Multivariate Time Series Classification With An Attention-Based Multivariate Convolutional Neural Network

Achyut Mani Tripathi, Rashmi Dutta Baruah
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati
Guwahati-781039, Assam, India
Email: {t.achyut, r.duttabaruah}@iitg.ac.in

***Abstract*—Classification of time series is an essential requirement of various applications that demand continuous monitoring of dynamical systems such as industrial process and health care monitoring. Feature extraction plays a vital role in deciding performance of the time series models. In recent years deep learning techniques have shown an excellent performance to extract highly discriminating features for the classification of the time series. A Convolutional Neural Network (CNN) is a unified framework that performs the feature learning and classification tasks simultaneously. Using the CNN to perform the classification of the multivariate time series is still a challenging task. In this paper, we propose an attention-based multivariate convolutional neural network (AT-MVCNN) that consists of the attention feature-based input tensor scheme to encode informations across the multiple time stamps. The method is capable of learning the temporal characteristics of the multivariate time series. The efficacy of the proposed method is tested on Human Activity Recognition (HAR) and Occupancy Detection datasets. The experiments and results show the proposed method outperforms the other deep learning and traditional machine learning models.**

I. Introduction and Prior Work

Time series (TS) is a collection of data samples arranged according to a time index. The TS with single-dimension is known as univariate time series (UTS) however, whereas TS contains more than one time series is named as multivariate time series (MTS). Classification of the TS is well studied problem in domains like climate behavior [1], robotics [2], health care system [3] and anomaly detection [4], [5]. The classification of the MTS is more challenging as compared to the UTS. In this paper we focus on the task of MTS classification. Numerous statistical-based methods have been proposed in recent years to perform the TS classification. Feature extraction plays a vital role in boosting the performance of the model to classify the MTS. In general, the time series is segmented in multiple segments of equal window size, and handcrafted features are extracted from these segments. The extracted features are further used to classify the time series. The extracted features from the segments are statistical properties [6] or transformation like Fourier [7], [8] and Wavelet transformations [9]. In [10], Bilal et al. proposed trend-value pair features to classify MTS. Bag of feature [11] is another popular feature extraction technique that is extensively used to classify time series. The handcrafted feature suffers from two major drawbacks. The first drawback is sometimes the handcrafted features fail to encode a relationship between the multiple time series, and the second is a preprocessing step i.e. the feature engineering imposes an additional computational cost to classify the MTS. The models that involve the preprocessing step are less desired for the application like the industrial monitoring that requires quick responses to make important decisions.

From the past decade, deep learning-based models have gained considerable attention from data scientists. The deep learning models are capable of extracting hidden features from the time series through hidden layers and substitutes the requirement of the additional feature preprocessing step mentioned above. The deep models are efficient in encoding nonlinear relations and temporal characteristics of the time series. Che et al. [12] proposed a recurrent neural network-based (RNN) framework to classify the multivariate time series . The RNN [13], Long Short Term Memory (LSTM) [14] and Convolutional Neural Network (CNN) [15] are the popular deep learning techniques that are well studied to classify the time series.

Among the above mentioned deep learning methods, the CNN is widely used to perform classification of the MTS. The CNN is well applied to perform the image classification [16]. Despite a great application of CNN to resolve the various challenges from the domains such as text mining, computer vision, and signal processing, use of CNN to classify the MTS is not well investigated. Characteristics of MTS is different from the text and image data. Temporal characteristics and uncertainty of the MTS impose various challenges while creation of the models to classify the MTS. An attempt has been made by Cui et al. to classify the MTS by designing of Multiscale Convolutional Neural Network (MSCNN) [17]. The MSCNN is an end to end neural network that applies multiple transformations on the time series at different scales, sampling rates, and frequencies. After the end of transformations, the MSCNN extracts the discriminating features using the convolution operation. At last, the extracted features are combined

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

to perform the classification. In [18], a unified framework was proposed by Yi et al. to perform MTSC. This framework combines multiple CNNs for the classification of MTS. The method uses a two-step algorithm. Initially, spectral clustering is applied to group the variables into different clusters. Latter coefficients of the created clusters are incorporated in the architecture, and a backpropagation technique is used to learn the model parameters. Borovykh et al. [19] proposed a Wavenet based model that makes the classification of the MTS by arranging the multiple dilated convolutional neural networks in a stacked layer. In [20], a fault diagnosis on semiconductor manufacturing process was performed using CNN. Gao et al. [21] explored the covariance structure between the multiple time series to classify the MTS. In [22], a deep model was proposed to identify a sleep disorder in MTS. A well structured review of the various deep learning-based methods proposed to perform the classification of univariate and multivariate time series can be found in [23].

