Enhancing Multivariate Time Series Classification Using LSTM and Evidence Feed Forward HMM

Achyut Mani Tripathi
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati
Guwahati-781039, Assam, India
Email: t.achyut@iitg.ac.in

***Abstract*—This paper presents a hybrid classifier that combines a Long Short Term Memory (LSTM) and an Evidence Feed Forward Hidden Markov Model (EFF-HMM) to classify multivariate time series (MTS). Learning of the EFF-HMM is performed based on mistakes of the LSTM. Confusion matrix obtained after classification of the MTS by the LSTM is employed during a learning process of the EFF-HMM. The EFF-HMM efficiently models a temporal characteristic and uncertainty of the MTS. The hybrid classifier combines strengths of the LSTM and EFF-HMM to enhance the accuracy of the MTS classification. The proposed method is tested on Human Activity Recognition (HAR) dataset to classify various human activities. The experiments and results show the proposed method outperforms as compared to state of the art methods.**

I. Introduction and related work

Technology advancement boosts demand for sensor-equipped monitoring applications that are capable of providing ongoing insight of any dynamical process such as an industrial process that requires continuous supervision of high volume data generated by different sensors associated with functional units. Robotics [1], Stock Market [2], Climate Forecasting [3], Human activity recognition [4] and anomaly detection [5], [6] are popular areas that require time series classification. The time series (TS) is a collection of data instances arrange according to time index. In a nutshell, the TS can be further classified into two classes first is univariate time series (UTS), and second is multivariate time series (MTS). The MTS is a collection of the multiple correlated TS that are recorded over time.

Extraction of informative features from the TS has gained considerable attention from data researchers. Various methods have been proposed to classify the TS. Bag of words (BOW) [7], Dynamic Time Wrapping (DTW) [8] and, Trend-Value pair [9] are popular features that have been successfully applied to measure similarity between the multiple TS. Classification of the UTS is less complicated as compared to classify the MTS. High speed, high volume, and dependencies among the multiple TS impose several challenges to classify the MTS.

In [10], DTW with a parametric deviation was used to perform multivariate time series classification (MTSC). In [11], author used a combination of DTW and Longest common subsequence (LCS) for MTSC. Wang et al. [12] proposed an echo state network with an evolutionary algorithm to classify the MTS. In [13], temporal abstraction in combination with an interval between the time series based method was proposed to classify MTS. Shapelet based features were used in [14] to extract compelling features for the early classification of the MTS. But all the techniques mentioned above are failed to model sequential, uncertainty, and temporal characteristics of the MTS. Hidden Markov Model (HMM) is successfully applied to learn and model the sequential characteristics of the sequential data. The TS can be considered as the sequential data with the temporal characteristics. The HMM models are well explored to classify the MTS. In [15], an imprecise HMM was proposed for the MTSC. Wang et al. [16] proposed a hybrid model by combining a GMM and the HMM for the MTSC. However, the majority of existing MTS classification techniques requires an additional feature engineering over the MTS for the MTSC.

In recent years deep learning has attracted data researchers to apply the deep learning-based methods to classify the MTS. A major advantage of the deep learning methods is that it does not require additional feature engineering to classify the MTS and capable of learning complex behavior of the MTS. In [17], a novel multi-channel based convolution neural network (CNN) was proposed to classify MTS. In [18], chin et al. proposed a novel CNN framework to classify MTS. Cui et al. [19] proposed a multi-scale CNN-based technique that automatically extracts time series features at different scales to classify MTS. In [20], author proposed a combination of LSTM and fully connected CNN to classify MTS. In [21], MTS Deep Net was proposed that extracts spatial and temporal features for MTSC. The other deep learning methods that have been well studied to classify the MTS can be found in [22], [23], [24], [25]. But the deep learning models fail to model the uncertainty of the MTS. The uncertainties of MTS can be easily modeled by probabilistic models such as the HMM [26] and an Evidence Feed Forward Hidden Markov Model (EFF-HMM) [27]. In the proposed work, a hybrid classifier is developed that combines strength of the deep learning and probabilistic technique to model the temporal characteristics and uncertainties of the MTS.

