Array 6 (2020) 100028

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/journals/array/2590-0056/open-access-journal

Role of sub-trajectories in online signature verification

Sudhir Rohilla [a], Anuj Sharma [b,1,*], R.K. Singla [b]

[a] Department of Computer Science, Gopichand Arya Mahila College, Abohar, India
[b] Department of Computer Science and Applications, Panjab University, Chandigarh, India

ARTICLE INFO

Keywords:
Online signature verification
Static feature
Structural feature
Kinematics feature
Statistical feature

ABSTRACT

In this paper, we have provided a partitioned based technique which can increase the efficiency of an existing technique in which each partition is called as a sub-trajectory. To implement it, eighty features are extracted from signature trajectories and categorized into four feature sets as static, kinematics, structural and statistical. We have used these four feature categories and their possible combinations on two different algorithms. An important outcome is observed as the EER decreases with increase in sub-trajectories to an optimum level and behaves in reverse direction afterwards which suggests that for any matching algorithm, the present technique can further reduce the error rate to an optimum level. These experiments are performed on the benchmark database SVC 2004 TASK 2 which contains forty genuine signatures of each forty writers and forty skilled forgeries from five different writers. The experiments are discussed in detail for change in the EER with change in each subsequent sub-trajectory level for all feature sets and the results prove that the technique using sub-trajectories improving the EER by a significant average amount of 1.18 with increase in one sub-trajectory level for all the eighty features and 1.5 for the features of categories kinematics and structural when taken together.

1. Introduction

Signature is a socially worldwide accepted biometric characteristic to represent any individual and hence the problem of automating the signature verification process has a significant importance. The online mode of signature verification is more robust than the offline mode as it captures the dynamic properties of signatures in the real time which is hard to replicate by imposters. The online signature verification process has two phases: feature extraction and the verification process. In literature, the features are classified as parametric and functional. In functional approach, signatures are categorized by the features which are functional in nature, especially function of time. On the other side, the parametric features are categorized into four categories on the basis of their physical behavior: static, kinematic, structural and statistical [1]. For the verification process, many classifiers and techniques have been explored in the previous works. Some of the important techniques are Support Vector Machine (SVM), Hidden Markov Model (HMM), Dynamic Time Warping (DTW), Symbolic data matching, Principal Component Analysis (PCA) etc which are used as the verification tools and techniques. The DTW is one of the most common technique in which the matching can be done between any two time sequences by considering the minimization of root mean square distances between them [2–5]. The best example of such type of technique is Euclidean distance based matching. Few classifier has also been considered as the signature verification tool like NN (Neural Network), SVM, HMM etc. The HMM tool is a stochastic model which is based on markov process containing hidden states [6–9]. It is considered to be potential verification approach despite the fact that it has a high computational complexity [10]. The other classifier, SVM, distinguishes any two class or multi-class objects by constructing a hyper plane or a number of hyper-planes in a high dimensional space which further maximizes the distance between different class objects and minimizes the classification error within the class [11–13]. The NN classifier is also used for the verification purpose [14,15]. But it requires a large number of input data for training which is difficult to get in case of signatures [16]. In symbolic data matching, signatures are represented by symbolic feature vectors and a reference feature vector (RFV) is evaluated to find out the forgery or genuineness of a test signature [16].

Algorithm 1 FEST: feature extraction using sub-trajectories
1: **procedure** FEST
2: **for** i = 1,2,…, all the writers, *U* **do**
3: **for** j = 1,2,…, all the samples of each writer, *S* **do**

(*continued on next page*)

* Corresponding
E-mail address: anujs@pu.ac.in (A. Sharma).
[1] https://sites.google.com/site/anujsharma25/.

https://doi.org/10.1016/j.array.2020.100028
Received 19 November 2019; Received in revised form 23 April 2020; Accepted 24 April 2020
Available online 4 May 2020

(*continued*)

4: **for** k = 1,2,…, STL(Sub-trajectory level, m) **do**
5: s _Vector(X,Y,T,P) ← signature captured
6: sig(X,Y,T,P) ← sizeNormalization(s_Vector(X,Y,T,P))
7: sig(X(k),Y(k),T(k),P(k)) ← sig(X,Y,T,P)
Quantification of a signature has been done by evaluating N number of features from the raw information
8: *(X,Y,T,P) of each signature*
9: $[f_1, f_2, \ldots, f_N]$ ← sig(X(k),Y(k),T(k),P(k))
10: st _Vector(k) ← $[f_1, f_2, \ldots, f_N]$;
11: sig _Sample_Vector(j)← [st_Vector(1), st_Vector(2),… st_Vector(m)];
end for
12: sig _User_Vector(i)← [sig_Sample_Vector(1); sig_Sample_Vector(2);… sig_Sample_Vector(S)]; **end for** 13: sig _Complete_Vector ← [sig_User_Vector(1); sig_User_Vector(2);… sig_User_Vector(U)];
end procedure

