

Leveraging IoT and Weather Conditions to Estimate the Riders Waiting for the Bus Transit on Campus

Ismail Arai ¹

Information Initiative Center
Nara Institute of Science and Technology
Nara, Japan
ismail@itc.naist.jp

Ahmed Elnoshokaty

Computer Information Systems College of Integrated Science and Engineering
Northern Michigan University
Michigan, USA
aelnosho@nmu.edu

Samy El-Tawab

James Madison University
Virginia, USA
eltawass@jmu.edu

Abstract—The communication technology revolution in this era has increased the use of smartphones in the world of transportation. In this paper, we propose to leverage IoT device data, capturing passengers’ smartphones’ Wi-Fi data in conjunction with weather conditions to predict the expected number of passengers waiting at a bus stop at a specific time using deep learning models. Our study collected data from the transit bus system at James Madison University (JMU) in Virginia, USA. This paper studies the correlation between the number of passengers waiting at bus stops and weather conditions. Empirically, an experiment with several bus stops in JMU, was utilized to confirm a high precision level. We compared our Deep Neural Network (DNN) model against two baseline models: Linear Regression (LR) and a Wide Neural Network (WNN). The gap between the baseline models and DNN was 35% and 14% better Mean Squared Error (MSE) scores for predictions in favor of the DNN compared to LR and WNN, respectively.

Index Terms—IoT (Internet of Things), Transit Systems, Machine Learning (ML), Data Analytics, Intelligent Transportation Systems (ITS)

I. INTRODUCTION

For the Transit Bus Management System, data-driven fleet management strategies empowered by precise models that predict the number of passengers waiting at the bus stops are essential. On a university campus, students often have to wait in additional time when a packed bus arrives. To solve the problem, we have developed the Internet of Things (IoT) system [1] that estimates the number of passengers by analyzing passengers’ smartphones’ Wi-Fi frames. Even though the system can grasp the bus stop’s occasion, it still has not been spun out from a research prototype with the lack of feasibility in the real field. Other data sources captured from the bus (e.g., camera feed) can be considered ground truth [2].

This paper proposes a machine learning utilization technique that employs the IoT passenger counter system to provide the labeled training data. The learning phase inputs the weather data, transportation information, campus schedule, and the number of passengers waiting at the bus stop. After building the machine learning model, it can estimate the number of passengers without the IoT system. Our group has noticed a significant relationship between the number of

passengers waiting for the bus at a particular bus station and the weather conditions. Using these machine learning models, we can achieve a realistic fleet management system without having IoT sensors installed permanently at each bus stop.

The experiment took place in public transit systems at James Madison University (JMU) in Harrisonburg, Virginia, USA, in Spring 2017. With a month of data collected at seven bus stops, the Mean Squared Error (MSE) of Deep Neural Network (DNN), Wide Neural Network (WNN), and Linear Regression (LR) were 1.15, 1.34, 1.77, respectively. As a result of the experiment, we achieved the best MSE with the DNN.

II. RELATED WORK

Several researchers have used IoT in the intelligent transportation world. Applications inside cities such as Smart Parking, Bus Monitoring have been easier to deploy with the integration of IoT devices to sense data [3], [4]. At the same time, other highway applications (e.g., highway monitoring and incident detection) advanced with the use of IoT devices [5], [6]. The IoT devices’ role has improved with the advance of power batteries and reliability of IoT [7]. Our system depends on IoT devices to improve the public transit system’s quality of service [8]. Several researchers have integrated the power of communication and public transportation to improve service quality (e.g., ridership and waiting time in public bus transit) [9], [10]. Dunlap et al. [9] estimated passenger origin and destination (OD) information for transit lines using IoT sensors to collect Wi-Fi and Bluetooth beacon.

The rise of data-mining-based studies and machine learning techniques has recently improved research quality in many fields. In Intelligent Transportation, Lathia et al. proposed a machine learning technique to enhance the the passenger’s ticket choice by studying travel history patterns and mining the public transport fare data collected from the bus system [11]. On the other hand, Amato et al. studied car parking occupancy with deep learning techniques [12]. Tahere et al. predicted five crowding levels with rich data such as ridership 15 minutes ago [13]. Machine learning techniques and deep learning will open the door toward more improvement to big data and data-driven systems.