In [24], author proposed a novel Multivariate Convolutional Neural Network (MVCNN) to perform the classification of the MTS of PHM Challenge 2015 dataset. The model transforms the input raw time series into a tensor. The convolutional layer of the MVCNN performs univariate and multivariate convolution over the input tensors to model the temporal characteristics of the MTS. The experiments and results show the method achieved better performance as compared to the existing deep learning models to classify the MTS. But the deep architecture of the MVCNN suffers from two major drawbacks. The MVCNN takes the raw time series to create the tensor to classify the MTS, thus failed to explain what type of role each input tensor plays during the classification of the MTS. The second is MVCNN fails to encode information across the multiple time stamp. An attention mechanism is well explored to learn relevant and useful part of subsequence to perform classification [25]. The attention mechanism assigns weights to features of the data to select the features that are more relevant to perform the classification. The attention mechanism is capable to encode the information across the multiple time stamp thus can be used to overcome the drawbacks of the MVCNN.

In this paper we incorporated an attention mechanism to resolve the issues mentioned above. A pioneer work related to the use of attention theory with CNN was performed by Yin et al. [26] to solve a research problem of the text mining. They presented three procedure to incorporate the attention features with the CNN model. In [27], author performed the human activity classification in videos by learning the temporal structures using the temporal attention filters. The attention mechanism have been well explored to solve the research problems from the domains such as computer vision [28] and text mining [26]. Tran et al. [28] proposed temporal attention features to classify the financial time series. In the [25] author performed an early classification of the MTS by combining the deep learning and attention mechanism. Shih et al. [29] performed the forecasting of MTS using temporal pattern attention and recurrent neural network.

In the proposed work we used the deep architecture of the MVCNN as the base architecture to develop Attention-Based Multivariate Convolutional Neural Network (AT-MVCNN) for the MTS classification. The proposed method incorporates the attention mechanism during the input tensor transformation step of the MVCNN and creates the attention feature tensors that are further used to train the model. To the best of our knowledge combination of the attention mechanism and MVCNN has not been proposed to classify the MTS. Our major aim in this paper is to study the effect of incorporation of the attention mechanism in the deep architecture of the MVCNN to classify the MTS.

The organization of the paper is as follows: section II presents the brief introduction of the MVCNN and the working of the AT-MVCNN. Section III describes experiments and obtained results, and finally, conclusions and future work are presented in section IV.

II. Methodology

This section presents the brief introduction of the TS, the MTS, the MVCNN and AT-MVCNN.

A. *Representation of Time Series*

As mentioned earliear, the TS is a collection of data samples arranged according to time index. It Ts denotes the TS and if length of the TS in N, then the value of TS at any time stamp is expressed as Ts_j, Where $1 \leq j \leq N$.

*1) **Univariate Time Series (UTS):*** The TS with single dimension is known as the univariate time series (UTS). If M is the dimension of time series then univariate time series can be expressed as Ts_j^d. Where $d = 1$. Fig.(1) shows the UTS with 9 time stamps(N).

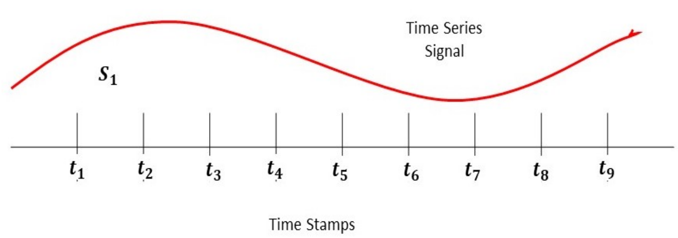

Fig. 1. Univariate Time Series

*2) **Multivariate Time Series (MTS):*** The time series with more than one dimension is known as the multivariate time series (MTS). The MTS is expressed as Ts_j^d. Where $1 \leq j \leq N$. and $1 \leq d \leq M$. Here M is the dimension of the MTS. Fig.(2) shows the MTS with four time series and 9 time stamps.

By considering the aforementioned notations, the MTS dataset can be expressed using three triplets $TSD = (t^j, Ts_j, Y^j)$. Here Y is a class label and the Ts_j is the time series and t_j is the time stamp of the MTS instance.

