ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 10 (2020) No. 4, pp. 1-11
DOI:10.33168/JSMS.2020.0409

Neural Network Structure and Hyper-parameter Optimization for Network Traffic Prediction

Min-Su Seok, JinYeong Um

Department of Computer Science & Engineering, Dongguk University, Seoul, 04620, Republic of Korea.

aksen123@dongguk.edu, mog07@dongguk.edu +

Abstract. This study suggests methods and possibilities for predicting the changes in network traffic. Although various devices operate for different purposes using network technology, determining the cause of packet loss in an unmanaged network environment is difficult. Many existing studies suggest protocols to reduce packet loss or transmission delay in limited power or network resources. However, all devices must use the same protocol to take effect. This paper aims to reduce packet loss or transmission delay by predicting network changes using neural networks. To this end, predictability and prediction results are presented using actual network traffic data. STL decomposition is used to check whether a prediction is possible, and prediction is performed using a feed-forward neural network and LSTM. Furthermore, we conclude that the prediction using LSTM is the most accurate.

Keywords: Network Prediction, Internet of Things, Time Series decomposition, STL decomposition, Neural Network, LSTM, MLP

1. Introduction

As the number of operational Internet-of-Things (IoT) devices increases, problems such as transmission delays occur due to traffic exceeding the network bandwidth. The IoT devices operating within limited powers and over wireless network environments may have less time wasted due to retransmission of data. Thus, their power consumption may be reduced, and the life of the device may be increased. This paper studies how to analyze current network traffic characteristics and predict future network traffic changes to maintain optimal network conditions. Network raffic is determined by the amount of data measured in a real network, and the prediction method is predicted using a neural network.

2. Related Works

Data transmission over a network assumes data loss. For example, a bus-type wired network uses Carrier-Sense Multiple Access with Collision Detection (CSMA/CD) while a wireless network uses CSMA with Collision Avoidance (CSMA/CA) to prevent frame collision. Even if a frame collision occurs, re-collision that may occur in retransmission is prevented Binary Exponential Back-off (BEB). However, the time delay required for retransmission and frequent delays in devices using limited resources such as IoT devices degrade the network performance. A similar example is TCP (Transmission Control Protocol) Incast (Elie, K., et al (2007)). This is a problem that overflows the buffer performance of a network device by simultaneously transmitting distributed data in a single network in a clustering environment in which one data is distributed and stored. This causes packet collision or queuing delay, which degrades the performance of the TCP itself.

To reduce retransmissions that occur beyond the allowable bandwidth of such networks, techniques for predicting network traffic changes have been proposed. The Smart Experts for Network State Estimation (SENSE) using Exponentially Weighted Moving Average (EWMA) formula and machine learning (Yalda, E. et al. (2016)), and prediction using Multi-Layer Perceptron (MLP) (Tiago, O. et al. (2014)) are presented in the figures below.

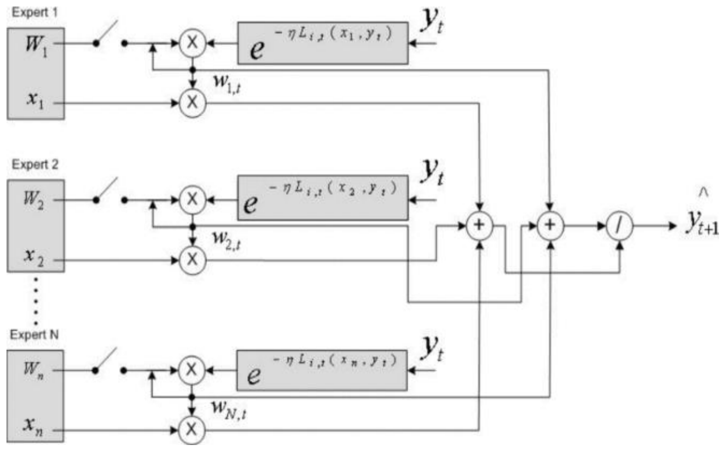

Fig. 1: SENSE Machine Learning Flowchart (Yalda, E. et al. (2016)).

