

Road State Inference via Channel State Information

Halit Bugra Tulay, *Member, IEEE*, Can Emre Koksal, *Senior Member, IEEE*

Abstract—A wide variety of sensor technologies are recently being adopted for traffic monitoring applications. Since most of these technologies rely on wired infrastructure, the installation and maintenance costs limit the deployment of the traffic monitoring systems. In this paper, we introduce a traffic monitoring approach that exploits physical layer samples in vehicular communications processed by machine learning techniques. We verify the feasibility of our approach with extensive simulations and real-world experiments. First, we simulate wireless channels under realistic traffic conditions using a ray-tracing simulator and a traffic simulator. Next, we conduct experiments in a real-world environment and collect messages transmitted from a roadside unit (RSU). The results show that we are able to predict different levels of service with an accuracy above 80% both on the simulation and experimental data. Further, the proposed approach is capable of estimating the number of vehicles with a low mean absolute error on both data. Our approach is suitable to be deployed alongside the current monitoring systems. It doesn't require additional investment in infrastructure since it relies on existing vehicular networks.

Index Terms—traffic monitoring, intelligent transportation systems, DSRC, C-V2X, vehicular ad-hoc networks

I. INTRODUCTION

A traffic monitoring system (TMS) is used to collect traffic data such as traffic density, types of vehicles, and speed to perform traffic analysis, predict future transportation needs, and improve the safety of transportation based on the collected data. Nowadays, intrusive sensors (e.g., inductive loops, magnetic detectors), vision-based systems, or radars are mostly used for traffic monitoring. Most of these systems need fixed wired infrastructure which results in high installation and maintenance costs. Consequently, the costs of the systems prevent the dense deployment on roads. According to the Georgia Department of Transportation, a TMS on two-lane roadway costs roughly \$25,000 and the cost of installation can go up to \$80,000 [1]. This motivates low-cost, non-intrusive approaches.

In this paper, we introduce a novel traffic monitoring approach that exploits the communication signals broadcast in a vehicular ad-hoc network (VANET). We solely rely on the channel state information, not the content of the messages. In VANETs, vehicles and traffic infrastructure exchange data periodically with each other via vehicle-to-infrastructure (V2I) communication links. The vehicles equipped with onboard units (OBUs) exchange data such as their position, speeds, and certain event-triggered messages with the other vehicles and roadside units (RSUs) in the traffic infrastructure. Similarly, RSUs inform vehicles about signal phases of traffic lights,

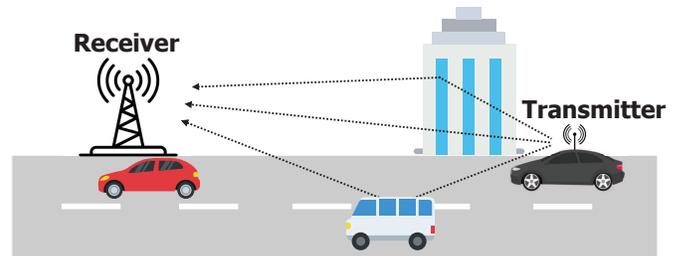


Fig. 1: A typical communication scenario in a vehicular network.

speed limits, or road work maps [2]. We utilize the signals transmitted from a vehicle and captured at deployed roadside units (RSUs) as shown in Fig. 1, or vice-versa. Our approach doesn't require additional hardware deployment while relying on the existing infrastructure. We also don't rule out the possibility of static transmitters or receivers placed on the side of a roadway. Since our approach is based on the use of mere channel state information over a single RF chain, it can be an effective candidate across different standards, including DSRC [3] and cellular vehicle-to-everything (C-V2X) which is defined by the 3GPP as part of its LTE and ongoing 5G families of standards. We also demonstrate that our method provides highly accurate results in sparse settings, even when there is a single transmitter-receiver pair. This may be highly valuable until we transition to dense deployments of V2X systems.

Our approach takes advantage of the radio propagation characteristics of vehicular networks. A transmitted wireless signal travels over multiple paths reflected from the surfaces. In vehicular networks, reflectors are mostly vehicles on a road and the channel state is shaped by the traffic conditions. The idea in our approach is to infer the traffic conditions on the road based on the channel state information, specifically the channel frequency response (CFR). We use supervised learning algorithms since a structured model-based mapping between the traffic conditions and the channel frequency response values is extremely difficult due to many possible traffic conditions. Unfortunately, there is no available signal-level data collected under a variety of traffic conditions to train the machine learning algorithms. To solve this problem, we use a ray-tracing simulator and a traffic simulator in conjunction to simulate the wireless propagation for a four-lane road. We first train the machine learning algorithms with the obtained simulation data and evaluate the performance of the proposed approach. Subsequently, we conduct DSRC-based V2I experiments in a real vehicular network at an intersection. Specifically, we collect signal phase and timing (SPaT) messages broadcast by a real roadside unit (RSU).

Halit Bugra Tulay and Can Emre Koksal are with the Department of Electrical and Computer Engineering, The Ohio State University, OH, 43202 USA (e-mail: tulay.1@osu.edu; koksal.2@osu.edu)

The messages have been collected from a static receiver and a moving vehicle using our software-defined radio, and we evaluate the performance with the collected experimental data.

Operational conditions in a traffic stream are characterized by quantitative measures specified in the highway capacity manual [4]. The highway capacity manual defines six levels of service (LOS), from letter A to F, for the density of traffic in terms of passenger cars per kilometer per lane while LOS A represents the best operating conditions and LOS F represents the worst conditions. Our approach estimates the LOS with over 80% classification accuracy on both the simulation data and the experimental data. We have estimated the number of vehicles with a mean absolute error (MAE) of 2.74 vehicles on a four-lane road in the simulations, and with a mean absolute error of 0.93 vehicles on a two-lane road in the real-world experiments. The main contribution of this paper can be summarized as:

- We propose a non-intrusive and cost-effective solution for traffic inference by exploiting the existing VANET infrastructure.
- We demonstrate the viability of the use of mere channel frequency responses within the existing vehicular communication technologies for high-performance traffic state inference, without requiring any additional hardware deployment.
- We collected a substantial amount of V2I communication data from a roadside unit and also generated simulation data using a ray-tracing simulator. The datasets are available at [5] for the research community use.
- We obtain multiple trained models with different machine learning algorithms, and we show that they are able to map the CFR values to traffic conditions with a low error.

This paper extends our previous work [6] by increasing the number of classes for the level of service prediction, using data preprocessing techniques, and with extended real-world measurements. This paper is organized as follows. In Section II, we briefly discuss the previous work done related to the topic and Section III describes the problem model. Section IV describes the details of our approach which include data creation, data preprocessing, and machine learning algorithms. In Section V, we demonstrate the experimental and simulation results. Finally, Section VI summarizes our work.

II. RELATED WORK

There has been extensive research on traffic monitoring/management systems, and different approaches have been proposed [7] [8]. The operation of a traffic monitoring system depends on various technologies. According to the used technology, we can categorize these systems into three groups: 1) Sensor-based systems 2) Aerial technology-based systems and 3) Wireless technology-based systems.

Sensor-based Systems: There is a wide variety of sensors used in traffic monitoring systems today. These sensors can be classified as intrusive and non-intrusive sensors. Intrusive sensors are installed directly into the road surface. Sensors such as inductive loops, magnetic detectors, and other weigh-in-motion devices constitute the popular intrusive sensors.

For example, the freeway performance measurement system (PeMS) is a system used by the California department of transportation (Caltrans). It is based on 30-second measurements from inductive loops in real time. The data comprises the number of vehicles crossing the loop and occupancy (the average fraction of time a vehicle is present over the loop). PeMS data can be accessed online [9]. PeMS achieves an average accuracy of 82% on the level of service classification task, while it is as low as 56% for LOS E [10]. The major drawback of the intrusive sensors is the interruption of traffic since they are installed under road surfaces. Also, the cost of the system increases if the sensor is able to monitor a single lane while the multiple lanes are to be monitored [11].

Non-intrusive sensors are deployed above the road level or on the side of a roadway. As a result, they can be easily deployed, and the installation of the systems does not interrupt traffic flow. Cameras, microwave radars, and passive infrared sensors are among the most popular non-intrusive sensors. Currently, many traffic monitoring systems incorporate cameras and video processing techniques [12]. As deep learning techniques are becoming more widely adopted, they offer great potential for traffic monitoring applications over traditional techniques. In [13], the authors evaluate the performance of different convolutional neural network (CNN) architectures on the level of service classification. They achieve a mean accuracy of more than 80% for tested CNN configurations, while the best architecture reaches a mean accuracy of 89%. Despite the wide usage of non-intrusive sensors, their deployment and maintenance costs are relatively high. Further, their performance can be affected by certain weather conditions [14].

Aerial technology-based systems: The aerial platforms, especially the unmanned aerial vehicles (UAVs), have become a cost-efficient solution for road traffic monitoring because of their mobility and large range [15]. These technologies don't require any hardware deployment under/on the roadways, and therefore, they are suitable for such a dynamic environment. In [16], the authors present a computer vision-based traffic surveillance system using an aerial camera array. They present that the deep learning combined with speeded up robust features (SURF)-based approach is able to achieve over 93% accuracy in density estimation.