Algorithm 2 GFE: General Feature Extraction without using sub-trajectories
1: **procedure** GFE
2: **for** i = 1,2,…, all the writers, *U* **do**
3: **for** j = 1,2,…, all the samples of each writer, *S* **do**
4: s _Vector(X,Y,T,P) ← signature captured
5: sig(X,Y,T,P) ← sizeNormalization(s_Vector(X,Y,T,P))
6: sig(X(k),Y(k),T(k),P(k)) ← sig(X,Y,T,P)
Quantification of a signature has been done by evaluating N number of features from the raw information
7: *(X,Y,T,P) of each signature*
8: $[f_1, f_2, \ldots, f_N]$ ← sig(X(k),Y(k),T(k),P(k))
9: sig _Sample_Vector(j) ← $[f_1, f_2, \ldots, f_N]$;
end for
10: sig _User_Vector(i)← [sig_Sample_Vector(1); sig_Sample_Vector(2);… sig_Sample_Vector(S)];
end for
11: sig _Complete_Vector← [sig_User_Vector(1); sig_User_Vector(2);… sig_User_Vector(U)];
end procedure

All these techniques are implemented to develop a better signature verification system in terms of the error rates or its performance. In this paper, we have provided a partitioned based technique which can increase the efficiency of an existing technique in which each partition is called as a sub-trajectory. To implement this technique, we have applied the algorithm FEST (Feature Extraction using Sub-Trajectories) as shown in Algorithm 1 and it is implemented along with the interval valued symbolic matching technique. Symbolic matching with sub-trajectories means that we are partitioning the signatures into a number of equal parts. Each part is considered as a sub-trajectory. This partition is done on the basis of captured data points. The sub-trajectory level two means that signature is broken into two halves, the sub-trajectory level three means that signature is broken into three equal parts and similarly, the sub-trajectory level *m* means that signature is broken into *m* equal parts. Each such part contains equal number of data points. After dividing into sub-trajectories, the symbolic matching technique has been applied on each of the sub-trajectory. Thus, we would have a feature set comprising *m* times the extracted features from the sub-trajectory level, (ST) = *m*, and in this way, we increase the total number of features each time by increasing the sub-trajectory levels. The concept of sub-trajectories can be understood with the help of Fig. 1. We have implemented the algorithm GFE (General Feature Extraction) as shown in Algorithm 1, to extract the features without using sub-trajectories. The implementation of GFE algorithm means that we are implementing any technique without implementing FEST algorithm and to compare the results of that technique while using the FEST algorithm in order to see the impact of sub-trajectories.

Moreover, in the next section of this paper, system design has been discussed to implement the proposed technique. Then feature classification has been discussed in section 3. In section 4, we have mentioned two different algorithms SDM (Symbolic Data Matching) and SDMSVM (Symbolic Data Matching using Support Vector Machine) on which the impact of FEST algorithm has been discussed. The last section concludes the paper by analysing the results.

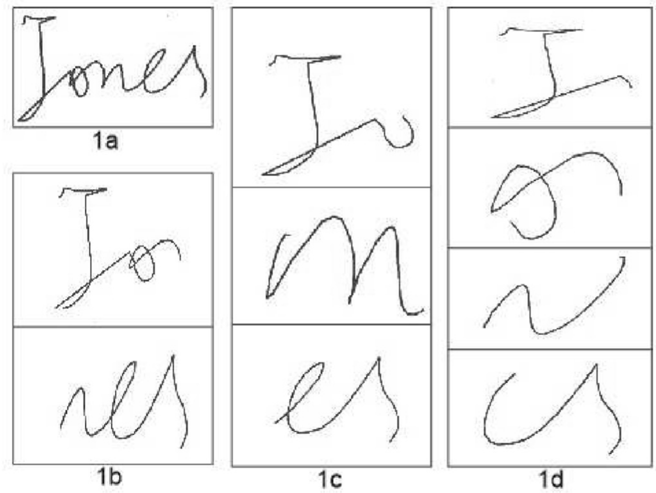

Fig. 1. 1a represents 1 partition means entire signature trajectory, 1 b shows 2 partitions, 1c shows 3 partitions and 1 d shows 4 partitions.

