

# Neural Network based Network Traffic Predictability for IoT Environment

Min-Su Seok<sup>1</sup> and JinYeong Um<sup>2\*</sup>

<sup>1,2</sup> Dept. of Computer Science & Engineering, Dongguk Univ., Seoul, Republic of Korea

<sup>1</sup>aksen123@dongguk.edu, <sup>2</sup>mog07@dongguk.edu

## Abstract

*The purpose of this paper is to predict of compare network changes using neural networks. Many devices operate for various purposes using network technology, but unmanaged network environments cause packet loss for a variety of reasons. This packet loss causes data retransmissions and delays, and in environment where limited battery or limited network resources must be utilized, various problems can occur. Experiments show how neural networks can be used to predict network changes and whether they can predict changes in real networks. Predictability of Network Traffic indicates that LSTM (Long Short-Term Memory) is higher than MLP (Multi-Layer Perceptron).*

**Keywords:** Network prediction, Internet of things, Deep learning, LSTM, MLP, Matlab

## 1. Introduction

As more IoT (Internet of Things) devices are connected to the Internet and operate, problems such as transmission delays occur due to traffic exceeding network bandwidth. In particular, IoT devices operate in limited power and wireless network environments, so short data transfer times can reduce power consumption and extend device life. This paper analyzes current network traffic characteristics and predicts future network traffic changes in order to maintain optimal network conditions. In order to predict the network traffic, the amount of data measured in the actual network is determined and the prediction method is predicted using a neural network.

## 2. Related work

Methods for transmitting network traffic in an optimal state have already been applied. The BEB algorithm in IEEE 802.11 WLAN that reduces the probability of frame collisions with arbitrary latency. CSMA / CA (Carrier Sense Multiple Access with Collision Avoidance) or CSMA / CD (CSMA / Collision Detection) is attempts to transmit after sensing for some time to prevent frame collisions. The Congestion Control algorithm of the TCP (Transmission Control Protocol) takes into account the RTT (Round Trip Time) of data packets and response packets to determine the appropriate data size to be sent at one time and prevents packet loss due to too much data. However, because these methods change after a collision or loss, retransmissions and delays occur for data that has already been lost.

---

### Article history:

Received (December 13, 2019), Review Result (January 23, 2020), Accepted (March 5, 2020)

For fast data transfer, the delay caused by retransmission must be reduced. To reduce the loss incurred by exceeding the allowable bandwidth of the network, a technique is proposed to predict network traffic changes. SENSE (Smart Experts for Network State Estimation) using EWMA (Exponentially Weighted Moving Average) equations and Machine Learning [1], and prediction using MLP (Multi-Layer Perceptron) [2]. However, since it is a predictive technique in limited network situation, it is difficult to apply to real network.

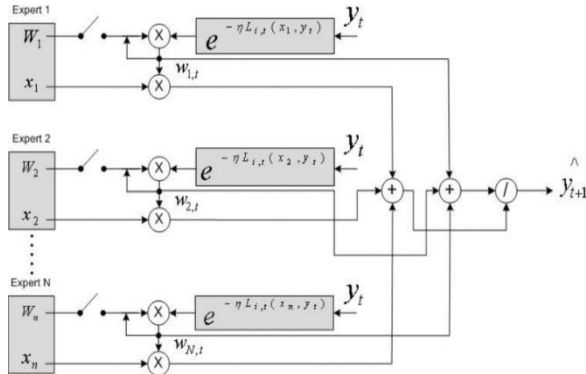


Figure 1. SENSE Machine Learning Flowchart [1]

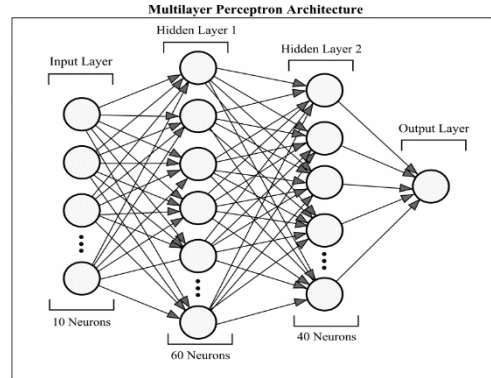


Figure 2. MLP network prediction algorithm structure [2]

This paper presents an experiment to predict network traffic changes. First, the network traffic is decomposed into time series components to determine whether they are predictable. Among the various decomposition methods, STL (Seasonal and Trend decomposition using Loess) decomposition [3] shows strong performance in decomposing time series data into Seasonal and Trend. If you decompose network traffic data into seasonality and trends and the remainder is too fluid, forecasting futures using that data can be difficult.