Wireless technology-based systems: The U.S. Department of Transportation's intelligent transportation systems (ITS) Joint Program Office encourages to deploy applications utilizing data captured from multiple sources (e.g., vehicles, mobile devices, and infrastructure) across all elements of the surface transportation systems [17]. To this end, connected vehicle technologies, DSRC and C-V2X, have been considered an important part of intelligent transportation systems. Architecture Reference for Cooperative and Intelligent Transportation (ARC-IT) [18] of the US Department of Transportation uses the collected data from connected vehicles to estimate traffic conditions. To this end, traffic monitoring using vehicle-to-vehicle (V2V) and vehicle-to-infrastructure communication links has been studied in several studies. In [10], the authors propose a method for estimating traffic density using connected vehicle technology and artificial intelligence. Using

VISSIM, a traffic microsimulation software, they demonstrate that the accuracy of the LOS classification is a minimum of 85% with 20% and greater connected vehicle penetration levels. In [19], the authors introduce CoTEC (COoperative Traffic congestion detECTION), a road congestion detection algorithm based on cooperative awareness messages (CAM) or beacon messages broadcasting the road traffic conditions periodically. CoTEC is evaluated under large-scale highway scenarios using iTETRIS, an open-source simulation platform created to investigate the impact of cooperative vehicular systems. In [20], the authors propose a method named SCORPION (System with COoperative Routing to imPROve traffic cONdition) that is based on V2I communication. They use the K-Nearest Neighbor (KNN) classifier which uses the average speed and the density of each road to classify the traffic conditions as free-flow, slightly congested, moderately congested, and severely congested.

Recent advances in wireless technology also enable a new sensing paradigm that is often used to recognize human behaviors using WiFi technologies in indoor settings [21]. Similarly, WiFi-based traffic monitoring systems have been proposed with the motivation of reducing deployment costs for large-scale deployment. The researchers study the feasibility of using WiFi devices for traffic monitoring applications using channel state information (CSI) [22] [23], received signal strength indicator (RSSI) [24] [25], link quality indicator (LQI) [26], and packet loss rate [26]. The proposed approaches exploit the RF propagation between a receiver and a transmitter that is affected by passing vehicles. Consequently, different patterns of the wireless channel metrics are observed at the receiver depending on traffic conditions and the types of vehicles. These patterns are used to classify and count passing vehicles. The system proposed in [22] utilizes a laptop and a router deployed on the roadside. The CSI powers of the passing vehicles are captured on local roads and highways to count and classify vehicles using machine learning techniques. They show that road lanes have different CSI patterns and this allows them to identify in which lane a vehicle is detected. A similar setup and deep learning techniques are used in their next study [23] to count and classify vehicles. In [24], the authors employ a system that exploits the attenuation of radio signals for the detection and classification of vehicles using machine learning techniques. Similarly, the authors in [25] proposed a system that exploits RSSI information to detect and classify vehicles. The authors in [26] set up a transmitter-receiver pair and classify traffic conditions as free-flow or congested using a decision tree-based classifier. In the proposed method, receivers collect packets from the transmitter placed on the opposite side of a road and different metrics (RSSI, LQI, packet loss) are used to classify the traffic conditions. They achieve a classification accuracy of 97% in their experiments.

The aforementioned approaches require specialized hardware (e.g., inductive loops, UAVs, camera arrays). In contrast, our approach relies on the samples collected in available infrastructure without a need to decrypt the message content, thereby preserving privacy as a bonus. Also, our approach is non-intrusive, easy to deploy, and cost-effective since it is

built upon the existing infrastructures. Thus, it can easily be integrated with other traffic monitoring systems and improve their performance. A unique component of our approach is that it doesn't observe the wireless link behavior (interruption of line-of-sight component) for counting each vehicle as opposed to the previous approaches. We estimate the number of vehicles from a single channel frequency estimate. Another unique component of our work is in the measurements. We rely heavily on actual data collected from a real roadside unit, as opposed to analysis or raw simulations.

III. PROBLEM MODEL AND STATEMENT

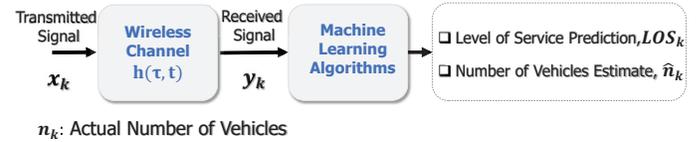


Fig. 2: Problem model.

The components of the system are shown in Fig. 2. x_k are transmitted signal samples of k^{th} frame from a static transmitter on the roadside or a vehicle, and y_k are the received signal samples after passing through the wireless channel. The receiver estimates the wireless channel using the received signal samples. We can represent the wireless channel as a linear time-varying channel filter and it can be characterized by its baseband impulse response as:

$$h_b(t, \tau) = \sum_i a_i(t) e^{-j2\pi f_c \tau_i(t)} g(\tau - \tau_i(t)) \quad (1)$$

where $a_i(t)$, $\tau_i(t)$, f_c are the path attenuation and delay of path i at time t , the carrier frequency, respectively. $g(\tau)$ is the impulse response of the transmit and receiver filters. The corresponding time-varying channel frequency response can be calculated as [27]:

$$H(t, f) = \int_{-\infty}^{\infty} h_b(\tau, t) e^{-j2\pi f \tau} d\tau = G(f) \sum_i a_i(t) e^{-j2\pi(f+f_c)\tau_i(t)} \quad (2)$$

where $G(f)$ is the frequency response of the transmit and receive filters and it can be assumed to be constant in the presence of guard subcarriers on both sides of the spectrum [28]. We consider an orthogonal frequency division multiplexing (OFDM) system. The channel frequency response is estimated using the preambles at the beginning of each OFDM frame. We can estimate the channel frequency response (CFR) at N subcarriers for k^{th} frame as:

$$\mathbf{H}_k \triangleq [H_{k,0}, H_{k,1}, \dots, H_{k,N-1}]^T \quad (3)$$

where $H_{k,i} \triangleq H(kT, f_i)$. T denotes the frame duration and f_i is the baseband frequency of i^{th} subcarrier. Note that $H_{k,i}$ is a complex number and represented by the magnitude $|H_{k,i}|$ and the phase $\angle H_{k,i}$ as $H_{k,i} = |H_{k,i}| e^{j\angle H_{k,i}}$. Considering the phase estimation errors and the phase noise, we only use the magnitude frequency response values, $|H_{k,i}|$, estimated from the received signal. The machine learning algorithms are trained with the magnitude response values and output the

level of service prediction, LOS_k , and estimated number of vehicles, \hat{n}_k .

Problem Statement: First, we predict the density of the traffic conditions and assign level of service labels $LOS_k \in \{A, B, C, D, E, F\}$ for each received frame. The highway capacity manual defines six levels of service according to the type of roadway. In this work, we use the level of service definitions that are based on density. The density of a roadway is expressed in terms of passenger cars per kilometer per lane (pc/km/ln). Table I summarizes the level of service thresholds based on density for a basic freeway segment. The machine learning algorithms map N-dimensional CFR vectors to level of service predictions, $c(|\mathbf{H}_k|) : \mathbb{R}^N \rightarrow LOS_k$, where $|\mathbf{H}_k|$ is the magnitude of CFR values for k^{th} received frame and $c()$ represents the classification algorithm.

TABLE I: Level of service definitions according to density

Level of service	Density, pc/km/ln
A	0-7
B	7-11
C	11-16
D	16-22
E	22-28
F	> 28

Next, we estimate the number of vehicles on a roadway. In this problem, a regression algorithm maps the CFR vector to the number of vehicles estimate, $r(|\mathbf{H}_k|) : \mathbb{R}^N \rightarrow \mathbb{R}$, where $r()$ represents the regression algorithm. To this end, we build a regression model that minimizes the error between the actual number of vehicles, n_k , and the estimated number of vehicles, \hat{n}_k .

IV. APPROACH

We need training data that include the wireless channel realizations under different traffic conditions to train the machine learning algorithms. Unfortunately, such data are not available. Therefore, we first generate data by integrating two simulators. The integration of the simulators helps us to obtain realistic wireless data under complex scenarios easily and evaluate the performance of the approach. Subsequently, we collect real experimental data in different scenarios as described in the next sections.

A. Data Creation Methodology

Simulation of the Wireless Channel

In a vehicular network, vehicle surfaces, buildings, and terrains are the main sources of reflections and diffractions. The wireless channel between the highly mobile nodes is hard to be captured by the probabilistic channel models. In this work, we utilize the ray-tracing approach rather than probabilistic channel models. The ray-tracing techniques represent the electromagnetic waves sent from a transmitter as a simple particle and estimate the paths between a receiver and the transmitter. Therefore, ray-tracing provides more accurate and spatially consistent results compared to the probabilistic models. More details about the principles of ray tracing can be found in [29]. In this work, we use Remcom's Wireless

Insite [30] as a ray-tracing simulator to simulate wireless propagation.