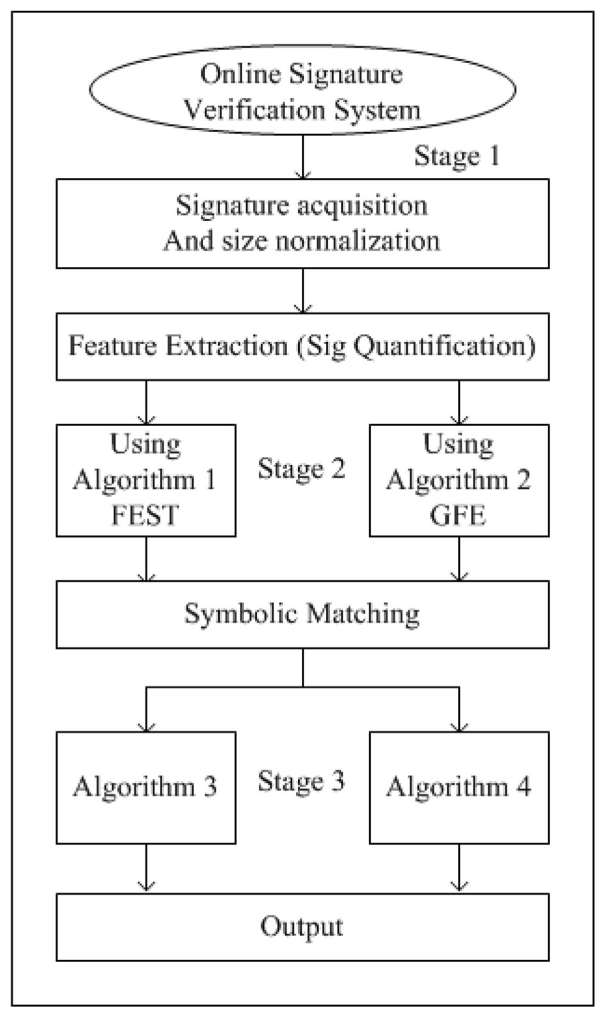

Fig. 2. System Design of signatures verification system.

2. System design

The system developed for this study contains 3 stages as shown in Fig. 2.

Stage I: The very first step in online signature verification is the data acquisition. The signature can be captured through various machines like pen tablet, tablet PCs and PDAs. As the signature is being done, we can dynamically capture its attributes like $x-y$ coordinate points, time sequence for the captured data points, pressure etc. The size normalization is then applied to the captured data points of each signature.

Stage II: The next stage is the feature extraction. The signatures are quantified by using the dynamic information captured at stage 1, some of the mathematical properties, called features, has been computed and the signatures are, then, represented in terms of these features. We have defined two ways of capturing the features: FEST (Feature Extraction using Sub-trajectories) and GFE (General Feature Extraction or without sub-trajectory) as explained in Algorithm 1 and 1 respectively.

Stage III: After evaluating the features, we have implemented the symbolic matching technique on both the feature vectors with sub-trajectories and without sub-trajectories. After that the output is stored as if the signature is authentic or genuine and termed as output 1 and 2 respectively.

3. Feature classification

The signature trajectories can be represented by some behavioral characteristics which are the numeric measure of the basic mathematical properties of each data point. These characteristics are termed as features of signature trajectories. In this paper, such type of eighty features have been described and categorized into four categories based on their behavioral nature. These categories are static, kinematics, structural and statistical [1]. We have identified 80 features from the literature. The Nelson and Kishon mentioned some formulas for the features path tangent angle, tangential and centripetal acceleration, jerk, curvature etc. mainly used in signature verification [17]. All of the used features along with their respective categories are mentioned in Table 1. These categories help us to find out the impact of each category in the process of signature verification.

4. Symbolic matching using sub-trajectory levels

4.1. Algorithm 3: symbolic data matching (SDM)

In symbolic data matching (SDM), signatures are represented by the feature vectors. Each feature vector contains the value of all the features define to characterize the signature such as a signature can be characterized by m features as $\{f_1, f_2, \ldots, f_m\}$. Then, on the basis of the mean and standard deviation of each feature in a number of sample signatures (feature vectors) of a writer, a feature level threshold is developed. This feature level threshold is used to obtain an interval valued reference feature vector (RFV) [16]. For example, the RFV of an i^{th} writer for m features is given by

$$RFV_i = \left\{ \left[f_{i1}^-, f_{i1}^+\right], \left[f_{i2}^-, f_{i2}^+\right], \ldots, \left[f_{im}^-, f_{im}^+\right] \right\}, \tag{1}$$

where the interval-valued m^{th} feature of i^{th} writer and $[f_{im}^-, f_{im}^+]$, is defined as

$$\left[f_{im}^- = \mu_{im} - \tau_{im}, f_{im}^+ = \mu_{im} + \tau_{im}\right], \tag{2}$$

where the μ_{im} is the mean of m^{th} feature of i^{th} writer's training sample and feature-level threshold, τ_{im}, is evaluated as

$$\tau_{im} = \alpha \sigma_{im}, \tag{3}$$

where the α is a scalar and σ_{im} is the standard deviation of m^{th} feature of i^{th} writer's training sample.

Each of the feature value of a test signature of a writer is compared to its corresponding interval in the reference feature vector of the same writer and is checked whether this value lies within the interval or not. If it lies within the interval, it is accountable for the genuineness of the signature otherwise for forgery of the particular signature. Then a common threshold, empirically evaluated, is applied on the total count of accountable features for a test signature. This threshold decides if the test signature is authentic or a forgery.

Table 1
A feature Set containing the eighty features under their respective categories. Here, T denotes a time interval and t specifies an instance of time.

