Comparison of Predictive Models for Forecasting Time-series Data

Serkan Özen
Middle East Technical University
Department of Computer Engineering
Ankara, Turkey
e154353@metu.edu.tr

Volkan Atalay
Middle East Technical University
Department of Computer Engineering
Ankara, Turkey
vatalay@metu.edu.tr

Adnan Yazıcı
Middle East Technical University
Department of Computer Engineering
Ankara, Turkey
yazici@ceng.metu.edu.tr

ABSTRACT
Dramatic increase in data size enabled researchers to study analysis and prediction of big data. Big data can be formed in many ways and one alternative is through the use of sensors. An important aspect of data coming from sensors is that they are time-series data. Although forecasting based on time-series data has been studied widely, it is still possible to advance the state-of-the-art by constructing new hybrid deep learning models. In this study, Random Forest, Convolutional Neural Network, Long Short Term Memory and hybrid Convolutional Neural Network-Long Short Term Memory models are applied and assessed on meteorological time-series data. Vector Auto-regression model and Multi-layer Perceptron model are used as the baseline forecasting methods for comparison purposes. Root Mean Square Error of the models for predictions are calculated for performance assessment which reveals the performance of these deep learning methods for forecasting based on time-series data.

CCS Concepts
• CCS → Computing methodologies → Machine learning → Learning paradigms → Supervised learning → Supervised learning by regression • CCS → Computing methodologies → Machine learning → Machine learning approaches → Neural networks..

Keywords
Deep learning; Time-series prediction; Supervised learning

1. INTRODUCTION
Time-series data generated by a physical device or process indicate a data type that changes over time. Time-series data is widely used in the field of Internet of Things (IoT) and in many systems such as basic operation and maintenance systems. Time-series forecasting is to predict future values based on previously observed values by using a model. Time-series forecasting has become a popular topic since 2000s because of the rise in the amount of available data. It has been used across various domains such as climate, finance, health, and agriculture. Linear regression such as autoregression and ARIMA models can be used to predict future values of univariate dataset. However, most of the time-series datasets consist of several variables. This type of dataset is called multivariate and revealing relationship among the variables of such a dataset cannot be extracted by simple univariate regression models. Instead, multivariate models such as Vector Auto-regression, Random Forest, Multi-layer perceptron (MLP) and deep neural networks (DNN) are used for revealing relationship among dataset variables. In this study, historical multivariate meteorological dataset between 1970 and 2015 is used to assess the performance of several predictive models for forecasting time-series data. With the aim of revealing the performances of deep neural network models for forecasting based on time-series data, we present the results of applying Convolutional Neural Network (CNN), Long-short term Memory (LSTM) and hybrid CNN-LSTM models on the meteorological time-series data. The models are designed for prediction of the future temperature values. As baseline predictors, a Vector Auto-regression (VAR) model, a Random Forest (RF) regressor and a multi-layer perceptron are used and compared with CNN, LSTM and CNN-LSTM models. The reason why these methods are used as baseline is that VAR model is a linear model for multivariate time-series data while MLP is the simplest form of neural networks which reveals the non-linear relationship among the variables of multivariate time-series dataset

ICBDR 2019, November 20–22, 2019, Cergy-Pontoise, France

ACM ISBN 978-1-4503-7201-5/19/11…$15.00
https://doi.org/10.1145/3372454.3372482

2. ARTIFICIAL NEURAL NETWORKS for TIME-SERIES ANALYSIS
Artificial neural networks have been widely used for forecasting in several domains particularly in meteorology, agriculture, energy, and finance. In one of the earliest studies, Lapades and Farber used shallow feed-forward neural networks (FFNN) for predicting non-linear time-series data and showed that FFNNs performed much better and provide higher accuracy than those of conventional methods [1]. FFNNs were then employed for forecasting precipitation [2, 3], bankruptcy of firms [4] and electricity consumption [5]. Subsequently, the use of hybrid approaches has been investigated for forecasting tasks. Khashei and Bijari and also Durdu combined ANN and auto-regressive integrated moving average (ARIMA) models for time-series analysis which gave better results than any single of them [6, 7]. Paulo et al. have employed the ensemble of 5 FFNNs each of which has different number of neurons at the hidden layer and different time lags [8]. The results show that ensemble of FFNNs perform better than ARIMA which is a classical linear time-series analysis method. Deep neural networks have recently gained popularity because of the radical increase in the size of available data and in the computational power as well. Kuremoto et al. have used Restricted Boltzmann Machines (RBM) and Deep Belief Networks (DBN) in order to form a time-series forecasting model [9]. They used 3 layers of RBMs to form a DBN. RBMs were pre-

