{i-1})$ and $E(x_{i+1}, y_{i+1})$. These two neighbor points C and E form

\overline{CE} which creates an angle α with the x-axis. The $\cos(\alpha)$ and $\sin(\alpha)$ are used as writing direction of point D .

- (b) **Slope:** The slope of point D is calculated as $\cos(\theta_i)$, where θ_i is the angle made by the straight line connecting point A with the last proximity point (Z).
- (c) **Curvature:** The curvature of D is calculated with the help of $\sin(\beta)$ and $\cos(\beta)$, where the angle β is generated with the support of two neighbor points of D i.e. B and F .
- (d) **Curliness:** It is calculated by dividing the length of the stroke by maximum side in the vicinity of the point D .
- (e) **Linearity:** Linearity is calculated as the average square distance between the straight line connecting two end points in the proximity of D and each point in the proximity. These features are illustrated in Fig. 3. More details about these features can be found in [12].

Different combination of features is tested in our framework. For this purpose, the features extracted from strokes are first quantized into one of 8 possible values. In this scheme, if the value lies between 0° and 45° then it is quantized as bin1, if the value lies between 46° and 90° then it is quantized as bin2 and so on. The feature values are normalized by dividing the count of each bin by the total number of points in the signature sample. Finally, 8 normalized feature values are recorded for each feature. Thus, a total of $8 \times 5 = 40$ feature values are calculated for 5 different features. Other bin divisions are also tested, but the one using $\pi/4$ provides the best accuracy. The feature variation for two different samples of two different signatures in Latin script and Devanagari script are depicted in Fig. 4.

Fig. 3 Illustration of different feature extraction steps

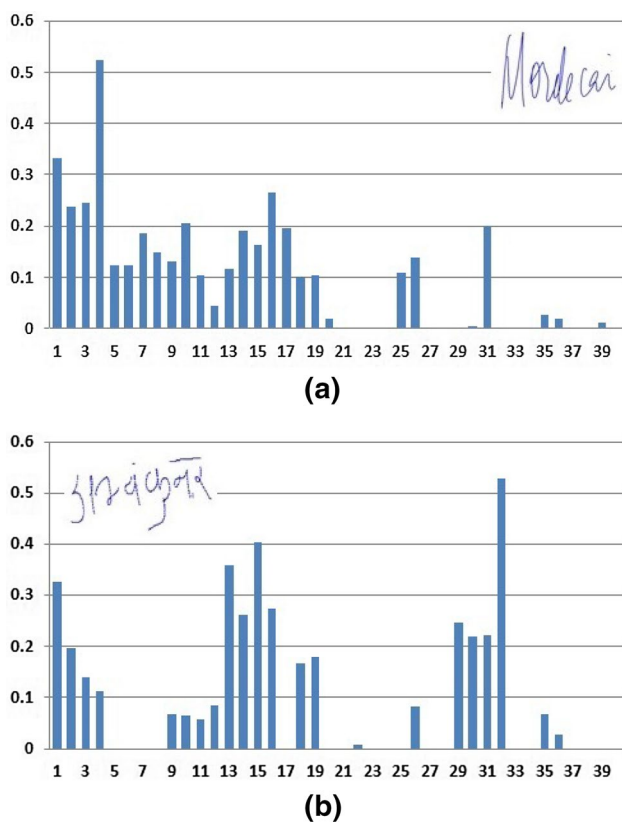
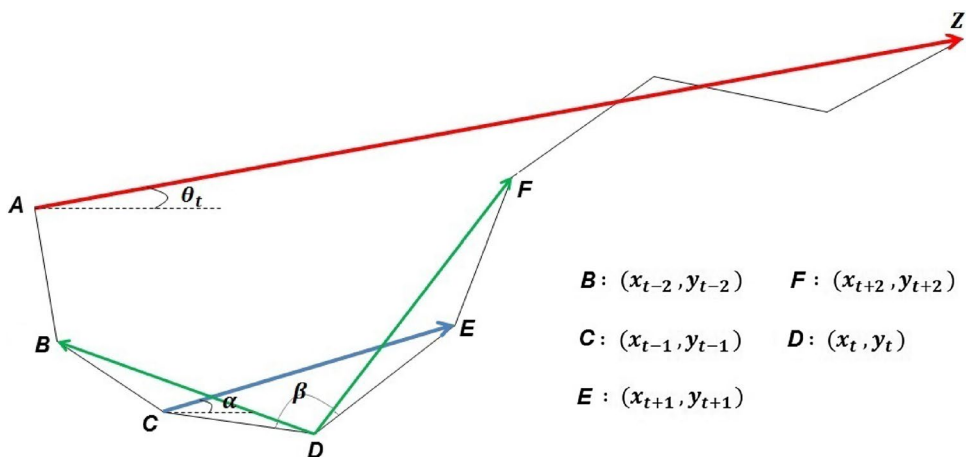


Fig. 4 Feature vectors are illustrated for two different signature samples—**a** for the signature in Latin script of one individual and **b** for the signature in Devanagari script of another individual. In this figure, X-axis represents the feature vector and Y-axis represents their values

3.3 Signature recognition and verification

Here, we present the details of the classifiers that have been used to implement our signature recognition and verification system. In our system, this process has been carried out using HMM and SVM classifiers. The details are as follows.

3.3.1 HMM based signature recognition and verification

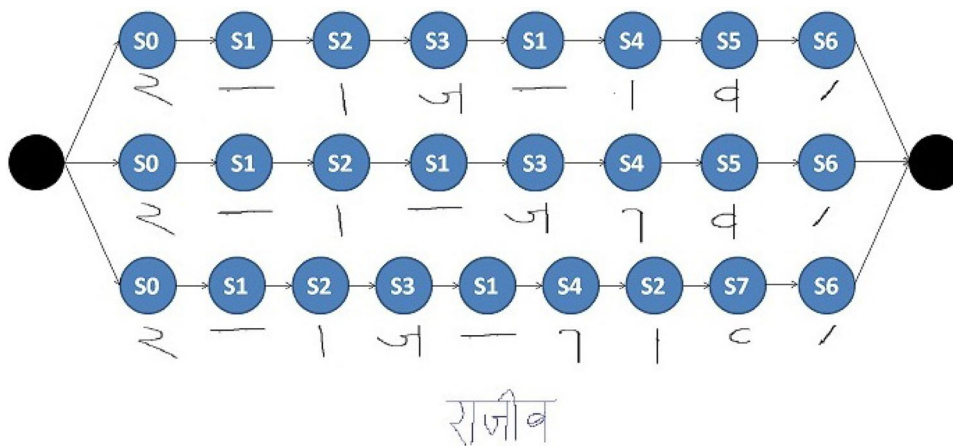
HMM is a stochastic sequential classifier that has been popular for modeling temporal sequences. HMM is defined using (π, A, B) , where π is the initial state probabilities, A is state transition matrix denoted as a_{ij} , where a_{ij} represents the transition probability between two states, e.g., from state i to state j . The term B refers the output probability matrix with density function $b_j(x)$. Here x represents the k dimensional feature vector [17, 30]. Gaussian mixture model (GMM) is defined separately for each state of model. Recognition of the sequences is performed using the Viterbi decoding algorithm. Baum–Welch algorithm is used for computing the state transition probabilities [16].