Actual predictions are conducted using LSTM (Long Short-Term Memory) [4], a type of RNN (Recurrent Neural Network). RNN are widely used for classification / prediction of high time series data types that are related to previous and now values. The reason is that unlike other neural network structures, the value of the hidden layer or the output layer is used as part of another layer input, which allows the previous value to be used for new prediction. However, too many previous values are not available for prediction, so you need to choose the appropriate number of previous values [5].

LSTM are calculated using three types of gates. The input gate determines how much of the data entered into the neural network. The Output Gate adjusts the output of the neural network. The Forget Gate determines the reflection rate for previous data. The number of gates depends on the experimental environment and requires more processing time than another RNN type.

### 3. Exploratory data analysis

The network traffic data used in the experiment utilizes The Time Series Data Library [6]. TSDL provides various time series data, and there are two main types of network traffic time series data. This paper uses data measured over 55 days from European private internet service provider. This data provides A1D measured daily, A1H measured every hour, and A5M measured every 5 minutes. [Figure 3] shows A1H dataset measured every hour. [Figure 4] shows the graph for one week and one day.

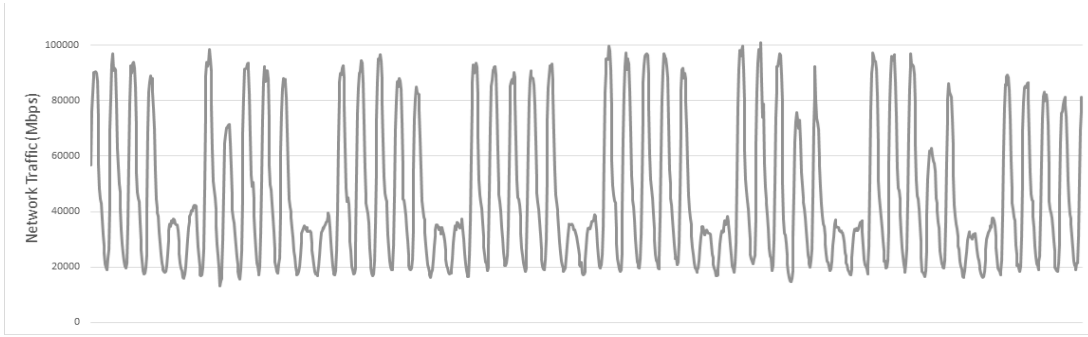


Figure 3. A1H dataset graph measured every hour for about 55 days

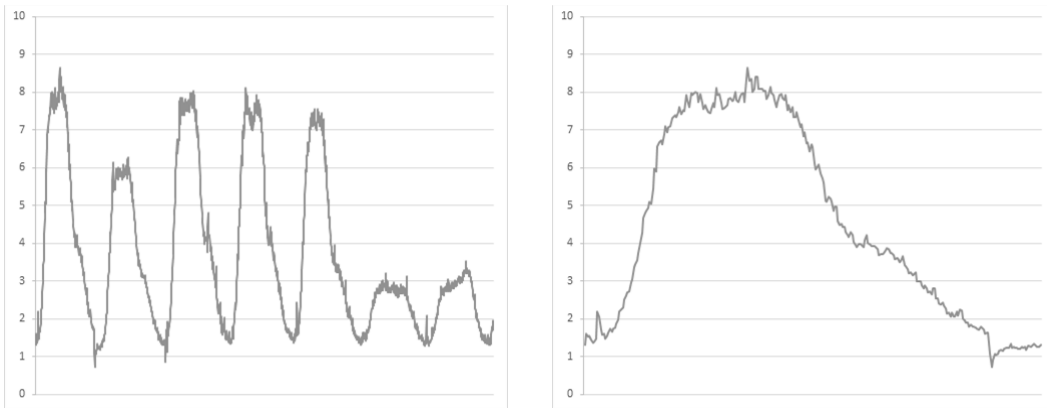


Figure 4. one week, one daily graph for the A1H dataset

[Figure 5] shows the decomposition result for the A1H dataset using R. The frequency is set to 24 hours, and one column of the X axis means one day. The overall trend is to repeat high traffic for five days and low traffic for two days. Seasonal has a period of about one day.