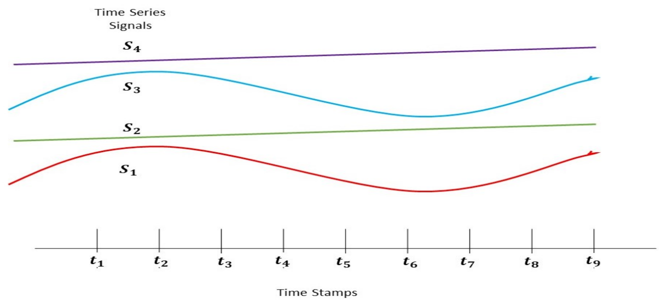

Fig. 2. Multivariate Time Series

B. Architecture of MVCNN

The MVCNN was first proposed by Liu et al.[24]. The primary objective of the MVCNN was to classify the MTS. The MVCNN contains unified framework that transforms the input MTS into tensors and later the convolutional operations are performed over the created tensors to learn the lagged characteristics of the MTS. Major components of the MVCNN are as follows:

*1) **Input Tensor Transformation**:* Initially the input TS is transformed into the tensor of size $(1*1*W_s)$. Here W_s is a size of considered sliding window. The created tensors are combined to form the input tensor for the CNN. The size of input tensor is given as $(h*d*W_s)$. Where h is the height of the tensor, d is the width of the tensor and the W_s is depth of the tensor. Fig.(3) shows a mechanism to create the input tensor from the MTS. The MTS contains four TS T_{S_1}, T_{S_2}, T_{S_3} and, T_{S_4} respectively. The sliding window of the size 4 samples ($W_s = 4$) is used to select the values of the time series from time stamp t_3 to t_6. Initially the size of each tensor is $(1*1*4)$. The tensors are combined to make the new tensor of the size $(2*2*4)$. The depth of the tensor is same as the size of the sliding window.

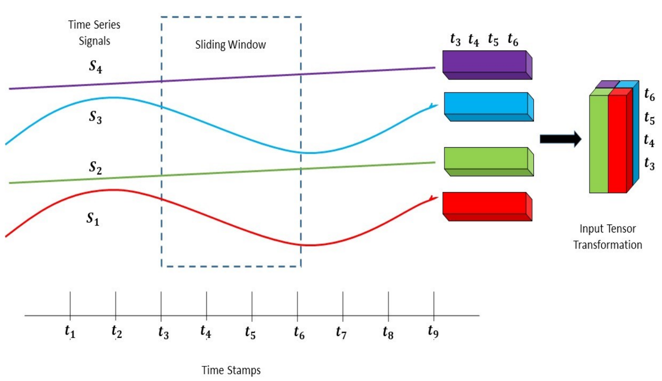

Fig. 3. Creation of Input Tensor from MTS in MVCNN

*2) **Convolution Operation**:* Fig.(4) shows the architecture of the MVCNN. The MVCNN contains convolutional layers to perform the convolution operation on the input tensor. The convolution operation of the MVCNN is performed in two stages. The first is univariate convolution step and the second is the multivariate convolution step. The primary aim of the univariate convolution step is to extract local feature from the individual time series. The multivariate convolution is performed to extract the features that are useful to explain the details regarding the interaction between the multiple time series. More details of the convolution step of the MVCNN can be found in [24].

*3) **Fully Connected Layer**:* At last the fully connected layer was applied at the output of the above explained convolution step. The dropout technique is used to prevent the overfitting problem in the model. Finally the softmax function is employed to transform the output into probability scores. An entropy-based loss function is used during the training of the MVCNN.

To resolve the drawbacks of the MVCNN (as mentioned in the introduction section) we proposed the attention mechanism during the creation of input tensors where we computed the attention matrix to form the attention feature tensors. Motivation to create the attention feature tensor has come from the work of Yin et al. [26]. At last the input tensor and computed attention feature tensor are given as the input to the convolutional layers.