The ray-tracing simulator provides the rays, each corresponding to a propagation path, between a receiver and a transmitter. The simulator also gives information on power, delay, phase, angle of arrival, and angle of departure of each path. Given this information, we can calculate CFR values according to Eq. 2. However, IEEE 802.11p utilizes a preamble-based channel estimation method from noisy received signals, and this results in an error in the channel estimation. We incorporate this error with the following model. Suppose that $\{t[n]\}_{n=0}^{N_{tr}-1}$ is a known training sequence, $\mathbf{y} = [y[L] y[L+2] \dots y[N_{tr}-1]]$ is the received signal samples, and $\mathbf{w} = [w[L] w[L+2] \dots w[N_{tr}-1]]$ is the noise samples after removing first L received samples. We can write the received signal in a matrix form as:

$$\mathbf{y}_k = \mathbf{t}_k \mathbf{h}_k + \mathbf{w}_k \quad (4)$$

where \mathbf{t}_k is the $(N \times L)$ circularly shifted training matrix with $N = N_{tr} - L$ the received sequence length. \mathbf{h}_k is an L-pair channel impulse response vector. The received frequency domain signal of the k^{th} OFDM frame, after removing the cyclic prefix and applying the discrete Fourier transform, can be written in the vector form as $\mathbf{Y}_k \triangleq [Y_{k,0}, Y_{k,1}, \dots, Y_{k,N-1}]^T$ can be written as:

$$\mathbf{Y}_k = \mathbf{T}_k \mathbf{H}_k + \mathbf{W}_k \quad (5)$$

where \mathbf{T}_k is $N \times N$ diagonal matrix with $\langle n, n \rangle^{th}$ element given as $T_{k,n}$ where $T_{k,n}$ is the pilot subcarriers of k^{th} frame. \mathbf{H}_k is the channel frequency response, and $\mathbf{W}_k \triangleq [W_{k,0}, W_{k,1}, \dots, W_{k,N-1}]^T$ is the frequency domain noise vector where $W_{k,i}$ is an additive Gaussian noise at subcarrier i of k^{th} frame, with zero mean and variance σ_W^2 . The commonly used channel estimation scheme for the IEEE 802.11p is the least-squares estimation and the least-squares channel estimate, which is also the maximum likelihood estimate under additive white gaussian noise (AWGN), is given by:

$$\begin{aligned} \hat{\mathbf{H}}_k &= \arg \min_{\mathbf{H}_k} \|\mathbf{Y}_k - \mathbf{T}_k \mathbf{H}_k\|^2 \\ &= (\mathbf{T}_k^H \mathbf{T}_k)^{-1} \mathbf{T}_k^H \mathbf{Y}_k = \mathbf{T}_k^{-1} \mathbf{Y}_k \\ &= \left[\frac{Y_{k,0}}{T_{k,0}}, \frac{Y_{k,1}}{T_{k,1}}, \dots, \frac{Y_{k,N-1}}{T_{k,N-1}} \right] \end{aligned} \quad (6)$$

where $\|\cdot\|$ denotes the norm of a vector, x^H is Hermitian transposition of x . Substituting Eq. 5 into Eq. 6, the LS channel estimate of subcarriers can be expressed as:

$$\hat{\mathbf{H}}_k = \mathbf{H}_k + \underbrace{\mathbf{T}_k^{-1} \mathbf{W}_k}_{\mathcal{E}=\text{Estimation Error}} \quad (7)$$

where $\mathcal{E} = \mathbf{T}_k^{-1} \mathbf{W}_k$ denotes the estimation error. Since $E[\hat{\mathbf{H}}_k] = E[\mathbf{H}_k] + \mathbf{T}_k^{-1} E[\mathbf{W}_k] = E[\mathbf{H}_k]$ forms an unbiased

estimator of \mathbf{H}_k , i.e., $E[\mathcal{E}] = \mathbf{0}$. The covariance matrix of the estimation error can be calculated as:

$$\begin{aligned} E[\mathcal{E}\mathcal{E}^H] &= E[\mathbf{T}_k^{-1}\mathbf{W}_p\mathbf{W}_p^H(\mathbf{T}_k^{-1})^H] \\ &= \mathbf{T}_k^{-1}E[\mathbf{W}_k\mathbf{W}_k^H](\mathbf{T}_k^{-1})^H \\ &= \sigma_W^2(\mathbf{T}_k^H\mathbf{T}_k)^{-1} \\ &= \frac{\sigma_W^2}{P_T}\mathbf{I}_N = \sigma_{\mathcal{E}}^2\mathbf{I}_N \end{aligned} \quad (8)$$

where P_T is equal to the transmitter power per subcarrier (i.e., $(\mathbf{T}_k^H\mathbf{T}_k) = P_T\mathbf{I}_N$, where \mathbf{I}_N is an identity matrix of size N) and $\sigma_{\mathcal{E}}^2$ is the error variance per subcarrier. Note that the error variance per subcarrier depends on the noise variance, σ_W^2 , and the transmitter power per subcarrier.

We estimate the CFR values using the least-squares channel estimator in our real-world experiments. For simulations, the propagation data obtained from the ray-tracing simulator are used to generate the actual CFR values. Since the estimation error is not incorporated in our simulations, we calculate the estimation errors with Eq. 8 and add them to the actual CFR values.

Simulation of the Traffic

Many models and simulation tools have been developed to generate realistic vehicular mobility [31]. In this work, we use Simulation of Urban Mobility (SUMO) [32] to simulate realistic traffic conditions for the ray-tracing simulator since it provides us great flexibility with various configuration files and allows us to import real roads from OpenStreetMap(OSM). The traffic control interface of SUMO, *TraCI* [33] is also utilized to obtain different parameters of the vehicles such as vehicle ID, position, speed, etc., and change these parameters.

Integration of the simulators

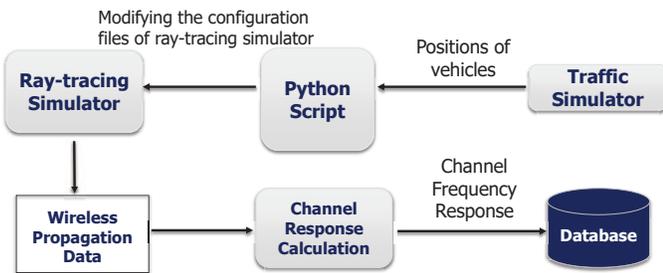


Fig. 3: The methodology that integrates the simulators.

We follow the methodology proposed in [34] as shown in Fig. 3 to integrate two simulators and simulate the wireless propagation under various traffic conditions. In this methodology, the positions of vehicles obtained from the traffic simulator are used to place vehicles in the form of *Objects* in the ray-tracing simulator. To enable this, we wrote a Python script that runs the traffic simulator for given mobility parameters and retrieves the position of vehicles. The script later modifies the positions of the objects in the configuration files of the ray-tracing simulator. After modifying the configuration files, the script uses the command line controls to run Wireless Insite's

calculation engine. Wireless Insite runs the simulations with given configurations and saves the requested output inside a folder. The script repeats this procedure for a certain number of simulations defined by us. Finally, we use MATLAB to post-process and create the channel frequency responses from the output folders and save them in a database. The roles of components can be summarized as follows:

- Traffic Simulator
 - Simulating traffic conditions for given mobility parameters.
 - Providing the positions of the vehicles.
- Ray-Tracing Simulator
 - Specifying the radio propagation environment with a configuration file.
 - Simulating the wireless propagation according to the positions of the vehicles.
 - Saving propagation information for each scenario.
- Script
 - Running the traffic simulator and obtaining the attitudes of the vehicles.
 - Modifying the configuration files of the ray-tracing simulator.
 - Running the ray-tracing simulator via the command line.

B. Data Preprocessing

Within the field of machine learning, data quality is a significant consideration. Data preprocessing is an essential procedure to improve the quality of data and the outcomes of the inference. Fig. 4 shows how the data obtained are preprocessed before being fed to machine learning algorithms. In the following sections, we introduce the steps of CFR data preprocessing.

Background Elimination

CFR does not only embody the reflections from the vehicles but also embodies the reflections from the static environment that should not be learned in machine learning algorithms. The static reflections might cause over-fitting, and degrade the generalization performance of the learning algorithms. Therefore, we need to remove the effect of the static environment from the CFR.

To this end, we estimate a sequence of CFR vectors when there is no vehicle on the road and calculate the average of the CFR vectors \mathbf{H}_b as the background CFR. We remove it from the CFR measurements as follows:

$$\bar{\mathbf{H}}_k = \mathbf{H}_k - \mathbf{H}_b$$

where $\bar{\mathbf{H}}_k$ is the CFR of k^{th} frame after the background elimination. This operation ensures that $\bar{\mathbf{H}}_k$ preserves only the dynamic reflections shaped by the vehicles.