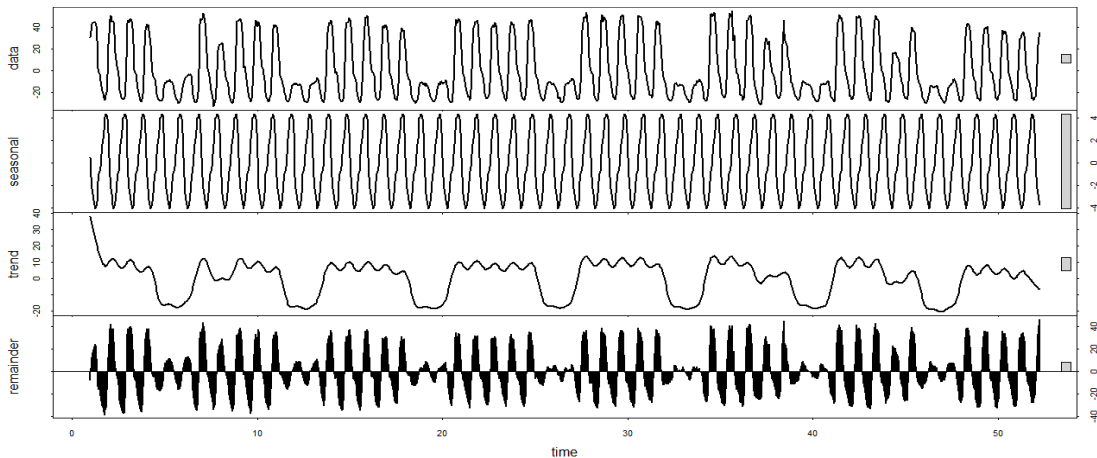


Figure 5. Time series decomposition of A1H Dataset

### 4. Experiment

Matlab 2018b and Deep Learning Toolbox [7] were used for network traffic prediction. Result and accuracy of A1H Dataset are shown in [Figure 6] and [Figure 7], LSTM configuration are shown in [Table 1].

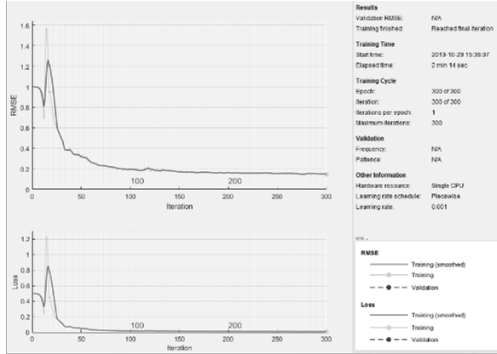


Figure 6. RMSE and Loss change with Epoch

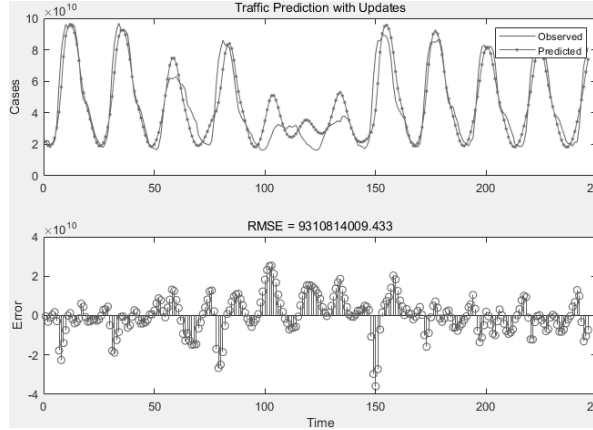


Figure 7. A1H dataset prediction results using LSTM

Table 1. LSTM parameter

Attribute	Value	Attribute	Value
number of Neuron for Layer 1	60	number of Neuron for Layer 2	40
Solver	adam	MaxEpochs	300
GradientThreshold	1	InitialLearnRate	0.005
LearnRateSchedule	piecewise	LearnRateDropPeriod	150
LearnRateDropFactor	0.2	Plots	training-progress

To compare prediction accuracy for LSTM, we used Feed-Forward neural network. Feed-Forward neural network configuration is shown in [Table 2].