C. Architecture of AT-MVCNN

*1) **Attention Feature Tensor Transformation**:* To create the attention feature map of the input tensor we computed the distance between the data values of the two consecutive sliding windows. The distance between the data samples of the two sliding windows is presented in a form of the attention matrix. For an example Fig.(5) shows the procedure of the computation of the attention matrix. Initially the two input tensors (of the size (1*1*4)) that represent the values of the time series for the two time stamp durations t_1 to t_4 and t_5 to t_8 are selected to compute the entries of the attention matrix. Each entry in the attention matrix of size 4*4 is expressed as follows

$$A_{ij} = \frac{1}{1 + dist(V_i, V_j)} \tag{1}$$

Where dist is an Euclidean distance. If I denotes the input tensor and A denotes the computed attention matrix, then the attention feature tensor F is calculated as

$$F = I * A \tag{2}$$

Fig.(6) shows the difference between the input tensor transformation in the MVCNNN and AT-MVCNN. The AT-MVCNN takes both the attention feature tensor and the inut tensor as the input for the convolutional layer. The first part of Fig.(6) shows creation of the four input tensors of size 1*1*4 that are created from the four input time series of the MVCNN. The same procedure is used in the At-MVCNN with an additional formation of the attention feature tensors. The attention matrix is computed for the each tensor and the four attention feature tensor are created. Now the eight input tensors are further used as input to the AT-MVCNN. Second part of the Fig.(6) shows eight input tensors created