The primary contribution of this paper is to enhance the MTSC performance using Long short term memory (LTSM) [28] and the EFF-HMM [27]. The EFF-HMM learns from mistakes of the LSTM and correctly classifies the MTS.

The remainder of this paper is organized as follows: Section

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

II provides preliminaries of techniques used to develop the hybrid classifier. Section III presents a framework of the hybrid classifier. Section IV describes experiments and obtained results, and finally, conclusion and future work are presented in section V.

II. Preliminaries

This section offers basics of the LSTM, HMM and EFF-HMM.

A. *Long Short Term Memory (LSTM)*

The LSTM [28] is a special kind of recurrent neural network (RNN) that resolves an issue of vanishing gradient problem of the RNN. The LSTM is capable of handling long term dependencies, thus well applied to model the sequential and temporal characteristics of the TS. Architecture of the LSTM consists of four neural networks interacting in a special fashion. Memory part of the LSTM, also named as cell state, consists of three major gates an input gate, an output gate, and a forget gate, respectively. Fig.(1) shows the memory cell of the LSTM. The forget gate performs identification of an undesired information need to exclude from the cell state. Eq.(1) shows computation of output of the forget date in the LSTM cell.

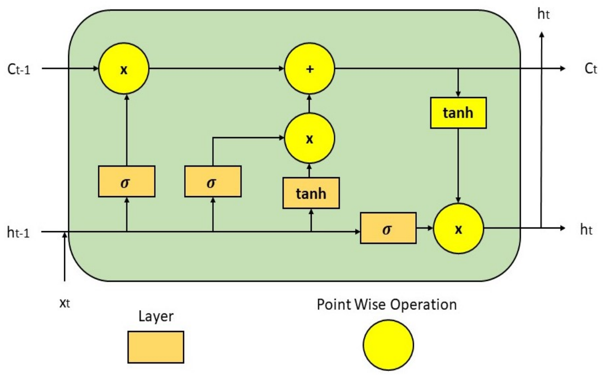

Fig. 1: Architecture of LSTM

$$f_t = \sigma(W_{f*}[x_t, h_{t-1}] + b_f) \tag{1}$$

Where f_t is current output of the forget gate, σ is a sigmoid function, W_f is weight, x_t is a current new input, h_{t-1} is output from the earlier time stamp and b_f is a bias of the forget gate.

Next step is to identify what new information required to be saved in the cell. To perform the task mentioned above the LSTM uses two layers i.e. an input gate layer and a tanh layer. Later output of the two layers are combined to store the new information in the LSTM cell. Eq.(2) shows computation of an output of the input gate layer. Eq.(3) shows computation of an output of the tanh layer.

$$I_t = \sigma(W_I * [x_t, h_{t-1}] + b_I) \tag{2}$$

Where I_t is a output of the input gate layer, W_I is a weight matrix, b_I is a bias of the input gate layer.

$$C_t^{'} = tanh(W_{c^{'}} * [h_{t-1}, x_t] + b_{c^{'}}) \tag{3}$$

Where $C_t^{'}$ is a output of the tanh layer, $W_{c^{'}}$ is a weight matrix, $b_{c^{'}}$ is a bias of the tanh layer. Combination of the output of input gate layer and tanh layer is performed using Eq.(4).

$$C_t = f_t * C_{t-1} + I_t * C_t^{'} \tag{4}$$

Where C_{t-1} is a cell output of the previous time stamp.

At last outputs of the cell i.e. O_t and h_t are calculated using Eq.(5) and Eq.(6) respectively.

$$O_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{5}$$

Here b_o is a bias of the output gate and W_o is a weight matrix of the output gate.

$$h_t = O_t * tanh(C_t) \tag{6}$$

B. *Hidden Markov Model (HMM)*

The HMM [26] is a popular probabilistic machine learning technique used to model observation sequences. The HMM is successfully applied to model the TS. The HMM efficiently models the uncertainty and temporal characteristics of the TS. HMM, parameters (θ) are learned using the Baum Welch algorithm [26], and later the Viterbi algorithm [26] is used to compute posterior probabilities of different hidden states. Fig.(2) shows the HMM model with two states S_1 and S_2 and two observations O_1 and O_2. Inference mechanism of the HMM requires three parameters $\theta = (\pi, a, b)$. Here a is matrix that denotes transition probabilities between the states, b is a matrix that denotes emission probabilities of the observations from the states and π is a matrix that denotes initial probabilities of the give states. Eq.(7) is used to perform inference using the HMM. Where t is a time stamp and T is a total time stamp.