trained through the use of their energy functions. The back propagation algorithm was then used for fine tuning of weights between visible and hidden layers of RBMs. Results showed that DBN performed better than classical ANNs and linear prediction model ARIMA. Liu et al. [10] have proposed a model for weather forecasting task consisting of 5 hidden layer DBN. Each layer of DBN is pretrained with an unsupervised auto-encoder to learn the non-linear transformation at its input. Results show that this hybrid model outperforms a single a support vector regressor (SVR) for the weather prediction task. RBMs are used for time-series analysis in a variety of studies [11, 12]. DBN is combined with Arima model for time-series analysis [13] and it is used for weather forecasting [14]. Cai et al. [15] have used RNN trained with a hybrid of particle swarm optimization (PSO) and evolutionary algorithm to predict 30 missing points in a dataset consisting of 5000 rows. Zhang and Xiao have used RNN to make prediction in chaotic time series which gives better results than traditional neural networks [16]. Hybrid deep learning models are widely used in the time-series prediction domain. Bao et al. used deep learning model for stock price forecasting [17]. In their model, as a first step, Wavelet Transform is applied on time-series data to remove the noise. The data is then fed to Stacked Denoising Auto-Encoder (SDAE) to get high-level variables. Finally, LSTM is used to obtain next day's closing stock price. Results show that hybrid models such that wavelet transform-SDAE-LSTM have better prediction performance than those of single LSTM and RNN models. Qiu et al. [18] proposed a model composed of DBNs and a support vector regressor (SVR) at the end of the model. Time-series input is fed into 20 DBNs each of which is trained with different number of epochs. Output of each DBN is then put into a single vector. This vector is fed into SVR to get the final prediction value. Results show that ensemble method outperforms single DBN, SVR and MLP models. In another study [19], SDAE is employed and different hyper parameter configurations are selected in order to forecast in-door temperature. Lv et al. [20] used stacked auto-encoders in order to predict traffic flow values from big data. In the proposed model, SDAE-logistic regressor combination is used for the prediction task. Results show that this hybrid model outperforms current traffic flow prediction models realized with Random Walk, MLP, Radial Basis Function. Hammerla et al. used CNN for extracting features from raw time-series data [21]. Liu et al. have used CNN model to predict time-series data [22]. Lin et al. have used hybrid CNN-LSTM model called TreNet in order to learn trends in time-series data [23] and they showed TreNet outperforms single CNN, LSTM and other traditional methods. Du et al. have used a hybrid CNN-LSTM framework for traffic flow prediction and showed that hybrid model has better prediction results than those of traditional and single models [24]. Wu et al. have used CNN-RNN hybrid approach to extract spatial features with CNN and temporal features with gated RNN which is claimed to outperform state-of-art traffic flow prediction models [25].

3. PREDICTION MODELS

Explanations of the models are given instead of detailed architectures and figures because of space constraints. m step sliding window of data is given as input and subsequent n steps of data are predicted by a model. The length m of sliding window is one of the hyper-parameters affecting the model prediction performance. After intensive experimental evaluation, $m = 20$ and $n = 10$ are found to be the best values for this dataset. The first baseline method for time-series prediction is selected to be Vector Auto-regression Model (VAR). It is multivariate version of Auto-regression model which is used for time-series prediction of a single variable. Future values of a variable is linear function of its past values. In the multivariate case, Auto-regression model learns the relationship between the past values of all variables. Since meteorological data is multivariate, VAR can be used for prediction of the temperature value. A shallow multi-layer perceptron (MLP) is also selected to be a baseline method particularly for the deep neural network models. First layer of MLP has 140 nodes since 20 time-steps of 7 variables require 20 x7 = 140 nodes. Input dimension of the simple CNN used in this study is 20x7 meaning that 20 rows (time-steps) from the dataset with 7 variables are given as input to the network. Two consecutive convolutional layers are added right after input layer so that a subset of the initial input is extracted. Then a max-pooling layer is added in order to extract more important high level features. Similarly, convolutional layer and max-pooling layer are used for the same reasons. Then, two dimensional output is flattened in order to obtain one dimensional output. In the final part, two dense layers, meaning fully connected to each other similar to a MLP are applied to one dimensional data in order to predict the next 10 time-steps. Input layer of the LSTM model accepts tensor of 20 time-steps of 7 variables. Then, the output is fed to Keras Repeat-vector in order to get hidden features for 10 future time-steps. Output's repeat vector is again fed to an LSTM layer in order to extract abstract features of previous time-steps. Then, two consequent dense layers are applied to each of output time-step in order to get the best guesses for future time-steps. As a result, the network gives output tensor of (10,1) dimensions meaning 10 future time-steps of a variable. On various domains hybrid systems have been shown to outperform single models. Therefore, an individual CNN and an individual LSTM models are combined to form a hybrid model. In this model, CNN is used as encoder and LSTM is used as decoder. First two layers of CNN are convolutional layers that are used to filter important features. Then, max-pooling layer is applied to get the highest score feature from the feature map. Finally, output vector is flattened to get 10 time-step values of a single feature. Random Forest is another model that can be used to predict future values of multivariate time series data. Historical data until time t is used to predict variable value at time t+1. Random Forest used in this experiment has 650 number of estimators (number of decision trees). This number is obtained through several experiments by observing the best results of random forest.