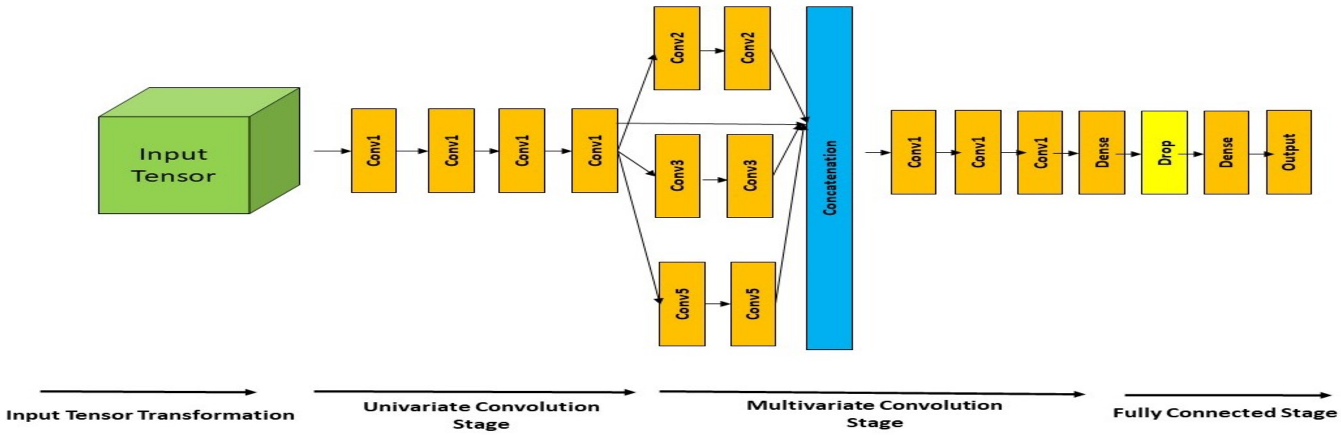

Fig. 4. Architecture of MVCNN

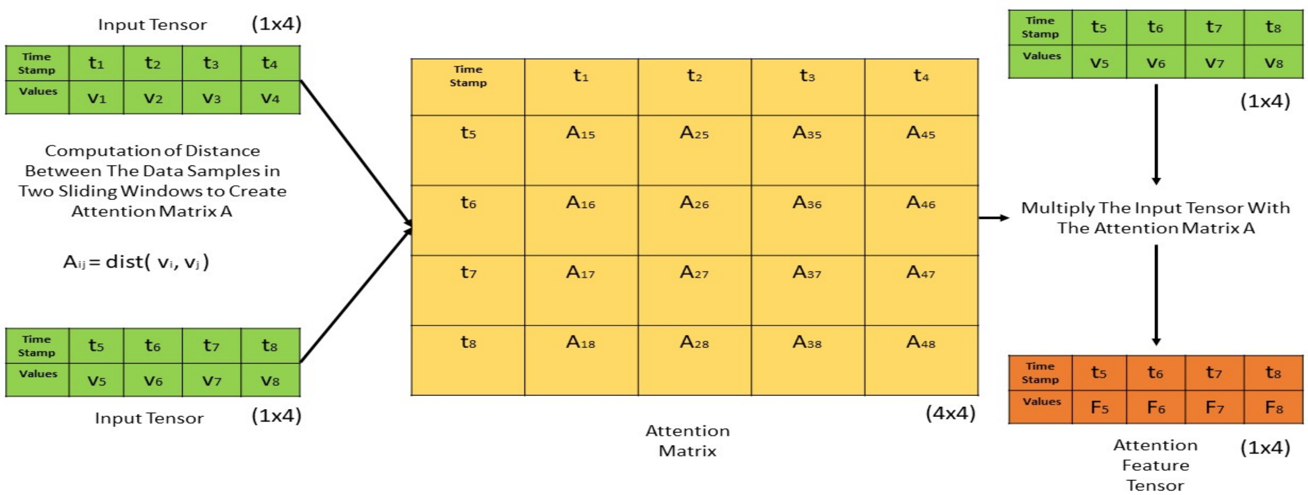

Fig. 5. Computation of Attention Matrix and Creation of Attention Feature Tensor

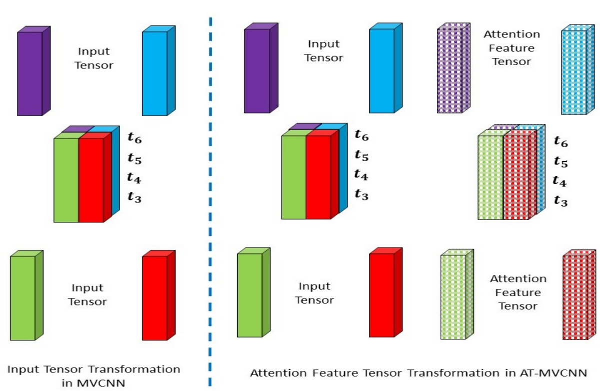

Fig. 6. Creation of Input Tensor for MVCNN and AT-MVCNN

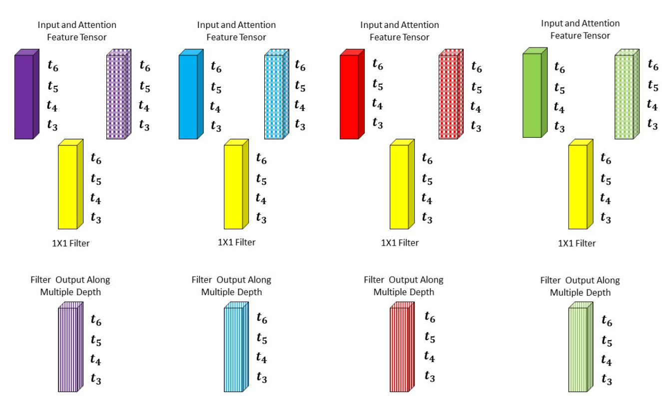

Fig. 7. Convolution Operation In AT-MVCNN

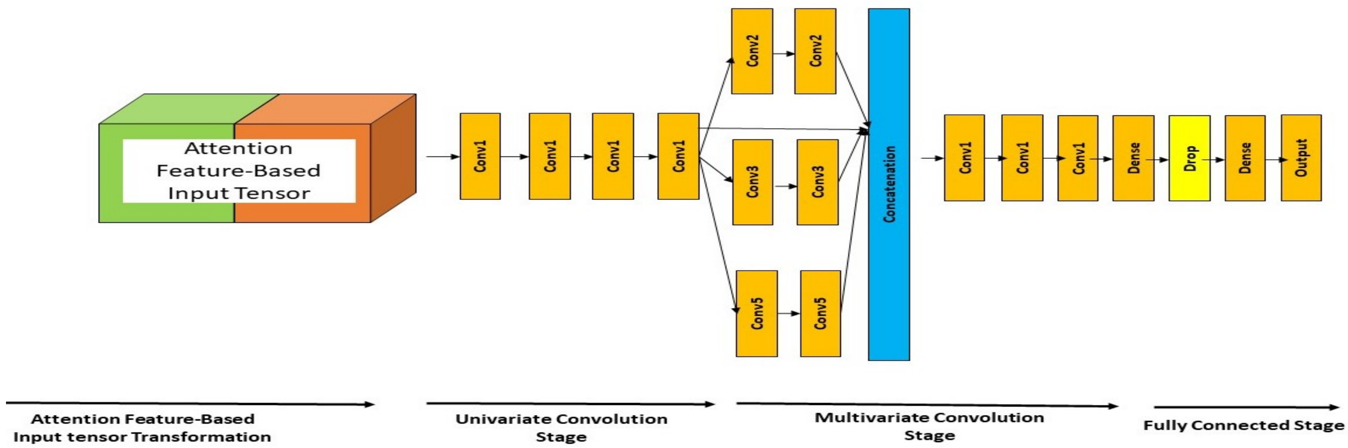

Fig. 8. Architecture of AT-MVCNN

in the input layer of the AT-MVCNN. Fig.(7) shows the procedure of convolution operation performed over the input tensors of the AT-MVCNN. The input and attention feature tensors are processed together with the convolutional filter of size 1*1 and the weights are shared between the input and attention feature tensors to reduce the number of parameters. The architecture of the AT-MVCNN is same as the MVCNN. The only part that differs AT-MVCNN from MVCNN is the input tensor transformation step. An additional computation of the attention feature map is performed in the AT-MVCNN. The AT-MVCNN contains three major layers the attention feature input tensor transformation stage, the convolution layers and the fully connected layers. Fig.(8) shows the architecture of the AT-MVCNN. We used the same base architecture of the MVCNN to implement the AT-MVCNN.

III. Experiments and Results

This section presents the experiments and obtained results.

A. *HAR Dataset*

To evaluate performance of the AT-MVCNN, we picked Human Activity Recognition (HAR) dataset established by Davide et al. [30]. Numerous human activities are recorded through Samsung Galaxy SII. A total of 30 humans of age in between 19 to 48 years are monitored through sensors to capture the different human activities. The sensor data corresponding to the six human activities Walking (W), Walking Down (WD), Walking Up (WU), Sitting (ST), Standing (SD), and Lie Down (LD) are measured and recorded. The recording frequency of the data was selected as 50Hz.

1) ***Training Data****:* The training data consist of a total of 7352 data instances recorded for the 21 subjects (Humans). The training data accommodates 1374 data samples for the W activity, 1407 data instances of the WD activity, 1286 data instances of the WU activity, 986 data samples belong to the SD activity, 1073 data instances of the LD activity and remaining 1226 data samples belonging to the ST activity.