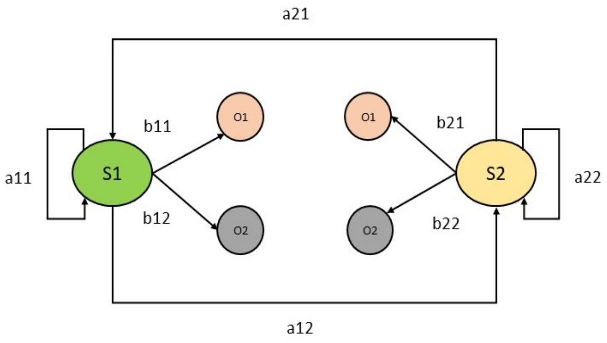

Fig. 2: Hidden Markov Model

$$P(S = s|O = o, \theta) = \pi_{s(1)} \prod_{t=1}^{T-1} a_{s(t)s(t+1)} \prod_{t=1}^{T} b_{s(t)o(t)} \tag{7}$$

C. *Evidence Feed Forward HMM (EFF-HMM)*

EFF-HMM [27] is an extension of the traditional HMM. In the HMM transition probability among the observations are not considered while the computation of the posterior probability. The EFF-HMM includes transition among the observations while computation of the posterior probability, thus enhances the classification accuracy. The EFF-HMM mainly requires learning of four parameters θ=(π, a, b, τ). Here an additional parameter τ is a matrix that denotes transition probability between the observation symbols O. Fig.(3) shows the EFF-HMM model with the two states (S_1, S_2) and two observation symbols (O_1, O_2).

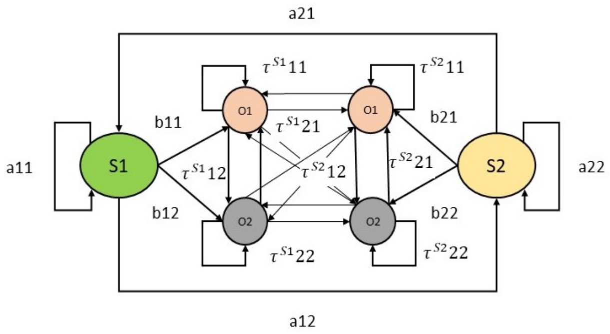

Fig. 3: Evidence Feed Forward Hidden Markov Model

1) **Initial Probability**

$$\pi_i = \text{Initial probability of being in the state i} \tag{8}$$

2) **Transition Probability**

$$a_{ij} = \frac{N(S_j|S_i)}{N(S_i)} \tag{9}$$

Where, a_{ij} is a transition probability of transition from the state S_i to S_j. $N(S_i)$= Number of times the state is S_i and $N(S_j|S_i)$=Number of times state changes from the state S_i to the state S_j

3) **Emission Probability**

$$b_{ik} = \frac{N(O_k|S_i)}{\sum_{i=1}^{N} N(O_k|S_i)} \tag{10}$$

Where, b_{ik} is an emission probability of the observation symbol k from state S_i and $N(O_k|S_i)$ is number of times observation O_k is emitted from the state S_i

4) **Observation Transition Probability**

$$\tau_{ij}^{pq} = \frac{N(O_{p(i)}^{t}, O_{q(j)}^{t+1})}{N(O_{p(i)}^{t})} \tag{11}$$

Where, $N(O_{p(i)}^{t}, O_{q(j)}^{t+1})$ is number of times O_p is emitted at time t from state S_i and O_q is emitted at time t+1 for the state S_j at time stamp t. $N(O_{p(i)}^{t})$ is number of times O_p emitted from the state S_i

