IEEE Open Journal of
Intelligent Transportation
Systems

Received 5 August 2021; revised 17 December 2021; accepted 17 March 2022. Date of publication 21 March 2022; date of current version 31 March 2022.

Digital Object Identifier 10.1109/OJITS.2022.3160934

Wireless-Signal-Based Vehicle Counting and Classification in Different Road Environments

RAOUL KANSCHAT [1], SHIVAM GUPTA [2], AND AURIOL DEGBELO [1]

[1] Institute for Geoinformatics, University of Münster, 48149 Münster, Germany

[2] Bonn Alliance for Sustainability Research, University of Bonn, 53113 Bonn, Germany

CORRESPONDING AUTHOR: A. DEGBELO (e-mail: auriol.degbelo@uni-muenster.de)

ABSTRACT Traffic monitoring is key to modern city planning. However, the costs associated with monitoring devices limit the large-scale deployment of existing traffic monitoring systems. In this article, we propose and evaluate an algorithm to automatically count the number of vehicles that have passed through a low-cost system for traffic monitoring. The system uses deviations in the Wi-Fi signals strength to predict the presence of a vehicle on the road and its type (car, bus). The study further systematically compares six analytical techniques for the classification of detected vehicles. The methods were tested with data from three road scenarios in the city of Münster, Germany. Vehicle classification accuracy ranged from 83% up to 100% in our study. We also observed that a higher Wi-Fi frequency (5 GHz) was superior to the 2.4 GHz for improving the overall vehicle detection and the results of the classification algorithms. The results suggest that the Wi-Fi-based techniques proposed in this study are promising for cost-efficient traffic monitoring in cities in a privacy-preserving manner.

INDEX TERMS Low-cost, machine learning, road traffic monitoring, smart cities, vehicle classification, vehicle counting, Wi-Fi.

I. INTRODUCTION

With the growing population and traffic in urban areas, there is a need to efficiently organize and improve road traffic. Numerous concerns exist when the road traffic is not efficiently planned, such as increased traffic jams, high gasoline consumption, air pollution, noise pollution [1], [2], [3]. Scarce management of traffic flow may result in extended waiting times at traffic lights or in congestion that influences the gasoline consumption as well as the drivers' mood and behavior. Access to timely, reliable traffic data with high spatial coverage of cities can help to minimize or address these problems. Information about the speed and direction of a vehicle or the number of vehicles that use specific roads can be useful to enhance and redirect the traffic flow. This data can further be helpful for more complex spatiotemporal models to infer harmful noise levels or air pollution in cities. Additional applications areas of timely and high-granular traffic data include road safety, traffic control systems, statistics, economic development and federal reporting, which could also benefit significantly from increased data availability.

Despite these potential benefits of timely, high-granular traffic monitoring data, the costs associated with installing and maintaining existing techniques for traffic monitoring (e.g., inductive loop detectors, video analysis) limit their large-scale deployment. For instance, inductive loops cost at least 5.000€.[1] The Georgia Department of Transportation has stated in a report that the installation of a Continuous Counting Stations (CSS) on a two-lane roadway costs approximately $25.000 and can go up to $80.000 [4]. There is thus a need for low-cost techniques that produce reliable traffic data. Along these lines, Gupta *et al.* [5] recently proposed an approach for traffic monitoring, which uses Wi-Fi signals to detect vehicles. Their approach was low-cost (they indicated a cost of less than $50 for it), and they reported promising results for vehicle classification using

The review of this article was arranged by Associate Editor Winnie Daamen.

[1] https://ct-technologyinfo.com/2020/11/09/traffic-detection-systems/ (last accessed: July 31, 2021).

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

Wi-Fi signals. Nonetheless, their work had a few limitations. First, the vehicle counting strategy used led to multiple false positives. It is desirable to avoid these false positive as much as possible, so that vehicle classification is done on relevant entities instead of noise. Second, the scenario of multiple vehicles passing at the same time was not covered. Third, their work reported that the k-nearest neighbor (kNN) technique used led to good classification accuracies, but it is unclear if other models would have performed better. These limitations are addressed in this work, with a focus on the following four research questions.

1) How to automatize the counting of vehicles on the road based on deviations of wireless signals? The emphasis here is on increasing the precision of vehicle counting strategies.
2) What are the respective merits of different types of signals (5GHz vs 2.4GHz) for detection and classification endeavours?
3) How to automatically classify the type of vehicles based on deflections of wireless signals? Here, the aim is to systematically compare several models for the classification tasks.
4) To which extent do models learned from a road environment can be applied to other road environments? This question touches on the generalization of the models. 'Other road environments' here can denote the same road at another time (in which case we talk about temporal generalization) or an entirely new road (in which case we talk about spatial generalization).

The key contributions of this article include (i) an algorithm for automated counting of the vehicles based on Wi-Fi signal; (ii) a comparative evaluation of the 2.4GHz and 5GHz signals for vehicle counting and classification, and (iii) a comparative evaluation of different classification algorithms to analyse Wi-Fi signals. The rest of the paper is organized in the following sections. Section II gives a brief overview of the related work, followed by Section III that describes the methods used during the work. The performances of six analytical approaches used to classify the types of vehicle are described in Section IV. A discussion of the results and their implications in presented in Section V before Section VI concludes the article.

II. RELATED WORK

Gupta *et al.* [5] proposed a grouping of traffic monitoring techniques into five classes: intrusive devices, non-intrusive devices, off-roadways devices, sensor combinations devices, and relatively low-cost devices. Related work in these categories in briefly reviewed in this section.

Intrusive devices: These devices are permanently installed into the pavement. They have high accuracy, but high installation and maintenance costs as well. Inductive loops, magnetic detectors, micro-loop probes, pneumatic road tubes, piezoelectric, and other weigh-in-motion sensors are examples of intrusive devices. Barbagli *et al.* [6] described the disruption of the traffic during the installation and repair as a huge drawback of intrusive devices along with increased costs for their installation. They state that "as a result, those solutions are not suitable for large-scale deployment and hence are restricted to small scale applications" [6].