For each signature data sample, the features are extracted stroke-wise and the resultant feature sequence is processed using left-to-right HMMs. In the proposed system, each individual signature model is a HMM. For each sequence of feature vectors, the likelihood of the belongingness of the sequence to different classes is computed. The class to which the HMM achieves the highest likelihood is considered as the final class of the feature vector sequence. Figure 5 shows the formation of signature in Devanagari script with different stroke combinations. As HMM performs stochastic matching of a model and the test signature sample using an order of probability distributions of the features of the signature sample, so HMM is able to imbibe the variability and similarity between the patterns. Due to this capability of HMM, the performance of the proposed system has been evaluated using HMM also apart from SVM.

3.3.2 SVM based signature recognition and verification

SVM has found its successful usage in the field of pattern recognition and regression tasks. In general, SVM employs the principle of Structural Risk Minimization (SRM) [41] along with structural learning theory to solve a problem.

Fig. 5 Signature modeling of the Devanagari signature 'RAJIB'. The signature is written by user in three different stroke combinations



SVM was primitively delineated for two-class problems, where it searches for the hyper-planes that maximize the distance between the positive and negative data samples.

If T_D denotes the training dataset in pairs of (x_i, y_i) , $i = 1, 2, \dots, n$, where $x_i \in R^n$ and $y_i \in (-1, 1)$. The term x_i is the feature vector corresponding to the input i th sample with y_i target value. The decision function for an input pattern x is given by (1), where b, α_i and $K(x, x_i)$ are the bias, Lagrange multiplier and kernel function, respectively.

$$f(x) = \text{sign} \left(\sum_{i=1}^n y_i \alpha_i K(x, x_i) + b \right) \tag{1}$$

To carry out signature recognition and verification using SVM, instead of extracting stroke wise features, features are extracted from the whole signature sample. Unique classes are created from samples of an individual signature. Different combinations of temporal order of strokes for a signature are stored during training.

4 Proposed approach of classifier combination

In this section, we discuss the approach to combine multiple classifiers using their individual probabilities, plausibility and belief which eventually help in making final decision. In this work, we have tested this approach on two classifiers namely, SVM and HMM. However, the approach is highly scalable and can be used to combine more classifiers as well. The details are as follows.

4.1 Basics of Dempster–Shafer theory (DST)

Dempster–Shafer Theory (DST) is a framework for dealing with uncertainty. The theory involves combining evidence from different sources and arriving at a degree of belief that

takes into account all the available evidences. It is effectively used for combining multiple information sources with incomplete, imprecise, biased, and conflict knowledge. In [14, 45], the authors have shown that DST can be employed to improve the accuracy of existing stand-alone handwriting recognition system. Similarly, the strategy can be further implemented on the combination of various classifiers. For this, a combination method has been proposed that combine the probabilistic outputs of various classifiers.

A DST based approach can be illustrated as follows. If $\Omega = \{w_1, \dots, w_v\}$ be a finite set, also known as frame, formed by exclusive classes for each individual signature. A mass function, referred as μ , can be defined on the power set of Ω , represented as $P(\Omega)$, that maps onto $[0, 1]$ so that $\sum \mu(A) = 1$ where $A \subseteq \Omega$ and $\mu(\phi) = 0$. Thus, μ is approximately a probability function defined on $P(\Omega)$ in lieu of Ω . It provides a broader description as the support of the function is enhanced: If $\|\Omega\|$ is the cardinality of Ω , then $P(\Omega)$ contains $2^{\text{exp } \|\Omega\|}$ elements [14].

The belief function *bel* is defined using (2).

$$\text{bel}(A) = \sum \mu(B); \forall A \subseteq \Omega, \text{ where } B \subseteq A, B \neq \phi \tag{2}$$

bel(A) refers to the probabilistic lower bound (i.e. all evidences that imply A). Similarly, the plausibility function *pl* is defined using (3).

$$\text{pl}(A) = \sum \mu(B); \forall A \subseteq \Omega, \text{ where } B \cap A \neq \phi \tag{3}$$

It refers to the probability of all the evidences that do not contradict A. Consequently, the difference between plausibility and belief i.e. $\text{pl}(A) - \text{bel}(A)$ corresponds to the *imprecision* associated with subset A of Ω .

Two mass functions μ_1 and μ_2 based on the evidences of two independent sources can be combined into a consonant mass function using (4).

$$M(Z) = \frac{\sum_{A \cap B = Z} \mu_1(A) \times \mu_2(B)}{1 - \sum_{A \cap B = \phi} \mu_1(A) \times \mu_2(B)} \tag{4}$$

where, $Z \neq \phi$, $Z \subseteq \Omega$, and A, B denotes two different sources. Evidential combination strategy [14] aims at combining the outputs of various classifiers, being utilized, in the best possible way. For this, the steps are—(1) building the frame, (2) calculating the mass function by finding the probabilistic output for each Q , (3) computation of conjunctive combination of Q mass functions, and (4) designing a decision function with the help of pignistic transform.

4.1.1 Preparation of dynamic frames

Since each signature is labeled with a class number. Therefore, number of classes are very large in comparison to the classical DST problems. If there are V number of signatures then there will be V number of classes. The mass functions for V classes involved 2^V values and the conjunctive combination of two mass functions required 2^{2^V} multiplications and 2^V additions. Hence, the cost for computation grows exponentially with the size of the signatures set. For making an efficient system, it is necessary to either cut-down the complexity or to reduce the size of the classes involved during computation of mass functions.

4.1.2 Probability computation

For combining and decision making, we must have the individual class probability from each classifier i.e. HMM and SVM classifiers. SVM gives the probability for every class that can be directly used for combination in DST. The probability score is calibrated using Platt scaling [29] from SVM score. Whereas HMM does not provide the probability directly for every class. Instead, it gives the log likelihood for every class [14]. This log likelihood score can be converted to probability by using a sigmoid function defined in (5).

$$p_q(\omega_i) = \frac{1}{1 + e^{-\lambda(l_q(\omega_i) - \neg l_q)}}, \tag{5}$$

where λ is defined in (6).

$$\lambda = \frac{1}{\max_i |L_q(\omega_i) - \neg l_q|} \tag{6}$$

where L_q, ω_i and $\neg L_q$ are the log likelihood, set of all lexicons and the median of $L_q(\omega_i) \forall q$, respectively.