Posterior probability of being in state S using the EFF-HMM is calculated using Eq.(12). The Eq.(12) has the additional term (τ) as compared to the Eq.(7). Table I shows size of the different probability tables for the EFF-HMM model.

$$P(S=s|O=o,\theta) = \pi_{s(1)} \prod_{t=1}^{T-1} a_{s(t)s(t+1)} \prod_{t=1}^{T} b_{s(t)o(t)} \prod_{t=1}^{T-1} \tau_{s(t)s(t+1)}^{o(t)o(t+1)} \tag{12}$$

The major reason behind the selection of the EFF-HMM is to explore the observation to observation linkage to recognize patterns present in a given data set. These patterns can be expressed in terms of the transition probability between the observations that further provides better classification as compared to the standard HMM.

TABLE I: Size of different probability tables in EFF-HMM

S.No.	Probability Table	Size (Rows,Columns)	

1	Π	(1,M)	M=No. of States
			N= No. of Emitted Observation Symbols

2	a	(M,M)	
3	b	(M,N)	
4	τ	(M*M),(N*N)	

III. Proposed Method

This section describes a methodology used to design the hybrid classifier for MTSC.

A. *Hybrid Classifier for MTSC*

The proposed hybrid classifier arranges the LSTM and EFF-HMM classifiers at two different layers. Here primary objective is to perform two rounds of classification using the LSTM and EFF-HMM. Initially, the MTS is classified by the LSTM, and in the second round the same MTS is reclassified by the EFF-HMM. At any time stamp misclassified MTS by the LSTM is correctly classified by the EFF-HMM.

Following are the major steps that describe working of the hybrid classifier.

1) Train the LSTM classifier using training data.
2) Classify the training data by the trained LSTM obtained from step-1.
3) Compute confusion matrix for the training data classified by the LSTM.
4) Use actual train labels to calculate the transition probability (a) of the EFF-HMM by the Eq.(9)
5) Use the confusion matrix to estimate values of the emission probability (b) and observation transition probability (τ) by the equations (10) and (11). Table II shows an example of learning the parameters of EFF-HMM for a dataset with 20 data instances. Table II shows the actual and predicted train labels of all 20 data instances. In our case, the predicted train labels are identified by the LSTM. Table III, Table IV and Table V shows the transition probability matrix, emission probability matrix and observation transition probability matrix for

TABLE II: Example of Learning of EFF-HMM

Time Stamp	Actual Label	Predicted Label
1	1	1
2	2	1
3	1	1
4	1	1
5	2	1
6	1	2
7	2	1
8	1	2
9	1	1
10	1	2
11	2	1
12	2	1
13	2	1
14	1	2
15	1	2
16	2	2
17	1	1
18	1	2
19	2	2
20	2	1

the EFF-HMM trained using the data shown in the Table II.

6) Use the trained LSTM and EFF-HMM to classify test MTS.

TABLE III: Transition Probability Table of EFF-HMM

States	1	2
1	0.45	0.55
2	0.62	0.38

TABLE IV: Emission Probability Table of EFF-HMM

States	Observation	
	1	2
1	0.35	0.65
2	0.69	0.31

TABLE V: Transition Probability Table Between Observations of EFF-HMM

State	State	1		2	
	Observation	1	2	1	2
1	1	0.33	0.67	1	0
	2	0.5	0.5	0.5	0.5
2	1	0.25	0.75	1	0
	2	1	0	1	0

Fig.(4) shows how the EFF-HMM model corrects the misclassified class label by the LSTM. Initially, the test MTS is classified by the LSTM, and in the second step, EFF-HMM reclassifies the test MTS to identify correct class label. Fig. (5) shows working procedure of the hybrid classifier.