Non-intrusive devices: These devices are "more reliable and cost-effective" [6] than the intrusive devices. Examples of these include: technologies such as video image processing, microwave radar, laser radar, passive infrared, ultrasonic, passive acoustic array, in which devices are mounted overhead on roadways or roadsides (see [5]). Recently, Asiain and Antolín [7] developed a Low Power Wide Area Network (LPWAN) based system for detecting traffic flow. However, like intrusive devices, most non-intrusive devices also have the downside of being expensive, energy intensive as well as being affected by the environment (e.g., they are prone to errors when environmental conditions like weather or daytime change, see [6]).

Off-roadway devices: These devices use remote sensing techniques for traffic monitoring. These techniques include aircraft and satellite monitoring as well as tracking phones or using probes within the vehicles. They are cheap and easy to deploy and offer a high spatial resolution but raise privacy concerns. Chourasia *et al.* [8] proposed Wi-Fi-based road-side sniffers to examine signals broadcasted by smartphones available inside vehicles to estimate traffic stats of road segments. Hoogendoorn *et al.* [9] conducted a study where they used grayscale imagery, which was recorded by a camera mounted on a helicopter. The study draw "insight into the behavior of drivers during$\cdots$ congestion, and to develop and test theories and models describing congested driving behavior". The assembly could detect and track 98% of the vehicle positions and their dimensions—the spatial resolution of 22cm and a temporal resolution of 8.6 Hertz. Overall, the outcome of their study suggests that the weather had a significant impact on the quality of the data collected. Also, the helicopter was affected by the wind strength, which also influenced the image quality. Another study conducted by Schreuder *et al.* [10] found a drop of reliability in the data from 98% of cars that were detected to 90% "after the weather conditions worsened."

The approach to counting vehicles with the help of probes uses Global Positioning System (GPS) signals to track the vehicles also belongs to the off-roadways devices class. The benefit of the system is that it can be easily applied to existing cars with GPS sensors and works with high accuracy. However, the obvious downside to this approach is the privacy concerns that come with it. Even with anonymous data collection, data mining algorithms have been shown to find out where the individuals live (see [11]). Nanthawichit *et al.* [12] proposed a method for treating probe vehicle data together with fixed detector data in order to estimate the traffic state variables of traffic volume, space mean speed, and density using a macroscopic model along with the Kalman filtering technique (KFT). The model combines data collected by probe vehicles and conventional data in a microscopic traffic flow model. The KFT was used to handle the inconsistencies of data collected by the probe vehicles. The researchers also had some experimental results on traffic prediction, which confirm "that the proposed method can provide reasonable estimation not only for traffic states but also, [· · ·], travel time can be effectively estimated and predicted" [12]. Recently, Ryan *et al.* [13] and Byun *et al.* [14] proposed the application of unmanned aerial vehicle (UAV) for estimating road traffic and vehicle speed automatically by analyzing video feed using machine learning approaches like deep neural network.

Sensor combination devices: These devices try to combine multiple traffic monitoring techniques such as passive infrared with ultrasound and Doppler microwave radar to enhance their overall accuracy. Even though they result in higher accuracy, these combinations are often highly complex to install, making it challenging to deploy them for high spatial coverage data collection processes.

Relatively low-cost devices: Examples of systems mentioned in Gupta *et al.* [5] belonging to this category were continuous-wave radar, computer vision low-cost sensors, and radio-wave technologies. Some of their disadvantages include the need for specialized hardware and procedures, limited computation capability for large data set analysis and privacy concerns. Forren and Jaarsma [15] proposed a low-cost system by using acoustic sensors to monitor the traffic. The study used microphones to count vehicles by analyzing their tire noise. Sen [16] investigated traffic monitoring of non-lane chaotic traffic using the noisiness and excessive use of vehicle honks. The author used the noise level, the number of honks, and their duration to check for congested roads. Two microphones were used along the road to measure the noise variation over time and distance. The system was able to detect different traffic conditions in real-time. However, it was also acknowledged in the study that the honking depends on the driver's behavior and that chaotic traffic jams result in many loud honks, but sometimes the road users also queued up in a traffic jam, which resulted in a quiet but still congested traffic situation. More recently, Kochláň *et al.* [17] proposed a low-cost vision-based traffic monitoring system using Raspberry pi and a high definition camera connected to a car battery in conjunction with a step-down transformer and an antenna to send the data to a remote server. The authors did not state the overall cost of the system. They reported that the system had a low power consumption because it was able to run on a single car battery for over one week. The computer vision algorithms were able to detect 95.7% and classify 93.2% of the vehicles. Ryan *et al.* [13] also proposed the computer vision-based approach assisted with small unmanned aerial systems (sUAS) to capture detailed data for collecting vehicle data.

A novel approach based on dedicated short-range communications (DSRC) signals to measure and classify traffic demand was introduced by Tulay and Koksal [18]. The DSRC is a method where vehicles communicate with other vehicles without the driver knowing the communication. This has the advantage that no wired infrastructure is needed to deploy the system. They used a static transmitter on a roadside, which captured the signals of bypassing cars and were able to distinguish different traffic intensities with an accuracy of 96.3% and 87.6% [18]. Another approach that uses a Microwave Doppler Radar Sensor connected to a Raspberry PI 3 Model B was proposed by Czyżewski *et al.* [19]. Their algorithm results were compared to a pneumatic tube counting system deployed on the same road. The study achieved an accuracy of 90% for vehicle counting, stating that it would be sufficient to use this approach to collect traffic statistics. One problem that the authors acknowledge was the difficulty of detecting vehicles with a high velocity (greater than 100km/h).

The Wi-Fi channel state information (CSI) was used in a field study conducted by Zhang *et al.* [20], Won *et al.* [21] to detect bypassing vehicles. To access the CSI particular chipsets and firmware are necessary, which was not available in the generation of the Raspberry PI which was used during this study. Furthermore this study's focus is on comparing the difference between the two different frequencies and the comparison of the different classification algorithms. The CSI is also used for human indoor gesture detection. They achieved an "average detection accuracy of 99.4% and an average classification accuracy of 91.1%." When a vehicle passes, the system detects peaks with the standard outlier detection technique using CNN. Homchan and Aswakul [22] further proposed the Wi-Fi packet measurement based approach for vehicular traffic monitoring using software-defined mesh network.

