Proceedings of the Federated Conference on Computer Science and Information Systems pp. 101–104
DOI: 10.15439/2020F137
ISSN 2300-5963 ACSIS, Vol. 21

Shallow, Deep, Ensemble models for Network Device Workload Forecasting

Cenru Liu
Ngee Ann Polytechnic, Singapore
liucenru@gmail.com

***Abstract*—Reliable prediction of workload-related characteristics of monitored devices is important and helpful for management of infrastructure capacity. This paper presents 3 machine learning models (shallow, deep, ensemble) with different complexity for network device workload forecasting. The performance of these models have been compared using the data provided in FedCSIS'20 Challenge. The R^2 scores achieved from the cascade Support Vector Regression (SVR) based shallow model, Long short-term memory (LSTM) based deep model, and hierarchical linear weighted ensemble model are 0.2506, 0.2831, and 0.3059, respectively, and was ranked 3^{rd} place in the preliminary stage of the challenges.**

***Index Terms*—Workload forecasting, Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Support Vector Regression (SVR), Hierarchical Linear Weighted Ensemble**

I. Introduction

EMCA Software is a Polish vendor of Energy Log server, which is capable of collecting data from various log sources and providing in-depth data analysis to its end-users. The objective of the FedCSIS'20 challenge is to explore reliable machine learning models to predict workload-related characteristics of monitored devices, based on historical data gathered from such devices, which is important for IT and technical teams to manage the capacity of their infrastructure [1].

Workload forecasting models have been developed based on machine learning methods in the literature. Future host load was predicted using 9 features extracted from historical workload values by using the Bayesian model in [2]. A forecasting model by combining neuro-fuzzy and Bayesian inference was developed for CPU workload forecasting in [3]. In [4], a workload forecasting model has been developed based on Artificial Neural Network (NN) and adaptive Differential Evolution (DE). Workload was predicted by using Autoregressive Integrated Moving Average (ARIMA) model in [5]. To consider temporal dependencies in workload sequence data, recently, Recurrent neural network (RNN) and its variant, Long short-term memory (LSTM), have been employed in workload forecasting and shown promising performance [8], [6], [7].

To explore the performance of shallow, deep and ensemble models and cater for FedCSIS'20 Challenge, we developed 3 network device workload forecasting models:

1) a cascade shallow model based on Support Vector Regression (SVR);
2) a deep learning model based on LSTM;
3) a hierarchical linear weighted ensemble model.

The performance of these 3 models were compared using the network device workload data provided in FedCSIS'20 Challenge. The hierarchical ensemble of LSTM achieved the highest R^2 score in the preliminary stage, while the cascade SVR model was more robust to overfitting.

This paper is organized as follows. The FedCSIS'20 challenge is briefly introduced in Section II. The cascade shallow model is presented in Section III, the LSTM based deep model is given in Section IV, and the hierarchical linear ensemble model is described in Section V. Section VI compares the performance of the 3 models. Conclusions are given in Section VII.

II. FedCSIS 2020 Challenge: Network Device Workload Prediction

In this section, we briefly introduced the FedCSIS'20 Challenge titled as Network Device Workload Prediction [1]. The task in this challenge is to predict future workload characteristics of a number of monitored devices based on the given historical data collected from these devices.

A. Data

The dataset provided in this challenge is in the format of a .csv file, which holds a table of over forty-four million rows and ten columns. The 10 columns include identifiers followed by the mean, standard deviation, and a candlestick aggregation of the corresponding values, as listed below:

- hostname: an ID of the device;
- series: a name of the considered characteristic;
- time window: a timestamp of the aggregation window;
- Mean: the mean of the values;
- SD: the standard deviation of the values;
- Open: a value of the first reading during the corresponding hour;
- High: the maximum of values;
- Low: the minimum of values;
- Close: a value of the last reading during the corresponding hour;
- Volume: the number of values.

B. Task

The data for each hostname-series pair can be arranged into 7 time series spanning over 80 days, which are values of mean, SD, open, high, low, close, and volume. The participants of the

challenge were required to forecast the mean of the workload values in each of the next 168 hours after the end of the training data for ten thousand hostname-series pairs selected from these over twenty-four thousands pairs.