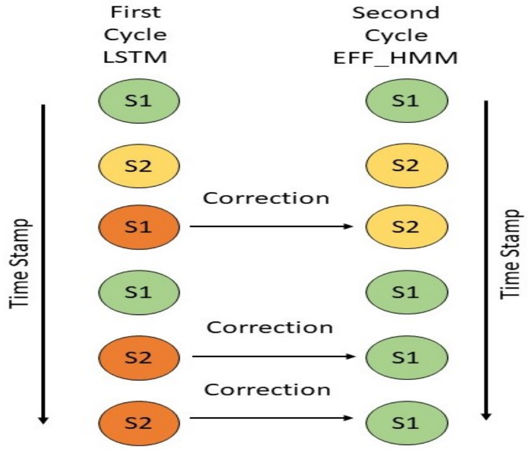

Fig. 4: Correction by Mistakes Technique

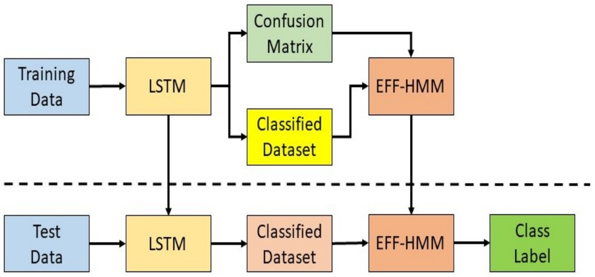

Fig. 5: Hybrid Classifier Using EFF-HMM and LSTM

IV. Experiments and Results

This section explains the experiments and results.

A. *Dataset Description*

To test the efficacy of the hybrid classifier, we selected the human activity recognition (HAR) dataset developed by Davide et al. [4]. Various sensors are used to record a movement of a person while performing numerous activities. The data set consists of recordings of a total of the 30 humans having age in between 19 to 48 years. The data set contains a total of six activities Walking (W), Walking Down (WD), Walking Up (WU), Sitting (ST), Standing (SD), and Lie Down (LD). Samsung Galaxy SII is used to record X, Y, Z data using accelerometer and gyroscope sensors with a recording frequency of 50Hz.

B. *Training Data*

Training data contains the recordings for 21 persons and has a total of 7352 data instances. The training data contains 1374 samples of the W activity, 1286 samples for the WU activity, 1407 samples for the WD activity, 1226 data instances for the ST activity, 986 data samples for the SD activity, and 1073 data instances for the LD activity. Table VI shows the number of data samples belong to the different human activities in the training and test data.

TABLE VI: Number of Training and Test Data Samples for Each Activity

S.No	Activity	Training Data)	Test Data
1	Walking (W)	1374	532
2	Walking Upstairs (WU)	1286	491
3	Walking Downstairs (WD)	1407	537
4	Sitting (ST)	1226	496
5	Standing (SD)	986	420
6	Lie Down (LD)	1073	471
	Total	**7352**	**2947**

C. Test Data

Test data consists of data gathered for 9 persons for all the six activities mentioned above. The test data contains a total of 2947 data samples. 532 data samples of the W activity, 491 data instances for the WU, 537 data instances for the WD, 496 data samples for the ST activity, 420 data samples of the SD activity, and 471 data instances for the LD activity.

D. Model Description

To show the efficiency of the proposed method, we compared the results of the hybrid classifier with three other supervised machine learning models. Total four models are developed to classify the MTS. Details of the four models are as follows:

(i) The first model is the LSTM. We train the LSTM for various mini-batch size and hidden units, and the best training accuracy is obtained for the LSTM with 200 hidden units, 100 epochs, a mini-batch size of 150, a dropout rate of 0.5, a learning rate of 0.0001, and with a RELU activation function. Table VII shows training accuracy of the LSTM with the different parameters.

TABLE VII: Training accuracy of LSTM for different mini-batch size and hidden units

No. of Hidden Units	Mini Batch Size (Training Accuracy (%))		
	50	100	150
25	94.15	93.92	92.32
50	94.56	95.69	95.67
75	95.81	95.92	96.12
100	95.95	96.06	95.29
125	95.84	95.99	95.78
150	95.72	95.79	95.34
175	95.39	95.18	95.21
200	95.67	95.63	**95.81**

(ii) The second model is a Multi class Support Vector Machine (MCSVM) with RBF kernel, and value of a parameter σ is selected as $\sigma = 0.6$. We used the same technique as used in [5] to extract <trend value> pair features from the MTS to train and test the MCSVM classifier. Windows size of 30 data samples is used to extract the features.