4.1.3 Mass function computation

After individual probability calculation, the probabilistic outputs of the two classifiers have been converted to a more complex mass function. The initial probability distribution p is converted into a consonant mass function μ with the help of *inverse pignistic* transform [5]. If p_i is the probability

value in a particular *frame* Ω corresponding to particular class. Initially, the elements of Ω has been ranked in decreasing order of probabilities defined in (7)

$$p(e_1) > \dots > p(e_{|\Omega|}). \tag{7}$$

Next, mass function μ is described using (8) and (9),

$$\mu(\{e_1, e_2, \dots, e_{|\Omega|}\}) = \mu(\Omega) = |\Omega| \times p(e_{|\Omega|}) \tag{8}$$

$$\forall i < |\Omega|, \mu(\{e_1, e_2, \dots, e_i\}) = i \times [p(e_i) - p(e_{i+1})] \tag{9}$$

In our framework, top four choices have been considered, hence, $|\Omega| = 4$. $\mu_1(e_i), \mu_1(e_i, e_{i+1}), \mu_1(e_i, e_{i+1}, e_{i+2}), \mu_1(e_i, e_{i+1}, e_{i+2}, e_{i+3})$ have been obtained from the resultant probability set of SVM classifier. Here, each subset of μ_1 is represented by X . Similarly, $\mu_2(e_j), \mu_2(e_j, e_{j+1}), \mu_2(e_j, e_{j+1}, e_{j+2}), \mu_2(e_j, e_{j+1}, e_{j+2}, e_{j+3})$ have been obtained from the resultant probability set of HMM classifier. Here, each subset of μ_2 is represented by Y . The above mass functions are combined using (10).

$$M(A) = \frac{\sum_{X \cap Y = A} \mu_1(X) \times \mu_2(Y)}{1 - \sum_{X \cap Y = \phi} \mu_1(X) \times \mu_2(Y)} \tag{10}$$

where, $A \neq \phi$ and $A \subseteq \Omega$. For decision making, *belief*, *plausibility* and *conflict* have been computed for each of the classes C using (11) and (12).

$$belief(C) = \sum_{A \subseteq C} M(A) \tag{11}$$

$$plausibility(C) = \sum_{A \cap C \neq \phi} M(A) \tag{12}$$

The calculated *belief* and *plausibility* have been used for further calculation of degree of conflict. Thus, the *conflict* of a class C is calculated using (13).

$$conflict(C) = plausibility(C) - belief(C) \tag{13}$$

At this point, the pignistic transform has been directly used to decide the best possible class of the signature sample to be classified. The class C with the lowest *conflict* value has been selected as the final class of the signature sample.

5 Experiment results and discussions

For our experiment of signature recognition and verification, we considered genuine and forged signatures in Devanagari and Latin scripts. The feature vectors of genuine signatures have been used to develop the training model. For validating the authenticity of online signatures, two separate sets have been created, one for genuine signatures and other for forged signatures.

Table 1 Details of the training and testing signature datasets

Script	Dataset	No. of users	Sample details	Skilled20%	Skilled80%
Devanagari	Own	100	No. of train samples	500 genuine	2000 genuine
			No. of test samples	2000 genuine, 2500 forgeries	500 genuine, 2500 forgeries
Latin	MCYT DB1	100	No. of train samples	500 genuine	2000 genuine
			No. of test samples	2000 genuine, 2500 forgeries	500 genuine, 2500 forgeries
	SVC2004 Task1	40	No. of train samples	160 genuine	640 genuine
			No. of test samples	640 genuine, 800 forgeries	160 genuine, 800 forgeries

5.1 Dataset description

The handwriting was sampled using a Wacom Bamboo Pad cth301k, touchpad with digital stylus. A total of 100 native Hindi writers were involved in the study for data collection. Each individual was prompted to provide 50 signatures, out of these, 25 were genuine and 25 were forgeries. Therefore a total of 2500 (i.e. 100×25) genuine and 2500 forged online signatures were collected. For signature recognition and verification in Latin script, two publicly available datasets, namely, MCYT biometric public database [26] and SVC2004 Task1 [44] were used. In MCYT, signature sub-corpus-100 dataset (DB1) was used that consist signatures of first 100 individuals. The dataset consists of 50 signatures from each individual, out of which, 25 are genuine and 25 are forgeries. On the other hand, in SVC2004 Task1 dataset, online signatures were collected from 40 different users, where each individual has contributed 40 signatures in all, Out of which, 20 were genuine and 20 were forgeries.

As followed in [10], each signature dataset was divided into training and test sets. The results were computed in two phases. First, the system was trained by randomly selecting 20% of genuine signatures and keeping the rest dataset for testing. Secondly, the system was trained using 80% randomly selected genuine signatures and keeping the rest for testing. Both first and second phase are denoted as Skilled20% and Skilled80%, respectively. An overview of the Skilled20% and Skilled80% is shown in Table 1. The test set consists of the remaining samples of genuine signatures and all the forgery signatures in both phases.

5.2 Signature recognition performance

Here, we present the results of the signature recognition on all the three datasets. The results have been computed using SVM and HMM classifiers for both Skilled20% and Skilled80% datasets as shown in Table 1.

5.2.1 Signature recognition using SVM

Signature recognition performance using SVM was carried out using different kernels, namely, Linear, the (Gaussian)

Table 2 Signature recognition results (in percentage) for Skilled20% category using SVM with different kernels

Script	Dataset	(Gaussian) RBF kernel	Linear kernel	Poly-nomial kernel	
Skilled20%					
Devanagari	Own	89.42	93.36	90.73	
	Latin	SVC2004 Task1	88.56	92.72	89.12
		MCYT DB1	88.21	91.06	88.61
Skilled80%					
Devanagari	Own	94.76	97.86	95.89	
	Latin	SVC2004 Task1	93.78	97.34	94.26
		MCYT DB1	93.12	96.64	93.56

Table 3 Optimal set of values of various hyper parameters in SVM

Soft-margin constant (C)	Width of (Gaussian) RBF kernel (γ)	Degree of polynomial kernel
8	2.2	2

Radial Basis Function (RBF) and Polynomial. The recognition accuracy of SVM-based system is shown in Table 2, where accuracies of 92.36%, 92.72% and 91.06% were recorded on Devanagari, SVC2004 Task1 and MCYT DB1 Skilled20% datasets, respectively. Similarly, recognition accuracies of 97.86%, 97.34% and 96.64% were recorded on Skilled80% datasets for Devanagari, SVC2004 Task1 and MCYT DB1, respectively. It can be seen from the Table 2 that the maximum recognition performance was recorded using SVM with linear kernel. In addition to this, the recognition performance was also evaluated by varying number of features for Skilled80% dataset using SVM with linear kernel. The optimal set of values of various hyper parameters in SVM is shown in Table 3. *Bayesian optimization technique* has been performed to find this optimal set. To evaluate the proposed system with varying number of features, we have applied a specific feature selection method, named 'Exhaustive Search'. This method has generated

Table 4 Signature recognition results for Skilled80% category datasets using SVM with linear kernel with varying number of features

Script	Dataset	Features	Accuracy (%)
Devanagari	Own	WD, SP, CV, CR, LR	97.86
		WD, SP, CV, LR	95.62
		WD, CV, CR, LR	85.56
Latin	SVC2004 Task1	WD, SP, CV, CR, LR	97.34
		WD, SP, CV, LR	95.86
		WD, CV, CR, LR	84.32
	MCYT DB1	WD, SP, CV, CR, LR	96.64
		WD, SP, CV, LR	94.86
		WD, CV, CR, LR	85.32

various feature subsets, the results using some of those are shown in Table 4, where maximum accuracies were recorded using all feature set combinations.