In this paper, we leverage machine learning models to study bus ridership in one of the USA’s college cities. We integrate

¹ Ismail Arai was also affiliated as a visiting scholar at James Madison University, Virginia, USA while working on parts of this paper.

several parameters such as class schedule and weather information to predict hourly passengers waiting at bus stops.

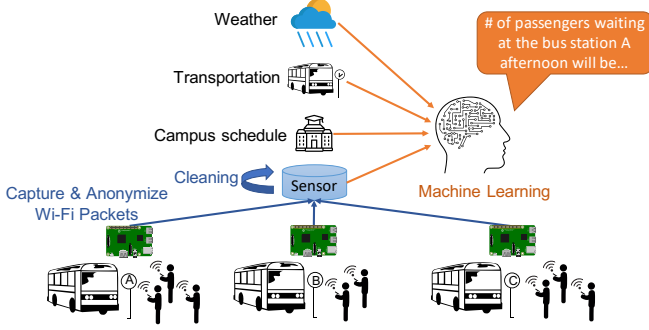


Fig. 1. System overview

III. FRAMEWORK

The proposed system estimates the number of passengers with a machine learning technique taking as an input anonymous and abstracted Wi-Fi capture data at each bus stop and weather data as shown in Figure 1. Some of the Wi-Fi captured data might correlate with the number of persons since they have a Wi-Fi enabled smartphone. On the other hand, we should keep in mind the privacy concerns during capturing Wi-Fi frames, including MAC addresses. Weather data is also informative for estimating the number of passengers. For example, students would not wait for a bus for several minutes on a sunny day. The research question is whether the weather data and the campus schedule data effectively work as an input to machine learning for estimating the number of passengers.

A. System Design and Rational

The proposed system is based on an IoT cloud computing model. The edge nodes are Raspberry Pi 3 (located at 7 different stations as shown in Figure 2) with a monitor mode enabled Wi-Fi interface that captures frames and uploads them to the cloud-based database. Since edge nodes do not require heavy computational need, it works with a solar panel and

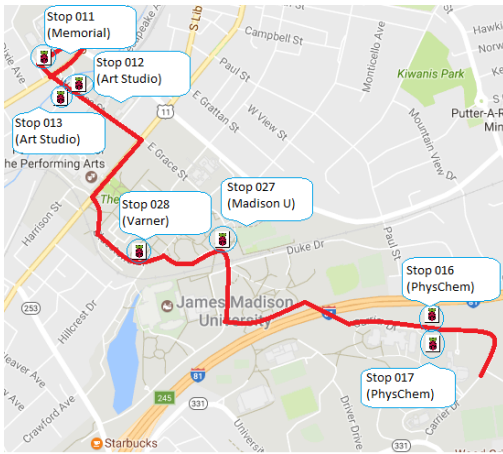


Fig. 2. Edge nodes located through the bus route

a rechargeable battery. These edge nodes' main functions are to capture the unencrypted frames, anonymize the MAC addresses with SHA-1, and upload the data to the cloud-based database. The other processes work on the cloud as follows:

- **Cleaning:** Filter only the passengers' smartphones frames. Many Wi-Fi frames have noises due to MAC randomization, smartphones passing by a bus stop, and devices in personal vehicles and buildings.
- **Learning:** Training the model by learning weather data, Wi-Fi sensor data, and some categorical information for estimating the number of passengers waiting for the bus at the bus stop.
- **Estimating:** Estimating the number of passengers from weather information and categorical information.

