Industrial Wi-Fi Traffic Forecasting Using Diverse Time-Series Models

Minjun Choi†, Jae Hong Shin*, and Jeongyeup Paek†

† Department of Computer Science and Engineering, Chung-Ang University, Seoul, Republic of Korea

* Manufacturing SW Platform R&D Team, Hyundai Motor Group, Uiwang-si, Gyeonggi-do, Republic of Korea

alswns2631@cau.ac.kr, shjh@hyundai.com, jpaek@cau.ac.kr

***Abstract*—Smart factories cannot be detached from wireless networks, which have become essential elements for real-time communication and control of various industrial devices. Accurate traffic prediction is indispensable for managing network resources efficiently, enabling proactive resource allocation, congestion avoidance, and load balancing. In this paper, we conduct an evaluation based on actual measurement data obtained from Wi-Fi Access Points (APs) and Automated Guided Vehicles (AGVs) in a real automobile manufacturing factory. We predict the AP's traffic volume using various deep learning models, including Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM), Transformer, and Graph Neural Network (GNN) architectures, for future traffic forecasting. Our results show that the independent models, which are trained for each AP, effectively capture individual traffic patterns.**

***Index Terms*—Time Series Forecasting, Traffic Prediction, Deep Learning, Wi-Fi, Wireless Network**

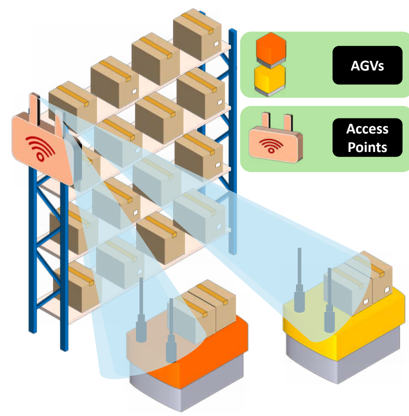


Fig. 1: Use cases of AGVs. AGVs are used to carry heavy objects in the factory. AGVs are controlled by a wireless network, and they communicate with APs using Wi-Fi 6.

I. Introduction

The evolution of wireless communication technologies, such as Wireless Local Area Networks (WLAN) and cellular networks, has enhanced flexible connectivity. This improves our convenience in daily life and boosts productivity in the industrial domain. In smart factory environments, autonomous transport systems such as AGVs and Autonomous Mobile Robots (AMRs) are used to carry heavy objects. Many aforementioned machines are controlled through wireless networks such as Wi-Fi and private 5G. Fig. 1 illustrates the AGV's operation scenario in a smart factory. AGVs are used to transport heavy objects, such as car parts, within the factory. Efficient management of wireless network resources is essential for reliable and effective factory operation, and to ensure the stable operation of these machines.

For intelligent network management, many network monitoring tools and control platforms, such as Cisco Meraki Go [1], Ruckus SmartZone [2], are widely used. Recently, many machine learning (ML) and deep learning (DL) techniques have been increasingly applied to wireless network management systems. These are being actively researched to improve resource allocation of wireless networks and optimize network performance. Among these, network traffic prediction has emerged as a key technology for enabling more efficient management of wireless network resources. The predicted traffic values can be utilized in various decision-making processes, for example, load balancing and resource scheduling. These will contribute to more reliable network operations, which will lead to productivity improvement in smart factories.

Various types of research related to network traffic prediction have been conducted, particularly focusing on non-industrial wireless cases [3], [4]. In this paper, we focus on traffic prediction in industrial Wi-Fi with various deep-learning models. Our contributions are as follows:

- We present a study based on a real industrial dataset collected from an actual automobile manufacturing factory environment. These datasets include traffic data from Wi-Fi Access Points (APs) and dmesg log data from AGVs.
- We conduct evaluations to predict future traffic volume using various deep learning models, and analyze the performance of these models in terms of various evaluation metrics (see §IV).

We aim to provide insights into the practical predictability and applicability of ML models for traffic forecasting.

This work was supported by the Hyundai Motor Group, also by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00359450), and also by the Institute of Information & Communications Technology Planning & Evaluation(IITP)-ITRC(Information Technology Research Center) grant funded by the Korea government(MSIT)(IITP-2025-RS-2022-00156353).

J. Paek is the corresponding author.

II. Related Work

Sone *et al.* [4] predicts the traffic load at an AP using real measurements from enterprise Wi-Fi network. They used probabilistic models such as Seasonal AutoRegressive Integrated Moving Average (SARIMA) and Holt-Winters. They also proposed a method using a time series forecasting model such as LSTM, Gated Recurrent Unit (GRU), and one-dimensional Convolutional Neural Network (1D-CNN). They further enhance prediction accuracy through a combined architecture that incorporates spatial feature extraction via 2D CNNs and temporal characteristics using Recurrent Neural Network (RNN)-based models such as LSTM and GRU. Shaabanzadeh *et al.* [3] updated the method using more models such as AutoRegressive Integrated Moving Average (ARIMA) and Transformer models.