Figure 1 presents that (1) the SENSE arranges EWMAs marked as Smart Experts in parallel, and (2) dynamically determines *Wi,t*, the importance of the EWMA, according to the prediction accuracy of each EWMA. Among them, (3) the prediction of various EWMA formulas are evaluated and the method of dynamically increasing the importance of the most accurate EWMA is determined. However, SENSE has issues because it is necessary to statically determine the values of operating parameters such as the number of EWMAs and the learning rate, regardless of the network state change.

The prediction paper using MLP compares the performance of predicting network traffic using Stacked Autoencoder and Convolutional Neural Network (CNN), which are widely used to extract the characteristics of the original data. Stacked Autoencoders require a similar computational amount to CNN, but CNN with higher accuracy with lower computations are more advantageous during training. However, since CNN operates with fixed parameters, it is difficult to flexibly cope in environments where network traffic changes frequently.

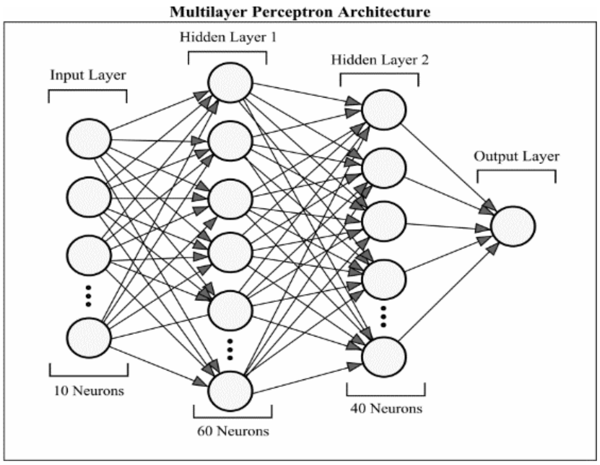

Fig. 2: MLP network prediction algorithm structure (Tiago, O. et al. (2014)).

It should be determined whether making predictions is possible. Even if the training set predictions are very high using machine learning or neural networks, it can also be predicted low by the actual test set. This is called Overfitting (Ian, G. et al. (2016)). Conversely, the lack of training set may make it difficult to predict. This is called Underfitting (Ian, G. et al. (2016)). This problem can increase the predictability of an actual test set by providing an additional training set, but it is difficult to know how much training set is needed before proceeding with the actual learning.

Therefore, the network traffic is decomposed into time series components to determine whether it is predictable. Among the various methods for decomposing time series data, Seasonal and Trend decomposition using Loess (STL) decomposition (Robert, B. C. et al. (1990)) has a strong performance in decomposing time series data into seasonal component and trend-cycle. If the remainder component is too large excluding seasonal component and trend-cycle, future predictions using the data will be difficult.

If it is determined that network traffic prediction is possible using STL, the actual prediction is performed using Long Short-Term Memory (LSTM), a kind of RNN (Recurrent Neural Network). The RNN is widely used for classification/prediction of time-series data types with high correlation between previous and current values because its calculation result of the hidden layer or output layer is used as another layer input part, unlike other neural network structures, so that the previous value can be used for new prediction. However, too many previous values cannot be used for prediction, so an appropriate previous value must be selected (Ian, G. et al. (2016)). The LSTM models (Sepp, H & Jurgen, S. (1997)) use three kinds of gates inside namely Input Gate, Output Gate, and Forget Gate.

The Input Gate determines how much the newly entered data will be reflected in the neural network. The Output Gate adjusts the output of the neural network. The Forget Gate determines the reflection rate for the previous data.