4. EXPERIMENTAL SETTING

The base dataset used in this study is the meteorological data including 7 different variables (features or attributes) from 27 different stations over 30 years. The variables are date, vapour pressure, average actual pressure, average sun duration (daily insolation), average cloudiness, average humidity, average daily wind speed, average daily temperature. Dataset is composed of daily measurements of these variables. Only December, January and February data of a certain city (Adana), which corresponds to 4062 items (samples or rows) of data, are used in this study. A few number of samples of the dataset is given in Table 1. The data is graphically shown for about 100 days of winter of 1970 is shown in Figure 1 while the data distribution of each variable is demonstrated in Figure 3. In Figure 1, the values for pressure variable is not shown since their range is high compared to the others. Correlation analysis of variables is shown in Figure 4. As it can be observed, correlation values are not close to 1 or -1 which means that the variables are not highly correlated.

Table 1. First 20 rows of dataset

Days	Vap. Pr.	Avg. Act. Pr.	Avg. Sun Dur.	Avg. Cloud.	Avg. Hum.	Avg. Wind Sp.	Avg. Temp.
1	8.9	1016.6	8	27	77.7	11	9.7
2	88	1018.6	5	4	66.3	21	11.2
3	8.3	1016.2	0	47	55.3	27	12.7
4	9.1	1016.4	0	65	58.7	22	13.6
5	8.6	1015.9	7	43	54.7	14	13.6
6	9.1	1015.7	8	17	6.9	12	11.4
7	10.2	1017.4	7	17	72.3	10	12.3
8	107	1017.2	7	3	74.7	10	12.5
9	108	1014.7	5	4	71.7	11	13.6
10	102	1013.7	6	5	6.1	14	14.6
11	8	1014.9	8	13	50.7	16	1.4
12	9.2	1014.7	6	3	6.4	13	11.9
13	10.5	1010.4	0	45	72.3	14	12.1
14	10.7	1013.5	0	5	76.7	26	12.7
15	8.1	1013.5	0	7	67.7	22	12.7
16	10.1	1017.5	7	47	68.7	13	13.1
17	11.1	1018.5	0	43	77.3	10	12.4
18	9.6	1015.4	5	4	6.8	13	13.9
19	9.4	1010	0	7	74.3	25	12.5
20	11.1	1008.6	0	7	87.3	23	9.5

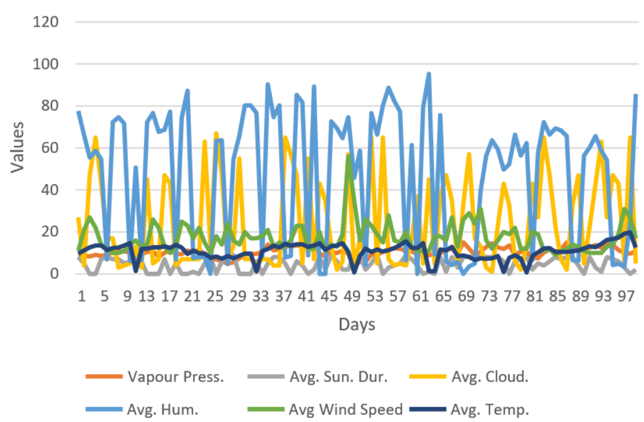

Figure 1. Graphical plot of the variable values for about 100 days of winter of 1970.