2) ***Test Data****:* The test data contains data gathered to examine the six human activities of the 9 subjects (Humans). The test data accommodates a total of 2947 data instances. Out of these 2947 data instances 532, 491, 537, 496, 420, and 471, data samples belong to the W, WU, WD, ST, SD, and LD activities respectively. The number of data samples for different activities in the training and test data is shown in Table I.

TABLE I
Number of Training and Test Data Samples for Each Activity in HAR Dataset

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

3) ***Model Description****:* Total four models are developed to compare the performance of the proposed method to classify the HAR dataset. We only compared the results of three deep learning models and multi-class support vector machine classifier. Table II shows the four models developed for the analysis of results.

TABLE II
Models Prepared to Compare Results of HAR Classification

S.No.	Model
1	Multi Class Support Vector Machine (MCSVM)
2	Long Short Term Memory (LSTM)
3	Multivariate Convolutional Neural Network (MVCNN)
4	Attention Based Multivariate Convolutional Neural Network (AT-MVCNN)

(i) **Multi Class Support Vector Machine (MCSVM)**: The first model is a multiclass support vector machine. The model is trained using a Gaussian kernel over the training data to identify the numerous humans activities. The window size is selected as 15 data samples to extract the <trend value> pair feature same as used in [4] from the time series of the HAR dataset.

(ii) **Long Short Term Memory (LSTM)**: The next model is a Long Short Term Memory (LSTM). The LSTM models is trained for various mini-batch size and hidden units, and best training accuracy is obtained for the LSTM with 200 hidden units, 100 epochs, mini-batch size of 150, dropout rate of 0.5, learning rate of 0.0001, and RELU activation function.

(iii) **MVCNN**: The MVCNN is trained to minimize the cross-entropy loss function by applying an adaptive moment estimation (Adam) algorithm [31], and a learning rate is selected as 10^-4 to train the model. The windows size (W_s) is selected as 20 data samples to form the input tensor.

(iv) **AT-MVCNN**: The AT-MVCNN is trained using the same parameters explained previously to train the MVCNN.

B. System Conflagration

All the developed models are run on NVIDIA GM107M GPU. The proposed method is developed using an open source Pytorch 1.4 library running on Ubuntu 16.04 LTS operating system.

C. Evaluation Metrics

We presented our analysis of the HAR dataset based on four evaluation metrics i.e. accuracy, precision ,and recall. The presented results are the best results obtained after multiple runs of the models to classify the given test dataset.