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	MaxEpochs	300
Min Gradient	1e-9	Max Fail	100

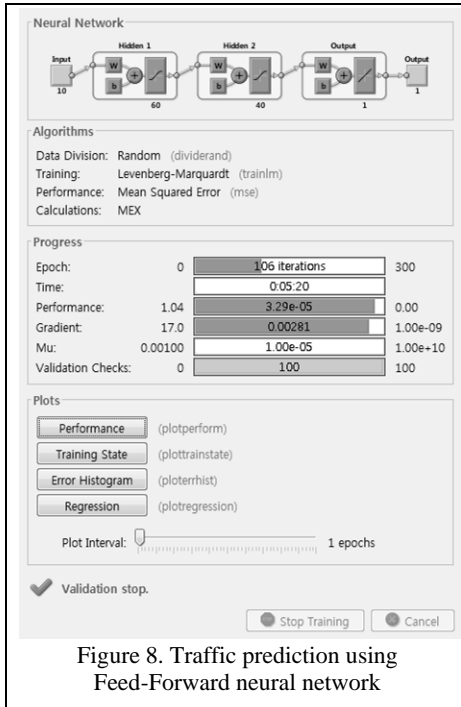


Figure 8. Traffic prediction using Feed-Forward neural network

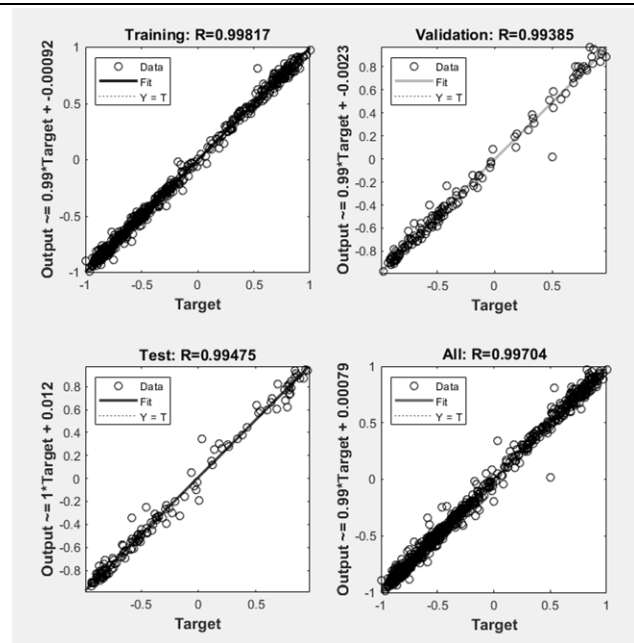


Figure 9. Regression analysis of predicted data using Feed-Forward neural network