Sr No	Static Features (1–18)	Kinematic Features (19–49)		Structural Features (50–64)	Statistical Features (65–80)
1	count(Stroke)	totalSignatureDuration	$t(Jerk_{min})/T_{pd}$	$\Theta_{initial} = tan^{-1}(V_y/V_x)$	avg(Jerk)
2	count(local x_{max})	totalPenDownTime, T_{pd}	$t(Jerk_{max})/T_{pd}$	$\Theta_{lastPenUp}$	Standard Deviation (sd) of a_x
3	count(local y_{max})	$T(V_x > 0)/T_{pd}$	$Jerk_{x,min}$	$\Theta_{lastPenUp} - \Theta_{initial}$	sd(a_y)
4	$((y_{max} - y_{min})*(x_{max} - x_{min}))/(\Delta_x*\Delta_y)$	$T(V_x < 0)/T_{pd}$	$Jerk_{x,max}$	directionChainCode, S_1	sd(v_x)
5	$((x_{max} - x_{min})*\Delta_y)/((y_{max} - y_{min})*\Delta_x)$	$T(V_y > 0)/T_{pd}$	$Jerk_{y,min}$	directionChainCode, S_2	sd(v_y)
6	$(x_{lastPenUp} - x_{max})/\Delta_x$	$T(V_y < 0)/T_{pd}$	$Jerk_{y,max}$	directionChainCode, S_3	sd(x)$/\Delta_x$
7	$(y_{lastPenUp} - y_{max})/\Delta_y$	$t(V_{x,min})/T_{pd}$	$Jerk_{min}$	directionChainCode, S_4	sd(y)$/\Delta_y$
8	$(x_{lastPenUp} - x_{min})/\Delta_x$	$t(V_{x,max})/T_{pd}$	$Jerk_{max}$	directionChainCode, S_5	avg(V)$/V_{x,max}$
9	$(y_{lastPenUp} - y_{min})/\Delta_y$	$t(V_{y,min})/T_{pd}$	$t(X_{min})/T_{pd}$	directionChainCode, S_6	avg(V)$/V_{y,max}$
10	$(x_{1stPenUp} - x_{max})/\Delta_x$	$t(V_{y,max})/T_{pd}$	$t(X_{max})/T_{pd}$	directionChainCode, S_7	avg(V)$/V_{max}$
11	$(y_{1stPenUp} - y_{max})/\Delta_y$	$t(V_{min})/T_{pd}$	$t(Y_{min})/T_{pd}$	directionChainCode, S_8	$(x_{max} - avg(x))/avg(x)$
12	$(x_{1stPenUp} - x_{min})/\Delta_x$	$t(V_{max})/T_{pd}$	$t(Y_{max})/T_{pd}$	dirChangeChainCode, C_1	$(y_{max} - avg(y))/avg(y)$
13	$(y_{1stPenUp} - y_{min})/\Delta_y$	$t(Jerk_{x,max})/T_{pd}$	$rms(Acc_{tan})/Acc_{max}$	dirChangeChainCode, C_2	avg($Jerk_x$)
14	count(V_x=0)	$t(Jerk_{x,min})/T_{pd}$	$rms(Acc_{cent})/Acc_{max}$	dirChangeChainCode, C_3	avg($Jerk_y$)
15	count(V_y=0)	$t(Jerk_{y,min})/T_{pd}$	$T((dx/dt)(dy/dt) > 0)/T((dx/dt)(dy/dt) < 0)$	dirChangeChainCode, C_4	sd(a_{tan})
16	count(V_x changes sign)	$t(Jerk_{y,max})/T_{pd}$			sd(a_{cent})
17	count(V_y changes sign)				
18	maxDistance$/((y_{max} - y_{min})*(x_{max} - x_{min}))$				

Table 2
The minimum EER for $\sum_{i=1}^{4} C_i = 15$ combinations for each category and their respective combinations with single sub-trajectory (ST1) as entire signature.

Sr No	Feature Set	MinEER		
		DS1	DS2	DS3
1	F_1	23.11	22.75	21.76
2	F_2	25.68	24.30	24.00
3	F_3	15.83	14.25	13.43
4	F_4	28.48	26.68	26.32
5	F_1F_2	19.31	26.32	17.67
6	F_1F_3	14.97	13.65	14.00
7	F_1F_4	20.16	19.82	19.17
8	F_2F_3	16.89	16.81	16.50
9	F_2F_4	24.56	22.78	22.64
10	F_3F_4	17.15	14.13	13.64
11	$F_1F_2F_3$	14.67	13.43	13.33
12	$F_1F_2F_4$	20.04	18.44	17.43
13	$F_1F_3F_4$	14.38	12.40	11.86
14	$F_2F_3F_4$	18.50	17.13	15.81
15	$F_1F_2F_3F_4$	16.47	14.54	13.89

4.2. Algorithm 4: SDM using support vector machine (SDMSVM)

In algorithm 4, the symbolic data matching has been used with SVM classifier (SDMSVM) [18]. In SDMSVM algorithm, two RFVs has been generated for forgery as well as genuine dataset which are described below:

$g.refVec = \{[min(g.f_1), max(g.f_1)], [min(g.f_2), max(g.f_2)], \ldots, [min(g.f_N), max(g.f_N)]\} f.refVec = \{[min(f.f_1), max(f.f_1)], [min(f.f_2), max(f.f_2)], \ldots, [min(f.f_N), max(f.f_N)]\}$

Then the remodeling of these RFVs has been done by using random numbers as described below:

$$g\_randInt = (max(g\_sigTr) - min(g\_sigTr))*(rand(I-1,1) + min(g\_sigTr))$$

$$f\_randInt = (max(f\_sigTr) - min(f\_sigTr))*(rand(I-1,1) + min(f\_sigTr))$$

$$g\_refVec = [min(g\_sigTr); g\_randInt; max(g\_sigTr)]$$

$$f\_refVec = [min(f\_sigTr); f\_randInt; max(f\_sigTr)]$$

In this technique, signatures are symbolized in 0 and 1 with respect to above generated RFVs and then two class SVM classifier is applied to check the genuineness of the test signatures.