Outlier Removal

Outlier removal is an important step since outliers could affect traffic inference performance. Thus, outliers should be

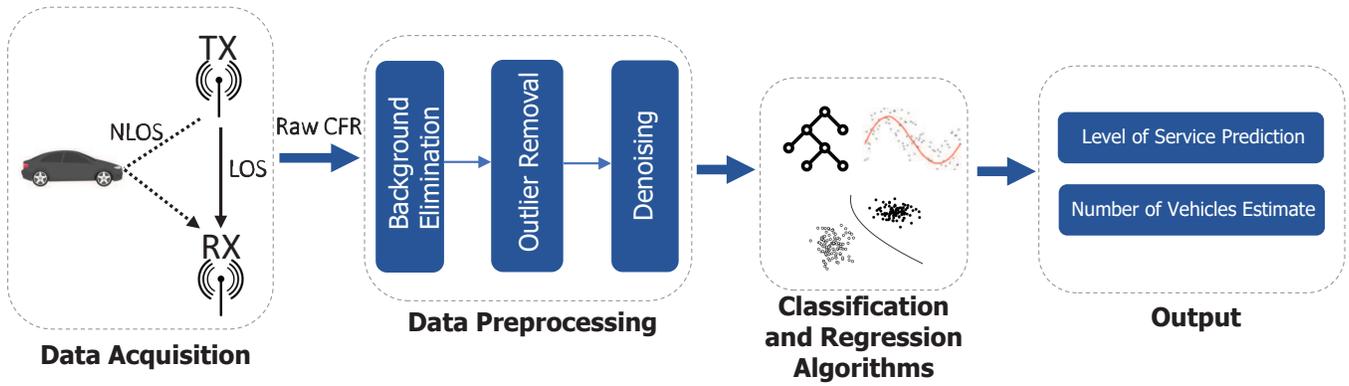


Fig. 4: Architecture overview of the system.

sifted out before further data processing. The purpose of outlier removal is to eliminate and replace outliers with their expected values.

To this end, linear filters are sometimes used for eliminating outliers but it is observed that the linear filters are generally ineffective in this regard and effective outlier removal filters are necessarily nonlinear [35]. Further, outlier removal is different from signal filtering. Signal filtering not only removes outliers but also changes the data structure by reducing the data variations. In this regard, outlier removal is more difficult than filtering since it tries to preserve data structure while removing outliers.

We utilize a Hampel filter, obtained by applying the Hampel identifier to a moving data window. In detail, given a sequence $x_1, x_2, x_3, \dots, x_n$ and a sliding window of length w , we can define the local median and the standard deviation as follows:

- $m_i = \text{median}(x_{i-w}, x_{i-w+1}, \dots, x_{i+w-1}, x_{i+w})$
- $\sigma_i = \kappa \text{median}(|x_{i-w} - m_i|, \dots, |x_{i+w} - m_i|)$, where $\kappa = \frac{1}{\sqrt{2} \text{erf}^{-1}(1/2)} \approx 1.4826$.

where m_i and σ_i are the local median and the standard deviation. The quantity σ_i/κ is known as the median absolute deviation (MAD). If a sample x_i is such that

$$|x_i - m_i| > n\sigma_i \quad (9)$$

for a given threshold n , then the Hampel identifier declares x_i an outlier and replaces it with m_i .

Fig. 5 shows the waveform of one subcarrier and the Hampel filtered version of it. The Hampel identifier adopted here uses the window size of 5 and the threshold $n = 3$.

Denoising

Internal state transitions (e.g. transmission strength changes, rate changes) in communication systems, electromagnetic interference, and thermal noise can be listed as the main source of noise in CFR samples. The noises in the raw CFR samples should be wiped out to avoid unnecessary complexity in the learning models and improve the performance of the algorithms. Also, the signal strength variation over distances of the order of the carrier wavelength, due to constructive and destructive interference of multipath components, should be

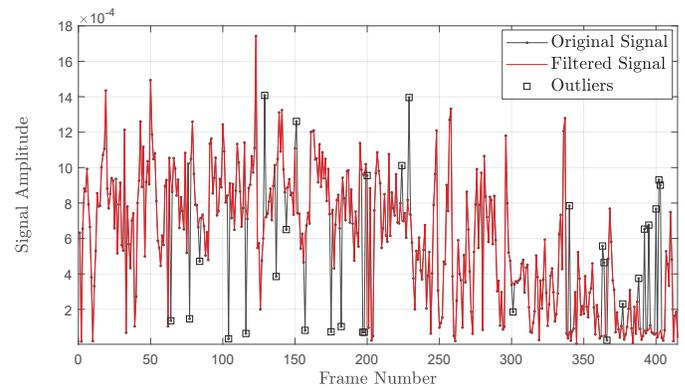


Fig. 5: Original and outlier filtered signals.

smoothed away to increase the performance of the approach. Since the signal fluctuation created by vehicular activities has low-frequency components, we can adopt a low-pass filter to eliminate the noise and multipath interference in CFR.

The filter to be used should not introduce a large delay to capture the exact time of the events and not distort the signal characteristics. We contend that it is not convenient to utilize traditional filters (e.g., the Butterworth and Chebyshev filters). Specifically, IIR filters present an undesired phase shift (delay) into the filtered signal which varies with the frequency of the signal. This delay can be prevented only if the complete signal is known in advance by using zero-phase filtering techniques which is impossible in real-time measurements. They also soften the rising/falling edges appeared in the signals, which are critical for traffic inference.

We utilize the wavelet filter [36] since it does not only smooth away the signal but also successfully preserves the sharp transitions. Fig. 6 shows the performance of the wavelet filter on a sub-carrier. The filtering is controlled by the selection of wavelet type and the decomposition level. The higher decomposition level means a lower frequency divider between the signal and noise. Specifically, we employ four levels 'sym4' wavelet transform on each sub-carrier signal with the decomposition level of 9. We observe that the wavelet filter captures the abrupt changes well while smoothing away the signal.

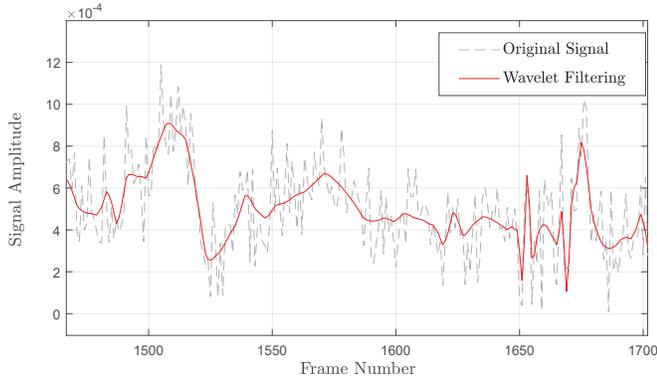


Fig. 6: Performance of the wavelet filter on the signal.

C. Machine Learning Algorithms

In this work, we use SciPy [37], an open-source Python library used for scientific computing, and scikit-learn [38] to construct our machine learning framework. Especially, scikit-learn features many machine learning algorithms for classification, regression, and more. In this paper, we use ensemble learning algorithms [39] (extremely randomized trees, gradient boosting, random forest), support vector machine (SVM), and k-nearest neighbors (KNN) algorithms to train learning models for classification and regression purposes. The hyperparameter optimization is performed with the grid search method to find the best parameters of the models. The parameters used for each data set can be found at [5].

Algorithm Evaluation

We use a stratified k-fold cross-validation approach to evaluate the performance of the different algorithms. The approach is similar to k-fold cross-validation. It divides the set of samples into k groups, or *folds*, of approximately equal size and uses $k - 1$ folds for the training and the remaining fold for the validation. This procedure is repeated k times and a different group of the samples is used as a validation set each time. In stratified k-fold cross-validation, the folds are formed in a way that each fold contains approximately the same proportion of predictor labels as the original data set to minimize the bias inherited from the random sampling. The stratified k-fold cross-validation estimate of a metric (CV_{Metric}) is computed by taking the average of the metric over k folds as:

$$CV_{Metric} = \frac{1}{k} \sum_{i=1}^k Metric_i$$

V. PERFORMANCE EVALUATION

A. Performance Metrics

In this work, the accuracy of the algorithms, the area under the receiver operating characteristic curve (AUC) [40], and the macro-averaged F_1 score are employed to evaluate the performance of the level of service prediction problem. The receiver operating characteristic (ROC) curve is plotted with the true positive rate (TPR) against the false positive rate

(FPR). Although the receiver operating characteristic (ROC) curve is typically used in the binary classification problem, we adapt it for multi-class classification with the one-vs-all approach and averaging techniques like macro averaging. For each class i , the TPR and FPR can be calculated as

$$TPR_i = \frac{TP_i}{TP_i + FN_i}, FPR_i = \frac{FP_i}{FP_i + TN_i}$$

The macro-averaged TPR and FPR can be calculated as

$$TPR_{macro} = \frac{\sum_{i=1}^C TPR_i}{k}, FPR_{macro} = \frac{\sum_{i=1}^C FPR_i}{k}$$

where C is the number of classes, which is 6 for the level of service prediction. So, the macro-averaged ROC can be obtained by plotting TPR_{macro} against FPR_{macro} for different threshold values. Here, AUC indicates the performance of the classifier independent of the threshold value and helps us evaluate how well the probabilities from the positive class are separated from the negative class since we have a balanced dataset. Macro-averaged F_1 score maintains a balance between the precision and the recall to measure the model's accuracy and it is calculated as the harmonic mean of the macro-averaged recall (R_{macro}) and macro-averaged precision (P_{macro}) as:

$$F_{1-macro} = \frac{2P_{macro}R_{macro}}{P_{macro} + R_{macro}}$$

where

$$P_{macro} = \frac{1}{C} \sum_{i=1}^C \frac{TP_i}{TP_i + FP_i}, R_{macro} = \frac{1}{C} \sum_{i=1}^C \frac{TP_i}{TP_i + FN_i}$$

We report the mean absolute error (MAE), weighted mean absolute percentage error (WMAPE) as a scale-independent metric, and Pearson correlation coefficient between estimated and actual values as performance metrics for the number of vehicles estimation problem. For M number of samples of X and Y , they are defined as

$$MAE = \frac{1}{M} \sum_{i=1}^M |x_i - y_i|$$

$$WMAPE = \frac{\frac{1}{M} \sum_{i=1}^M |x_i - y_i|}{\frac{1}{M} \sum_{i=1}^M x_i}$$

Pearson correlation coefficient between X and Y (ρ_{xy}):

$$\rho_{xy} = \frac{\sum_{i=1}^M (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^M (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^M (y_i - \bar{y})^2}}$$

where x_i and y_i are the actual and estimated value of a sample point, respectively. \bar{x} and \bar{y} are the sample mean of the actual and estimated values. The Pearson correlation coefficient has a value between +1 and -1. The higher correlation between the predicted and actual values implies a better fit of the model to the data.