C. Evaluatoin

The solutions were assessed by the R^2 measure. The forecasts of each time series are compared to ground truth values and assessed using the R^2 score that is defined as:

$$R^2(f, y) = 1 - \frac{RSS(f, y)}{TSS(y)}. \tag{1}$$

RSS is the residual sum of squares of forecasts and TSS is the total sum of squares, given as

$$\begin{aligned} RSS &= \sum_i (y_i - f_i)^2, \\ TSS &= \sum_i (y_i - \frac{1}{N}\sum_i y_i)^2, \end{aligned} \tag{2}$$

where y_i and f_i are the ground truth and their prediction, respectively, and N is length of the time series. The score of a submitted solution is the average R^2 value over all time series from the test set.

The preliminary scores of the submitted solutions were evaluated externally and published on the challenge leaderboard computed on a small subset (10%) of the test time series that are fixed for all participants. The final evaluation will be published after completion of the competition using all of the test time series.

III. Shallow Model

Support Vector machine (SVM) was proposed by Vladimir Vapnik and his co-workers based on the statistical learning theory (or VC theory) [9], [10], [11], [12], [13], [14], [15], [16], [17]. The SVM has shown competitive generalization ability over many existing machine learning models in a number of fields, e.g. optical character recognition (OCR), object recognition, time series prediction, etc. [13], [18], [19], [20], [21]. The Support Vector Regression (SVR) is a powerful regression approach and successfully applied in numerous applications [22], [23], [24], [25]. In this work, a cascade shallow model has been developed based on the SVR with the Radial basis function kernel (RBF) for workload prediction.

Although 7 types of hourly aggregated workload values were provided in the challenge, only the hourly mean of the data was used in our method. The data were organized in a matrix, in which each row represents the time series of a hostname-series pair and each column stores the mean of workloads in one hour. The data were standardized to have zero mean and unit standard deviation, which is essential to non-linear machine learning models.

One difficulty in this challenge is arising from the fact that the devices considered in the data were not uniform and some of the devices were a part of the same system and it is likely that their workloads were highly correlated and cross-dependent [1]. To increase the diversity of training, the cascade

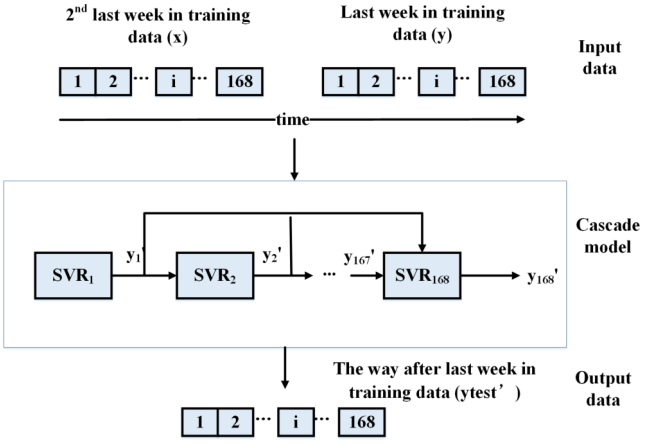

Fig. 1. Structure of the cascade SVR model and composition of its input and output.

SVR-based models are trained on the following two parts of the data provided:

- training set 1: the 10k time series involved in testing;
- training set 2: the time series selected from the other 14k sequences based on the following rules: having less missing values and closer to the 10k testing sequences.

Instead of using all data in over 80 days, only the values in the last 2 weeks, i.e. 336 average hourly values, were used in training.

The cascade shallow model is composed of a series of connected single SVR models denoted as hourly models, each of which was trained to predict the mean of workloads in one hour. Let SVR_t represent an hourly model predicting the mean workloads in the t^{th} hour, where $t \in [1, 168]$. The input features to SVR_t can be the values in all of the hours before t. Assuming weekly periodic property of workloads, we took only the values from the previous 168 hours. Therefore, the feature length of an hourly model is 168.

In the cascade shallow model, there are a series connected hourly models that are trained one by one. The outputs from the previous hourly models will be used as partial inputs to all subsequent models. Fig. 1 illustrates the structure of the cascade SVR model and the composition of its input and output. In this way, the latter model can be adapted based on the predictions from its previous models, which conforms to the cognition that the previous values are correlated to the latter ones.