5. Experimentation

5.1. Algorithm 3: SDM

The experiments are performed on the benchmark dataset, SVC TASK 2, used in first signature verification competition held in 2004. The dataset contains 20 genuine signatures of forty writers and 20 forgery signatures from five skilled forgers other than the forty writers. We have used three combination of training and testing datasets. The first dataset, DS1, contains 5 genuine signatures for training and other 15 genuine signatures for testing. Thus, testing set comprises 35 signatures (15 genuine signatures and 20 forgery signatures). In the second dataset, DS2, 10 genuine signatures are used for training and 30 signatures (10 genuine signatures other than those used in training and 20 forgery signatures) are used for testing purpose. Similarly, the third dataset, DS3, contains 15 genuine signatures for training and 25 signatures for testing (5 genuine signatures other than those used in training and 20 forgery signatures).

5.1.1. Symbolic matching without sub-trajectories: SDM with GFE

We have extracted eighty features from any signature trajectory and categorized into four categories as mentioned in Table 1. The features are evaluated as mentioned in Algorithm 1 and the verification experiments are performed on each category as well as on all the possible combinations of these categories. Thus, it gives us the fifteen experiment set which are to be performed on the three training-testing dataset as mentioned above. The results are given in Table 2.

Table 3
The minimum EER for three datasets (DS1, DS2 and DS3) is reported in the experiments performed considering $F_1F_2F_3F_4$ as a feature set with seven sub-trajectory levels (ST1, ST2, ST3, ST4, ST5, ST6 and ST7).

Sr No	Sub-trajectory Levels	MinEER DS1	MinEER DS2	MinEER DS3
1	ST1	16.47	14.54	13.89
2	ST2	13.43	12.68	12.38
3	ST3	13.00	11.28	10.86
4	ST4	12.00	10.92	10.14
5	ST5	11.57	11.96	11.10
6	ST6	12.46	12.25	12.30
7	ST7	14.57	13.66	13.41

5.1.2. Symbolic matching with sub-trajectories: SDM with FEST

Symbolic matching technique has been implemented along with the idea of sub-trajectory levels. The experiments for sub-trajectory level are conducted on all the experiment sets by taking all the combinations of mentioned feature categories and it is observed that the experiment set (F_2F_3) gives the optimum results. The results of the experiment set that contains all the features taken together $(F_1F_2F_3F_4)$ and the experiment set (F_2F_3), for which we have achieved the best EER, are reported to see the effect of sub-trajectories in the process of verification. The minimum EER at sub-trajectory level for experiment sets $(F_1F_2F_3F_4)$ and (F_2F_3) with all the three datasets (DS1, DS2 and DS3) has been mentioned in Tables 3 and 5 respecively. The change in EER with each sub-trajectory level for the experiment sets $(F_1F_2F_3F_4)$ and (F_2F_3) has also been mentioned in Tables 4 and 7. The average change in EER for the respective experiment set with each dataset is also reported in these respective Tables. Thus, the average change in EER as per sub-trajectory level for the experiment sets $(F_1F_2F_3F_4)$ and (F_2F_3) as $1.18((1.32 + 1.06 + 1.17)/3)$and $1.5((1.45 + 1.31 + 1.73)/3)$, respectively. This behavior of change in the EER for the corresponding experiment sets, $(F_1F_2F_3F_4)$ and (F_2F_3), with DS1, DS2 and DS3 are shown in Fig. 3 and Fig. 4, respectively.

5.2. Algorithm 4: SDMSVM

We have used the dataset $DS1$ to see the impact of the algorithm 4 by using the sub-trajectories. By using two class classifier, it is showing lower values of EER as compared to the Algorithm 3. The results are presented in Table 6 in which results with respect to ST1 level shows the implementation without sub-trajectory and remaining column shows the minimum EER obtained after implementing subsequent levels of sub-trajectories.

Table 4
The absolute change in the minimum EER for three datasets (DS1, DS2 and DS3) has been mentioned by considering $F_1F_2F_3F_4$ as a feature set with change in sub-trajectory levels.