Summary: Overall, many efforts are underway to develop novel methods to count and classify traffic in a transport system. The central objective has been to reduce the cost of an individual counting station to allow a large-scale deployment by using existing infrastructures or cheap sensor technology. Our work is in line with this objective. In particular, as mentioned in Section I, we extend and address several limitations of Gupta *et al.* [5] to develop a more automated and reliable Wi-Fi based approach for vehicle detection and classification.

III. METHOD

This section presents an overview of the data and methods used in the study.

A. DATA COLLECTION SCENARIOS

The focus of this study was on investigating the performance of the system in urban areas, and three road types that are common in German cities were chosen to provide a realistic test setting. The first type is the one-way roads, where vehicles are only allowed to drive on one lane and in one direction. The second type is a road with two lanes and the vehicles driving in one direction, and the third type is the roads with two lanes where the vehicles drive in both directions. The speed limits of all scenarios are 50 km/h. Highways were excluded at this point for safety reasons. The measurements took place in November and December 2020 in the city of Münster (North Rhine-Westphalia, Germany). The data collection activities were approved by the city council of Münster, and necessitated the deployment of additional traffic signs to ensure the safety of the drivers. The three scenarios are now described in detail.

Scenario 1 - The one-way street: For this scenario, the Austermannstraße was chosen. It is a busy road in the city, with cars and busses as the most common vehicle types passing. The two units of the hardware system were placed approximately 6 meters apart across the road. Two measurements were collected from this road on 16.12.2020 and 17.12.2020. The first-day measurement started at 16:00 and ended at 19:00 with weather conditions of 9°C, cloudy and 87% relative humidity 1015 hPa. The second-day measurement started at 16:30 and ended at 19:30 with weather conditions of 10°C, cloudy, 79%, and 1021 hPa. The sunset time for the first measurement was at 16:17 and for the second at 16:18.

Scenario 2 - Two lanes one direction: Here, the Corrensstraße was chosen. It is a large but not a busy road in front of a university building. The most common vehicle types using this road were cars and busses. The hardware units were approximately 11 meters apart across the road. Three measurements were collected on 19.11.2020 (16:15-17:30), 24.11.2020 (17:00-18:00) and 25.11.2021 (17:20-18:40). The sunset times for the measurements were at 16:32, 16:26 and 16:25, respectively. One noteworthy observation for this particular scenario was that few busses drove extremely slowly through the system because the hardware installation was located in proximity to a bus stop, and as the busses started, they needed some time to accelerate. The weather conditions for the first measurement were 2°C, cloudy, 85% relative humidity and 1027 hPa; the second measurement 0°C, cloudy, 90% and 1020 hPa, and for the third measurement, the conditions were 5°C, cloudy, 82% relative humidity and 1018 hPa.

Scenario 3 - Two lanes two directions: The third scenario with two lanes and two directions was located at the Mendelstraße with a distance between the two parts of the installation by approximately 10 meters across the road. The two most frequent vehicle classes over this road were cars and busses with the possibility of two cars passing at the same time. Two measurements with 2.5 hours length were collected on 08.12.2020 (starting 16:00) and 09.12.2020 (starting 16:20). The sunset times for both measurements were at 16:17. The weather conditions for the first measurement were 1°C, foggy, 97% relative humidity and 1006 hPa and for the second 0°C, foggy, 100% and 1010 hPa.

B. DATA COLLECTION APPROACH

1) SENDER AND RECEIVER

The data was collected using a system extended and much improved from [5]. A key novelty here was the addition of the 5 GHz frequency. The sending unit consists of multiple elements which are portrayed in Figure 2. A "TP-Link WLAN-Router model Archer C7 AC1750" was used, which uses the dual band frequencies 2.4 GHz and 5 GHz. The SSIDs of the router was separated into the 2.4 GHz and 5 GHz frequencies to allow the receiver to distinguish between the two signals. The Receiver, which is displayed in Figure 3 consists of a Raspberry Pi 3 B, two USB WI-FI dongles, two dual band directional antennas and a light sensor. The graphical representation of whole the installation is illustrated in Figure 1 and the annotated configuration of the installation is presented in Figure 2 and 3.

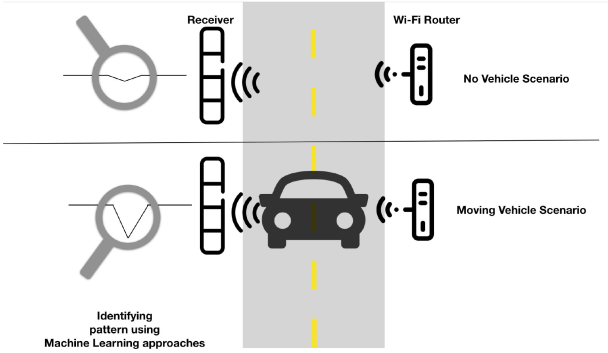

FIGURE 1. Graphical representation of the installed system and its working (Icons made by Freepik.com).

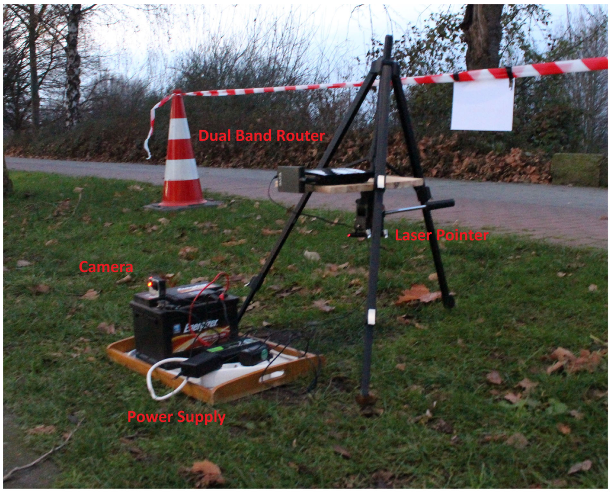

FIGURE 2. A picture of the system's sending unit. The different hardware components (Router, laser pointer, camera and power supply) are labeled.

The signal strength of the frequencies was measured in decibel (dB). The setup helped us capture the interruptions in the signal caused by vehicle driving between the sender and the receiver on the road.

In contrast to systems discussed in the previous section, our prototype can be installed within a couple of hours without stopping or distracting the traffic flow on the road. To install the system, the two units need to be placed on opposite sides of the road. As the hardware system can be accessed and restarted without interfering with the traffic, detecting errors and restarting the system can be performed remotely.