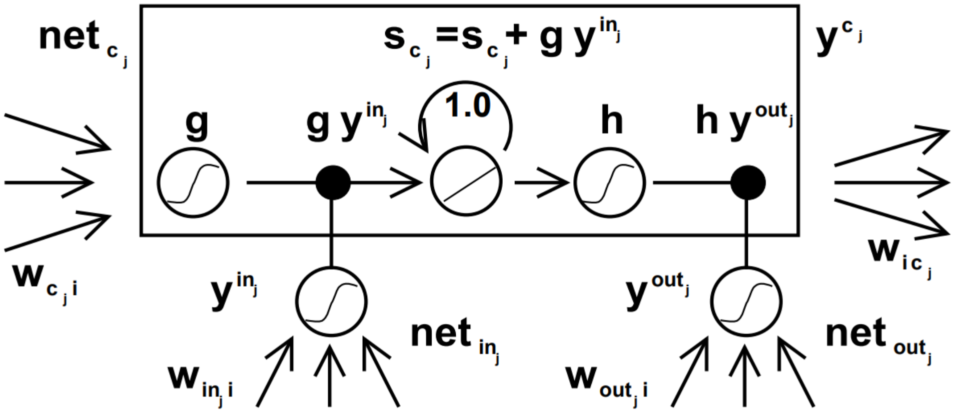

Fig. 3: LSTM structure (Sepp, H & Jurgen, S. (1997)).

In [Fig. 3], the multiplicative of variables corresponding to in_j is the Input Gate, and the multiplicative of variables corresponding to out_j is the Output Gate. The 1.0 loop in the center corresponds to the Forget Gate because the circuit is connected if *1.0* and the circuit is broken if *0.0*.

The number of gates varies depending on the experimental environment and requires more processing time than RNN. However, since the weight calculation for Input / Output / Previous data is effective compared to RNN, prediction accuracy is high. The prediction accuracy is compared using the Feed-Forward Neural Network (Tiago, O. et al. (2014)), which is a completely different neural network structure than LSTM.

3. Exploratory Data Analysis

The network traffic data used in the experiment uses The Time Series Data Library (TSDL) (Hyndman, R. J. (2020)). It provides various time series data, of which there are two types of network traffic time series data.

This paper uses (1) the A Dataset aggregated for 53 days by a private Internet Service Provider (ISP) connecting 11 European cities. In addition to (2) the B Dataset aggregated by the UK academic network for 70 days. These data are measured daily (A1D and B1D), hourly (A1H and B1H), and every 5 minutes (A5m and B5m). [Fig. 4] is a graph showing the A1H Dataset measured every hour, and [Fig. 5] is a graph corresponding to one week and one day as part of [Fig. 4].

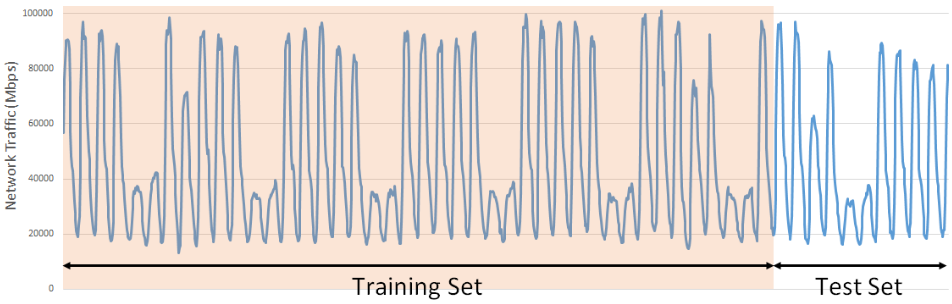

Fig. 4: A1H dataset graph measured every hour for about 55 days.