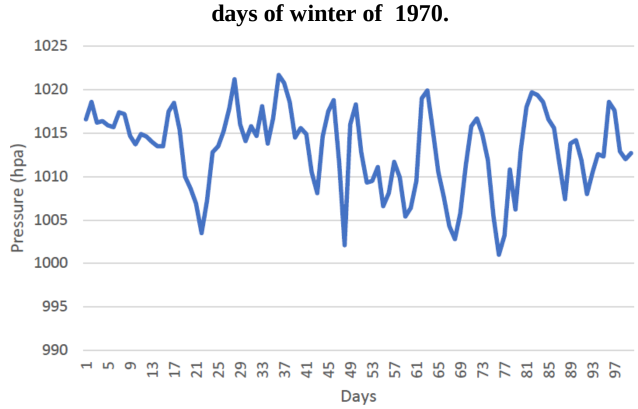

Figure 2. Graphical plot of the first 100 days' average pressure since 1970.

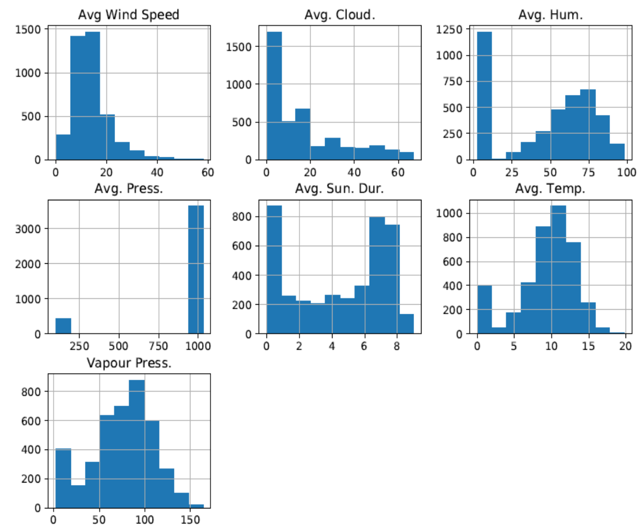

Figure 3. Data distribution of each variable.

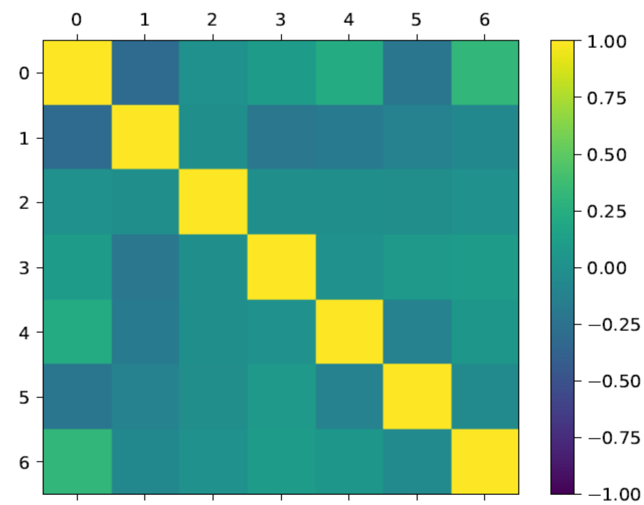

Figure 4. Correlation analysis of the variables. 0: Vapour Pressure, 1: Sunshine Duration, 2: Average Pressure, 3: Average Cloudiness, 4: Average Humidity, 5: Average Wind Speed, 6: Average Temperature

5. EXPERIMENTS and EVALUATION

The future values of temperature are predicted based on historical meteorological data. The relationship among the variables are learned by the models and future values of temperature are then predicted. The number of inputs to the model is the variable values for 20 days and predictions are the temperature values for the next 10 days. When the models are trained on the historical data, in 20 days input and 10 days output manner, then model can predict the temperature values for the next 10 days given the variable values of the previous 20 days as the input. Tests are performed with the test data which is not used to train models. Performance results of different models are assessed in terms of Root Mean Squared Error (RMSE) which is given as follows.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(p_i - a_i)^2} \tag{1}$$