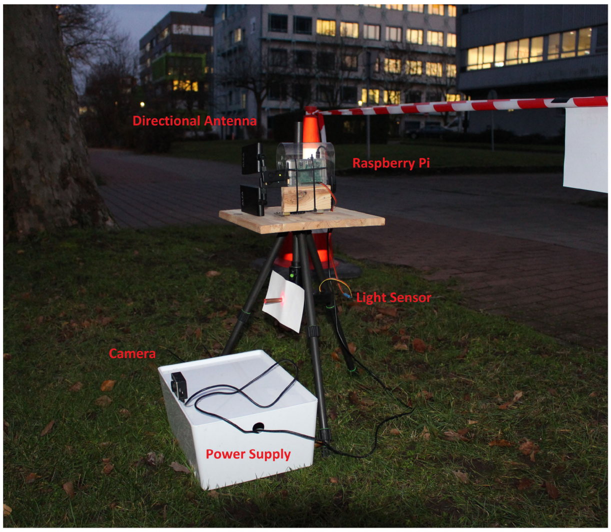

FIGURE 3. A picture of the receiving unit of the system. The different hardware components (directional antennas, Raspberry Pi, light sensor, camera and power supply) are labeled.

As it consists of multiple components, defective parts can be replaced individually with no effect on traffic flow, making it more promising in terms of easy maintenance and cost-effectiveness. Considering the safety approvals required before installation and the fact that the data collection can only be initiated with some programming skills, the system in the current state is not suitable for citizen science initiatives. The approximate cost of the installation was 200€, which is a lot cheaper than the state of the art monitoring systems [4] or even newer approaches [21].

2) GROUND TRUTH

A laser tripwire system was used to collect ground truth data about vehicles passing or not, and when. In addition, the types of vehicle passing were recorded through cameras mounted on the ground (max 20 cm above the ground). That way, only the wheels of the cars were recorded to ensure that no personal data about the drivers was collected. This was a mandatory requirement by the local authorities. The video data were subsequently manually annotated with the class of vehicle passing to generate the ground truth data.

A subsequence of the data is shown in Figure 4. In the visualization, an extract of the different collected time series is shown for the time that a car drove by the system. In both frequencies a noticeable drop in signal strength is visible. Corresponding to the signal drop, the laser was interrupted, which can be seen in Figure 4 (Bottom). The system, which recorded 10 datapoints per second picked up three datapoints of the disrupted laser, which correspond to 3 datapointts (middle) and 4 datapoints (top).

C. VEHICLE COUNTING APPROACH

As mentioned in Section I, one of the aim of this work is to reduce noise during the vehicle detection task. In previous work [5], *peaks* were used to execute the detection of vehicles. A peak starts when the Wi-Fi signal strength drops more than a given threshold and ends when it recovers to the starting dB value also within the threshold. A peak has a time window, i.e., the duration between its start and end. As the main idea of Gupta *et al.* [5]'s method was to use the point in time the signal recovers (i.e., returns to its value before deflection) as an indicator for the end of a peak, their approach is called *recovery-based method* to peak extraction at this point. The threshold of 2dB was used for peak detection in [5] and that value was also used in this work while implementing the recovery-based approach as a baseline for comparison. One weakness of the recovery-based approach

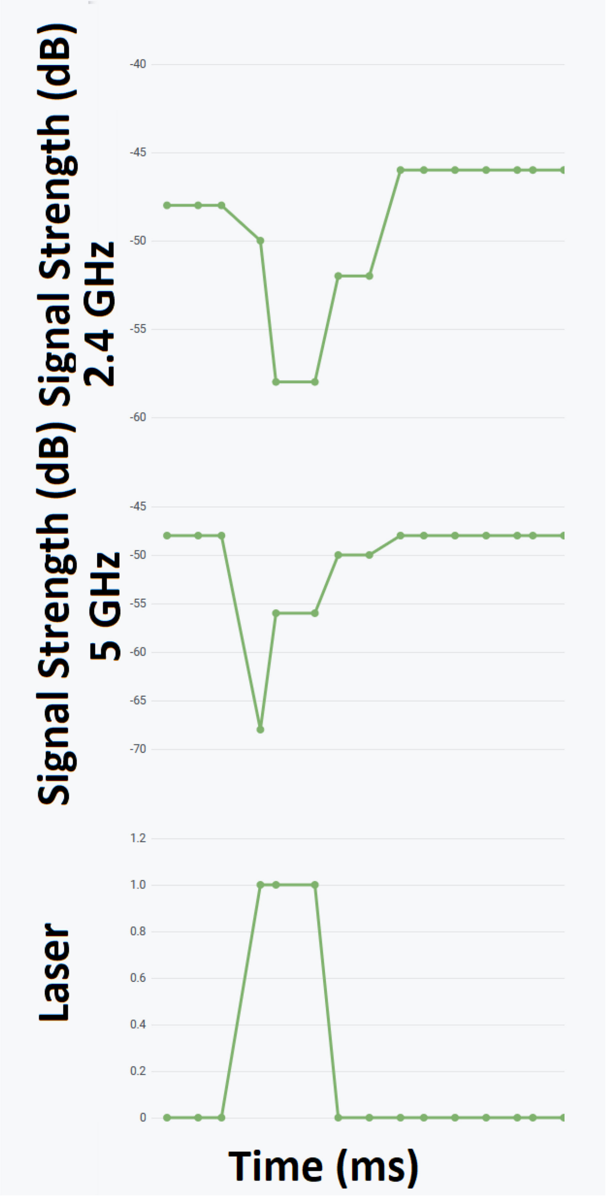

FIGURE 4. Visualization of the change in signal strength in dB when a car drives through the system. Top: 2.4 GHz signal strength; Middle: 5 GHz signal strength; Bottom: Laser Value (0 = The laser is not interrupted, 1 = The Laser is interrupted).

is its limited ability to detect more complex patterns in signals. For example, in some cases when a bus passes the system, the signal strength recovers momentary and goes down again, causing the algorithm to detect two different busses instead of one.