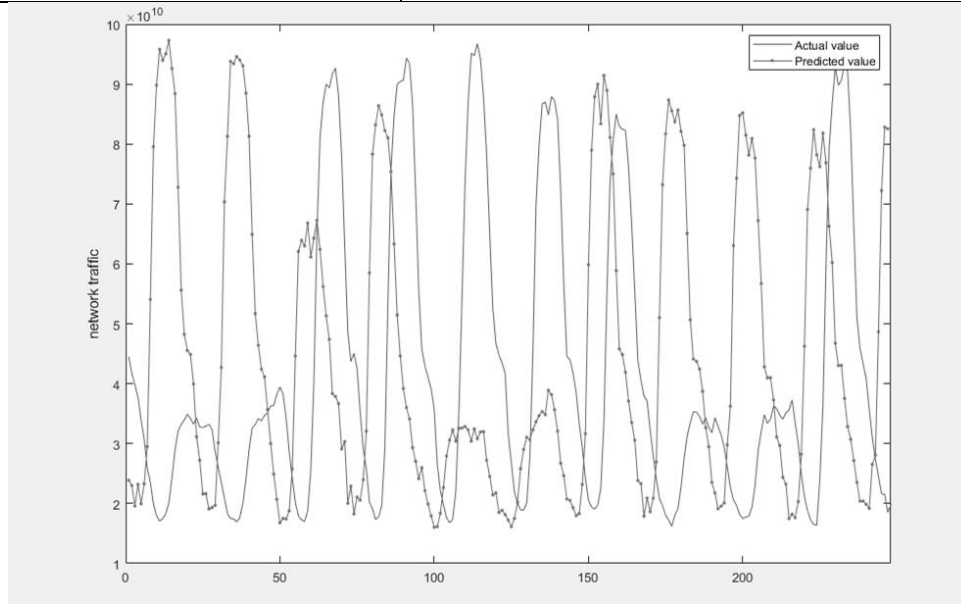


Figure 10. A1H dataset prediction results using feed-forward neural network

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 RMSE result means higher accuracy. 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.

## 5. Conclusion

If all nodes share the same network resource, data collision can be prevented. Similarly, there is S-MAC (Sensor MAC) [8]. However, even if there is no data to transfer, it is not efficient because it is used.

In other words, a wireless network has many nodes sharing the same channel, so collisions are unavoidable. In particular, ISM band (Industrial, Scientific and Medical band) used by most IoT devices have severe interference between channels.

To solve this problem, it is more effective to predict the time when the channel is not busy to reduce the collision. To this end, this paper uses past network changes to determine if the network traffic time is predictable. Using this, it is determined that the transmission of non-real-time or non-critical data in low traffic time will improve IoT performance.

Finally, since network changes have time series characteristics, it is more accurate to predict using LSTM in the form of RNN than MLP in the form of Feed-Forward neural network.

## 6. Future work

Future research will not predict the future network traffic based on the amount of data in the network, but rather increase the accuracy of prediction with various parameters such as packet collision rate and latency.

In addition, network protocols can recognize packets for the same channel or the same encryption scheme, but packets that use different protocols, adjacent channels, or different encryption schemes are difficult to recognize. In other words, even if the data reception signal strength is high, the collision rate is also high, thereby avoiding the waste of power generated by IoT devices by avoiding other channels.

## Acknowledgements

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

- [1] Yalda E., Jong-Suk A. and Katia O., "Smart experts for network state estimation," IEEE Transactions on Network and Service Management, vol.13, no.3, pp.622-635, (2016) DOI: 10.1109/TNSM.2016.2586506
- [2] Tiago O, Jami B. and Alessandro S., "Multilayer perceptron and stacked autoencoder for internet traffic prediction," 11th IFIP International Conference on Network and Parallel Computing (NPC), pp.61-71, (2014) DOI: 0.1007/978-3-662-44917-2\_6
- [3] Robert B. C., William S. C. Jean E. M, and Irma T., "STL: A seasonal-trend decomposition procedure based on loess," vol.6, no.1, pp.3-73, (1990)
- [4] Sepp H and Jürgen S., "Long short-term memory," Neural Computation, vol.9, no.8, pp.1735-1780, (1997) DOI: 10.1162/neco.1997.9.8.1735
- [5] Geon-Myeong L., "Artificial intelligence: from turing test to deep learning," Saeng Neung Publish, (2018)

- [6] Rob J. H., Data Market, <https://robjhyndman.com/hyndsight/tsdl/>, Jan 1, **(2020)**
- [7] Haytham M. F., “Mat DL: A lightweight deep learning library in MATLAB,” *Journal of Open Source Software*, vol.2, no.19, **(2017)** DOI: 10.21105/joss.00413
- [8] Wei Y., John H., and Deborah E., “An energy-efficient MAC protocol for wireless sensor networks, Proceedings,” *Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, June 23-27; New York, USA, **(2002)** DOI: 10.1109/INFCOM.2002.1019408

***This page is empty by intention.***