Sr No	Sub-trajectory Level change	change in EER, DS1	change in EER, DS2	change in EER, DS3
1	ST2 to ST1	3.04	1.86	1.51
2	ST3 to ST2	0.43	1.40	1.52
3	ST4 to ST3	1.00	0.36	0.72
4	ST5 to ST4	0.43	1.04	0.96
5	ST6 to ST5	0.89	0.29	1.20
6	ST7 to ST6	2.11	1.41	1.11
	Average	1.32	1.06	1.17

Table 5
The minimum EER for three datasets (DS1, DS2 and DS3) is reported in the experiments performed considering F_2F_3 as a feature set with seven sub-trajectory levels (ST1, ST2, ST3, ST4, ST5, ST6 and ST7).

Sr No	Sub-trajectory Levels	MinEER DS1	MinEER DS2	MinEER DS3
1	ST1	16.89	16.81	16.50
2	ST2	13.84	13.11	12.50
3	ST3	12.10	10.93	11.25
4	ST4	11.51	10.31	10.70
5	ST5	10.60	10.56	9.47
6	ST6	12.89	10.91	11.88
7	ST7	12.76	11.65	12.82

5.3. Analysis of results

On the basis of results obtained, we have analysed the outcomes which are as follows:

5.3.1. For algorithm 3: SDM

• EER decreases with increase in sub-trajectory level, obtain an optimum value and then begins to rise with further increase in sub-trajectory level. This behavior is shown in Figs. 3 and 4 for $F_1F_2F_3F_4$ and F_2F_3 with DS1, DS2 and DS3, respectively.

• The EER decreases with increase in training data. This can be observed from results reported in Tables 2, 3 and 5.

• The results also prove that the technique using sub-trajectories changes the EER by a significant average amount of 1.18 with increase in each sub-trajectory level for all the eighty features and 1.5 for the features of categories kinematics and structural when taken together as we have achieved the best EER for this experiment set. The results are mentioned in Tables 4 and 7, respectively for the $F_1F_2F_3F_4$ and F_2F_3 feature sets with DS1, DS2 and DS3.

5.3.2. For algorithm 4: SDMSVM

• The categorization of features in this study helps us in the selection of the optimum feature set among 15 available feature set. The optimum error$=0.0008$ has been observed at sub-trajectory level 7 for F_2F_3 feature set which shows that the combination of kinematics and structural produces the best EER among other features.

• Without sub-trajectory (ST $=1$), we have obtained an EER of 3.6979 for the optimal feature set $F_1F_2F_3F_4$ as given in Table 6. But after implementing the sub-trajectories (ST $=2,3,4,5,6,7$), the performance of the system increases as EER reported is 0.0008 at $ST=7$ for F_2F_3 feature set. This reduction of EER occurs because with increase in each subsequent sub-trajectory level, there is an increase in feature values of a genuine signatures that can be compared with the forgery test signature and more the number of features values to compare, the better is the performance of the system.

• The error rates are decreased with each subsequent implementation of sub-trajectory level and it goes to an optimum level. With the use of SVM, it is observe that the behavior is kept stable after implementing $ST=7$ level for all the experimental sets as they show the same trend for each sub-trajectory level. We can see this behavior in Table 6.

• The results also prove that the technique using sub-trajectories changes the EER by a significant average amount of 1.6556188893 with increase in each sub-trajectory level for overall 15 experiments. The mean of change in EER per sub-trajectory (ST) level for the optimum feature set F_2F_3 is 1.599. The results are mentioned in Table 6. This lower value of standard deviation for optimum feature set among other experiment sets shows the consistency of the features with **Algorithm** 3.

6. Conclusion

In this way, the overall conclusion is that the categorization helps us to see how each set of features when taken together can be useful in evaluating the performance of the system rather than taking the single feature. Also by implementing sub-trajectory, we have reduced the EER to an optimal level which can be implemented with any state of the art verification technique in signature verification and hence, it can improve the EER further. This shows the effectiveness of the discussed categorization and sub-trajectories in signature verification system.

Further in comparison to previous work, the performance of this technique is quite satisfactory and we have achieved the minimum EER

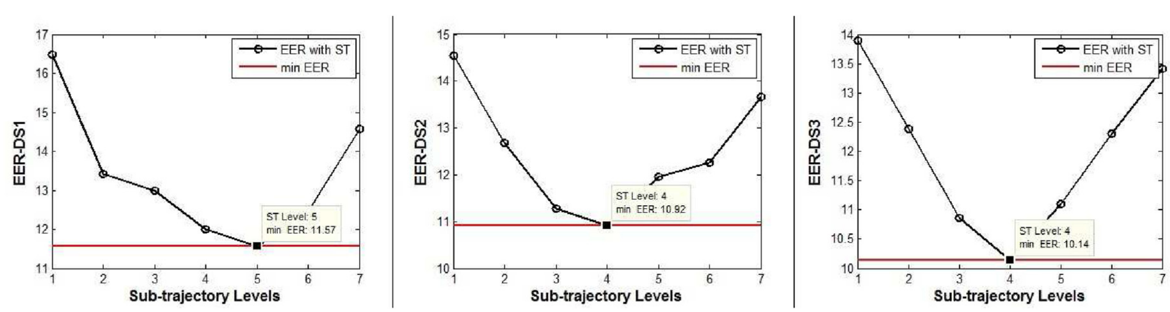

Fig. 3. EER with respect to Sub-trajectory levels for $F_1F_2F_3F_4$ with DS1, DS2 and DS3.