B. Simulation Setup

Fig. 7 shows the urban canyon scenario that corresponds to a region in Virginia. The ray-tracing simulator simulates the wireless channel on the four-lane road under different traffic conditions. Two receivers are placed to capture signals transmitted from a transmitter at an intersection that represents a roadside unit. The gray rectangular objects in Fig. 7 correspond to the vehicles on the road.

We import the real-world map of the region from OpenStreetMap (OSM) [41] and convert OSM files to the SUMO road network format using the network converter tool of SUMO as shown in Fig. 8. Three types of vehicles: car (1.80 m×4.60m×1.60m), bus (2.40m×9.00m×3.20m) and trucks (2.50m×12.00m×4.30m) are simulated in SUMO. We change the probability of injecting a vehicle to the SUMO network to simulate the different traffic conditions on a road segment.

The material decisions of buildings, terrain, and vehicles are vital for simulating realistic simulations. Wireless Insite has a material database that consists of common buildings, terrain, and a few generic material types. We choose the materials according to the ITU (International Telecommunication Union)



Fig. 7: Four-lane road simulation in an urban canyon scenario.

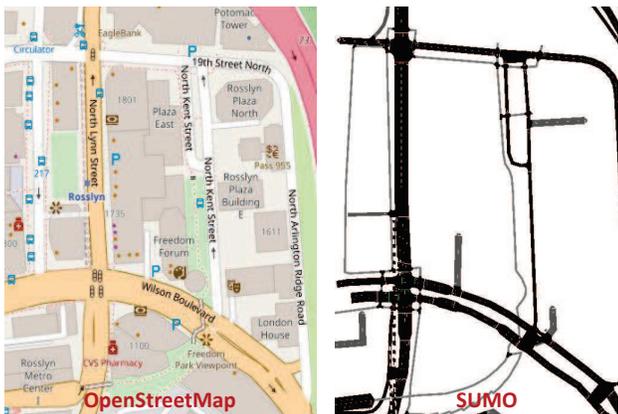


Fig. 8: The road simulated in the ray-tracing simulator is converted to the SUMO network from OpenStreetMap (OSM).

recommendations and Table II shows the details of the ray-tracing parameters.

TABLE II: The ray-tracing simulator parameters

Wireless Insite parameters	
Propagation Model	X3D
Total Number of Rays	25
Building Material	ITU Layered drywall 5GHz
Terrain Material	Asphalt
Vehicle Material	Metal
Antenna	Half-wave dipole
Transmit Power (P_{Total})	30 dBm
Tx-Rx Antenna Height	2 meters
Carrier Frequency	5.9 GHz
Bandwidth (B)	10 MHz

Simulation Results

We have created a data set that includes 12,000 simulations under different traffic conditions using the approach described in Section IV and performed the stratified 10-fold cross-validation on the data set to obtain the results. For each simulation, we obtain the number of vehicles between the transmitter and receivers from SUMO. Next, we calculate the traffic density in terms of passenger cars per kilometer per lane from the number of vehicles. We observe that the density ranges between 0 and 40 pc/km/lane, and we convert the density values to the corresponding level of service, from A to F, according to Table I. We have created a balanced data set that includes 2000 samples from each level of service.

Table III shows the accuracy, macro-AUC, and macro-F1 results for the LOS prediction problem. We achieve the best average accuracy of 88.5% with the extremely randomized trees algorithm. Fig. 9a shows the distribution of 10-fold cross-validation classification results on a box plot. Fig. 9b shows the confusion matrix for the extremely randomized trees algorithm. We notice that the mislabeled levels are higher for LOS B-C compared to the other levels. It is an anticipated result considering the narrow boundaries of these levels.

Afterward, the number of vehicles on four lanes between the transmitter and receivers in Fig. 7 is estimated. Instead of classification, the algorithms now are used to build a regression model that maps the CFR values observed at two receivers to the number of vehicle estimates. Table IV shows MAE, WMAPE, and the correlation coefficients of the algorithms. We again observe that the extremely randomized trees algorithm outperforms others with a mean absolute error of 2.21 vehicles and a WMAPE of 13.8%. Fig. 10 shows the actual and estimated number of vehicles for 50 frames obtained using the extremely randomized trees algorithm.

TABLE III: LOS prediction results on the simulation data.

Algorithm	Accuracy	Macro-AUC	Macro- F_1
Extra Trees	88.5%	0.98	0.88
Random Forest	87.9%	0.98	0.88
Gradient Boosting	80.7%	0.95	0.80
SVM	86.7%	0.96	0.86
KNN	83.5%	0.96	0.83

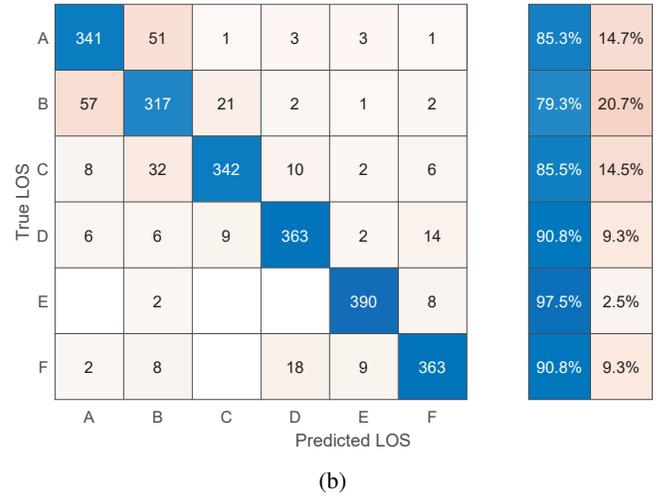
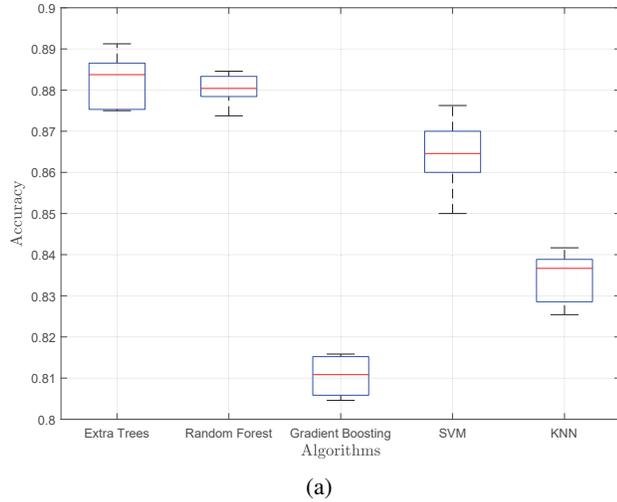


Fig. 9: a) Box-plot distributions of the classification accuracy resulting from the 10-fold cross-validation of the algorithms. b) Confusion matrix of the extra randomized trees algorithm.

TABLE IV: Number of vehicles estimation results on the simulation data.

Algorithm	MAE	WMAPE	Correlation Coef.
Extra Trees	2.21	13.8%	0.96
Random Forest	2.31	14.4%	0.96
Gradient Boosting	2.60	15.9%	0.95
SVR	2.70	16.3%	0.94
KNN	2.48	15.4%	0.94

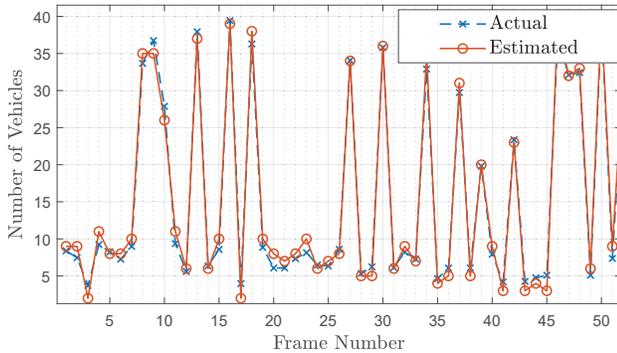


Fig. 10: The actual and estimated number of vehicles for 50 frames in the simulation.