5.2.2 Signature recognition using HMM

In HMM, the models were trained with all five features defined in Sect. 3.2. Experiments were carried out by tuning the parameters e.g. number of state and number of Gaussian mixture components per state. The number of states was varied from 2 to 6 and the numbers of Gaussian mixture components were considered from 4 to 128. In the present work, online stroke-based classes are not considered, rather than features are extracted from the entire online signature sample which leads to the framing of entire signature-based classes. As a consequence, in the present study, stroke-based HMMs are not considered, rather than each individual signature model is a HMM. So, in the present work, estimating the number of states does not depend on the size of the signature. Maximum accuracies for all three datasets were recorded at 3 HMM states with 64 and 32 Gaussian mixtures components for Skilled20% and Skilled80% data categories, respectively. The recognition results are depicted in Fig. 6a, b for Skilled20% and Skilled80% datasets, respectively.

Table 5 shows the signature recognition rates for Skilled80% category for both the scripts with varying number of features. Signature recognition results based on the top 5 choices were also computed for Skilled80% category for both the scripts as shown in Fig. 7, where SVM classifier results outperform HMM in all three datasets.

5.2.3 Signature recognition using classifier combination

In this section, we present the signature recognition rates using the proposed DST approach of classifier combination as discussed in Sect. 4.1. The performance was evaluated on both Skilled20% and Skilled80% datasets by combining the results of SVM and HMM classifiers.

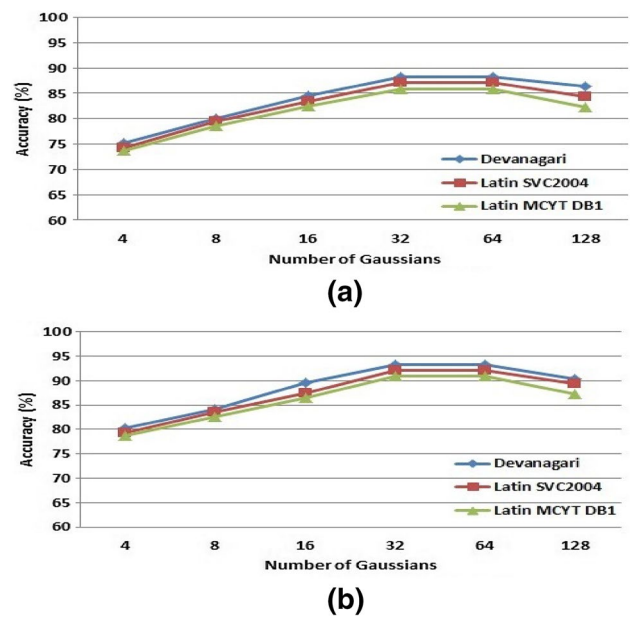


Fig. 6 Signature recognition performance using HMM with varying GMM components: a for Skilled20%; b for Skilled80% category

Table 5 Signature recognition results for Skilled80% datasets using HMM with varying number of features

Script	Dataset	Features	Accuracy (%)
Devanagari	Own	WD, SP, CV, CR, LR	93.23
		WD, SP, CV, LR	90.62
		WD, CV, CR, LR	85.78
Latin	SVC2004 Task1	WD, SP, CV, CR, LR	92.12
		WD, SP, CV, LR	89.66
		WD, CV, CR, LR	83.26
	MCYT DB1	WD, SP, CV, CR, LR	90.91
		WD, SP, CV, LR	87.56
		WD, CV, CR, LR	81.52

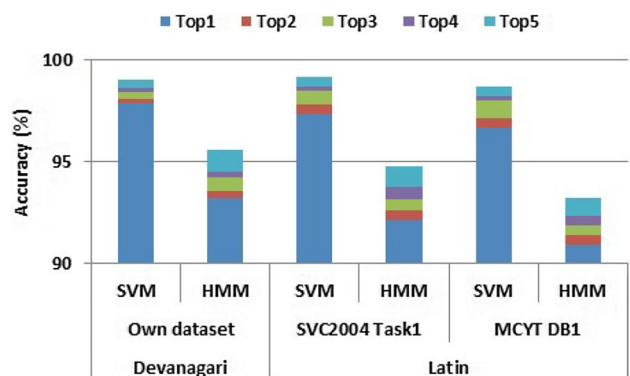


Fig. 7 Signature recognition performance using SVM and HMM for Skilled80% category for both the scripts. The results are shown considering different Top choices

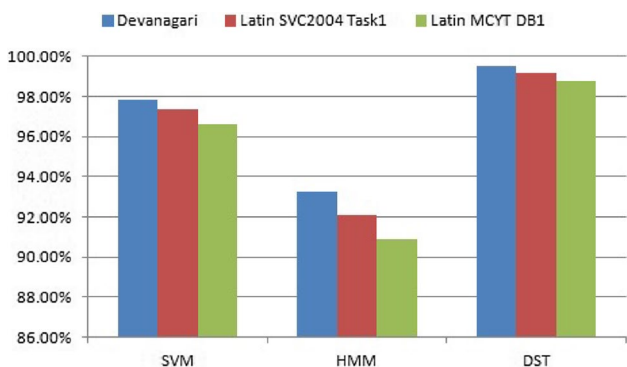
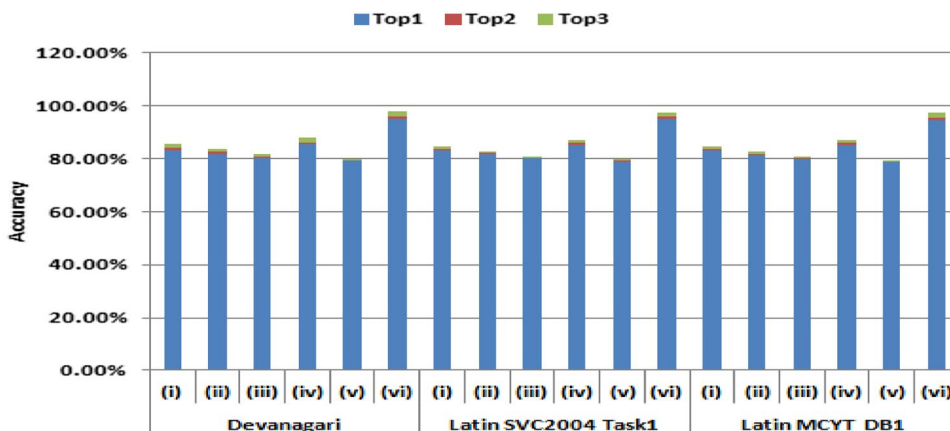


Fig. 8 Comparative performance analysis of signature recognition results using SVM, HMM and DST for Skilled80% category

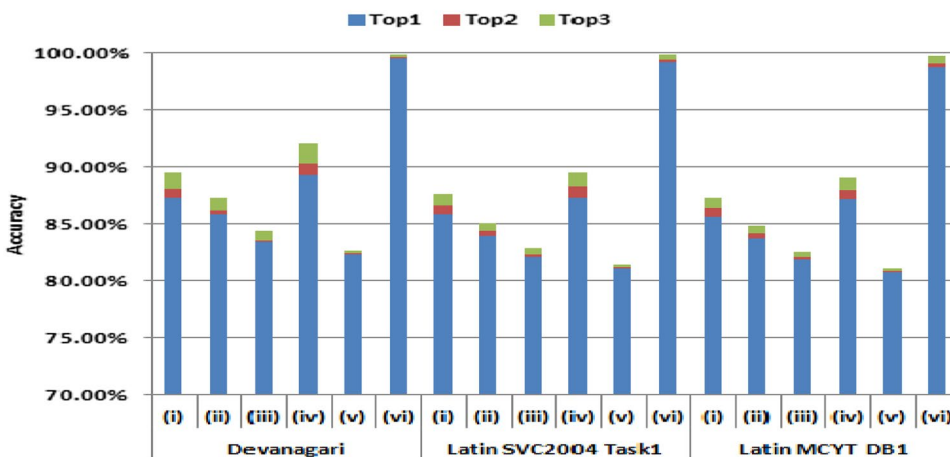
95.12%, 95.04% and 94.52% were recorded on Devanagari, SVC2004 Task1 and MCYT DB1 datasets, respectively. Similarly, Fig. 9b shows the recognition results using DST for Skilled80% dataset for the first three top choices where 99.51%, 99.17% and 98.81% accuracies were recorded on Devanagari, SVC2004 Task1 and MCYT DB1 datasets, respectively.