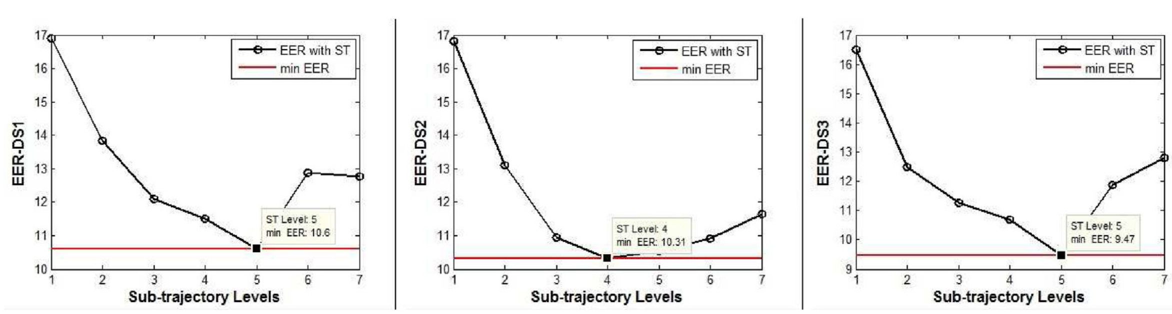

Fig. 4. EER with respect to Sub-trajectory levels for F_2F_3 with DS1, DS2 and DS3.

Table 6

Result of **Algorithm** 4: The total Error evaluated by considering $\sum_{i=1}^{4} C_i = 15$ the various combinations for F_1 (static), F_2 (kinematics), F_3 (structural) and F_4 (statistical) with sub-trajectory levels (ST = 1,2,3,4,5,6,7).

Sr No	Feature Set	Min EER for ST1	ST2	ST3	ST4	ST5	ST6	ST7	mean of change in change in EER per ST level
1	F_1	14.0475	9.7133	6.4662	4.5921	3.5042	2.6892	2.0187	2.0048
2	F_2	15.3342	6.3304	3.3596	2.0613	1.0787	0.6392	0.5408	2.465566667
3	F_3	20.3121	11.425	6.2946	3.0417	1.7558	0.5575	0.1929	3.3532
4	F_4	20.7025	12.9229	8.2621	5.2821	3.7746	2.4738	1.9267	3.1293
5	F_1F_2	7.4592	2.8092	1.2296	0.7671	0.3821	0.2229	0.1654	1.215633333
6	F_1F_3	8.6792	3.9933	1.9008	1.0408	0.51	0.3004	0.0896	1.4316
7	F_1F_4	9.2825	4.7513	2.5792	1.6013	0.9879	0.7096	0.5404	1.457016667
8	F_2F_3	9.7004	3.0058	0.945	0.3967	0.1771	0.0325	**0.0008**	1.599
9	F_2F_4	10.1738	3.4346	1.4871	0.7588	0.3379	0.2313	0.1879	1.664316667
10	F_3F_4	12.2021	5.2188	2.2096	0.8946	0.3975	0.1	0.02	2.03035
11	$F_1F_2F_3$	4.9688	1.4092	0.4125	0.2029	0.0675	0.0196	0.0038	0.8275
12	$F_1F_2F_4$	5.5079	1.6163	0.6321	0.3471	0.145	0.0925	0.0742	0.905616667
13	$F_1F_3F_4$	6.0154	2.2383	0.8667	0.4271	0.1363	0.0804	0.0146	1.000133333
14	$F_2F_3F_4$	6.7712	1.7017	0.4504	0.1617	0.0567	0.0087	0.0058	1.127566667
15	$F_1F_2F_3F_4$	3.6979	0.8987	0.2279	0.1008	0.0221	0.0071	0.00083	0.616178333
Average(Mean)									1.6556188893

Table 7

The absolute change in the minimum EER for three datasets (DS1, DS2 and DS3) has been mentioned by considering F_2F_3 as a feature set with change in sub-trajectory levels.

Sr No	Sub-trajectory Level change	change in EER, DS1	change in EER, DS2	change in EER, DS3
1	ST2 to ST1	3.05	3.70	4.00
2	ST3 to ST2	1.74	2.18	1.25
3	ST4 to ST3	0.59	0.62	0.55
4	ST5 to ST4	0.91	0.25	1.23
5	ST6 to ST5	2.29	0.35	2.41
6	ST7 to ST6	0.13	0.74	0.94
	Average	1.45	1.31	1.73

of 9.47 for DS3, 10.31 for DS2 and 10.60 for DS1 with the feature set F_2F_3 and the comparison of similar work to literature is shown in Table 8 which shows the promising results achieved with our approach. We have shown the comparison for only Algorithm 3 as it comprises the similar conditions at training phase. Algorithm 4 considers the forgery signatures at training phase which shows the contrast change in the minimum EER reported for both the algorithms. By considering all the results, we conclude that the sub-trajectory technique can be implemented effectively in online signature verification.