To enhance the threshold method and make it more robust to fluctuations, we use a new method called *median-based peak extraction*. We begin by first computing the median of the time series, which we then use as a reference value. In addition a 3 data point time window was also used to allow the detection of more complex patterns in the signal stream. By this new approach new peaks are identified when the mean of the signal strength within the windows differs more than 1.5 dB to the median. The peak ends when the mean of the signal strength of the three data points recovers back within the 1.5 dB. The threshold of 1.5 dB was chosen after multiple iterations during the data analysis. Table 1 shows the results across all three scenarios. Note that the detection algorithm does not perform any classification at this stage. Since the outcomes of the extraction step are the input values for the classification methods, the detection algorithm determines the maximum vehicle count that can potentially be classified. The assignment of the classes of vehicles to the peaks has been done manually to generate the data in this table. As the table illustrates, the median-based peak extraction allowed a more complex and robust peak extraction and is more resilient to noise detection across scenarios. Nonetheless, it might still be prone to noise error when the idle signal strength changes or a vehicle stands for a long time between the sensors. In all three scenarios, the median-based approach has reduced the number of NaV (false positive) dramatically, leading to substantial improvements in the precision and overall F-score, especially when applied to 5GHz signals.

TABLE 1. Comparison of the peak extraction methods: Nav = Not a vehicle indicates a false positive.

Mendelstraße		2.4GHz	5 GHz
	Vehicle class	Count	Count
Ground Truth	NaV	0	0
	Cars	1416	1416
	Busses	6	6
	2 Cars	37	37
2 dB Threshold	NaV	14966	6645
	Cars	1095	1381
	Busses	6	6
	2 Cars	27	37
Precision		0.070	0.176
Recall		0.773	0.976
F-score		***0.129***	***0.299***
Interval	NaV	2151	3251
	Cars	1131	1364
	Busses	6	6
	2 Cars	37	37
Precision		0.353	0.302
Recall		0.805	0.964
F-score		***0.491***	***0.460***
Austermanstrasse		2.4GHz	5 GHz
	Vehicle class	Count	Count
Ground Truth	NaV	0	0
	Cars	1306	1306
	Busses	16	16
	2 Cars	0	0
2 dB Threshold	NaV	18196	1252
	Cars	1198	1230
	Busses	16	16
	2 Cars	0	0
Precision		0.063	0.499
Recall		0.918	0.943
F-score		***0.117***	***0.652***
Interval	NaV	4493	238
	Cars	1264	1302
	Busses	16	16
	2 Cars	0	0
Precision		0.222	0.847
Recall		0.968	0.997
F-score		***0.361***	***0.916***
Correnstraße		2.4GHz	5 GHz
	Vehicle class	Count	Count
Ground Truth	NaV	0	0
	Cars	320	320
	Busses	25	25
	2 Cars	0	0
2 dB Threshold	NaV	9719	9186
	Cars	194	320
	Busses	25	25
	2 Cars	0	0
Precision		0.022	0.036
Recall		0.635	1.0
F-score		***0.043***	***0.070***
Interval	NaV	978	152
	Cars	140	284
	Busses	25	25
	2 Cars	0	0
Precision		0.144	0.670
Recall		0.478	0.896
F-score		***0.222***	***0.767***

D. VEHICLE CLASSIFICATION TECHNIQUES

Once the peaks were extracted from the frequencies with the new approach discussed above, a classification of the peaks was intended. The goal here is to find out the type of vehicle that led to the disturbance in the Wi-Fi signal. In order to accomplish the classification process different algorithms were used and compared. Initially, the classification algorithm k-nearest neighbor classifier utilized in the previous work by [5] was implemented and used as a comparison baseline. Afterwards different preprocessing techniques were applied to the data and various parameters for the kNN were investigated. In addition, a Random Forest classifier was used to classify the peaks. Ultimately, a new approach using the matrix profile was implemented in our study for the vehicle classification. A brief description of each method, as well as the rationale for their choice is presented next.

1) K-NEAREST NEIGHBOR

K-nearest neighbor is a “simple but effective” [23] classification method, suitable as a basis of comparison for following classification algorithms because it “should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data” [24]. This classification method which is a supervised machine learning technique [25] learns by simply storing the training samples and their features. The classification is then based on the Euclidean distance between the training and test samples [24]. For each sample that will be classified the method computes the Euclidean distances between the testing sample and all the training samples. The "k" samples that have the least distance to the tested sample then determine the predicted class based on a majority voting [26].

In a first step, the peaks have been extracted from the two frequencies (see Section III-C). From these peaks the amplitude and the length of the peak define the features which, were used to classify the vehicles. These features are the same that were used in previous work [5]. The amplitude is determined by the maximum deflection of the signal strength during the peak. The length of the system indicates how long the signal needed to recover to its standard strength. The resulting data is then split into 70% training data and 30% testing data and the kNN is trained and the test data is classified. After testing the values 3, 5 and 7 for "k", the value 3 was chosen because it resulted in the highest accuracy.

To enhance this method further, firstly, an additional preprocessing step was conducted by applying feature scaling to the input data. The different features were normalized which can impact the overall accuracy of the kNN. The standardization method from scikit-learn's preprocessing package [27] was used to perform this step. Secondly the Euclidean Distance (ED) of the peak was added as a third feature for the kNN algorithm for classification. The ED was computed by adding up the distances between each of the datapoints of a peak.

Since there was a greater amount of cars compared to the other vehicle classes passing the sensors during the measurement, an oversampling technique was also used to improve the accuracy of the kNN even further. Chawla *et al.* [28] stated that a class imbalance is present when the number of instances of the different classes are not approximately equal. This class imbalance problem results in wrong detection of the dominant class which in this case are cars [29]. The fact other vehicle classes occur less often corresponds to the explanation by Chawla *et al.* [28] that "real-world data sets are predominately composed of "normal" examples with only a small percentage of "abnormal" or "interesting" examples" [29]. The Synthetic Minority Oversampling Technique (SMOTE) provided by the python package "imblearn.over_sampling" [30] was used to calculate the additional samples of the minority classes. Only classes with a minimum instance occurrence of 4 were included. Tests with the oversampling technique did not result in significant improvements, and are not reported in this article.