*1) **Results of the HAR Dataset**:* The MCSVM model shows precision rate of 78.95%, 74.34%, 93.11%, 81.25%, 80.95%, and 86.20% to classify the W, WU, WD, ST, SD and LD activities respectively. The MCSVM classifier also shows the lowest accuracy of 82.59% to classify the different human activities in the HAR dataset. The LSTM model performs better than MCSVM and yields the accuracy of 88.06% to classify the human activities. The MVCNN achieves the accuracy of 91.34% to classify the numerous human activities. The AT-MVCNN yields the precision rate of 85.92%, 96.33%, 99.52%, 99.79%, 98.80%, and 98.93% to classify the W, WU, WD, ST, SD and LD activities. The AT-MVCNN also yields the highest classification accuracy of 96.27% among all the four models.

TABLE III
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 IV
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 V
Confusion Matrix for MVCNN

Activity	W	WU	WD	ST	SD	LD	Precision (%)
W	**445**	80	0	0	3	4	**83.64**
WU	62	**415**	0	0	0	14	**84.52**
WD	0	0	**523**	0	0	14	**97.39**
ST	0	12	0	**458**	7	19	**92.23**
SD	5	5	2	1	**399**	8	**95.00**
LD	6	1	4	5	3	**452**	**95.96**
Recall (%)	**85.90**	**80.89**	**98.86**	**98.70**	**96.84**	**88.45**	**91.34**

TABLE VI
Confusion Matrix for AT-MVCNN

Activity	W	WU	WD	ST	SD	LD	Precision (%)
W	**455**	72	0	0	2	3	**85.92**
WU	10	**473**	0	0	0	8	**96.33**
WD	0	0	**533**	0	0	4	**99.52**
ST	0	1	0	**495**	0	0	**99.79**
SD	1	1	0	0	**415**	3	**98.80**
LD	1	1	0	1	2	**466**	**98.93**
Recall (%)	**97.43**	**86.31**	**100**	**99.79**	**99.04**	**96.28**	**96.27**

Table III, Table IV, Table V and Table VI shows the confusion matrix obtained by the MCSVM, LSTM, MVCNN and AT-MVCNN classifiers respectively. Table VII shows the results of the various state of the art methods to classify the HAR dataset. It is clear from the Table VII that the AT-MVCNN outperformed the state of the art methods to classify the HAR dataset. The best results are obtained for the $Ws = 20$ data samples for the models LSTM, MVCNN and AT-MVCNN.

D. Occupancy Dataset and Results

We tested the performance of the AT-MVCNN on the MTS of the occupancy dataset. The occupancy dataset contains recording of the temperature, light, CO2 and humidity sensors to detect the occupancy of the humans. All the sensor reading are recorded with a sampling rate of 1 minute. Identification of the occupancy is a binary classification task. The training data contains total 8144 data instances belong to different events. Two test data set are prepared with 2666 and 9753 data instances respectively. We used the same procedure as used in the [24] to classify the MTS of the occupancy dataset. Table VIII shows performance of the three models. The proposed method outperforms the RF classifier and MVCNN. The AT-

TABLE VII
State of the art methods to classify HAR dataset

Paper	Dataset	Method	Accuracy(%)
[32]	UCI_HAR	Handcrafted Features+ RF	77.81
[33]	UCI_HAR	HMM	83.51
[34]	UCI_HAR	CNN+FFT	95.75
[35]	UCI_HAR	Hierarchical HMM	93.18
[36]	UCI_HAR	CNN	90.89
[37]	UCI_HAR	SVM + Stacked Auto encoder	92.16
-	UCI_HAR	MCSVM	82.59
-	UCI_HAR	LSTM	88.06
[24]	UCI_HAR	MVCNN	91.34
-	UCI_HAR	**AT-MVCNN**	**96.27**

MVCNN yields the highest accuracy of 98.34% and 98.69% to classify the first and second test dataset respectively.

The attention mechanism introduced during the input tensor formation step of the AT-MVCNN inspects the information at previous time stamp and selects the relevant information to enhance the classification of the MTS. It is clear from the results that the incorporation of the attention-based mechanism in the MVCNN enhances the accuracy of the deep model and capable to encode temporal information across the multiple time stamps.