Table 8

Comparison with previous work for SVC 2004 TASK2.

Authors	Features used	Technique used	EER reported
[19]	Positional, pressure, path tangent angle, path velocity magnitude, log curvature radius acceleration and their first derivatives	DTW and HMM	10.91 (for skilled forgery-user independent)
[20]	Time Encoded Signal Processing and Recognition (Tespar) and Tespar DZ (set of differential descriptors)	Wavelet analysis + WLSVM (Weka libSvm)	6.96 (for skilled forgery)
[21]	Positional, pressure and velocity	velocity and pressure partition + Neuro-fuzzy classifier of the Mamdani-type	11.58
Our Study	**Static, Kinematics, Structural and Statistical**	**Symbolic interval valued based verification at sub-trajectory level** *Algorithm*3	**9.47 (for skilled forgery)**

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Sudhir Rohilla: Conceptualization, Methodology, Software, Data curation, Writing - original draft, Visualization, Investigation, Software, Validation, Writing - review & editing. **Anuj Sharma:** Conceptualization, Methodology, Software, Data curation, Writing - original draft, Visualization, Investigation, Software, Validation, Writing - review & editing. **R.K. Singla:** Supervision.

References

[1] Rohilla S, Sharma A, Singla R. Online signature verification at sub-trajectory level, Springer International Publishing, Smart Innovation. Systems and Technologies: Advanced Computing, Networking and Informatics- Volume 2014;2 28:369–74.

[2] Mariano LG, Rafael RL, Enrique CN. Embedded system for biometric online signature verification. IEEE Transaction on Industrial Informatics 2014;10(1):491–501.

[3] Fang P, Wu ZC, Shen F, Ge YJ, Fang B. Improved dtw algorithm for online signature verification based on writing forces. Proceedings of ICIC 2005:631–40.

[4] Huang K, Yan H. Stability and style variation modeling for online signature verification. Pattern Recogn 2003;36(10):2253–70.

[5] Bovino L, Impedovo S, Pirlo G, Sarcinella L. Multi expert verification of handwritten signatures. Proceeding IEEE 7th International Conference Document Analysis and Recognition; 2003. p. 932–6.

[6] Ortega-Garcia J, Fierrez-Aguilar J, Martin-Rello J, Gonzalez-Rodriguez J. Complete signal modeling and score normalization for function based dynamic signature verification. Lecture Notes in Computer Science in Audio and Video Based Biometric Person Authentication 2003;2688:658–67.

[7] Shafiei MM, Rabiee HR. A new online signature verification algorithm using variable length segmentation and hidden markov models. Seventh International Conference on Document Analysis and Recognition 2003;1:443–6.

[8] Yoon HS, Lee JY, Yang HS. An online signature verification system using hidden markov model in polar space. Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR) 2002:329–33.

[9] Coetzer J, Herbst BM, Preez JA. Offline signature verification using discrete radon transform and a hidden markov model. EURASIP J Appl Signal Process 2004;4:559–71.

[10] L. R. Rabiner, B. H. Juang, An introduction to hidden markov models, IEEE ASSp Magazine.

[11] Kholmatov A, Yanikoglu B. Identity authentication using improved online signature verification method. Pattern Recogn Lett 2005;26(15):2400–8.

[12] Lv H, Wang W, Wang C, Zhuo Q. Offline Chinese signature verification based on support vector machines. Pattern Recogn Lett 2005;26(15):2390–9.

[13] Fuentes M, Salicetti SG, Dorizzi B. Online signature verification: fusion of a hidden markov model and a neural network via a support vector machine. Eighth International Workshop on Frontiers in Handwriting Recognition 2002:253–8.
[14] Dimauro G, Impedevo G, Pirlo G. Component oriented algorithms for signature verification. Int J Pattern Recogn Artif Intell 1994;8:771–94.
[15] Draouhard JP, Sabourin R, Godbout M. A neural network approaches to on-line signature verification using directional pdf. Pattern Recogn 1996;29:415–24.
[16] Guru DS, Prakash HN. Online signature verification and recognition: an approach based on symbolic representation. IEEE Trans Pattern Anal Mach Intell 2009;31(6):1059–73.
[17] Nelson W, Kishon E. Use of dynamic features for signature verification. Proc IEEE Int Conf Syst Man Cybern 1991;1:201–5.
[18] Rohilla S, Sharma A. Svm based online signature verification technique using reference feature vector. Proc Natl Acad Sci India: Physical Sciences 2017;87(1):125–36.
[19] Fierrez-Aguilar J, Krawczyk S, Ortega-Garcia J, Jain AK. Fusion of local and regional approaches for on-line signature verification. Advances in Biometric Person Authentication, Lecture Notes in Computer Science 2005;3781:188–96.
[20] Emerich S, Lupu E, Rusu C. A new set of features for a bimodal system based on on-line signature and speech. Digit Signal Process 2013;23(3):928–40.
[21] Cpalka K, Zalasinski M, Rutkowski L. New method for the online signature verification based on horizontal partitioning. Pattern Recogn 2014;47(8):2652–61.