2) RANDOM FOREST

Random forest is a supervised machine learning classification and regression method that uses a divide and conquer approach [31]. Multiple classifiers are used to achieve a robust classification. The number of classifiers, which are decision trees in the case of random forest, decide the accuracy and computation time for the algorithm. More trees lead to more computation time [32]. For each new instance, the individual trees vote on a class, and the decision is then based on the majority of votes [33]. Previous work [34] reported that random forest models perform well on a wide variety of classification problems. The fact that many weak classifiers most of the time perform better than a single classifier [35], and that this method, in contrast to the kNN is efficient and able to "operate quickly over larger datasets" [33] is the reason that we chose to utilize it for our study.

In our study, the random forest used the same input data as the new kNN approach to make a comparison possible. First, the peaks for both frequencies are extracted using the new counting method. The used features are the peaks duration, maximum amplitude, and Euclidean distance. The features were also normalized. The dataset was then split into the same training and testing data using the same random seed, which was used for the kNN classification to generate meaningful results. The scikit-learn's random forest classifier was then used with 100 trees. Increasing the trees further up to 512 did not change the accuracy of the classification.

3) MATRIX PROFILE

The matrix profile technique was proposed in 2016 by Yeh *et al.* [36] and is based on the all-pairs-similarity-search. This method, also known as similarity join, retrieves the nearest neighbors for each object in a data collection. In the context of time series analysis, the method can be used to identify patterns or outliers. Yeh *et al.* [36] created a "simple, fast, parallelizable and parameter-free" algorithm called Scalable Time series Anytime Matrix Profile (STAMP) which uses the concept of the matrix profile. Their algorithm was, at the time of publishing in 2016, the fastest to detect motifs and anomalies in a time series. We found this technique to be relevant and worth of exploration in this study, as Wi-Fi-based signals can be modeled as time series. To utilize the matrix profile, the first step is to count the peaks in the time series. A peak in the context of the matrix profile would be defined as a discord or anomaly. However, multiple peaks with similar patterns occur within the time series which is the reason why the algorithm is not able to detect them as an anomaly. This is called the "twin freak problem" [37]. This problem was investigated by He *et al.* [37] suggesting that the algorithm "fails to identify rare subsequences when it occurs more than once in the time series."

Due to the "twin freak problem," the matrix profile cannot simply be used for the peak extraction and classification. Therefore a different approach was used in our study. The python library "STUMPY" [38] provides different algorithms for motif and discord detection using the concept of the matrix profile. The initial step is to split the data into a training set which was chosen to be 70% of the data and a testing set with the remaining 30%. Next, the new peak extraction was used on the testing dataset, and a distance

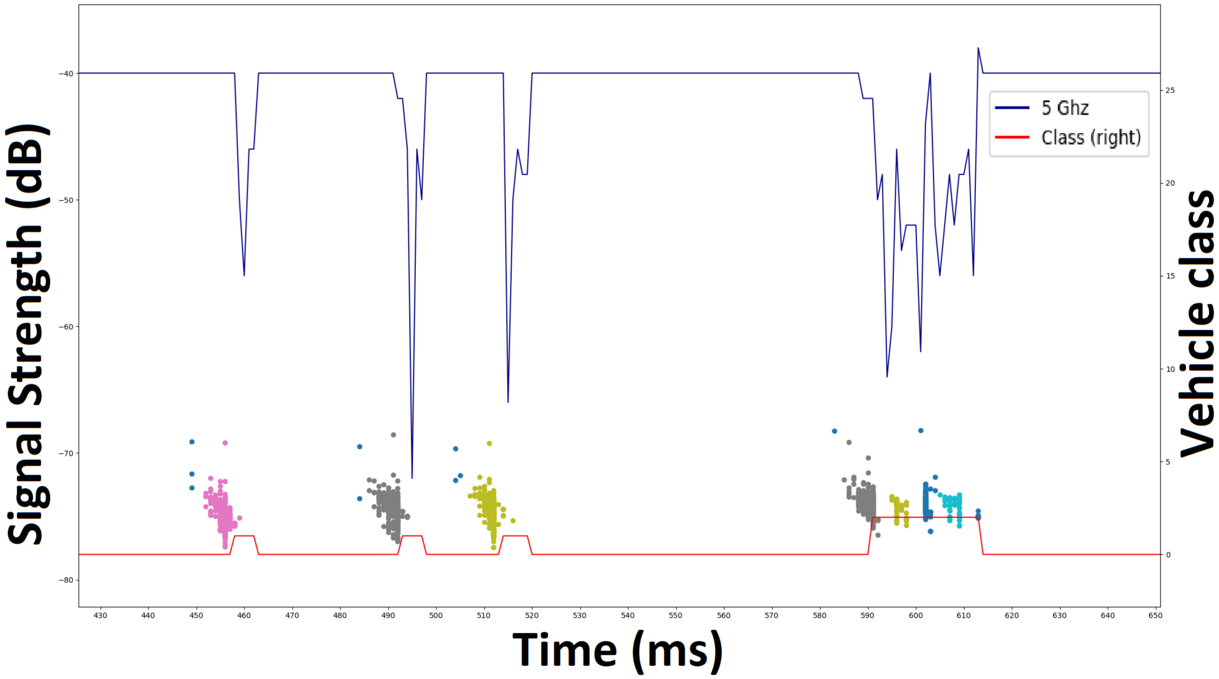

FIGURE 5. Matrix profile clustering approach. The signal strength of the 5 GHz frequency is visualized in blue. The vehicle class shown in red indicate 3 cars (left) with a class of and a bus (right) with a class of 2. The clustering algorithm found 7 different clusters from the votes.

profile to the training dataset for each peak was computed, which is basically an all-pairs-similarity-search. In this distance profile, the global minimum is then sought-after. The result is the index of the most similar subsequence in the training dataset to the extracted peak. If a class was annotated within the length of subsequence, it was assigned to be the predicted class for the particular peak.

To make the detection more robust not only the global minimum of the distance profile was chosen, but also the three lowest minima in the testing set were used in conjunction with a majority voting. With the previously explained approach using the counting method in conjunction with the pattern search, a major advantage of using the matrix profile is left out. All the previous methods need a counting step and a classification step. The matrix profile provides the possibility to combine those steps. A first attempt utilizing this advantage but avoiding the “twin freak problem” is proposed in the following text.