Our system makes sure that only light processing tasks are done on the edge nodes, while the heavy tasks of machine learning and computing are done on the cloud. We installed an IoT node at each bus stop to capture the Wi-Fi frames of passengers' smartphones. There are mainly two ways to set the IoT devices in the field. One is to install them inside the bus (hereafter call it mobile sensing), the other is to place them at each bus stop (fixed point sensing). Estimating the ridership is relatively easy with the mobile sensing against the fixed point sensing because the mobile sensing can track all the passengers inside the bus during driving. However, mobile sensing has no way to know the number of passengers waiting at the bus stop when the bus is not at the bus stop. The number of required devices is the same as the number of buses and bus stops with mobile sensing and fixed point sensing. The fixed point sensing is reasonable in our field because the number of bus stops is less than the buses.

TABLE I
WEATHER DATA

Name	Content
dt	Time of data calculation, unix, UTC
temp	Temperature
feels_like	Accounting for the human perception of weather
temp_min	Minimum temperature among sensors in city
temp_max	Maximum temperature among sensors in city
pressure	Atmospheric pressure (on the sea level), hPa
sea_level	Sea Level, meters
grnd_level	Ground Level, meters
humidity	Humidity, %
wind_speed	Wind speed. Unit Default: meter/sec
wind_deg	Wind direction, degrees (meteorological)
rain_1h	Rain volume for the last hour, mm
rain_3h	Rain volume for the last 3h, mm
snow_1h	Snow volume for the last hour in liquid state, mm
snow_3h	Snow volume for the last 3h in liquid state, mm
clouds_all	Cloudiness, %
weather_id	Weather condition id
weather_main	Rain, Snow, Extreme etc.
weather_description	Weather condition within the group

B. Data Preparation

As described in Figure 1, the proposed system employs four types of data for estimating the number of passengers: the weather data (Table I) obtained from OpenWeather [14],

TABLE II
RAW SENSOR DATA

Name	Type
Bus stop	String
Date and time in UTC	YYYY-MM-DD hh:mm:ss
MAC Address	6 Octet
Signal Strength (dBm)	Integer

TABLE III
THE NUMBER OF PEOPLE DATA

Name	Type
Bus stop	String
Date and time in UTC	YYYY-MM-DD hh:mm:ss
Count	Integer

the transportation data (e.g., the locations of the bus stops), the campus schedule (e.g., class schedule), and the number of people data converted from the raw data. Table II shows the specification of the raw data. The IoT sensor data is cleaned and transformed into the number of people data (Table III) to feed the machine learning. In the training phase, the machine learning model inputs all the data, while in the testing phase, it does not input the passenger count data since it is the expected output of the model.

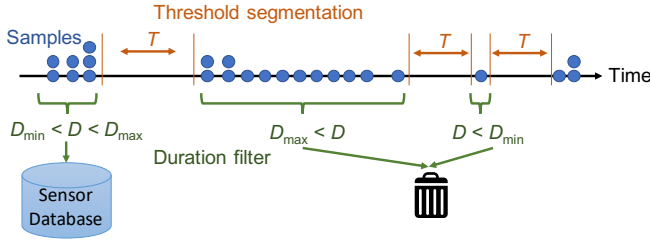
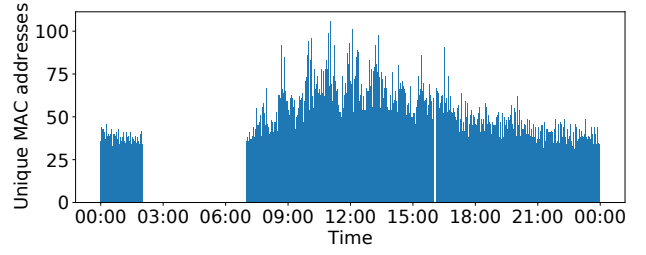


Fig. 3. Overview of the segmentation and the filtering

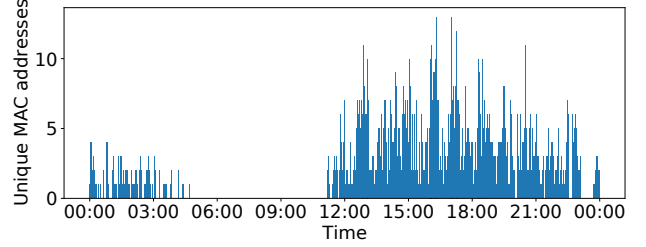
We have cleaned the data as follows:

- 1) **Random MAC Address:** For a privacy measure, iOS 8 and Android 8 started to provide a MAC randomization feature, randomizing a source MAC address when the device is trying to find a Wi-Fi network with a probe request frame. When we identify the devices with the probe request frame, the random MAC address will be noise. We have removed such frames by checking G/L and I/G bits in the MAC address.¹
- 2) **MAC Addresses observed at only one bus station:** The passengers' smartphone's Wi-Fi frames should be captured at two or more bus stations. If a MAC address is found at only one bus stop, we assume that a person only passed by the bus stop or a device is installed around the bus stop (e.g., Wi-Fi enabled PC in a building). We know that a passenger phone battery may die while riding the bus; we assume that this case can be ignored with big data.

¹Our data collection was done while iOS versions implementing randomization were not yet released.



(a) Raw data



(b) Cleaned data

Fig. 4. MAC addresses every minutes at 017 (PhysChem) on 4/21

- 3) **Segmentation and filtering based on the duration:** As shown in Figure 3, samples are split into segments with a given threshold time. If the segment's period is shorter than D_{min} or longer than D_{max} , it is discarded. The rest of the segments will be stored in the cloud-based database.
- 4) **Unrealistic RSSI:** Usually, the waiting passengers' length is within ten or more meters. In this condition, Received Signal Strength Indication (RSSI) will be in a certain range.

Figure 4 shows the result of cleaning the data of bus stop 017 (PhysChem) on April 21st in 2017. We used 2 and 30 minutes for D_{min} and D_{max} , respectively. And we assumed the RSSI range for capturing Wi-Fi frames from smartphones within 10 meters is between -30 dBm and -80 dBm. Against the raw data shows the unrealistic numbers of unique MAC addresses all over the time, the cleaned data shows like a transition of the number of passengers in a day. Though we have not confirmed the cleaned data's correctness due to the lack of the ground truth, it will be no problem because it is generally consistent with the trends we have seen visually at the bus stop.

To prepare machine learning data, we derived features (e.g., academic week) as a week of the semester from the date. Also, we derived the morning/evening feature from time. We further created dummy variables for all values of categorical features (e.g., weather_description and Bus_stop) in Tables I and II, respectively. For example, the weather_description having "Rain" and "Cloud" will be True/False values of weather_description_Rain, and weather_description_Cloud.

C. Machine Learning

For the Transit Bus Management System, data-driven fleet management strategies empowered by precise models that predict the number of passengers waiting at transit bus stops based

on real-time weather conditions are essential. For example, the system could suggest dispatching many buses to a particular route depending on the weather conditions, instead of having near-empty buses some days and full one's others. We highlight the following machine learning techniques: Linear Regression (LR) analysis is a predictive modeling technique that estimates the relationship between two or more variables. Recall that a correlation analysis does not assume the causal relationship between two variables.

$$Y_i = f(X_i, \theta) + \epsilon_i \quad (1)$$

where for k features and n data points, $i = 1, \dots, n$ and $\theta = (\theta_1, \dots, \theta_k)^T$ is the vector of the parameters to be estimated. The error ϵ_i are with a mean equal to zero and an unknown σ^2 . LR can be used when the features measured have a linear correlation with the dependent variable. Since not all features are having a linear relationship with the passengers' count to be predicted, it is expected that good results cannot be obtained using a plain LR model.

Classification and regression trees (CART) are binary decision trees. The tree is constructed by splitting the entire data into subsets by using all the independent variables. The goal is to produce terminal leaves that are as homogeneous as possible with respect to the target variable. Regression trees can be notable accurate in the case of nonlinear problems. For every node t , $1/n \sum_{i=1}^n (Y_i - \hat{Y}_i(t))^2$ is the node sum of squares. In other words, it is the total squared deviations of Y_i in t from their average. The regression tree is formed by splitting the nodes iteratively so that the decrease in $R(T)$ is maximized, where $R(T)$ sums up all the sums of squares within all the nodes.