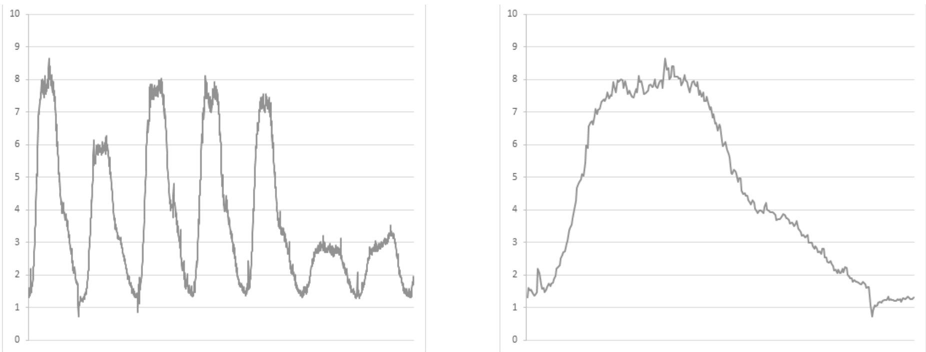

Fig. 5: one week, one daily graph for the A1H dataset.

[Fig. 6] shows the result of STL Decomposition for A1H Dataset using *R*. When analyzing the Seasonal component, the frequency is about 24 hours. There is a repetition of high traffic (five days) and low traffic (two days) in the trend-cycle. When seasonal component and trend-cycle are combined, the high traffic for five days and low traffic for two days are regularly repeated. [Fig. 7] shows the result of STL Decomposition for one week corresponding to [Fig. 6]. The seasonal component is similar to [Fig. 7], but the trend-cycle lacks meaningful information. By this method, it can be proved that A1H Dataset is a predictable Dataset.

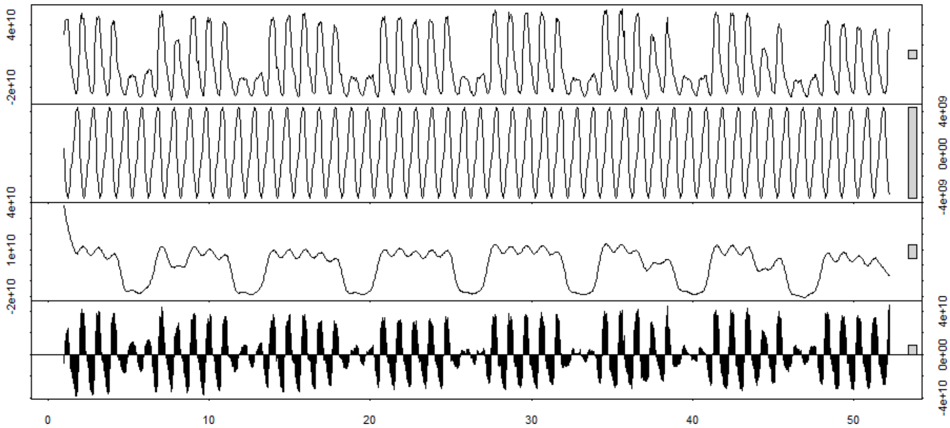

Fig. 6: Time series decomposition of A1H Dataset.

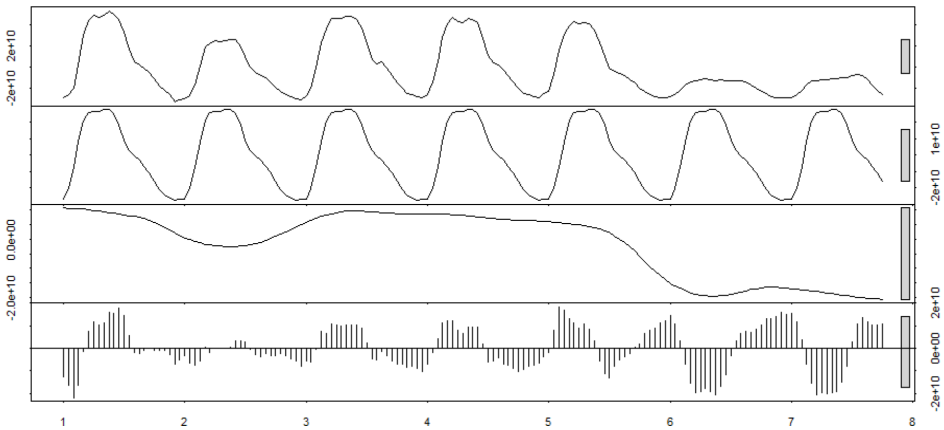

Fig. 7: Time series decomposition for one week of A1H Dataset.