Effect of Signal-to-Noise Ratio (SNR): The signal quality is an important consideration for machine learning algorithms. Especially, the noise in data can prevent knowledge extraction from the data. The noise can impair the models trained with such data since the algorithm can interpret the data noise as a pattern and try to generalize from it [42]. We evaluate how the signal quality (i.e., signal-to-noise ratio) affects our results by varying the transmitter power level. The SNR of k^{th} frame at a receiver can be defined as:

$$SNR_k = \frac{P_R}{\sigma_W^2} = \frac{P_T \|\mathbf{H}_k\|^2}{k_B T N_F B}$$

where P_R is the total received power, and P_T is the transmitter power per subcarrier which is equal to P_{Total}/N with

uniform power allocation across the subcarriers. $k_B = 1.38 \times 10^{-23} J/K$ is the Boltzmann's constant, T is the temperature in Kelvin (K), N_F is the receiver noise figure, and B is the total bandwidth.

To evaluate the performance of the algorithms with different SNR levels, we decrease the total transmit power to 20 dBm from 30 dBm which results in a 10 dB decrease in SNR. The instantaneous SNR ranges from -35.3 dB to 50.6 dB at the first receive antenna with a mean value of 19.7 dB, by using $P_{Total} = 20$ dBm, $B = 10$ MHz, and $T = 300$ K, $N_F = 2$. Table V shows the accuracy, macro-AUC, and macro-F1 results for the LOS prediction problem with 20 dBm transmitter power. We observe that the classification accuracies decrease by 1-5.2% compared to Table III. Since SVM is sensitive to noisy data [43] [44], especially with a low-bias kernel, the performance of SVM weakens significantly among the other algorithms.

Table VI shows the number of vehicles estimation results with 20 dBm transmit power. Again, we observe a performance degradation from the results in Table IV. Fig. 11 shows the cumulative distribution function (CDF) of MAE with the extremely randomized trees algorithm. We note the right shift of the curve that indicates the increasing error when we decrease the transmit power.

Performance with a single antenna: We also evaluate the single antenna scenario in which it is not possible to obtain signals from multiple receivers. To this end, the single antenna scenario where only Rx#1 in Fig. 7 is used for

TABLE V: LOS prediction results with 20 dBm transmit power.

Algorithm	Accuracy	Macro-AUC	Macro-F1
Extra Trees	87.1%	0.98	0.87
Random Forest	86.9%	0.98	0.87
Gradient Boosting	79.7%	0.95	0.79
SVM	81.5%	0.87	0.81
KNN	81.4%	0.94	0.81

TABLE VI: Number of vehicles estimation results with 20 dBm transmit power.

Algorithm	MAE	WMAPE	Correlation Coef.
Extra Trees	2.31	14.53%	0.96
Random Forest	2.42	15.3%	0.95
Gradient Boosting	2.69	16.9%	0.95
SVR	4.92	31.1%	0.75
KNN	2.84	17.9%	0.92

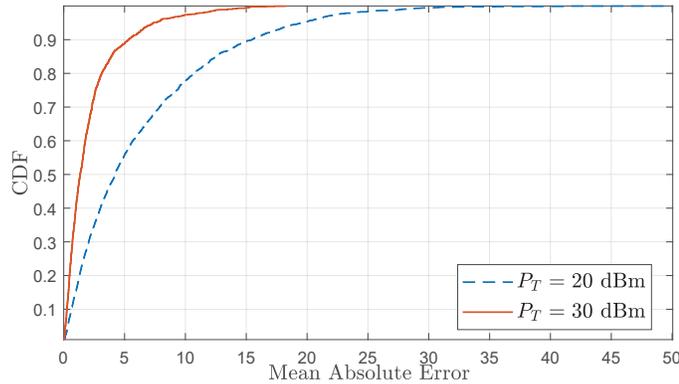


Fig. 11: The CDF of the mean absolute error under different transmit power levels.

data creation by disabling Rx#2 in the simulations. Table VII and Table VIII show the performance of the algorithms with the single antenna for the level of service prediction and the number of vehicles estimation problems, respectively. When we compare the results with Table III and IV, we observe that the results worsen when we use a single antenna. Specifically, the classification accuracies drop by 2.6-3.4%, and WMAPEs of the algorithms increase by 2.9-5% with the single antenna. Hence, we observe a significant improvement in the performance by utilizing a second antenna. With this motivation, we evaluate the performance with a third and a fourth antenna in the simulations. However, the performance improvement is not significant and we observe a diminishing return while increasing the number of antennas.

TABLE VII: LOS prediction results with the single antenna.

Algorithm	Accuracy	Macro-AUC	Macro- F_1
Extra Trees	86.6%	0.98	0.86
Random Forest	86.1%	0.98	0.86
Gradient Boosting	74.3%	0.93	0.74
SVM	82.9%	0.95	0.82
KNN	81.7%	0.94	0.81

TABLE VIII: Number of vehicles estimation results with the single antenna.

Algorithm	MAE	WMAPE	Correlation Coef.
Extra Trees	2.74	17.1%	0.93
Random Forest	2.86	17.9%	0.93
Gradient Boosting	3.16	20.1%	0.92
SVR	3.30	20.5%	0.90
KNN	3.01	18.5%	0.92

C. Real-world Experiments

The feasibility of the proposed approach is tested under a real-world DSRC communication scenario by collecting signal phase and timing (SPaT) messages broadcasted from a roadside unit. In the experiments, a software-defined radio is used for data acquisition and an action camera recorded the road to obtain the ground truth of the number of vehicles. Our experiments include two types of scenarios:

- 1) Static transmitter-static receiver experiments.
- 2) Static transmitter-moving receiver experiments.

The details of the scenarios and the experimental setup are summarized in the following sections.

Hardware and Software Setup

Even though there are commercial IEEE 802.11p modems, they only provide minimal access to the physical layer. For this reason, we prefer working with an open-source software tool, GNU Radio [45], and building our receiver prototype using a software-defined radio. X300 of Ettus Research with a TwinRX daughterboard is used for the experiments. We choose the TwinRX daughterboard since it is sufficient for the DSRC spectrum that is allocated from 5.850 to 5.925 GHz with 10 MHz subchannels [3]. Table IX shows the hardware used in the experiment.

GNURadio is a framework that contains signal processing blocks for software-defined radios. The authors in [46] presented the first OFDM receiver for the GNU Radio which supports IEEE 802.11a/g/p and channel bandwidth up to 20MHz. We modified this implementation to enable us to obtain the CFR values of subcarriers and the correct time of each frame reception. The time information is subsequently used to synchronize the frame reception time with the video records. While the receiver provides different algorithms for channel estimation, we use the least-squares channel estimation algorithm to estimate the channel frequency response vector using the long training sequence of the received frames.

TABLE IX: Hardware used for the experiments.

Component	Type
CPU	Intel Core i7-4720 HQ CPU 2.6GHz
USRP	Ettus X300
RF Daughter Board	Ettus TwinRx
RF Antenna	VERT 2450
Camera	Akaso 12 MP Action Camera 1080p

Static Transmitter-Receiver Setup

This experiment aims to imitate the simulation setup for a single receiver-transmitter pair and evaluate the performance of our approach in the real world. For this experiment, we have equipped a vehicle with our software-defined radio and experimented on the 33 Smart Mobility Corridor. The corridor is a 35-mile highway test corridor that aims to test real-world autonomous and connected vehicle technologies. To this end, roadside units are placed at intersections to broadcast SPaT messages and other safety messages. Fig. 12 shows



Fig. 12: The roadside unit at the intersection. Photographed by City of Dublin [47].



Fig. 13: The vehicle collected SPaT messages 120 meters away from the roadside unit.

the roadside unit that is utilized during our experiments. Our experiment setup is shown in Fig. 13. We have collected SPaT messages from the RSU while our vehicle is parked 120 meters away from the roadside unit. 5700 SPaT messages are collected during the experiment and the vehicles on two lanes close to the parking location are counted from the video record to obtain the actual number of vehicles. The number of vehicles on the road ranges between 0 and 11 which results in a maximum of 46 pc/km/lane on two lanes along the experiment.

Static Transmitter-Receiver Results

We time-stamp each received frame from the RSU to log the time of frame reception. With the help of the timestamps, we determine the number of vehicles at the frame reception time from the video records. Fig. 14 shows the power of the first pilot sub-carrier and the number of vehicles. It is seen that there is a correlation between the power level and the number of vehicles on the road.

First, we predict the level of service on the road between the receive antenna and the roadside unit. The stratified 10-fold cross-validation results of the accuracy, macro-AUC, and macro-F1 are shown in Table X. We obtain accuracy results above 81.7% and the extremely randomized trees algorithm achieves the best accuracy with 91.8%.