The DST approach based results are also compared with the existing classifier fusion strategies such as Sum rule, Product rule, Borda count rule, Simple weighted averaging and Majority voting rule. The comparison of these results are shown in Fig. 9, where the proposed DST approach outperforms the other strategies. In addition to this, we have also computed the results using DST and the afore-said classifier fusion methods by varying number of fea-

Fig. 9 Comparative signature recognition performance analysis of various classifier combination methods, namely, (i) Product, (ii) Borda Count, (iii) Sum, (iv) Simple weighted averaging, (v) Majority voting, (vi) Proposed approach (using DST), **a** for Skilled20%, **b** for Skilled80% category



(a)



(b)

Figure 8 shows the comparative performance analysis of signature recognition results using SVM, HMM and DST for Skilled80% category. Figure 9(a) shows the recognition results using DST for Skilled20% dataset when computed for the first three top choices where accuracies of

95.12%, 95.04% and 94.52% were recorded on Devanagari, SVC2004 Task1 and MCYT DB1 datasets, respectively. Similarly, Fig. 9b shows the recognition results using DST for Skilled80% dataset for the first three top choices where 99.51%, 99.17% and 98.81% accuracies were recorded on Devanagari, SVC2004 Task1 and MCYT DB1 datasets, respectively.

Table 6 Signature recognition using DST and other classifier fusion strategies for Skilled80% category with varying number of features

Script	Dataset	Strategies	Features			
			WD, SP, CV, CR, LR (%)	WD, SP, CV, LR (%)	WD, CV, CR, LR (%)	
Devanagari	Own	Product	87.33	85.48	84.32	
		Borda count	85.89	84.28	82.33	
		Sum	83.43	81.28	80.41	
		Simple weighted averaging	89.31	87.48	85.17	
		Majority voting	82.31	80.34	78.47	
		Proposed approach (using DST)	99.51	97.38	96.04	
	Latin	SVC2004 Task1	Product	85.89	84.51	82.94
			Borda count	83.94	82.18	80.54
			Sum	82.13	80.33	78.42
			Simple weighted averaging	87.34	85.58	83.37
Majority voting			81.08	79.43	77.72	
Proposed approach (using DST)			99.17	97.91	96.08	
MICYT DB1		Product	85.67	84.12	82.59	
		Borda count	83.72	81.88	80.54	
		Sum	81.89	80.13	78.06	
		Simple weighted averaging	87.18	85.51	83.91	
		Majority voting	80.74	79.94	77.54	
		Proposed approach (using DST)	98.81	97.64	95.61	

Table 7 Results of one-way ANOVA in Devanagari script

ANOVA						
Source of variation	SS	df	MS	F	P value	F crit
Between Groups	0.057212	5	0.011442	36.22347	7.78E-07	3.105875
Within Groups	0.003791	12	0.000316			
Total	0.061002	17				

SUMMARY						
Groups	Count	Sum	Average	Variance		
Product	3	2.5713	0.8571	0.00023		
Borda Count	3	2.525	0.841667	0.000318		
Sum	3	2.4512	0.817067	0.000242		
Simple weighted averaging	3	2.6196	0.8732	0.00043		
Majority voting	3	2.4112	0.803733	0.000369		
Proposed approach (using DST)	3	2.9293	0.976433	0.000306		

Statistical analysis has also been performed to validate the performance of the proposed method using DST. Correspondingly, a parametric test using one-way ANOVA has been performed to validate the performance of the proposed system. In this test, each classifier fusion strategy has been considered as one group and top three choices based on posterior probabilities have been included in each group. Results of the one-way ANOVA test are reported in Tables 7,

8 and 9 for both the scripts which has been found statistical significant.

5.3 Signature verification results

The experiment of signature verification are performed using Skilled20% and Skilled80% categories of three different datasets i.e. Devnagari, SVC2004 Task1 and MICYT DB1.

Table 8 Results of one-way ANOVA on SVC2004 Task1 dataset in Latin script

Anova: single factor						
SUMMARY						
Groups	Count	Sum	Average	Variance		
Product	3	2.5334	0.844467	0.000218		
Borda Count	3	2.4666	0.8222	0.000289		
Sum	3	2.4088	0.802933	0.000344		
Simple weighted averaging	3	2.5629	0.8543	0.000396		
Majority voting	3	2.3823	0.7941	0.000282		
Proposed approach (using DST)	3	2.9316	0.9772	0.000241		
ANOVA						
Source of variation	SS	df	MS	F	P value	F crit
Between Groups	0.067014	5	0.013403	45.41848	2.19E-07	3.105875
Within Groups	0.003541	12	0.000295			
Total	0.070555	17				

Table 9 Results of one-way ANOVA on MCVT DB1 dataset in Latin script

Anova: single factor						
SUMMARY						
Groups	Count	Sum	Average	Variance		
Product	3	2.5238	0.841267	0.000237		
Borda Count	3	2.4614	0.820467	0.000255		
Sum	3	2.4008	0.800267	0.000368		
Simple weighted averaging	3	2.566	0.855333	0.000267		
Majority voting	3	2.3822	0.794067	0.000277		
Proposed approach (using DST)	3	2.9206	0.973533	0.000262		
ANOVA						
Source of variation	SS	df	MS	F	P value	F crit
Between Groups	0.065405	5	0.013081	47.0977	1.79E-07	3.105875
Within Groups	0.003333	12	0.000278			
Total	0.068737	17				

First, the results have been computed using traditional verification approach which is followed by the proposed DST based signature verification performance.

5.3.1 Signature verification using individual classifier

The authenticity of genuine signatures is performed by testing forged signatures. A common threshold value is computed for the lowest value of Conflict (c) defined in (13). Our experimentation suggests that, for Skilled80% category datasets, this threshold value should be lesser than or equal to 0.4 to mark a test signature sample as accepted. Similarly, for Skilled20% category datasets, this threshold value should

be lesser than or equal to 0.2 to for being a test signature as accepted. Two types of errors were obtained in automatic signature verification, namely false acceptance of forged signatures [i.e. False Acceptance Rate (FAR)] and false rejections of genuine signatures [i.e. False Rejection Rate (FRR)]. FAR indicates the rate of accepting forgeries as genuine signatures whereas FRR indicates the rate of false rejections of genuine signatures. For measuring FAR, we have used forged signatures in test dataset. A trade-off between both errors usually has to be established by adjusting a decision threshold. Figure 10 shows the signature verification results in terms FAR and FRR for both SVM and HMM classifiers in both categories of dataset—Skilled20% and Skilled80%.