Neural networks (NN) are models in which input features flow through hidden layers towards the output [15]. The neural network learns new feature spaces by first computing the affine (linear) transformations of the given inputs and then applying a nonlinear function (rectified linear unit ReLU), which will be the next layer's input. This process will continue just before the output layer when a linear transformation will be applied for predicting the hourly passenger demand. NN with no hidden layers can only learn how to solve linearly separable problems. NN with two or more hidden layers capture non-linearity is a natural aspect. NN with three hidden layers is usually considered a deep-neural network. For this study, we expect the performance of a deep neural network (DNN) model that consists of an input layer, three hidden layers, and an output layer to outperform LR, CART, and Neural Networks with one hidden layer (WNN).

In the first layer, layer 1 of the NN architecture, each neuron receives a set of X -values (numbered from 1 to n) from input vector X and computes the predicted \hat{Y} value. Vector X contains the weather's value and the bus features, for one example, from the training set. Each node in layer 1 has its own set of parameters, usually referred to as W (column vector of weights) and b (bias), as shown in Figure 5. In each iteration, the neuron calculates a weighted average of

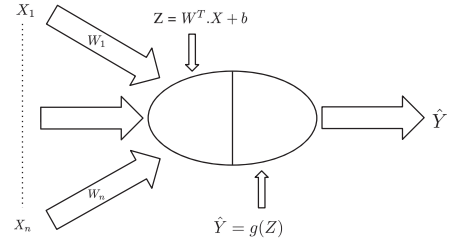


Fig. 5. Neural Network Neuron

TABLE IV
EXPERIMENTAL DATA SPECIFICATION

Name	Value
Date	From 2017-04-05 to 2017-05-04
Bus stops	7
Raw data records	210,161,203
Cleaned data records	9,981,883

the vector X , based on its current weight vector W . Then, it adds bias where weights and bias change during the learning process to minimize prediction error. Finally, the result of this calculation is passed through a ReLU.

The equation in Figure 5 used X and \hat{Y} , which are the column vector of features and the predicted value, respectively, for a node in layer 1. The X vector is, therefore, layer 0 (input layer). When switching to the general notation for layer k , like layer 2 and layer 3, we use $a^{[k-1]}$ and $a^{[k]}$, which are the input to the hidden layer k and the activated output predicted value of layer k , respectively (activation function g used is ReLU). Accordingly, for any hidden layer k with $n^{[k]}$ neuron, each neuron performs a similar calculation according to the following equations:

$$Z_i^{[k]} = W_i^T \cdot a^{[k-1]} + b_i \quad (2)$$

$$a_i^{[k]} = g(Z_i^{[k]}) \quad (3)$$

The activation function g for any hidden layer is ReLU, except for the output layer, which is a linear function to predict the hourly passenger demand. The model learns through iterations of forward-feeding and backward propagation from the input layer, hidden layers, to output and back. Back-propagation is seen as a common approach where random weights are assigned; the output seen is compared with the test data; the output error is calculated comparing the two (i.e., actual output vs. expected output in MSE loss function). The layer immediately closer to the output layer adjusts its weights, leading to weight adjustments in the subsequent inner layers until the error rate is reduced [15]. NN models extract features by weight allocation and weight decay through iterations of forward-feeding and backward propagation techniques.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

We have collected the experimental data for a month, as shown in Table IV. IoT devices were installed at the seven bus stops shown in Figure 2 from April 5th to May 4th in