(iii) The third model (LSTM-HMM) is a combination of the LSTM and HMM with six hidden states .

(iv) The fourth model (LSTM-EFF-HMM) is a combination of the LSTM and EFF-HMM with six hidden states.

In the HMM and EFF-HMM number of the different emitted observations, and the number of the hidden states is same and equal to six (number of different human activities).

E. Results

Initially, the four models are trained using the training data, and later the test data is used to inspect the performance of the four models. The accuracy attains by the MCSVM classifier is 82.59%. The MCSVM classifier also shows the lowest precision rates to classify the six human activities. The LSTM classifier obtained the accuracy of 88.06% and shows the precision rate of 81.39%, 80.45%, 95.34%, 87.30%, 90.48% and 93.84% to classify the W, WU, WD, ST, SD and LD activities. The classification accuracy of the LSTM classifier is improved by 1.08% when the HMM is used in combination with the LSTM for the classification. The LSTM-HMM attains the accuracy of 89.14%. The highest accuracy 96.17% is achieved by the hybrid classifier and also shows the highest precision rates 93.98%, 93.69%, 98.14%, 96.77%, 96.90%, and 97.66% to classify the W, WU, WD, ST, SD and LD activities. Tables VIII, Table IX, Table X and Table XI shows confusion matrix for the MCSVM, LSTM, LTSM-HMM and LSTM-EFF-HMM.

An interesting observation can be seen by comparing the performance of the LSTM-HMM and the LSTM-EFF-HMM. The LSTM-EFF-HMM improves the accuracy of the LSTM by 8.11%. However, LSTM-HMM improves the same by 1.08%. It is clear from the performance of the LSTM-HMM and the LSTM-EFF-HMM that the transition probability between the observation symbols adds more useful information to the model and enhances the classification accuracy as compared to traditional HMM model that lacks the use of the observation transition probabilities during the inference.

The result of the proposed method is compared with state of the art methods to classify the numerous human activities in the HAR dataset. Table XII shows the classification accuracy and the existing methods to classify the human activities in the HAR dataset. It is clear from Table XII that the proposed method attains the second position in terms of the classification accuracy and may attains more accuracy if more inspection is performed in the feature representation step.

TABLE VIII: Confusion Matrix for MCSVM

Activity	W	WU	WD	ST	SD	LD	Precision(%)
W	**420**	102	0	0	8	2	**78.95**
WU	103	**365**	0	0	12	11	**74.34**
WD	0	0	**500**	10	12	15	**93.11**
ST	17	37	0	**403**	12	27	**81.25**
SD	30	20	5	4	**340**	21	**80.95**
LD	15	2	12	11	25	**406**	**86.20**
Recall(%)	**71.79**	**69.39**	**96.71**	**94.16**	**83.13**	**84.23**	**82.59**

TABLE IX: Confusion Matrix for LSTM

Activity	W	WU	WD	ST	SD	LD	Precision (%)
W	**433**	89	0	0	5	5	**81.39**
WU	73	**395**	0	0	0	23	**80.45**
WD	0	0	**512**	0	0	25	**95.34**
ST	2	22	0	**433**	17	22	**87.30**
SD	12	7	0	0	**380**	21	**90.48**
LD	0	2	0	8	19	**442**	**93.84**
Recall (%)	**83.27**	**76.70**	**100**	**98.19**	**90.26**	**82.16**	**88.06**

TABLE X: Confusion Matrix for LSTM-HMM

Activity	W	WU	WD	ST	SD	LD	Precision (%)
W	**437**	88	0	0	3	4	**82.14**
WU	68	**399**	0	0	0	24	**81.26**
WD	0	0	**512**	0	0	25	**95.34**
ST	4	16	4	**446**	15	11	**89.92**
SD	8	3	0	1	**387**	21	**92.14**
LD	0	2	0	3	22	**446**	**96.69**
Recall (%)	**84.53**	**78.85**	**99.22**	**99.11**	**90.63**	**83.99**	**89.14**