TABLE VIII
The Performance of Different Models on OCCUPANCY Dataset

S.No	Model	First Test Set	Second Test Set
1	**Random Forest [24]**	95.05	97.16
2	**MVCNN [24]**	97.40	97.72
3	**AT-MVCNN**	**98.34**	**98.69**

IV. Conclusion and Future Work

The classification of the TS is an essential task required by various real-world applications. The proposed method is aiming to classify the MTS by incorporating the attention mechanism in the deep architecture of the MVCNN. The new attention feature-based tensors are efficient in encoding the information across the multi time stamp and relationship between the multiple time series. The results and experiments show the proposed method outperforms the MVCNN, the LSTM and MCSVM to classify the MTS.

An investigation of the effect of the attention mechanism during the multivariate convolution operation of the AT-MVCNN would be significant future work. In future, we would like to explore application of the AT-MVCNN to identify the anomalies in the MTS.

References

[1] N. Maknickienė, A. V. Rutkauskas, and A. Maknickas, "Investigation of financial market prediction by recurrent neural network," *Innovative Technologies for Science, Business and Education*, vol. 2, no. 11, pp. 3–8, 2011.

[2] C. Pérez-D'Arpino and J. A. Shah, "Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 6175–6182.

[3] S. S. Jones, R. S. Evans, T. L. Allen, A. Thomas, P. J. Haug, S. J. Welch, and G. L. Snow, "A multivariate time series approach to modeling and forecasting demand in the emergency department," *Journal of biomedical informatics*, vol. 42, no. 1, pp. 123–139, 2009.

[4] 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.

[5] ——, "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.

[6] A. Nanopoulos, R. Alcock, and Y. Manolopoulos, "Feature-based classification of time-series data," *International Journal of Computer Research*, vol. 10, no. 3, pp. 49–61, 2001.

[7] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," in *Proceedings of the 23rd ACM international conference on Multimedia*. Acm, 2015, pp. 1307–1310.

[8] K. Samiee, P. Kovacs, and M. Gabbouj, "Epileptic seizure classification of eeg time-series using rational discrete short-time fourier transform," *IEEE transactions on Biomedical Engineering*, vol. 62, no. 2, pp. 541–552, 2014.

[9] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices," *IEEE journal of biomedical and health informatics*, vol. 21, no. 1, pp. 56–64, 2016.

[10] 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.

[11] M. G. Baydogan, G. Runger, and E. Tuv, "A bag-of-features framework to classify time series," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2796–2802, 2013.

[12] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, p. 6085, 2018.

[13] M. Hüsken and P. Stagge, "Recurrent neural networks for time series classification," *Neurocomputing*, vol. 50, pp. 223–235, 2003.

[14] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "Lstm fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2017.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

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

[18] S. Yi, J. Ju, M.-K. Yoon, and J. Choi, "Grouped convolutional neural networks for multivariate time series," *arXiv preprint arXiv:1703.09938*, 2017.

[19] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," *arXiv preprint arXiv:1703.04691*, 2017.

[20] K. B. Lee, S. Cheon, and C. O. Kim, "A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes," *IEEE Transactions on Semiconductor Manufacturing*, vol. 30, no. 2, pp. 135–142, 2017.

[21] J. Gao, Y. L. Murphey, and H. Zhu, "Multivariate time series prediction of lane changing behavior using deep neural network," *Applied Intelligence*, vol. 48, no. 10, pp. 3523–3537, 2018.

[22] S. Chambon, M. N. Galtier, P. J. Arnal, G. Wainrib, and A. Gramfort, "A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 4, pp. 758–769, 2018.

[23] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[24] 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.

[25] E.-Y. Hsu, C.-L. Liu, and V. S. Tseng, "Multivariate time series early classification with interpretability using deep learning and attention mechanism," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2019, pp. 541–553.

[26] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "Abcnn: Attention-based convolutional neural network for modeling sentence pairs," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 259–272, 2016.

[27] A. Piergiovanni, C. Fan, and M. S. Ryoo, "Learning latent subevents in activity videos using temporal attention filters," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[28] D. T. Tran, A. Iosifidis, J. Kanniainen, and M. Gabbouj, "Temporal attention-augmented bilinear network for financial time-series data analysis," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 5, pp. 1407–1418, 2018.

[29] S.-Y. Shih, F.-K. Sun, and H.-y. Lee, "Temporal pattern attention for multivariate time series forecasting," *Machine Learning*, vol. 108, no. 8-9, pp. 1421–1441, 2019.

[30] 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.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] 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.

[33] 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.

[34] 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.

[35] ——, "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.

[36] 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.

[37] 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.