The hyper-parameters of the non-linear SVR models with RBF kernel were set as follows.

ϵ in the ϵ-insensitive loss function was set to be an estimate of a tenth of the standard deviation using the inter-quartile range of the response variable y, expressed as:

$$\epsilon = iqr(y)/13.49, \tag{3}$$

where $iqr(y)$ is the inter-quartile range of y.

The parameter C controls the trade off between training error and model complexity, which was set to be an estimate

of the standard deviation of the response variable, expressed as:

$$C = iqr(y)/1.349. \quad (4)$$

γ is a free parameter used in the radial kernel. The radial basis function kernel, or RBF kernel on two samples x_i and x_j is defined as

$$K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2). \quad (5)$$

The value of γ was optimized by the heuristic procedure using sub-sampling [26].

IV. Deep Learning Model

Considering temporal dependencies in workload sequence data, a workload forecasting model based on LSTM, a special kind of RNN, has been developed. The RNNs, derived from feedforward neural network, use memory to process sequence signals, which exhibit temporal dynamic behavior by connecting nodes to form a directed graph along a temporal sequence [27], [28], [29]. The LSTM, proposed by Hochreiter and Schmidhuber in 1997, unlike standard feedforward neural networks, has feedback connections, which allows the LSTM to process not only single data point, e.g. image, but also an entire sequence of data, e.g. speech or video [28], [30].

Similarly to the cascade SVR model, the average hourly workload values were used to train the LSTM model. The data given in the challenge were separated into two parts, one was used for training the LSTM networks, and the other for the purpose of validation in order to prevent overfitting:

- training set: the 10k time series involved in testing;
- validation set: the time series selected from the other 14k sequences based on the following rules: having less missing values and closer to the 10k testing sequences.

The data were standardized to have zero mean and unit standard deviation.

Due to the limited computation resource available for training sophisticated deep networks with multiple layers, just the data in the last 4-8 weeks were used in training and validation. The length of the input sequence was dependant on the size of the network, which was fixed in one LSTM network. The LSTM model was trained with sequence-input-sequence-output mode. The LSTM models have multiple LSTM-layers ranging from 2-5. Each layer has different number of hidden neurons, ranging from 128-640.

V. Hierarchical Linear Weighted Ensemble

In machine learning, ensemble of multiple independently trained models is expected to perform better than any base model by combining the advantage of base models and diluting their self-errors. In our ensemble model, the base models were linearly combined with different weights to yield final output, where the weights were estimated by linear regression. Note that only the deep models were used as base models since they gave highest preliminary scores. The dataset employed to train the linear regression models is the same as that used to train the cascade SVR model. A set of weights were trained

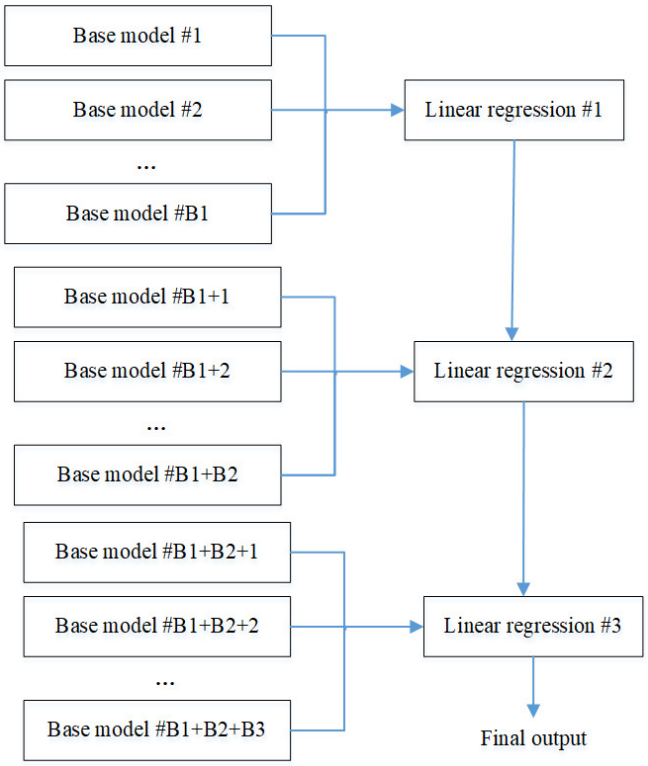

Fig. 2. Flowchart of hierarchical linear weighted ensemble.

for each of the 168 hours on the predictions from the base models, based on which the final output was generated.