Next, we estimate the number of vehicles on two lanes closest to the receive antenna while the number of vehicles on the road ranges between 0 and 11. The results are shown in Table XI. The extremely randomized trees algorithm reaches a mean absolute error of 0.93 vehicles and a WMAPE of

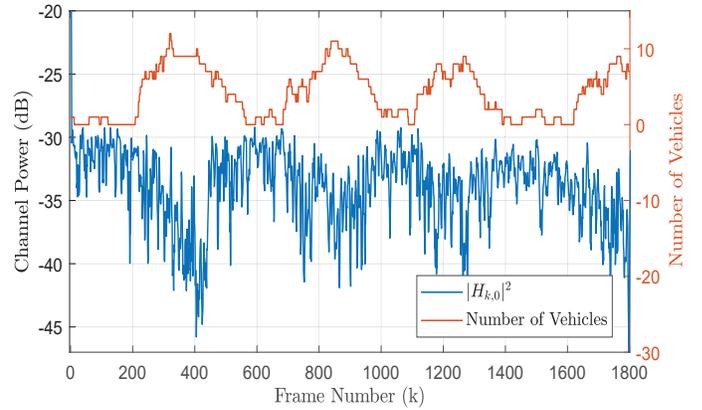


Fig. 14: The power of the first pilot subcarrier and the number of vehicles on the road.

TABLE X: LOS prediction results on the experimental data.

Algorithm	Accuracy	Macro-AUC	Macro-F ₁
Extra Trees	91.8%	0.99	0.91
Random Forest	91.4%	0.99	0.90
Gradient Boosting	81.7%	0.94	0.80
SVM	82.9%	0.95	0.85
KNN	87.3%	0.96	0.86

TABLE XI: Number of vehicles estimation results on the experimental data.

Algorithm	MAE	WMAPE	Correlation Coef.
Extra Trees	0.93	23.4%	0.85
Random Forest	0.98	24.5%	0.85
Gradient Boosting	1.07	26.8%	0.82
SVR	1.22	32.2%	0.80
KNN	0.97	26.7%	0.83

23.4%. Fig. 15 shows the estimated and actual values for 50 received frames obtained using the extremely randomized trees algorithm.

When we compare the simulation and the static transmitter-receiver results on the estimation of the number of vehicles, we observe that the simulation results are better than the experimental results in terms of WMAPE and the correlation coefficient. However, we note that a small error in the exper-

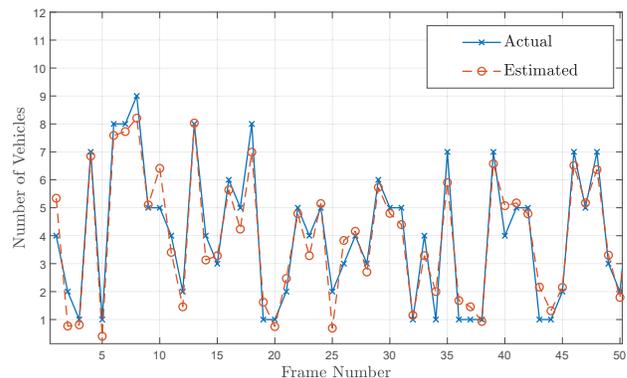


Fig. 15: The actual and estimated number of vehicles for 50 frames.

iment results translates into a large percentage error since the actual number of vehicles on the road is mostly close to 0 and WMAPE is sensitive to such numbers. This is one of the reasons we have a larger percentage error as compared to the simulation results.

Static Transmitter - Moving Receiver Setup

After static transmitter-receiver experiments, we have extended our experiments with a more challenging scenario, a moving receiver scenario. With the help of this experiment, we aim to evaluate the performance of our approach when the signals are received from a moving vehicle at a known location. During the experiments, we drive our vehicle in the right lane toward the location where we collected the static data and record the road using the camera on the vehicle. Fig. 16 shows a caption from the video recorded while we are collecting data along the road. The video can also be found at [5].

Static Transmitter- Moving Receiver Results

Since collecting enough data from a moving vehicle to train a machine learning model may not be feasible for some roads, we propose using models trained with the data collected from a static receiver or simulation data. In this experiment, we use our best model trained with the data obtained from the static receiver scenario to directly estimate the number of vehicles on the road. So, the data obtained from the moving vehicle is merely used for test purposes with the best model.

We drive in the right lane and collect data while driving 5 meters away from the location where we collected the static data. The frames received from the moving vehicle are later used for testing. We have passed on the road 8 times and collected 140 frames in total. In these experiments, we achieve to estimate the number of vehicles with a mean absolute error of 0.95 and a WMAPE of 24%. Fig. 17 shows the estimated number of vehicles of two passes alongside the video records. We round the estimates to the nearest integer to ease the comparison with the actual values. The frames received in the proximity (under 5 meters away from the static receiver location) are indicated in rectangular frames on the plots. We



Fig. 16: SPaT messages collected from a moving vehicle. The received signal and the constellation diagram are seen at the upper left.

observe that the performance of the trained model is affected by the location mismatch. Specifically, the estimation error increases with the location mismatch. When we approach the static receiver location, the outputs of the algorithm get closer to the actual number of vehicles as expected.

VI. CONCLUSION

In this work, we introduce a novel traffic monitoring approach that is based on signal level measurements at a receiver and machine learning techniques. It is a cost-effective and non-intrusive approach. We infer the level of service and the number of vehicles on a roadway using the channel frequency response estimated at a receiver. Since there is no available data, including the traffic conditions and the channel frequency response values, we first create a dataset using a ray-tracing simulator and a traffic simulator. Next, we conduct real-world experiments by collecting DSRC messages broadcast from a roadside unit. Both simulation and experimental results show that the proposed approach is capable of estimating the number of vehicles and predicting the level of service. Our system aims to exploit the infrastructures of vehicular networks, and it doesn't need the deployment of a dedicated device. It can enhance the performance of a current traffic monitoring system when used alongside the system.

ACKNOWLEDGEMENT

We thank Frank Barickman (NHTSA), John Martin (NHTSA), Sughosh Rao (NHTSA), Gavin Howe (NHTSA), and Keith A. Redmill (OSU) for useful discussions in our meetings.

REFERENCES

- [1] Georgia Department of Transportation, "Challenges of the day to day operation of a traffic monitoring program," last accessed 6 November 2022. [Online]. Available: <http://onlinepubs.trb.org/onlinepubs/conferences/2016/NATMEC/Wiegand-TannerPPT.ppt>
- [2] I. Soto, M. Calderon, O. Amador, and M. Uruena, "A survey on road safety and traffic efficiency vehicular applications based on c-v2x technologies," *Vehicular Communications*, vol. 33, p. 100428, 2022.
- [3] J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the united states," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.
- [4] T. R. Board, "Highway capacity manual," *Washington, DC*, vol. 2, p. 1, 2000.
- [5] H. B. Tulay. (2022) Road state inference using wireless signals. [Online]. Available: <https://github.com/bugratulay/Traffic-Inference>
- [6] H. B. Tulay and C. E. Koksall, "Increasing situational awareness in vehicular networks: Passive traffic sensing based on machine learning," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–7.
- [7] K. Nellore and G. P. Hancke, "A survey on urban traffic management system using wireless sensor networks," *Sensors*, vol. 16, no. 2, p. 157, 2016.
- [8] A. M. De Souza, C. A. Brennand, R. S. Yokoyama, E. A. Donato, E. R. Madeira, and L. A. Villas, "Traffic management systems: A classification, review, challenges, and future perspectives," *International Journal of Distributed Sensor Networks*, vol. 13, no. 4, p. 1550147716683612, 2017.
- [9] California Department of Transportation, "Performance measurement system (pems)," last accessed 6 November 2022. [Online]. Available: <https://pems.dot.ca.gov/>
- [10] S. M. Khan, K. C. Dey, and M. Chowdhury, "Real-time traffic state estimation with connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1687–1699, 2017.