However, most prior studies on traffic prediction are for non-industrial environments. Moreover, they typically use coarse-grained sampling intervals (e.g. 15 minutes), whereas our study employs fine-grained measurements collected every 90 seconds. To the best of our knowledge, there is a lack of research that applies traffic prediction techniques to real-world factory Wi-Fi environments. We address this gap by conducting an evaluation using actual data collected from an operational factory. Our goal is to provide some insights that can support factory network management and contribute to more stable and efficient industrial operations.

III. Dataset & Characteristics

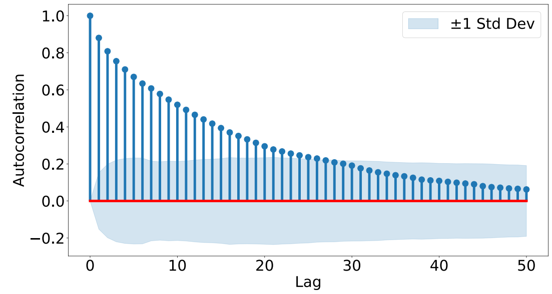
We use a dataset collected from an actual car manufacturing environment. In factory environments, AGVs transport massive objects and communicate with APs using the 802.11ax (Wi-Fi 6) standard. There are a few tens of APs deployed in the factory, and several tens of AGVs that are connected to the APs. The dataset was collected in real-world production environments, where AGVs are carrying heavy materials as part of the production process. The dataset used in this study comprises two primary sources:

- **AP logs**: includes AP's traffic volume (rx, tx bytes), client number (AGVs number), and channel-related data (e.g. 90 seconds moving average airtime total and airtime busy).
- **AGV dmesg logs**: Handover scanning log and connections log generated by the AGV's operating system.

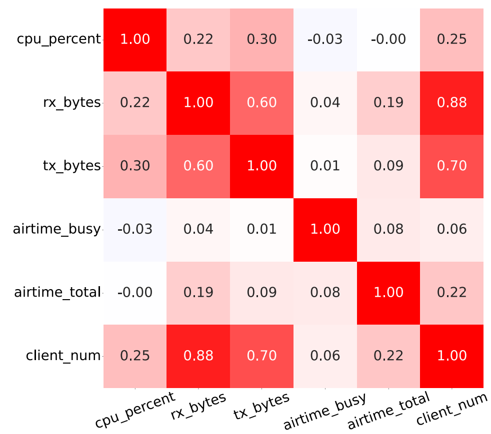
We collected the aforementioned data over a period spanning from December to March, and the AP log's sampling rate is 90 seconds. Because traffic variation is not significant during the non-operational hours, and this research focuses on the efficient operation of a manufacturing environment, our prediction target only includes the operational hours.

One key characteristic of the dataset is that AP's traffic (rx, tx bytes) does not exhibit strong long-term temporal dependencies. This is supported by the sample autocorrelation function (ACF) [5] plot, which shows that short-term patterns are more prominent. Fig. 2(a) shows the ACF plot. So, we focus on short-term traffic prediction. Specifically, we aim to predict one step ahead of traffic values(i.e., predicting traffic volume 90 seconds into the future). Another characteristic of the dataset is that the volume of the traffic is highly correlated with the number of clients connected to the AP. Fig. 2(b) shows the Pearson correlation of numeric values.

We use this dataset in our prediction model design and evaluations.



(a) ACF of AP's received traffic (rx bytes). High autocorrelation at low lags indicates short-term dependencies.



(b) Pearson correlation of AP02's numeric values. rx bytes, tx bytes, and client number are highly correlated.

Fig. 2: Autocorrelation of traffic and inter-feature correlation patterns in AP numeric value.

IV. Model Assessment & Benchmarking Methodology

In this section, we present the experimental setup and results of our traffic prediction evaluation. We evaluate the performance of various deep-learning models using the dataset described in §III. To preprocess the input features, we utilized several normalization schemes, including min-max scaling, Z-score normalization, and IQR-based scaling, for input feature preprocessing but failed to achieve a satisfactory result. The validation loss remained nearly constant without a simultaneous decrease, and the model outputs collapsed to a single value (typically the dataset's mean). Consequently, we applied a logarithmic transformation using $\log_{1p}(x)$ to the traffic features to reduce skewness and stabilize the variance. The logarithmic normalization effectively reduced multimodality and enhanced the stability of the distribution of traffic features.