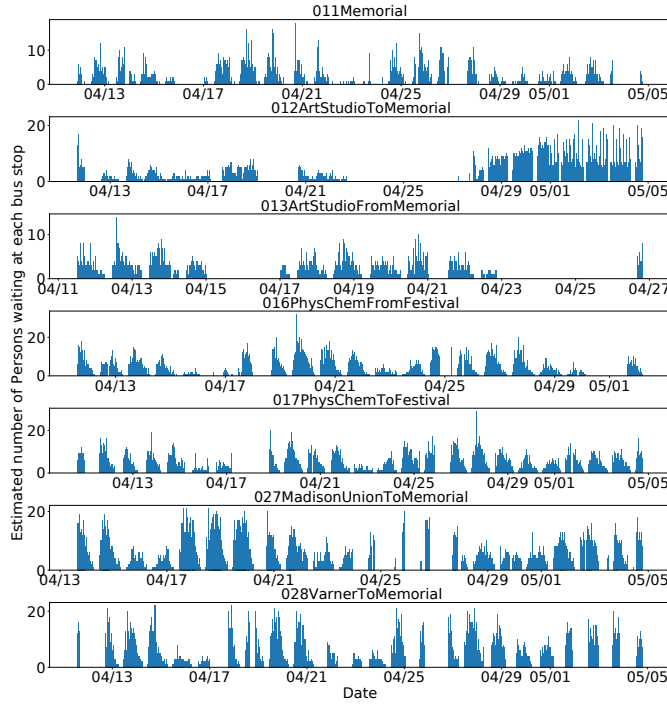


Fig. 6. Estimated numbers of passengers waiting at each bus stop with the cleaned data.

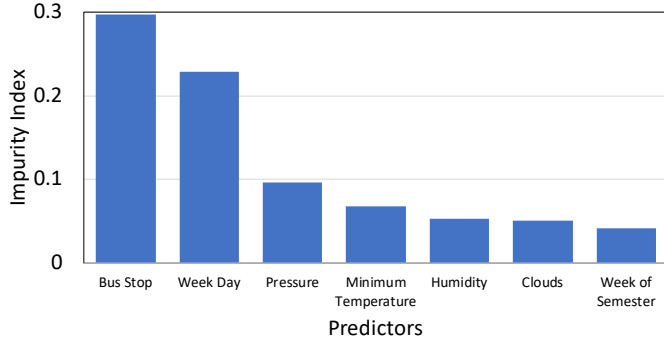


Fig. 7. Predictors Impurity Index

2017. Figure 6 shows the estimated numbers of passengers waiting at each bus stop with the cleaned data. The numbers of passengers were around 20.

The weather predictors (Table I) and the passenger counts (Table III) are recorded hourly and every second, respectively. We aggregated the passenger counts data on an hourly basis and averaged the reading for the predictors. We then merged the aggregated bus data with the weather data to have our granularity of analysis on the hourly level. We further normalized our data and derived new predictors like weekdays/weekends and morning/evenings. We also extracted the week of the academic semester from the date, which is shown to be informative predictive features, as in Figure 7.

After the data preparation and transformation stage, we benchmarked our DNN model (an input layer, three hidden