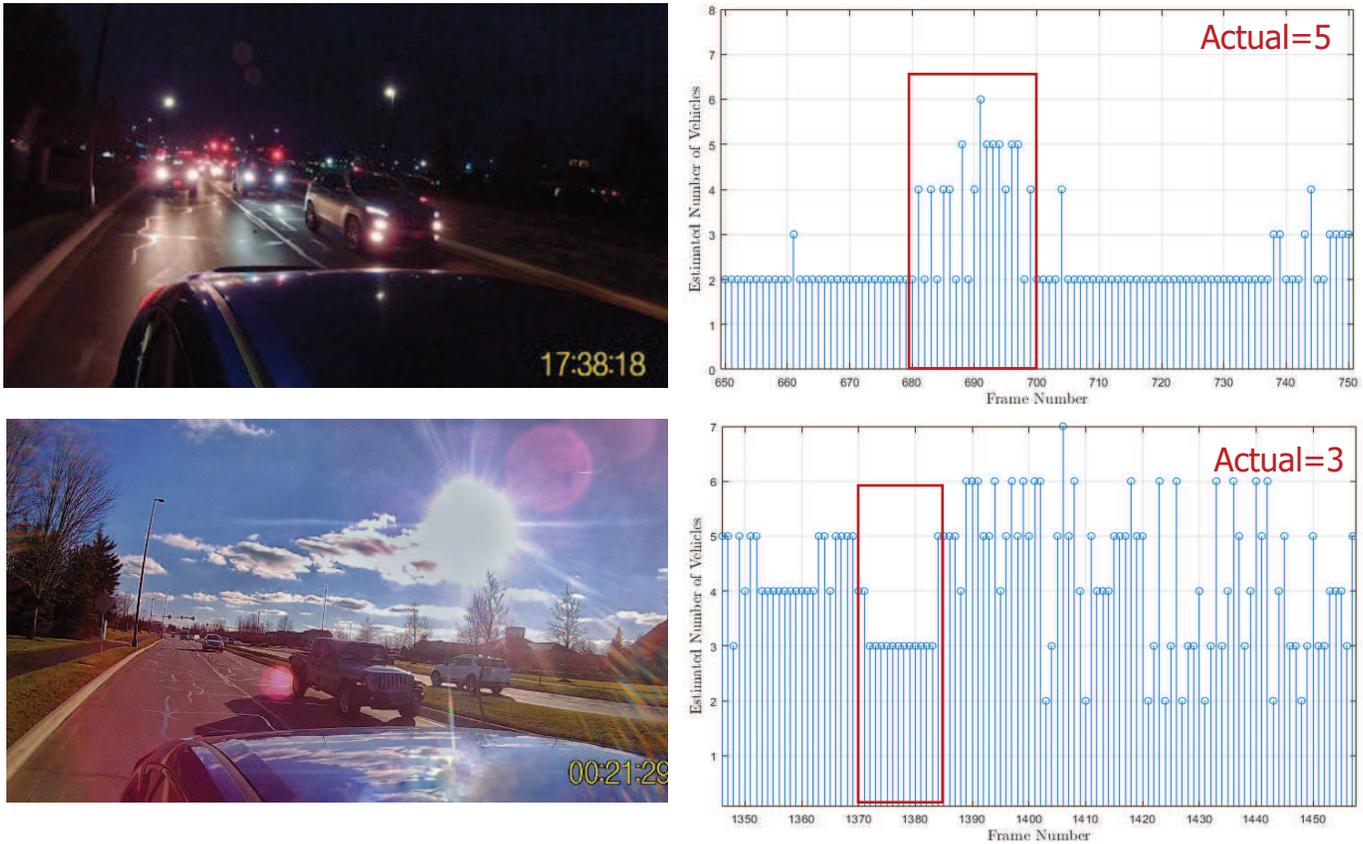


Fig. 17: The results of the two passes using the frames received from the vehicle.

- [11] L.-E. Y. Mimbela, L. A. Klein *et al.*, "Summary of vehicle detection and surveillance technologies used in intelligent transportation systems," 2007.
- [12] S. R. E. Datondji, Y. Dupuis, P. Subirats, and P. Vasseur, "A survey of vision-based traffic monitoring of road intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2681–2698, 2016.
- [13] T. Pamula, "Road traffic conditions classification based on multilevel filtering of image content using convolutional neural networks," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 3, pp. 11–21, 2018.
- [14] D. R. Middleton, R. Longmire, S. Turner, D. Middleton *et al.*, "State of the art evaluation of traffic detection and monitoring systems. volume 1, phases a & b: design," Arizona. Dept. of Transportation, Tech. Rep., 2007.
- [15] K. Kanistras, G. Martins, M. J. Rutherford, and K. P. Valavanis, "A survey of unmanned aerial vehicles (uavs) for traffic monitoring," in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2013, pp. 221–234.
- [16] X. Zhao, D. Dawson, W. A. Sarasua, and S. T. Birchfield, "Automated traffic surveillance system with aerial camera arrays imagery: Macroscopic data collection with vehicle tracking," *Journal of Computing in Civil Engineering*, vol. 31, no. 3, p. 04016072, 2017.
- [17] Carl Andersen, "Connected Vehicle Pilots Program," Oct. 2014. [Online]. Available: https://www.its.dot.gov/research_archives/aeris/pdf/02_CVPilotsProgram.pdf
- [18] United States Department of Transportation, "The national its reference architecture," last accessed 6 November 2022. [Online]. Available: <https://www.arc-it.net/>
- [19] R. Bauza and J. Gozálviz, "Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications," *Journal of Network and Computer Applications*, vol. 36, no. 5, pp. 1295–1307, 2013.
- [20] A. M. de Souza, R. S. Yokoyama, L. C. Botega, R. I. Meneguette, and L. A. Villas, "Scorpion: A solution using cooperative rerouting to prevent congestion and improve traffic condition," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, pp. 497–503.
- [21] Y. Ma, G. Zhou, and S. Wang, "Wifi sensing with channel state information: A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 3, pp. 1–36, 2019.
- [22] M. Won, S. Zhang, and S. H. Son, "Witrafic: Low-cost and non-intrusive traffic monitoring system using wifi," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2017, pp. 1–9.
- [23] M. Won, S. Sahu, and K.-J. Park, "Deepwitrafic: Low cost wifi-based traffic monitoring system using deep learning," in *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2019, pp. 476–484.
- [24] M. Haferkamp, M. Al-Askary, D. Dorn, B. Sliwa, L. Habel, M. Schreckenberg, and C. Wietfeld, "Radio-based traffic flow detection and vehicle classification for future smart cities," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*. IEEE, 2017, pp. 1–5.
- [25] S. Gupta, A. Hamzin, and A. Degbelo, "A low-cost open hardware system for collecting traffic data using wi-fi signal strength," *Sensors*, vol. 18, no. 11, p. 3623, 2018.
- [26] S. Roy, R. Sen, S. Kulkarni, P. Kulkarni, B. Raman, and L. K. Singh, "Wireless across road: Rf based road traffic congestion detection," in *2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011)*. IEEE, 2011, pp. 1–6.
- [27] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.
- [28] Y. Liu, Z. Tan, H. Hu, L. J. Cimini, and G. Y. Li, "Channel estimation for ofdm," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1891–1908, 2014.
- [29] Z. Yun and M. F. Iskander, "Ray tracing for radio propagation modeling: Principles and applications," *IEEE Access*, vol. 3, pp. 1089–1100, 2015.
- [30] Remcom, "Wireless insite 3d wireless prediction software," last accessed 6 November 2022. [Online]. Available: <https://www.remcom.com/wireless-insite-em-propagation-software>
- [31] M. J. Silva, G. I. Silva, C. M. Ferreira, F. A. Teixeira, and R. A. Oliveira,

"Survey of vehicular network simulators: A temporal approach," in *International Conference on Enterprise Information Systems*. Springer, 2018, pp. 173–192.

- [32] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.
- [33] "Traci," last accessed 6 November 2022. [Online]. Available: <https://sumo.dlr.de/docs/TraCI.html>
- [34] H. B. Tulay and C. E. Koksall, "Robust sybil attack detection in vehicular networks," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. IEEE, 2021, pp. 1–7.
- [35] R. K. Pearson, "Outliers in process modeling and identification," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 1, pp. 55–63, 2002.
- [36] S. Sardy, P. Tseng, and A. Bruce, "Robust wavelet denoising," *IEEE Transactions on Signal Processing*, vol. 49, no. 6, pp. 1146–1152, 2001.
- [37] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–, [Last accessed 6 November 2022]. [Online]. Available: <http://www.scipy.org/>
- [38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [39] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.
- [40] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.
- [41] OpenStreetMap contributors, "Planet dump retrieved from <https://planet.osm.org>," <https://www.openstreetmap.org>, 2017.
- [42] S. García, J. Luengo, and F. Herrera, "Dealing with noisy data," in *Data preprocessing in data mining*. Springer, 2015, pp. 107–145.
- [43] A. Atla, R. Tada, V. Sheng, and N. Singireddy, "Sensitivity of different machine learning algorithms to noise," *Journal of Computing Sciences in Colleges*, vol. 26, no. 5, pp. 96–103, 2011.
- [44] M. Sabzekear and S. M. H. Hasheminejad, "Robust regression using support vector regressions," *Chaos, Solitons & Fractals*, vol. 144, p. 110738, 2021.
- [45] GNU Radio Website, Last accessed 6 November 2022. [Online]. Available: <http://www.gnuradio.org>
- [46] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, "An ieee 802.11 a/g/p ofdm receiver for gnu radio," in *Proceedings of the second workshop on Software radio implementation forum*. ACM, 2013, pp. 9–16.
- [47] "Connected vehicle technology installed at two dublin intersections," Jan. 2019. [Online]. Available: <https://dublinohiousa.gov/dev/newsroom/connected-vehicle-technology-installed-at-two-dublin-intersections/>



Can Emre Koksall (S'96–M'03–SM'13) received the M.S. and Ph.D. degrees in electrical engineering and computer science from MIT, in 1998 and 2002, respectively. He is professor at the Electrical and Computer Engineering Department of The Ohio State University since 2006. He researches on wireless communication, cybersecurity, information theory, stochastic processes. He served as an Associate Editor for the IEEE Transactions on Information Theory, the IEEE Transactions on Wireless Communications, and Computer Networks.



Halit Bugra Tulay (S'19) received the B.S. degree in the electrical and electronics engineering from Hacettepe University in 2016. He is currently pursuing the Ph.D. degree in the department of electrical and computer engineering at The Ohio State University. His research interests include wireless communication, cybersecurity and machine learning.