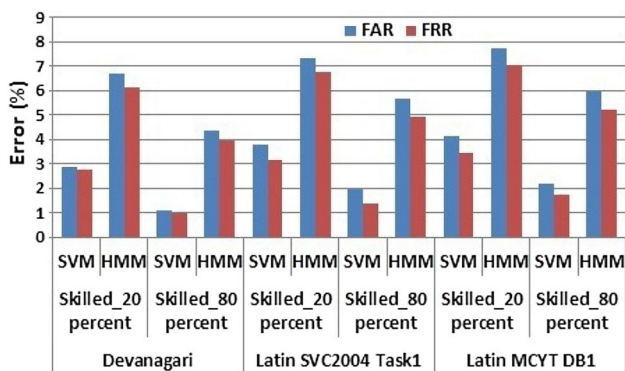


Fig. 10 Signature verification using HMM and SVM classifiers

5.3.2 Signature verification using DST approach

This section presents the results that have been computed using the proposed DST approach of classifier combination. For measuring FAR, we have used forged signatures in test dataset. Figure 11 shows the signature verification results in terms of FAR, FRR and EER after combining classifiers for both Skilled20% and Skilled80% datasets. The detailed description of the EER on all three datasets is shown in Table 10 for both Skilled20% and Skilled80% category.

To measure the performance of the proposed system when DST is applied, we use *Receiver operating characteristic (ROC) analysis*, and *F1-Score* as performance measurement parameters. ROC curve is generated by plotting the

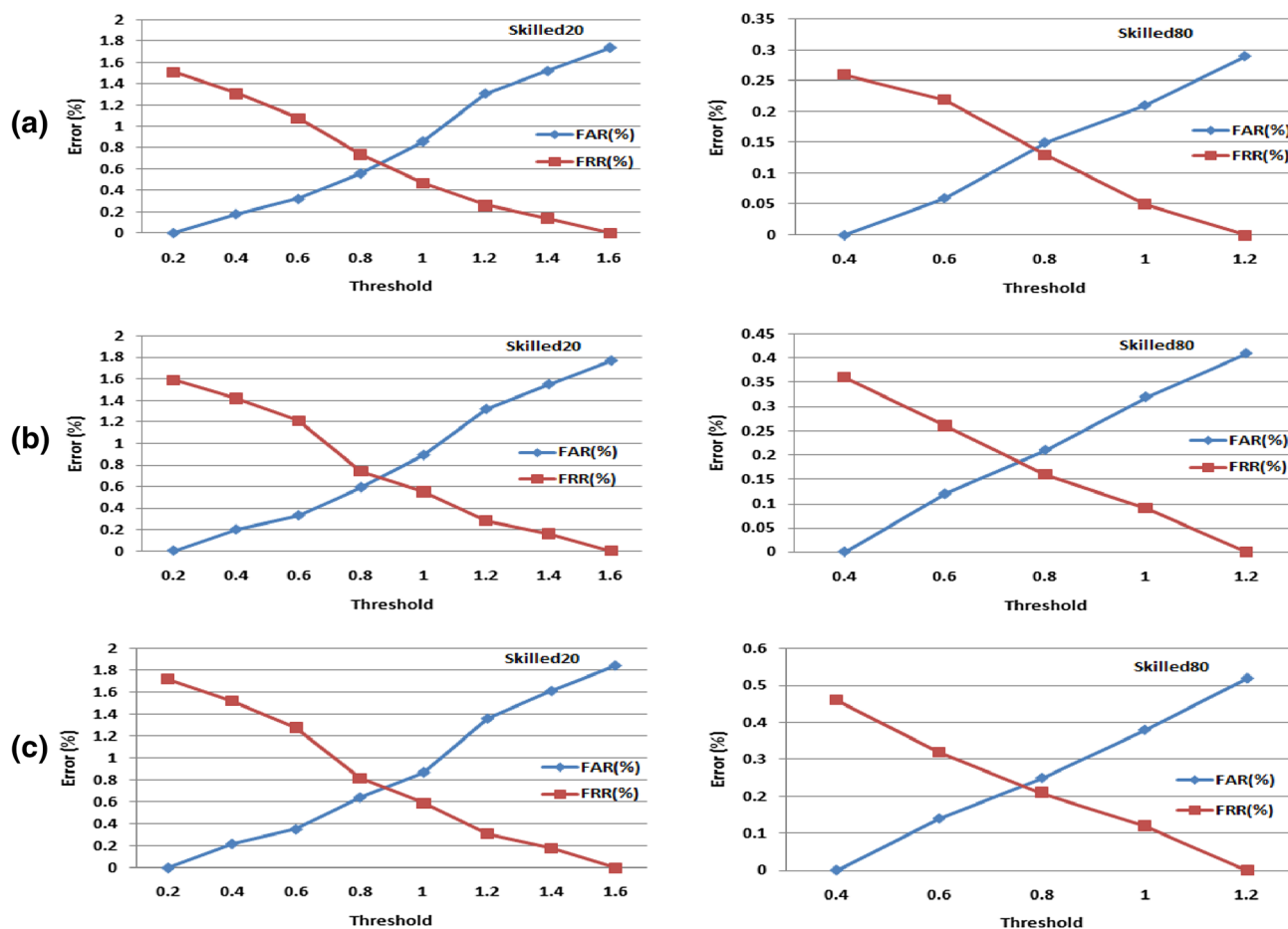


Fig. 11 Signature verification performance using DST approach for both Skilled20% and Skilled80% category (row-wise) for : a Devanagari Scrip; b SVC2004 Task1 (Latin Script); c MCYT DB1 (Latin script)

Apart from these two parameters, another measurement, called Equal Error Rate (EER) is also used to measure the overall error of the system. EER is defined as the error rate when $FRR = FAR$.

True Positive Rate (TPR) against the False Positive Rate (FPR). *F1-Score* is defined using (14). *ROC analysis*, and *F1-Score* are shown in Figs. 12 and 13 respectively.