The aim was to first use a manual extraction of peaks from the training dataset. This extraction uses the same technique that was used to find the ground truth of vehicles. Therefore, no further changes had to be made to the data and the extraction is not prone to errors except the ones that were made during the data annotation or sensor errors. For each manual extracted peak the distance profile to the testing dataset is then computed. Thereafter all local minima were extracted from the distance profile. For each minima, a vote with the extracted peak’s class and the distance profiles value at that specific index is assigned to the same index on the testing dataset. All votes then form a cluster on the testing dataset. A Density-Based Spatial Clustering (DBSCAN) was used for robustness and to ignore some outliers. Each cluster contains the votes of classes and has to be assigned to one peak (see Figure 5).

4) SUMMARY

In summary, six algorithms were compared during the study:

- kNN (baseline): kNN without any preprocessing using the peaks duration and maximum amplitude as features;
- kNN (normalization): kNN baseline with duration and amplitude as features, both normalized;
- kNN (ED+normalization): kNN baseline with amplitude, duration, and Euclidean distance as a third feature added. All three features were normalized;
- Random Forest (ED+normalization): random forest with amplitude, duration, and Euclidean distance as features, all three normalized;
- Matrix profile 1: STUMPY Fast Pattern Search for each peak. The global minimum was chosen to for the classification.
- Matrix profile 2: STUMPY Fast Pattern Search for each peak. Three lowest minimum where chosen in conjunction with a majority voting for the classification.

E. GENERALIZABILITY

Ferguson [39] summarizes generalizability as the combination of internal and external validity of the findings. Validity is a criteria for the quality of results and can be separated into internal and external validity. To interpret the results of an experiment, internal validity is necessary. However, the “external validity pertains to the generalizability of the

TABLE 2. **Accuracy results of the different classification methods for the measurement at the 08.12.2020 at the Mendelstraße.**

Mendelstraße	2.4GHz				5GHz			
08.12.2020	vehicle class	f-score	support	overall accuracy	vehicle class	f-score	support	overall accuracy
kNN	NaV	0.90	250	**0.87119**	NaV	0.98	502	**0.96676**
	Car	0.84	173		Car	0.95	214	
	Bus	x	x		Bus	0.00	1	
	2 Cars	0.00	4		2 Cars	0.29	5	
kNN	NaV	0.89	250	**0.86651**	NaV	0.98	502	**0.95845**
(normalization)	Car	0.84	173		Car	0.94	214	
	Bus	x	x		Bus	0.00	1	
	2 Cars	0.00	4		2 Cars	0.09	5	
kNN (ED +	NaV	0.90	250	**0.87353**	NaV	0.97	502	**0.95291**
normalization)	Car	0.85	173		Car	0.93	214	
	Bus	x	x		Bus	0.00	1	
	2 Cars	0.10	4		2 Cars	0.00	5	
Random Forest	NaV	0.89	250	**0.86417**	NaV	0.98	502	**0.96944**
(ED +	Car	0.84	173		Car	0.95	214	
normalization)	Bus	x	x		Bus	x	1	
	2 Cars	0.00	4		2 Cars	0.00	5	
Matrix profile 1	NaV	0.56	112	**0.59783**	NaV	0.66	770	**0.56765**
	Car	0.65	160		Car	0.42	172	
	2 Cars	0.00	4		2 Cars	0.00	4	
Matrix profile 2	NaV	0.38	112	**0.59058**	NaV	0.67	770	**0.57928**
	Car	0.70	160		Car	0.44	172	
	2 Cars	0.00	4		2 Cars	0.00	4	

treatment effect to other populations, settings, treatment variables, or measurement variables" [39].

Generalizability in the context of the proposed system has multiple dimensions. Based on Ferguson's [39] definition, hardware and software generalizability are desirable in the different scenarios or settings. Also, different measurement variables like weather conditions could be investigated regarding generalizability. For the system to allow a large scale deployment generalizability is also important. For the generalizability of our proposed system, we tested it in three different road environments. An additional step was to investigate the generalizability of the different counting and classification models. Temporal and spatial generalizability were tested. Temporal generalizability relates to whether the counting and classification methods and their parameters apply to the same scenario but at a different time, or if an adjustment to the parameters or learning data is needed for a new measurement. The spatial generalizability related to whether or not the counting and classification methods and models can be used in other, entirely different road scenarios.

To test generalizability, first the training data from the 16.12.2020 at the Austermannstraße was used to train the models of the classification methods. Then testing data from the 17.12.2020 at the Austermannstraße was classified (temporal generalizability). For the spatial generalizability, the training data from the Mendelstraße and the testing data from the Austermannstraße were used. The Mendelstraße is the two lane street and the three vehicle classes car, bus and two cars at the same time occur. On the Austermannstraße which has only one lane, only cars and busses occurred. This test generalizes the counting and classification methods spatially because the training and testing data were collected in different locations with a different distance between the sensors, different numbers of lanes for each scenario as well as having different vehicle classes appearing.

IV. RESULTS

Tables 2 to 6 show the results. The support values in the tables indicate the amount of instances that were tested for each method.

A. MENDELSTRAßE

This scenario, which had two lanes two directions, had two separate measurements of about 3 hours each. Two noteworthy observations are that the distance between the sensors was smaller and that some cars slowed down significantly after seeing the traffic warning signs. In total 1416 cars and 6 busses were counted. The results are shown in Table 2.

Mendelstraße - 08.12.2020: A first measurement at the Mendelstraße was conducted on 08.12.2020. A few observations from the table is that the overall accuracies of the machine learning approaches classifying the 5 GHz peaks are almost 10% higher than the 2.4 GHz classifications. Regarding the classification techniques, the normalization did not improve classification accuracies over the baseline (quite unexpectedly). The use of the Euclidean distance increased the accuracy for the 2.4 GHz slightly, but not for

TABLE 3. Accuracy results of the different classification methods for the measurement at the 09.12.2020 at the Mendelstraße.