TABLE XI: Confusion Matrix for LSTM-EFF-HMM

Activity	W	WU	WD	ST	SD	LD	Precision (%)
W	**500**	30	0	0	0	2	**93.98**
WU	20	**460**	0	0	0	11	**93.69**
WD	0	0	**527**	0	0	10	**98.14**
ST	1	5	0	**480**	5	5	**96.77**
SD	3	0	0	0	**407**	10	**96.90**
LD	0	0	0	1	10	**460**	**97.66**
Recall (%)	**95.42**	**92.93**	**100**	**99.79**	**96.45**	**92.37**	**96.17**

TABLE XII: State of the art methods to classify HAR dataset

Paper	Dataset	Method	Accuracy(%)
[29]	UCI_HAR	Handcrafted Features+ RF	77.81
[30]	UCI_HAR	HMM	83.51
[31]	UCI_HAR	CNN+FFT	95.75
[32]	UCI_HAR	Hierarchical HMM	93.18
[33]	UCI_HAR	CNN	90.89
[34]	UCI_HAR	SVM + Stacked Auto encoder	92.16
[35]	UCI_HAR	CNN+ Local Features	97.63
-	UCI_HAR	MCSVM	82.59
-	UCI_HAR	LSTM	88.06
-	UCI_HAR	LSTM-HMM	89.14
-	UCI_HAR	**LSTM-EFFHMM**	**96.17**

V. Conclusion

The hybrid classifier is presented in this paper that combines the strength of the LSTM and EFF-HMM. The LSTM classifies the human activities using the raw data of the HAR dataset, and the EFF-HMM efficiently models the temporal assets and uncertainty of the MTS. The proposed models shows the better classification accuracy as compared to the MCSVM, LSTM-HMM, and, LSTM. In future, we would like to extend the method to classify anomalies in the multivariate time series data.

References

[1] C. Yang, C. Zeng, Y. Cong, N. Wang, and M. Wang, "A learning framework of adaptive manipulative skills from human to robot," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1153–1161, 2018.

[2] D. G. Torres and H. Qiu, "Applying recurrent neural networks for multivariate time series forecasting of volatile financial data," 2018.

[3] A. Eymen and Ü. Köylü, "Seasonal trend analysis and arima modeling of relative humidity and wind speed time series around yamula dam," *Meteorology and Atmospheric Physics*, pp. 1–12, 2018.

[4] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones." in *Esann*, 2013.

[5] A. M. Tripathi and R. D. Baruah, "Anomaly detection in multivariate time series using fuzzy adaboost and dynamic naive bayesian classifier," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2019, pp. 1938–1944.

[6] ——, "Anomaly detection in data streams based on graph coloring density coefficients," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2016, pp. 1–7.

[7] J. Wang, P. Liu, M. F. She, S. Nahavandi, and A. Kouzani, "Bag-of-words representation for biomedical time series classification," *Biomedical Signal Processing and Control*, vol. 8, no. 6, pp. 634–644, 2013.

[8] R. J. Kate, "Using dynamic time warping distances as features for improved time series classification," *Data Mining and Knowledge Discovery*, vol. 30, no. 2, pp. 283–312, 2016.

[9] B. Esmael, A. Arnaout, R. K. Fruhwirth, and G. Thonhauser, "Multivariate time series classification by combining trend-based and value-based approximations," in *International Conference on Computational Science and Its Applications*. Springer, 2012, pp. 392–403.

[10] T. Górecki and M. Łuczak, "Multivariate time series classification with parametric derivative dynamic time warping," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2305–2312, 2015.

[11] T. Górecki, "Classification of time series using combination of dtw and lcss dissimilarity measures," *Communications in Statistics-Simulation and Computation*, vol. 47, no. 1, pp. 263–276, 2018.

[12] L. Wang, Z. Wang, and S. Liu, "An effective multivariate time series classification approach using echo state network and adaptive differential evolution algorithm," *Expert Systems with Applications*, vol. 43, pp. 237–249, 2016.

[13] R. Moskovitch and Y. Shahar, "Classification of multivariate time series via temporal abstraction and time intervals mining," *Knowledge and Information Systems*, vol. 45, no. 1, pp. 35–74, 2015.