4. Experiment

Matlab 2018b and Deep Learning Toolbox (Haytham, M. F. (2017)) are used as the program for network traffic prediction. Deep Learning Toolbox is a Matlab framework that has been changed from the existing Neural Network Toolbox to design and implement not only neural networks but also deep neural networks. The apps and plots in Matlab show the training process and visualize the training results.

The training set and the test set consisted of 80% and 20% of the entire datasets respectively. The LSTM parameters for predicting A1H are shown [Table 1]. The result and accuracy of the prediction using this are shown in [Fig. 8].

Table. 1: LSTM Parameter

Attribute	Value	Attribute	Value
number of Neuron for Layer 1	60	number of Neuron for Layer 2	40
Solver	adam	Max Epochs	300
Gradient Threshold	1	Initial Learn Rate	0.005
Learn Rate Schedule	piecewise	Learn Rate Drop Period	150
Learn Rate Drop Factor	0.2	Plots	training-progress

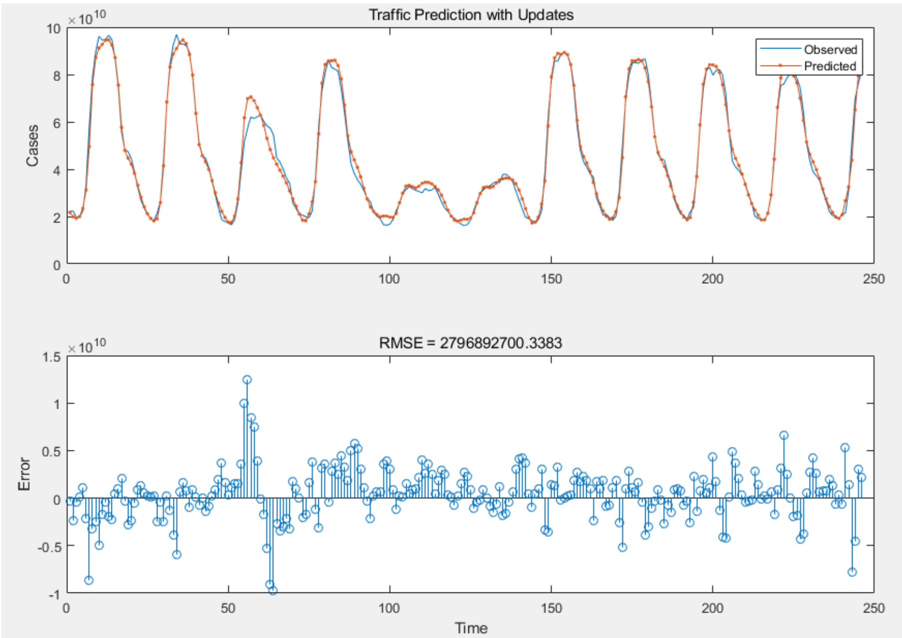

Fig. 8: A1H dataset prediction results using LSTM (Min-Su, S. et al. (2020)).

The reason for setting the LSTM neural network in [Table 1] is to compare it with the same setting as the feed-forward neural network. The feed-forward neural network setting is shown in [Table 2], and this setting is the most predictable feed-forward neural network structure studied in the paper (Tiago, O. et al. (2014)).