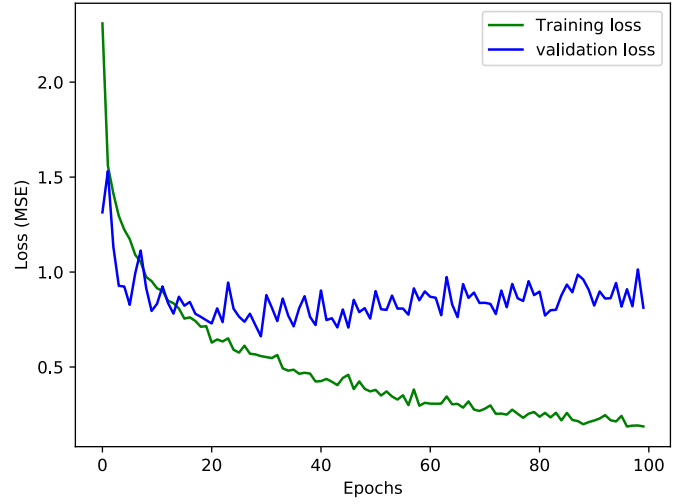


Fig. 8. Deep Neural Network MSE

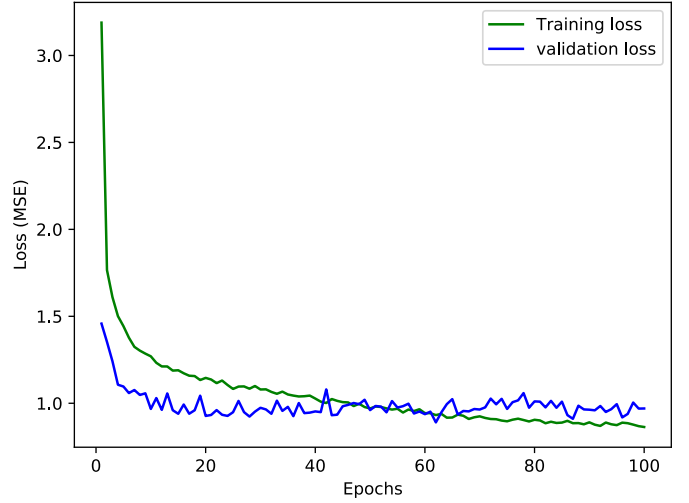


Fig. 9. Wide Neural Network MSE

layers, and one output layer) against baseline models: a WNN model with one hidden layer only and traditional LR. Unlike traditional computational intelligence approaches, deep learning predictive performs much better with large datasets. The model learns features to look for and to make better predictions. We split the data randomly to 80% training and 20% testing.

For better prediction, we further fine-tuned the DNN and WNN through another round of random splitting of the training dataset into 80% training and 20% validation. We then plotted and analyzed the MSE of the training and validation datasets for WNN and DNN over 100 epochs, as shown in Figures 8 and 9.

As a final step, and after fine-tuning the parameters of NN models using the validation dataset, we compared the DNN model against the two baseline models using the testing dataset. The MSE for the testing dataset for the DNN model was 1.15, compared to 1.34 and 1.77 for WNN and LR

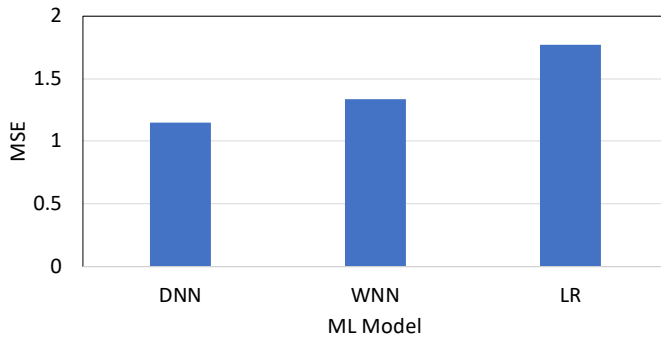


Fig. 10. Mean Squared Error (MSE) of Models.

models, respectively, as shown in Figure 10. We offer empirical evidence on DNN models' effectiveness for predicting the hourly number of passengers waiting in bus stations by achieving 35% and 14% better prediction for the DNN model over LR and WNN models, respectively.

Furthermore, to study the influence of our dataset's informative features, we use an ensemble of gradient boosting decision trees. A benefit of using ensembles of decision tree methods like gradient boosting is that they can automatically estimate feature importance from a trained predictive model. Decision trees provide more information about the relationships of the features than a standard correlation index. We rank the predictive power of the input features, and we find that there are seven most decisive features for partitioning the decision tree to predict the number of passengers waiting at the bus stop. The seven features as shown in Figure 7 are Bus Stop, Week Day, Pressure, Minimum Temperature, Humidity, Clouds, and Week of Semester. It summarizes the reduction in the impurity index over all trees when a particular feature is pointed during the trees' internal space partition over several epochs.

V. CONCLUSION

This paper studied the correlation between the passengers waiting at bus stations and the weather conditions using a deep learning algorithm. This paper gives empirical evidence on the importance of incorporating predictive modeling in intelligent transportation to maximize fleet utilization. We highlighted the importance of applying deep learning models to precisely predict the number of passengers waiting at bus stops in intelligent transportation systems. This research also leveraged a broad set of features from IoT devices like smartphone Wi-Fi data in conjunction with detailed weather information. Our results show four of the top seven decisive features (Pressure, Minimum Temperature, Humidity, and Clouds) were related to temperature and weather. It gives empirical evidence on the importance of weather information in influencing riders' behavior and, hence, better predicting the number of passengers waiting at the bus stops.