where n is number of predictions of day's feature, pi is predicted value and ai is actual value of ith prediction. Here, n is the number of times the prediction is made for each of the days, that is, n = (testSize - dataSize) - outputSize+1. For example, if test dataset has 10 days (rows) and model accepts 3 days and predicts next 2 days, then n becomes 6 because n = (10 - 3) - 2 + 1. Test is performed in an iterative manner. Test data is split into portions composed of 30 days with 1 sliding window, i.e. sample portions are p1 = day1, day2, .., day30; p2 = day2, day3, …, day31 etc. By this way, meteorological values for 20 days are given to the model and the temperature values for the subsequent 10 days are obtained. The actual temperature values of these 10 days and the predicted ones are fed to the RMSE equation to find the prediction accuracy of the specific model. The RMSE values of all days are plot in Figure 5. Our dataset consists of 4062 rows. We have used 10-fold cross validation to increase the accuracy validation. Data is split into 10 static groups and experiments are repeated by selecting a different group as test data. As a result, training data size is 9 x 406 = 3654 while test dataset size is 1 x 406 = 406. This means that there are 377 predictions made for each of days, since n = (406 - 20) - 10 + 1 = 377. 10 fold cross validation is performed to test the models on the different portions of data. It enables models to be trained and test on different portions of data and remove the danger of learning the local trends in dataset. Increasing the number of folds increases accuracy of models' performance.

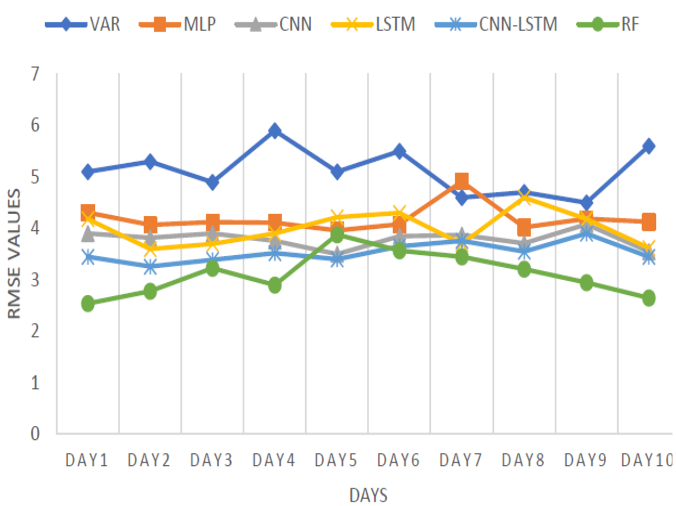

Figure 5. RMSE values of all models.

6. DISCUSSION

Overall RMSE values of 10 days' prediction of the models are given in Figure 5. For the purpose of forecasting meteorological data, neural network models perform better than the baseline VAR model, since meteorological multivariate data have non-linear relationship among the variables. In addition, deep learning models have smaller forecasting error values than that of the shallow neural network model. Furthermore, hybrid models improve prediction results more than the single models do. Hybrid CNN-LSTM model outperformed single CNN model and single LSTM model, because hybrid structures behave like encoder-decoder models, where first part extract abstract features from data and second part attempts to map these features to real prediction results. Random-forest seems to perform better than hybrid CNN-LSTM model for this problem. This is because amount of data is not enough for deep neural network to capture relevance among the variables whereas random-forest performs well on small amount of data where seasonality or trending don't exist in data. As a result, it can be seen that hybrid deep neural network performs better than those of single ones and random-forest performs even better than deep hybrid models when dataset is not large enough. Our next study will be on data fusion. We will analyze effect of multiple data sources on data prediction performance. Amount of data and predicted days' count will be increased.

7. REFERENCES

[1] A. S. Lapedes and R. M. Farber, "How neural nets work," in Neural Information Processing Systems, pp. 442–456, 1988.

[2] K.-l. Hsu, X. Gao, S. Sorooshian, and H. V. Gupta, "Precipitation estimation from remotely sensed information using artificial neural networks," Journal of Applied Meteorology, vol. 36, no. 9, pp. 1176–1190, 1997.

[3] R. J. Kuligowski and A. P. Barros, "Localized precipitation forecasts from a numerical weather prediction model using artificial neural networks," Weather and Forecasting, vol. 13, no. 4, pp. 1194–1204, 1998.

[4] M. D. Odom and R. Sharda, "A neural network model for bankruptcy prediction," in 1990 IJCNN International Joint Conference on Neural Networks, pp. 163–168, IEEE, 1990.