Table. 2: Feed-Forward neural network Parameter

Attribute	Value	Attribute	Value
number of Neuron for Layer 1	60	number of Neuron for Layer 2	40
number of Input	10	Max Epochs	300
Min Gradient	1e-9	Max Fail	100

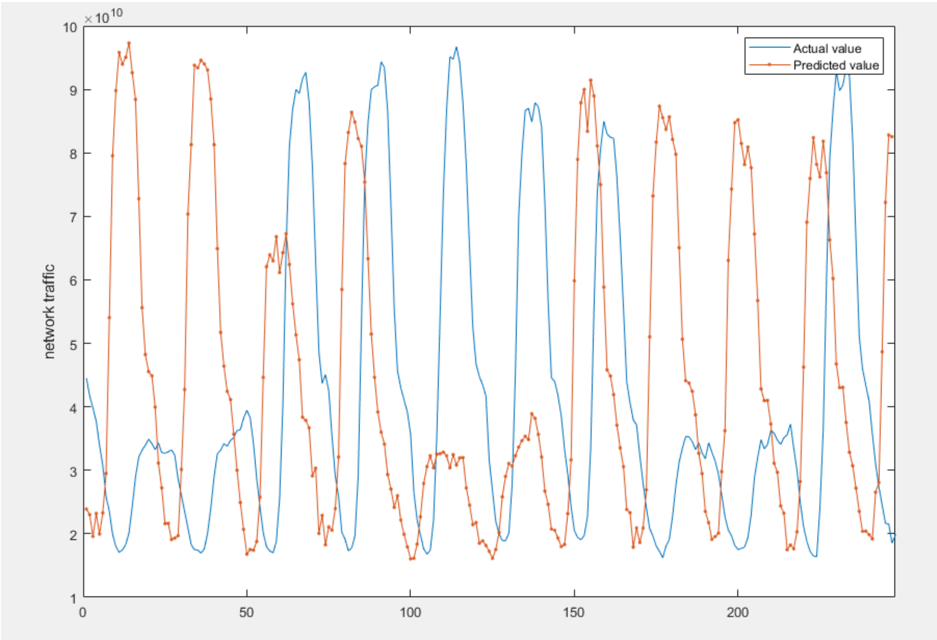

Fig. 9: A1H dataset prediction results using Feed-Forward neural network (Min-Su, S. et al. (2020)).

The accuracy of the two neural networks utilizes Root Mean Square Error (RMSE), which uses the difference between the actual and predicted values. Equation (1) is used to compare one actual and predicted value, and Equation (2) is used to compare several actual and predicted values.

$$\sqrt{E((\theta - \hat{\theta})^2)} \quad (1)$$

$$\frac{\sum_{i=0}^{n}(x_{1,i} - x_{2,i})^2}{n} \quad (2)$$

The lower the RMSE result means the higher accuracy. The experimental results show that the same network traffic dataset is more accurate than LSTM using 2.7969e + 10 and 3.5422e + 10 using Feed-forward neural network (Min-Su, S. et al. (2020)).

5. Conclusion and Future Work

If the limited network resources are divided and used equally, packet collisions between nodes can be prevented, which is similar to the study on Sensor MAC (S-MAC) (Wei, Y., et al. (2002)). However, it is not efficient because it uses network resources even if there is no data to transmit. A study by Qian, Z., et al. (2014) stated that only the node with the most power can communicate to reduce the total packet, reducing the collision rate, but it can be used only for nodes using the same protocol.

In a wireless network, collision is inevitable because multiple nodes share the same channel. In particular, the industrial, scientific, and medical bands used by most IoT devices have very high interference between channels.

To solve this problem, it is more effective to reduce the collision by predicting the time that the channel is not busy. In other words, if the node predicts the probability of data transmission (Hui, W., et al. 2015), collisions can be reduced, but this must use the same protocol. However, if it is predicted by using the entire network traffic, it is determined that it will improve network performance within the IoT environments that use different protocols by transmitting network data in lesser time than the same protocol or with lesser importance.

Finally, since the network change has a time series feature, then it is more accurate to predict using the LSTM of the recurrent neural network type than the MLP of the Feed-Forward neural network type.