When we observed the public scores of the solutions from the individual base models and the weights generated from the linear regression, it was found that some solutions with high scores got very low weights indicating that the importance of these solutions has been weakened, partially due to the variation of the given and unknown data. To address this issue, a hierarchical linear regression that combined various individual models in different stages has been developed. An example structure is shown in Fig. 2, where there are 3 linear regression stages, having B1, B2, and B3 base models, respectively. The B1 base models are firstly linearly combined, the output of which is then combined with the additional B2 base models, and similarly, the output from the second stage is then combined with the other B3 base models to yield the final output. The base models are arranged in ascending order of their public scores, e.g. the score of the model indexed B2+1 is higher than those of the models indexed from B1+1 to B2, by which the models with higher scores are combined in later stages so that the high-scored models are likely to have more priority in combination.

VI. Experiment Results

In this section, the performance of these 3 models is compared by using the data provided in the challenge.

A. Results of SVR models

When the cascade shallow model was trained on partial of the given data, e.g. the training set 2, the preliminary R^2 score was 0.2153. This was increased to 0.2506 if both training sets were used.

B. Results from LSTM networks

We have trained various LSTM models using different network structures. The performance was rather different. The highest preliminary R^2 score was 0.2831, which was achieved from a LSTM network having 3 LSTM layers each with 336 hidden neurons.

C. Results for Linear weighted ensemble

The preliminary R^2 score from the hierarchical linear weighted ensemble model was 0.2990 when trained on partial data, i.e. training set 1, while it was increased to 0.3059 when the LSTM network was trained on both training sets.

D. Discussion

Although the preliminary R^2 score, which was assessed based on 10% of the testing data, from the cascade SVR model is lower than those from both single and ensemble of LSTM models, its final score evaluated on the full testing set is 0.2365 that is higher than the baseline and published top score being 0.2295 and 0.1630, respectively. This indicates that the cascade shallow model is robust to overfitting. Both single and ensemble of LSTM can achieve higher preliminary R^2 score while they are likely to fall into overfitting. This implies that suitable implementation of shallow models can outperform deep models.

VII. Conclusions

This paper addresses forecasting workloads of network devices from historical sequence data. Three machine learning models, which are cascade SVR-based shallow model, LSTM-based deep model, and hierarchical linear weighted ensemble model, have been developed and verified using the data provided in the FedCSIS'20 Challenge. The preliminary evaluation on R^2 scores achieved from the shallow, deep and ensemble models are 0.2506, 0.2831, and 0.3059, respectively. Both the single and ensemble of LSTM models achieved much higher preliminary scores than the SVR model, while the SVR is more robust to overfitting.

References

[1] FedCSIS 2020 Challenge: Network Device Workload Prediction, *https://knowledgepit.ml/fedcsis20-challenge/*.

[2] S. Di, D. Kondo, W. Cirne, "Host load prediction in a Google compute cloud with a Bayesian model," *Proc. of IEEE Int. Conf. on High Performance Computing, Networking, Storage and Analysis*, 2012.

[3] F. Benhammadi, Z. Gessoum, A. Mokhtari, "CPU load prediction using neuro-fuzzy and Bayesian inferences," *Neurocomputing*, vol. 74, pp. 1606–1616, 2011.

[4] J. Kumar, A. Singh, "Workload prediction in cloud using artificial neural network and adaptive differential evolution," *Futur. Gener. Comput. Syst.*, vol. 81, pp. 41–52, 2018.

[5] R. Calheiros, E. Masoumi, R. Ranjan, R. Buyya, "Workload prediction using ARIMA model and its impact on cloud applications' QoS," *IEEE Trans. Cloud Comput.*, vol. 3, no. 4, pp. 449–458, 2014.

[6] Z. Huang, J. Peng, H. Lian, J. Guo, and W. Qiu, "Deep recurrent model for server load and performance prediction in data center," *Complexity*, 2017.