[5] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," Neural Computation, vol. 3, no. 2, pp. 246–257, 1991.

[6] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and arima models for time series forecasting," Applied Soft Computing, vol. 11, pp. 2664–2675, Mar. 2011.

[7] D. Ömer Faruk, "A hybrid neural network and arima model for water quality time series prediction," Engineering Applications of Artificial Intelligence, vol. 23, no. 4, pp. 586 – 594, 2010.

[8] C. Paulo, R. Miguel, R. Miguel, and S. Pedro, "Multi-scale internet traffic forecasting using neural networks and time series methods," Expert Systems, vol. 29, no. 2, pp. 143–155, 2012.

[9] T. Kuremoto, S. Kimura, and Kobayashi, "Time series forecasting using a deep belief network with restricted boltzmann machines," Neurocomputing, vol. 137, pp. 47–56, 2014.

[10] J. N. K. Liu, Y. Hu, J. J. You, and W. M. Chan, "Deep neural network based feature representation for weather forecasting," 2014.

[11] R. Hrasko, A. G. Pacheco, and R. A. Krohling, "Time series prediction using restricted boltzmann machines and backpropagation," Procedia Computer Science, vol. 55, pp. 990–999, 2015.

[12] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time series forecasting using restricted Boltzmann machine," in Communications in Computer and Information Science, vol. 304, pp. 17–22, 07 2012.

[13] T. Hirata, T. Kuremoto, M. Obayashi, S. Mabu, and K. Kobayashi, "Time series prediction using dbn and arima," in 2015 International Conference on Computer Application Technologies, pp. 24–29, Aug 2015.

[14] H. Takaomi, T. Kuremoto, M. Obayashi, S. Mabu, and K. Kobayashi, “A novel approach to time series forecasting using deep learning and linear model,” IEEJ Transactions on Electronics, Information and Systems, vol. 136, pp. 348–356, 03 2016.

[15] X. Cai, N. Zhang, G. K. Venayagamoorthy, and D. C. Wunsch, “Time series prediction with recurrent neural networks trained by a hybrid pso–ea algorithm,” Neurocomputing, vol. 70, no. 13, pp. 2342 – 2353, 2007.

[16] J.-S. Zhang and X.-C. Xiao, “Predicting chaotic time series using recurrent neural network,” Chinese Physics Letters, vol. 17, no. 2, p. 88, 2000.

[17] W. Bao, J. Yue, and Y. Rao, “A deep learning framework for financial time series using stacked autoencoders and long-short term memory,” PLoS ONE, vol. 12, no. 7, p. e0180944, 2017.

[18] X. Qiu, L. Zhang, Y. Ren, P. N. Suganthan, and G. Amaratunga, “Ensemble deep learning for regression and time series forecasting,” in 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL), pp. 1–6, IEEE, 2014.

[19] P. Romeu, F. Zamora-Mart ńez, P. Botella-Rocamora, and J. Pardo, “Time-series forecasting of indoor temperature using pre-trained deep neural networks,” in Artificial Neural Networks and Machine Learning – ICANN 2013 (V. Mladenov, P. Koprinkova-Hristova, G. Palm, A. E. P. Villa, B. Appollini, and N. Kasabov, eds.), (Berlin, Heidelberg), pp. 451–458, Springer Berlin Heidelberg, 2013.

[20] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, “Traffic flow prediction with big data: A deep learning approach,” IEEE Transactions on Intelligent Transportation Systems, vol. 16, pp. 865–873, April 2015.

[21] N. Y. Hammerla, S. Halloran, and T. Ploetz, “Deep, convolutional, and recurrent models for human activity recognition using wearables,” CoRR, vol. abs/1604.08880, 2016.

[22] J. Liu, K. Zhao, B. Kusy, J.Wen, and R. Jurdak, “Temporal embedding in convolutional neural networks for robust learning of abstract snippets,” CoRR, vol. abs/1502.05113, 2015.

[23] T. Lin, T. Guo, and K. Aberer, “Hybrid neural networks for learning the trend in time series,” tech. rep., 2017.

[24] S. Du, T. Li, X. Gong, Y. Yang, and S. J. Horng, “Traffic flow forecasting based on hybrid deep learning framework,” in 2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), pp. 1–6, IEEE, 2017.

[25] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, “A hybrid deep learning based traffic flow prediction method and its understanding,” Transportation Research Part C: Emerging Technologies, vol. 90, pp. 166 – 180, 2018.