We studied only one month of campus-level data in 2017. In future work, data collection could take place yearly and on the city scale to confirm the generalization of the predictive model's performance. A limitation of this study is that we

did not take into consideration any unusual event like a pandemic. In future work, we will incorporate data of 2020 and investigate more informative features. Among the features to be explored are the government's restriction levels (stay-at-home, business is open with restrictions or no restrictions, etc.) The study should also incorporate features like the number of daily reported cases reported in the city and lockdown dates to improve the prediction of the number of passengers waiting at bus stops during the COVID-19 pandemic.

ACKNOWLEDGMENT

This work is funded through a 4-VA research grant (<https://4-va.org>) and JSPS KAKENHI Grant Numbers JP20K11789, JP20H04183.

REFERENCES

- [1] S. El-Tawab, R. Oram, M. Garcia, C. Johns, and B. B. Park, "Data analysis of transit systems using low-cost IoT technology," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2017, pp. 497–502.
- [2] S. Ryu, B. B. Park, and S. El-Tawab, "WiFi Sensing System for Monitoring Public Transportation Ridership: A Case Study," *KSCE Journal of Civil Engineering*, vol. 24, no. 10, pp. 3092–3104, 2020.
- [3] A. Khanna and R. Anand, "IoT based smart parking system," in *2016 International Conference on Internet of Things and Applications (IOTA)*. IEEE, 2016, pp. 266–270.
- [4] M. Garcia, P. Rose, R. Sung, and S. El-Tawab, "Secure Smart Parking at James Madison University via the Cloud Environment (SPACE)," in *2016 IEEE Systems and Information Engineering Design Symposium (SIEDS)*, 2016, pp. pp–271.
- [5] A. Salem, T. Nadeem, M. Cetin, and S. El-Tawab, "Driveblue: Traffic incident prediction through single site bluetooth," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 725–730.
- [6] O. Popescu, S. Sha-Mohammad, H. Abdel-Wahab, D. C. Popescu, and S. El-Tawab, "Automatic incident detection in intelligent transportation systems using aggregation of traffic parameters collected through V2I communications," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 2, pp. 64–75, 2017.
- [7] H. V. Chand and J. Karthikeyan, "Survey on the Role of IoT in Intelligent Transportation System," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 11, no. 3, pp. 936–941, 2018.
- [8] S. El-Tawab, I. Arai, A. Salman, and B. B. Park, "A Framework for Transit Monitoring System Using IoT Technology: Two Case Studies," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2020, pp. 1–6.
- [9] M. Dunlap, Z. Li, K. Henrickson, and Y. Wang, "Estimation of Origin and Destination Information from Bluetooth and Wi-Fi Sensing for Transit," in *Transportation Research Board 95th Annual Meeting*, no. 16-6837, 2016.
- [10] T. Oransirikul, R. Nishide, I. Piumarta, and H. Takada, "Feasibility of Analyzing Wi-Fi Activity to Estimate Transit Passenger Population," in *IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, 2016, pp. 362–369.
- [11] N. Lathia, C. Smith, J. Froehlich, and L. Capra, "Individuals among commuters: Building personalised transport information services from fare collection systems," *Pervasive and Mobile Computing*, vol. 9, no. 5, pp. 643–664, 2013.
- [12] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo, "Car parking occupancy detection using smart camera networks and deep learning," in *2016 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, 2016, pp. 1212–1217.
- [13] T. Arabghalizi and A. Labrinidis, "How full will my next bus be? a framework to predict bus crowding levels," in *Urbcomp 2019*, 2019.
- [14] "Openweather," <https://openweathermap.org/>, [Online; accessed 16-December-2020].
- [15] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*. Elsevier, 1992, pp. 65–93.