[7] J. Kumar, R. Goomer, and A. Singh, "Long short term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters,". *Procedia Comput.(Elsevier)*, vol. 125, pp. 676–682, 2018.

[8] B. Song, Y. Yu, Y. Zhou, Z. Wang, and S. Du, "Host load prediction with long short-term memory in cloud computing," *The Journal of Supercomputing*, vol. 74, 6554–6568, 2018.

[9] B.E. Boser, I.M. Guyon, V. Vapnik, "A training algorithm for optimal margin classifiers," *Proceedings of the Annual Conference on Computational Learning Theory, ACM*, pp. 144–152, Pittsburgh, PA 1992.

[10] I. Guyon, B. Boser, and V. Vapnik, "Automatic capacity tuning of very large VC-dimension classifiers," *Advances in Neural Information Processing Systems 5*, pp. 147–155, Morgan Kaufmann Publishers, 1993.

[11] C. Cortes, and V. Vapnik, Support vector networks, Machine Learning, vol. 20, pp. 273–297, 1995.

[12] B. Schölkopf, C. Burges, and V. Vapnik, "Extracting support data for a given task," *Proceedings of First International Conference on Knowledge Discovery and Data Mining*, AAAI Press, 1995.

[13] B. Schölkopf, C. Burges, and V. Vapnik, "Incorporating invariances in support vector learning machines," *Artificial Neural Networks, Springer Lecture Notes in Computer Science*, Vol. 1112, pp. 47–52, Berlin, 1996.

[14] V. Vapnik, S. Golowich and A. Smola, "Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing," in M. Mozer, M. Jordan, and T. Petsche (eds.), Neural Information Processing Systems, vol. 9, MIT Press, Cambridge, MA., 1997.

[15] V. Vapnik and A. Chervonenkis, "Theory of Pattern Recognition" (in Russian), Nauka, 1974.

[16] V. Vapnik, "Estimation of dependences based on empirical data," Springer Verlag.

[17] V. Vapnik, "The Nature of Statistical Learning Theory," Springer, New York.

[18] B. Schölkopf, P. Simard, A. Smola, and V. Vapnik, "Prior knowledge in support vector kernels," *M.I. Jordan, M.J. Kearns, and S.A. Solla (Eds.), Advances in Neural Information Processing Systems 10*, MIT Press, Cambridge, MA, pp. 640–646, 1998.

[19] V. Blanz, B. Schölkopf, H. Bulthoff, C. Burges, V. Vapnik, and T. Vetter, "Comparison of view-based object recognition algorithms using realistic 3D models," *Artificial Neural Networks, Springer Lecture Notes in Computer Science*, vol. 1112, pp. 251–256, Berlin, 1996.

[20] B. Schölkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2758–2765, 1997.

[21] K.R. Muller, A. Smola, G. Ratsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, "Predicting time series with support vector machines," *Artificial Neural Networks, Springer Lecture Notes in Computer Science*, vol. 1327, pp. 999–1004, Berlin, 1997.

[22] H. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in Neural Information Processing Systems 9*, pp. 155–161, MIT Press, Cambridge, MA, 1997.

[23] M. Stitson, A. Gammerman, V. Vapnik, V. Vovk, C. Watkins, and J. Weston, "Support vector regression with ANOVA decomposition kernels," *Advances in Kernel Methods—Support Vector Learning*, MIT Press Cambridge, MA, pp. 285–292, 1999.

[24] A. Smola, and B. Schölkopf, "A Tutorial on Support Vector Regression," *STATISTICS AND COMPUTING*, vol. 14, pp. 199-222, 2003.

[25] D. Basak, S. Pal, and D. Patranabis, "Support Vector Regression," *Neural Information Processing – Letters and Reviews*, vol. 11, Non. 10, pp. 203-224, October 2007.

[26] fitrsvm: Fit a support vector machine regression mode, https://www.mathworks.com/help/stats/fitrsvm.html.

[27] S. Dupond. "A thorough review on the current advance of neural network structures,". *Annual Reviews in Control*, vol. 14, pp. 200–230, 2019.

[28] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[29] M. Miljanovic, "Comparative analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction," *Indian Journal of Computer and Engineering*, vol. 3, no. 1, 2012.

[30] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.