Table 10 Signature verification performance (in percentage) using DST approach for both Skilled20% and Skilled80% categories

Script	Dataset	EER	EER
		Skilled20%	Skilled80%
Devanagari	Own	0.65	0.14
Latin	SVC2004 Task1	0.68	0.19
	MCYT DB1	0.63	0.23

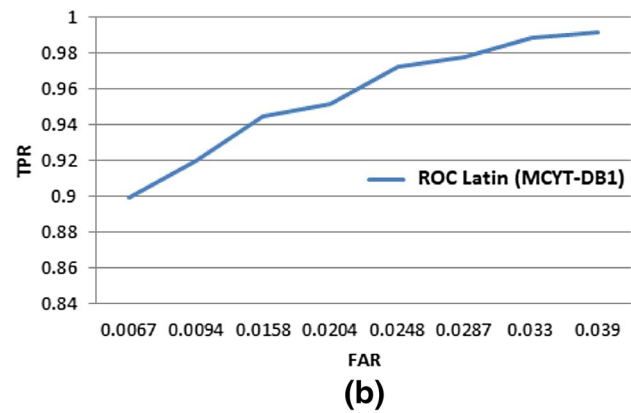
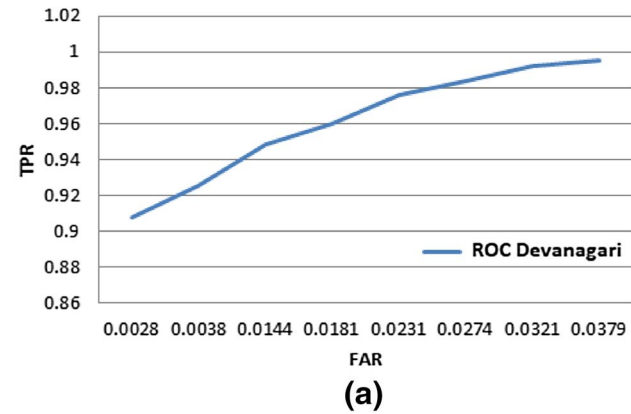


Fig. 12 ROC analysis of the proposed system for: **a** Devanagari script; **b** Latin (MCTY-DB1)

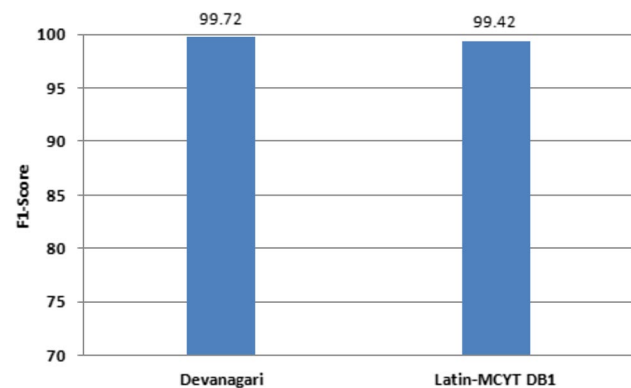


Fig. 13 F1-Score of the proposed system for Devanagari and Latin

Table 11 Comparison of signature verification results (in percentage) with existing state-of-the-art works in Latin script

References	Dataset	EER
Guru et al. [10]	MCYT DB1	5.84
Garcia et al. [26]	MCYT DB1	1.21
Nanni et al. [24]	MCYT DB1	9.7
Aguilar et al. [8]	SVC2004 Task1	2.41
Liu et al. [21]	SVC2004 Task1	3.98
Xinghua et al. [42]	MCYT DB1	2.17
	SVC2004 Task1	2.60
Xinghua et al. [43]	MCYT DB1	2.15
	SVC2004 Task1	2.63
Diaz et al. [4]	MCYT DB1	13.56
	SVC2004 Task1	18.25
Zanuy et al. [7]	MCYT DB1	4.86
	SVC2004 Task1	5.5
Dutta et al. [32]	MCYT DB1	5.66
	SVC2004 Task1	11.91
Proposed methodology	MCYT DB1	0.23
	SVC2004 Task1	0.19

$$F1\text{-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (14)$$

5.4 Comparative study

To our knowledge, there exist no benchmark verification system for online signatures in Devanagari script. For signature recognition and verification in Latin script, two publicly available datasets, namely MCYT biometric public database and SVC2004 Task1 database have been used in this work. Both datasets have been used by many researchers [8, 10, 26] to get the same platform. Therefore, in this section, a comparative analysis of signature recognition and verification have been presented with some state of the art techniques. Table 11 shows the comparative analysis performance for signature verification in terms of EER where the proposed approach performs better in comparison to other existing techniques.

6 Conclusion

In this paper, we have proposed a novel and efficient approach for online signature biometric system. The proposed recognition and verification system involves DST based classifier combination to improve the traditional stand-alone systems. The approach has been applied on two different scripts, namely, Latin and Indic script Devanagari.

DST has been used to combine the decisions of signature identification and verification made by two classifiers, i.e. HMM and SVM. The theory has been proved to be remarkably advantageous in improving the overall efficiency of the system. The robustness of the system has been tested on three different datasets where, two of them are publicly available and the third dataset collected in Indic script. From experiments, the results outperform other existing studies. The proposed approach is expected to explore a new direction of research with the help of uncertainty based models towards online signature recognition in other cursive scripts.

Compliance with ethical standards

Conflict of interest The authors have no conflict of interest regarding the publication of this paper.