We compare the following models in our traffic prediction evaluation:

- *MLP* [6]: A basic neural network that receives flattened features as input. It does not consider temporal or spatial dependencies, with lack of inductive bias.
- *LSTM* [7]: A RNN-based model that captures temporal dependencies by maintaining hidden and cell states across time steps.
- *LSTM Cell* [8]: A variant of LSTM that allows step-by-step processing over hidden and cell state updates at each time step.
- *Transformer* [9]: A sequence model that leverages self-attention mechanisms to capture dependencies across all time steps, suitable for modeling long-range patterns.
- *GAT + LSTM Cell*: A hybrid model combining GAT [10] and LSTM Cell. GAT is used to model spatial relationships between APs, followed by an LSTM cell to capture temporal dependencies.

In the case of GNN, we constructed a graph where each node represents an AP and the edges have two features: the Euclidean distance between APs and the handover count between APs due to the AGVs' movement.

The evaluation metrics we use to assess the performance of these models are as follows:

- *RMSE (Root Mean Square Error)*: Measures the square root of the average squared differences between predicted traffic value (y_i) and ground-truth traffic values ($\hat{y}_i$).

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- *MAE (Mean Absolute Error)*: Represents the average of the absolute differences between predicted and actual values.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- *R^2 Score (Coefficient of Determination)*: Indicates how well the regression model has predicted the target values. A value closer to 1 implies better performance.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

- *MAPE (Mean Absolute Percentage Error)*: Expresses prediction error as a percentage, calculated by averaging the absolute percentage differences between predicted and actual values. To avoid distortion caused by division-by-zero, data points with zero traffic are not included in the MAPE calculation.

$$\text{MAPE} = \frac{100\%}{n'}\sum_{i=1}^{n'}\left|\frac{y_i - \hat{y}_i}{y_i}\right|, \quad \text{where } y_i \neq 0$$

We also measure the average inference time per sample for each model to evaluate the computational efficiency of the models.

TABLE I: Performance comparison of a shared, per-AP models and graph-aware model including average inference time per sample.

Model	RMSE	MAE	MAPE	R^2	Time(ms)
Parameter sharing					
MLP	1.88	1.08	9.17%	0.856	0.1
LSTM	1.89	1.01	8.63%	0.854	1.0
LSTM Cell	1.88	1.06	9.03%	0.855	1.4
Transformer	1.89	1.08	8.63%	0.857	1.2
Per-AP (no parameter sharing)					
MLP	1.58	0.65	6.50%	0.897	1.5
LSTM	1.53	0.62	5.86%	0.903	3.8
LSTM Cell	1.53	0.61	5.94%	0.903	16.6
Transformer	1.56	0.70	6.70%	0.900	10.4
Graph aware model					
GAT+LSTM cell	1.53	0.62	6.00%	0.903	1.6

V. Evaluation Results

In this section, we explain our evaluation environment and the results of our traffic prediction evaluations.

A. Evaluation Setup

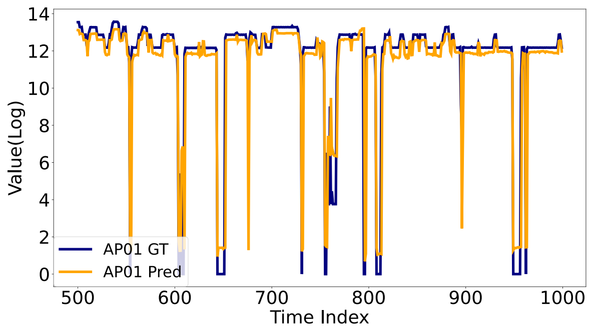
Evaluation was conducted on both the NVIDIA RTX 4090 GPU and RTX 4070 SUPER. We used sequence models using PyTorch [11] modules, such as LSTM and Transformer, and GNN models using PyTorch Geometric [12] modules. We used CUDA 11.8 with PyTorch 2.3.0 (+cu118) and PyTorch Geometric 2.6.1. We chose an input sequence length of 10 time steps. All the sequence models were configured with two layers and a hidden dimension of 128, and the dropout rate was set to 0.1. In the case of GAT + LSTM cell model, we used one GAT layer with four heads and one LSTM cell layer with a hidden dimension of 128.

All models were trained using the Adam optimizer with a learning rate of 1e-4 and weight decay of 1e-5. The batch size is set to 32, and training is performed for 200 epochs with early stopping based on validation loss (patience = 30).