[14] G. He, Y. Duan, R. Peng, X. Jing, T. Qian, and L. Wang, "Early classification on multivariate time series," *Neurocomputing*, vol. 149, pp. 777–787, 2015.

[15] A. Antonucci, R. De Rosa, A. Giusti, and F. Cuzzolin, "Robust classification of multivariate time series by imprecise hidden markov models," *International Journal of Approximate Reasoning*, vol. 56, pp. 249–263, 2015.

[16] M. Wang, S. Abdelfattah, N. Moustafa, and J. Hu, "Deep gaussian mixture-hidden markov model for classification of eeg signals," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 4, pp. 278–287, 2018.

[17] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Exploiting multi-channels deep convolutional neural networks for multivariate time series classification," *Frontiers of Computer Science*, vol. 10, no. 1, pp. 96–112, 2016.

[18] C.-L. Liu, W.-H. Hsaio, and Y.-C. Tu, "Time series classification with multivariate convolutional neural network," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 6, pp. 4788–4797, 2018.

[19] Z. Cui, W. Chen, and Y. Chen, "Multi-scale convolutional neural networks for time series classification," *arXiv preprint arXiv:1603.06995*, 2016.

[20] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate lstm-fcns for time series classification," *Neural Networks*, vol. 116, pp. 237–245, 2019.

[21] X. Wang, Y. L. Murphey, and D. S. Kochhar, “Mts-deepnet for lane change prediction,” in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 4571–4578.

[22] S. Karimi-Bidhendi, F. Munshi, and A. Munshi, “Scalable classification of univariate and multivariate time series,” in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 1598–1605.

[23] Y. Yuan, G. Xun, F. Ma, Y. Wang, N. Du, K. Jia, L. Su, and A. Zhang, “Muvan: A multi-view attention network for multivariate temporal data,” in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 717–726.

[24] S. D. Bhattacharjee, W. J. Tolone, M. Elshambakey, I. Cho, A. Mahabal, and G. Djorgovski, “Context-aware deep sequence learning with multi-view factor pooling for time series classification,” in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 959–966.

[25] S. Imani, F. Madrid, W. Ding, S. Crouter, and E. Keogh, “Matrix profile xiii: Time series snippets: A new primitive for time series data mining,” in *2018 IEEE International Conference on Big Knowledge (ICBK)*. IEEE, 2018, pp. 382–389.

[26] L. R. Rabiner and B.-H. Juang, “An introduction to hidden markov models,” *ieee assp magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[27] M. DelRose, C. Wagner, and P. Frederick, “Evidence feed forward hidden markov model: A new type of hidden markov model,” *arXiv preprint arXiv:1102.0899*, 2011.

[28] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] B. Kolosnjaji and C. Eckert, “Neural network-based user-independent physical activity recognition for mobile devices,” in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2015, pp. 378–386.

[30] Y.-J. Kim, B.-N. Kang, and D. Kim, “Hidden markov model ensemble for activity recognition using tri-axis accelerometer,” in *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2015, pp. 3036–3041.

[31] C. A. Ronao and S.-B. Cho, “Human activity recognition with smartphone sensors using deep learning neural networks,” *Expert systems with applications*, vol. 59, pp. 235–244, 2016.

[32] ——, “Recognizing human activities from smartphone sensors using hierarchical continuous hidden markov models,” *International Journal of Distributed Sensor Networks*, vol. 13, no. 1, p. 1550147716683687, 2017.

[33] C. A. Ronaoo and S.-B. Cho, “Evaluation of deep convolutional neural network architectures for human activity recognition with smartphone sensors,” , pp. 858–860, 2015.

[34] Y. Li, D. Shi, B. Ding, and D. Liu, “Unsupervised feature learning for human activity recognition using smartphone sensors,” in *Mining intelligence and knowledge exploration*. Springer, 2014, pp. 99–107.

[35] A. Ignatov, “Real-time human activity recognition from accelerometer data using convolutional neural networks,” *Applied Soft Computing*, vol. 62, pp. 915–922, 2018.