References

- Aritra D, Ghosh S, Sarkhel R, Choudhuri S, Das N, Nasipuri M (2018) Combining multi-level contexts of superpixel using convolutional neural networks to perform natural scene labeling. In: Second international conference on computing and communication 2018, Sikkim Manipal Institute of Technology, Sikkim. Springer, pp 1–9
- Bouguelia MR, Nowaczyk S, Santosh K, Verikas A (2018) Agreeing to disagree: active learning with noisy labels without crowdsourcing. *Int J Mach Learn Cybern* 9(8):1307–1319
- Cpałka K, Zalasniński M, Rutkowski L (2014) New method for the on-line signature verification based on horizontal partitioning. *Pattern Recognit* 47(8):2652–2661
- Diaz M, Fischer A, Ferrer M, Plamondon R (2016) Dynamic signature verification system based on one real signature. *IEEE Trans Cybern* 99:1–12
- Dubois D, Prade H, Smets P (2001) New semantics for quantitative possibility theory. In: European conference on symbolic and quantitative approaches to reasoning and uncertainty. Springer, pp 410–421
- Emerich S, Lupu E, Rusu C (2010) On-line signature recognition approach based on wavelets and support vector machines. In: International conference on automation quality and testing robotics (AQTR), vol 3. IEEE, pp 1–4
- Faundez-Zanuy M, Pascual-Gaspar JM (2011) Efficient on-line signature recognition based on multi-section vector quantization. *Pattern Anal Appl* 14(1):37–45
- Fierrez-Aguilar J, Krawczyk S, Ortega-Garcia J, Jain AK (2005) Fusion of local and regional approaches for on-line signature verification. In: Advances in biometric person authentication. Springer, Berlin, pp 188–196
- Fischer A, Diaz M, Plamondon R, Ferrer MA (2015) Robust score normalization for DTW-based on-line signature verification. In: 13th international conference on document analysis and recognition (ICDAR), IEEE, pp 241–245
- Guru D, Prakash H (2009) Online signature verification and recognition: an approach based on symbolic representation. *IEEE Trans Pattern Anal Mach Intell* 31(6):1059–1073
- Hinckley K, Sinclair M (1999) Touch-sensing input devices. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 223–230
- Jaeger S, Manke S, Reichert J, Waibel A (2001) Online handwriting recognition: the npn++ recognizer. *Int J Doc Anal Recognit* 3(3):169–180
- Kaur B, Singh D, Roy PP: A novel framework of EEG-based user identification by analyzing music-listening behavior. *Multimedia tools and applications*, pp 1–22
- Kessentini Y, Burger T, Paquet T (2015) A dempster-shafer theory based combination of handwriting recognition systems with multiple rejection strategies. *Pattern Recognit* 48(2):534–544
- Khazae M, Ahmadi H, Omid M, Moosavian A, Khazae M (2014) Classifier fusion of vibration and acoustic signals for fault diagnosis and classification of planetary gears based on dempster-shafer evidence theory. *Proc Inst Mech Eng Part E J Process Mech Eng* 228(1):21–32
- Kumar P, Saini R, Roy PP, Dogra DP (2016) Study of text segmentation and recognition using leap motion sensor. *IEEE Sens J* 17(5):1293–1301
- Kumar P, Saini R, Roy PP, Dogra DP (2017) 3D text segmentation and recognition using leap motion. *Multimed Tools Appl* 76(15):16491–16510
- Kumar P, Saini R, Roy PP, Dogra DP (2017) A bio-signal based framework to secure mobile devices. *J Netw Comput Appl* 89:62–71
- Lee J, Yoon HS, Soh J, Chun BT, Chung YK (2004) Using geometric extrema for segment-to-segment characteristics comparison in online signature verification. *Pattern Recognit* 37(1):93–103
- Lejtman DZ, George SE (2001) On-line handwritten signature verification using wavelets and back-propagation neural networks. In: Sixth international conference on document analysis and recognition. IEEE, pp 992–996
- Liu Y, Yang Z, Yang L (2015) Online signature verification based on DCT and sparse representation. *IEEE Trans Cybern* 45(11):2498–2511
- 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 (sigcomp2011). In: International conference on document analysis and recognition (ICDAR). IEEE, pp 1480–1484
- López-García M, Ramos-Lara R, Miguel-Hurtado O, Cantó-Navarro E (2014) Embedded system for biometric online signature verification. *IEEE Trans Ind Inf* 10(1):491–501
- Nanni L, Lumini A (2005) Ensemble of parzen window classifiers for on-line signature verification. *Neurocomputing* 68:217–224
- Ortega-Garcia J, Fierrez-Aguilar J, Martin-Rello J, Gonzalez-Rodriguez J (2003) Complete signal modeling and score normalization for function-based dynamic signature verification. In: Audio-and video-based biometric person authentication. Springer, pp 1058–1058
- Ortega-Garcia J, Fierrez-Aguilar J, Simon D, Gonzalez J, Faundez-Zanuy M, Espinosa V, Satue A, Hernaez I, Igarza JJ, Vivaracho C (2003) Mcyt baseline corpus: a bimodal biometric database. *IEE Proc Vis Image Signal Proces* 150(6):395–401
- Parizeau M, Plamondon R (1990) A comparative analysis of regional correlation, dynamic time warping, and skeletal tree matching for signature verification. *IEEE Trans Pattern Anal Mach Intell* 12(7):710–717
- Parodi M, Gómez JC (2014) Legendre polynomials based feature extraction for online signature verification. Consistency analysis of feature combinations. *Pattern Recognit* 47(1):128–140
- Platt J (1999) Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In: Advances in large margin classifiers, pp 40–61
- Roy PP, Dey P, Roy S, Pal U, Kimura F (2014) A novel approach of bangla handwritten text recognition using hmm. In: 14th international conference on frontiers in handwriting recognition (ICFHR). IEEE, pp 661–666

31. Rúa EA, Castro JLA (2012) Online signature verification based on generative models. *IEEE Trans Syst Man Cybern Part B (Cybern)* 42(4):1231–1242
32. Dutta S, Saini RPK, Roy PP (2017) An efficient approach for recognition and verification of on-line signatures using PSO. In: 4th Asian conference on pattern recognition (ACPR) (Accepted)
33. Sae-Bae N, Memon N (2014) Online signature verification on mobile devices. *IEEE Trans Inf Forensics Secur* 9(6):933–947
34. Santosh K, Lamiroy B, Wendling L (2013) Dtw-radon-based shape descriptor for pattern recognition. *Int J Pattern Recognit Artif Intell* 27(03):1350,008
35. Santosh K, Nattee C, Lamiroy B (2010) Spatial similarity based stroke number and order free clustering. In: 12th international conference on frontiers in handwriting recognition. *IEEE*, pp 652–657
36. Santosh K, Nattee C, Lamiroy B (2012) Relative positioning of stroke-based clustering: a new approach to online handwritten devanagari character recognition. *Int J Image Gr* 12(02):1250,016
37. Santosh K, Wendling L (2015) Character recognition based on non-linear multi-projection profiles measure. *Front Comput Sci* 9(5):678–690
38. Sharma A, Sundaram S (2018) On the exploration of information from the DTW cost matrix for online signature verification. *IEEE Trans Cybern* 48(2):611–624
39. Tolosana R, Vera-Rodriguez R, Ortega-Garcia J, Fierrez J (2015) Preprocessing and feature selection for improved sensor interoperability in online biometric signature verification. *IEEE Access* 3:478–489
40. Vajda S, Santosh K (2016) A fast k-nearest neighbor classifier using unsupervised clustering. In: International conference on recent trends in image processing and pattern recognition. Springer, pp 185–193
41. Wu QZ, Lee SY, Jou IC (1998) On-line signature verification based on logarithmic spectrum. *Pattern Recognit* 31(12):1865–1871
42. Xinghua X, Xiaoyu S, Fangun L, Jungang Z, Zhili C, Xiaofu M (2018) Discriminative feature selection for on-line signature verification. *Pattern Recognit* 74:422–433
43. Xinghua X, Zhili C, Fangjun L, Xiaoyu S (2017) Signature alignment based on gmm for on-line signature verification. *Pattern Recognit* 65:188–196
44. Yeung D.Y, Chang H, Xiong Y, George S, Kashi R, Matsumoto T, Rigoll G (2004) Svc2004: first international signature verification competition. In: Biometric authentication. Springer, pp 16–22
45. Zadeh LA (1986) A simple view of the dempster-shafer theory of evidence and its implication for the rule of combination. *AI Mag* 7(2):85
46. Wang X, Xing HJ, Li Y, Hua Q, Dong CR, Pedrycz W (2015) A study on relationship between generalization abilities and fuzziness of base classifiers in ensemble learning. *IEEE Trans Fuzzy Syst* 23(5):1638–1654
47. Wang XZ, He YL, Wang DD (2014) Non-naive bayesian classifiers for classification problems with continuous attributes. *IEEE Trans Cybern* 44(1):21–39
48. Wang XZ, Wang R, Feng HM, Wang HC (2014) A new approach to classifier fusion based on upper integral. *IEEE Trans Cybern* 44(5):620–635
49. Wang XZ, Wang R, Xu C (2018) Discovering the relationship between generalization and uncertainty by incorporating complexity of classification. *IEEE Trans Cybern* 48(2):703–715
50. Wang XZ, Zhang T, Wang R (2017) Noniterative deep learning: incorporating restricted boltzmann machine into multilayer random weight neural networks. *IEEE Trans Syst Man Cybern Syst* 99:1–10
51. Wang R, Wang XZ, Kwong S, Xu C (2017) Incorporating diversity and informativeness in multiple-instance active learning. *IEEE Trans Fuzzy Syst* 25(6):1460–1475

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.