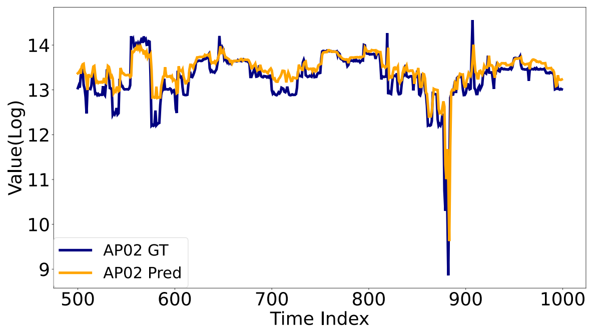
B. Evaluation Results

With the aforementioned setup environment, as described in V-A, we conducted evaluations to predict the future traffic values of dozens of APs using the above-mentioned models. Table I summarizes the averaged performance metrics obtained from traffic predictions across dozens of APs. Fig. 3 shows the ground truth and prediction results for two selected APs (AP01 and AP02) among dozens, focusing on the time index range of 500-1000.

We conducted evaluations under two different scenarios: shared model case and independent model cases. In the shared model case, a single model was trained to predict the traffic of all APs simultaneously. Otherwise, in the case of the independent cases, we trained a separate model, allowing each model to focus on AP's traffic characteristics. The results show that the independent model case outperforms the shared model



(a) AP01 Prediction



(b) AP02 Prediction

Fig. 3: (a) Traffic prediction for AP01, (b) AP02 (time index 500-1000). The x-axis represents time indices from a test dataset, showing a subset of the factory operation hours in March 2025. Each sample point is collected at 90-second intervals. The y-axis shows the log-transformed traffic values.

case. RMSE loss has reduced by approximately 0.3, MAE loss has reduced around 0.3 $\sim$ 0.4, and MAPE has improved by 2 $\sim$ 3% . The results indicate that the independent model case is more effective because it can consider AP's unique traffic patterns. However, the limitation of the parameter sharing model just considers the average and most dominant traffic patterns of all APs. But these models are cost-effective and fast at training and interference time because they share parameters across all APs

Adding GAT to the LSTM cell model shows a similar performance to the independent case of the LSTM cell. Fundamentally, a GNN extracts embeddings for all nodes, and thus, models can inherently consider the independent traffic patterns of each AP. Because we use a Discrete Time Dynamic Graph (DTDG) for GNN, it cannot capture the asynchronous behavior of AGVs, which leads to a similar performance to the independent case of the LSTM cell.

VI. Summary and Future Work

In this paper, we evaluated the ability to predict future traffic intensity on Wi-Fi 6 networks in an industrial environment. We used a dataset collected from an actual industrial environment, evaluated the performance of various deep learning models and conducted multiple evaluations to assess their performance. Our evaluation shows that maintaining an individual model for all APs is more effective than training a single model.

However, the current implementation does not yet perform full batch processing, so the evaluated results are quite conservative and could improve with proper batching. Another limitation of our study is restricted to discrete time series data, and we cannot reflect the asynchronous pattern of the AGV's movement. Future work will focus on addressing these limitations by using continuous time series data, such as handling continuous-time dynamic graphs, for example, TGN [13] and TGAT [14] for improving the prediction performance. Additionally, precise traffic forecasting can be utilized for anomaly detection by comparing the predicted value with the actual traffic volume, referred to as forecasting-based anomaly detection. This would be our next step.

Acknowledgement

We thank the Hyundai Motor Group's Manufacturing SW Platform R&D Team within the Manufacturing Solution Division for providing us with real AGV–AP Wi-Fi communication logs from their car manufacturing factory in operation[1]

References

[1] Cisco Meraki. (2025) Meraki Go Documentation. Accessed: 2025-07-04. [Online]. Available: https://documentation.meraki.com/Go

[2] Ruckus Networks, *RUCKUS SmartZone (LT-GA) Network Management Guide, Version 7.1.0*, CommScope, Inc., 2025, accessed: 2025-07-04. [Online]. Available: https://support.ruckuswireless.com/documents/4953-ruckus-smartzone-lt-ga-network-management-guide-7-1-0

[3] S. S. Shaabanzadeh and J. Sánchez-González, "A spatio-temporal prediction methodology based on deep learning and real Wi-Fi measurements," *Computer Networks*, vol. 250, p. 110569, 2024.

[4] S. P. Sone, J. J. Lehtomäki, and Z. Khan, "Wireless traffic usage forecasting using real enterprise network data: Analysis and methods," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 777–797, 2020.

[5] P. J. Brockwell and R. A. Davis, *Introduction to time series and forecasting*. Springer, 2002.

[6] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[7] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, 11 1997.

[8] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[10] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[11] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[12] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," *arXiv preprint arXiv:1903.02428*, 2019.

[13] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," *arXiv preprint arXiv:2006.10637*, 2020.

[14] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," *arXiv preprint arXiv:2002.07962*, 2020.

[1]Exact details are not disclosed due to corporate secrecy.