Future studies will not predict the future network traffic based on the amount of data in the network but will increase prediction accuracy with the various parameters such as packet collision rate and latency. The study of Somayeh, K., et al. (2012) points out the need for accurate parameters to improve the performance of the scheduling algorithm in a grid computing environment. However, based on Zengkai, L., et al. (2014), this study intends to reduce the additional cost loss by predicting fault diagnosis using different pressure measurement values measured, as shown in this paper’s data analysis.

In particular, the network protocol can recognize packets for the same channel or the same encryption method, but it is difficult to recognize packets using different protocols, adjacent channels, and different encryption methods. Even if these problems are roughly recognized using signal strength, etc., it is intended to reduce power waste generated in IoT devices by utilizing avoidance of other channels. Similarly, it increases the reliability of the intrusion detection system by monitoring packets in different sections of the agent (Zhai, S., et al. (2014)).

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2017R1D1A1B03032736).

References

Elie, K., Vijay, V., Amar, P., David, G. A., Gregory, R. G., Garth, A. G., Srinivasan, S. (2007). On Application-level Approaches to Avoiding TCP Throughput Collapse in Cluster-Based Storage Systems. *Proceedings of the 2nd international Petascale Data Storage Workshop (PDSW '07)*. Reno, NV.

Yalda., E., Jong-Suk, A. & Katia, O. (2016). Smart Experts for Network State Estimation. *IEEE Transactions on Network and Service Management*, 13(3), 622-635.

Tiago, O., Jami, B., & Alexsandro, S. (2014). Multilayer Perceptron and Stacked Autoencoder for Internet Traffic Prediction. *11th IFIP International Conference on Network and Parallel Computing (NPC)*, 61-71.

Ian, G., Yoshua, B. & Aaron, C. (2016). *Deep Learning*. MIT Press

Robert, B. C., William, S. C., Jean, E. M. & Irma, T. (1990). *STL: A Seasonal-Trend Decomposition Procedure Based on Loess*, 6(1), 3-73.

Sepp, H & Jurgen, S. (1997). Long Short-term Memory. *Neural Computation*, 9(8), 1735-1780.

Hyndman, R. J. (2020). Time Series Data Library, https://datamarket.com/data/list/?q=provider:tsdl.

Haytham, M. F. (2017). MatDL: A Lightweight Deep Learning Library in MATLAB. *Journal of Open Source Software*, 2(19).

Min-Su, S. & JinYeong, U. (2020). Neural Network based Network Traffic Predictability for IoT Environment. *International Journal of Internet of Things and its Applications*, 4(1).

Wei, Y., John, H. & Deborah, E. (2002). An Energy-Efficient MAC Protocol for Wireless Sensor Networks. *Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies,* June23-27. New York. USA.

Qian, Z., Wen, H. & Guangsheng, F. (2014). An Improved ARP Protocol for Enhanced Lifetime in Wireless AD Hoc Networks. *International Journal of Grid and Distributed Computing*, 7(2), 173-184.

Hui, W., Xuefeng, L. & Yunfeng, W. (2015). A New MAC Protocol PB-TDMA for WSNs Based on Event-driven. *International Journal of Security and Its Applications*, 9(11), 55-62.

Somayeh, K., Saeed, J. & Nasrolah, M. C. (2012). Predicting Job Wait Time in Grid Environment by Applying Machine Learning Methods on Historical Information. *International Journal of Grid and Distributed Computing*, 5(3), 11-22.

Zengkai, L., Yonghong, L., Baoping, C., Dawei, Z. & Junlei, L. (2014). Fault Diagnosis of Subsea Blowout Preventer Based on Artificial Neural Networks. *International Journal of Security and Its Applications*, 8(2), 49-58.

Zhai, S., Hu, C. & Zhang, W. (2014). Multi-Agent Distributed Intrusion Detection System Model Based on BP Neural Network. *International Journal of Security and Its Applications*, 8(2), 183-192.