Mendelstraße	2.4GHz				5GHz			
09.12.2020	vehicle class	f-score	support	overall accuracy	vehicle class	f-score	support	overall accuracy
kNN	NaV	0.86	265	**0.81995**	NaV	0.63	27	**0.88235**
	Car	0.75	142		Car	0.93	168	
	2 Cars	0.00	4		2 Cars	0.00	9	
kNN (normalization)	NaV	0.86	265	**0.81752**	NaV	0.65	27	**0.88725**
	Car	0.74	142		Car	0.94	168	
	2 Cars	0.00	4		2 Cars	0.00	9	
kNN (ED + normalization)	NaV	0.87	265	**0.82725**	NaV	0.63	27	**0.88235**
	Car	0.75	142		Car	0.93	168	
	2 Cars	0.00	4		2 Cars	0.00	9	
Random forest (ED + normalization)	NaV	0.88	265	**0.83455**	NaV	0.65	27	**0.88235**
	Car	0.76	142		Car	0.93	168	
	2 Cars	0.33	4		2 Cars	0.00	9	
Matrix profile 1	NaV	0.77	208	**0.67619**	NaV	0.45	27	**0.8**
	Car	0.51	107		Car	0.88	128	
	2 Cars	x	x		2 Cars	x	x	
Matrix profile 2	NaV	0.72	208	**0.66032**	NaV	0.40	27	**0.82581**
	Car	0.57	107		Car	0.90	128	
	2 Cars	x	x		2 Cars	x	x	

the 5 GHz. The accuracies of both matrix profile techniques were much lower, compared to the previous models.

Mendelstraße - 09.12.2020: A second measurement at the Mendelstraße was conducted on 09.12.2020. The results of classification methods are shown in Table 3. Again, the 5 GHz frequency counted more cars, and the classification accuracy is higher in all cases. The highest overall accuracy was scored by the kNN with normalized feature values with the 5 GHz frequency with 0.88725. Furthermore, the amount of 2 cars again were very low, and none was detected. The feature normalization decreased the accuracy of the kNN algorithm for 2.4 GHz but improved it for 5GHz. After adding the peaks Euclidean distance, the accuracy was increased for 2.4 GHz but decreased for 5GHz. The random forest had similar results to the kNN classification with a slight improvement in 2.4 GHz. This time, the matrix profile results were closer to the machine learning results with 0.67619 for 2.4 GHz and 0.82581 for 5 GHz.

B. AUSTERMANNSTRAß E

The next scenario, which was a one way street had two separate measurements of approximately three-hour long each. Two observations worth mentioning are that the distance between the sensors was smaller and that some cars slowed down significantly after seeing the traffic warning signs. In total, 1306 cars and 18 busses were counted.

Austermannstraße - 16.12.2020: The results of the first measurement at the Austermannstraße are shown in Table 4. Higher accuracy than the previous two-lane street can be seen, with the enhanced kNN method even scoring 100% accuracy in the 5 GHz peaks. The number of counted cars for both frequencies this time was the same, but the 2.4 GHz frequency had more NaV peaks extracted. In the 2.4 GHz frequency, no bus was detected compared to the 2 correctly detected bus instances in the higher frequency. This time the normalization improved the accuracy for the kNN, and the Euclidean distance also increased the accuracy of the 2.4 GHz classification. The random forest became slightly worse compared to the kNN baseline in 2.4 GHz and the same accuracy for 5 GHz. The matrix profile's overall accuracies were again worse than the machine learning approaches. For the 5 GHz peaks, the accuracy of the method got closer to the previous methods, with the improved matrix profiling version scoring 0.96226. However, in the 2.4 GHz frequency, the majority voting system decreased the accuracy.

Austermannstraße - 17.12.2020: The results of the classification methods for the second measurement at the Austermannstraße can be found in Table 5. Similar to the first measurement's results in Table 4 the difference in peak extraction of the frequencies are visible. The lower frequency 693 peaks without a vehicle class were extracted compared to zero at the higher frequency. Furthermore, 8 cars and 6 busses, which the 2.4 GHz frequency did not measure, were perceived in the higher frequency. The best accuracy was achieved with the kNN method in combination with the feature normalization with 0.99061. The baseline algorithms overall accuracy was 0.97648 for 2.4 GHz and 0.98592 for 5 GHz. Normalizing the data decreased the result for the lower frequency but increased it for the 5 GHz. This time adding the peaks Euclidean distance to the input feature increased both frequencies accuracies. The random forest classifier worked best for the 2.4 GHz peaks and was especially good for bus detection. The enhanced matrix profiling classification scored 0.98889 on the 5 GHz peaks

TABLE 4. Accuracy results of the different classification methods for the measurement at the 16.12.2020 at the Austermannstraße.

Austermannstraße	2.4 GHz				5 GHz			
16.12.2020	vehicle class	f-score	support	overall accuracy	vehicle class	f-score	support	overall accuracy
kNN	NaV	1.00	290	**0.99366**	NaV	0.93	8	**0.99482**
	Car	0.99	183		Car	1.00	183	
	Bus	x	x		Bus	1.00	2	
kNN	NaV	1.00	290	**0.99577**	NaV	1.00	8	**1.00**
(normalization)	Car	1.00	183		Car	1.00	183	
	Bus	x	x		Bus	1.00	2	
kNN (ED +	NaV	1.00	290	**0.99788**	NaV	1.00	8	**1.00**
normalization)	Car	1.00	183		Car	1.00	183	
	Bus	x	x		Bus	1.00	2	
Random forest	NaV	0.99	290	**0.98732**	NaV	1.00	8	**0.99482**
(ED +	Car	0.99	183		Car	1.00	183	
normalization)	Bus	x	x		Bus	0.67	2	
Matrix profile 1	NaV	0.73	392	**0.66298**	NaV	0.00	3	**0.93082**
	Car	54	148		Car	0.97	153	
	Bus	0.00	3		Bus	0.00	3	
Matrix profile 2	NaV	0.59	392	**0.54512**	NaV	0.00	3	**0.96226**
	Car	0.50	148		Car	0.98	153	
	Bus	0.00	3		Bus	0.00	3	

TABLE 5. Accuracy results of the different classification methods for the measurement at the 17.12.2020 at the Austermannstraße.

Austermannstraße	2.4GHz				5GHz			
17.12.2020	vehicle class	f-score	support	overall accuracy	vehicle class	f-score	support	overall accuracy
kNN	NaV	0.99	693	**0.97648**	NaV	x	x	**0.98592**
	Car	0.95	197		Car	0.99	205	
	Bus	0.00	3		Bus	0.77	9	
kNN	NaV	0.99	693	**0.97536**	NaV	x	x	**0.99061**
(normalization)	Car	0.94	197		Car	1.00	205	
	Bus	0.00	3		Bus	0.86	9	
kNN (euclid	NaV	0.99	693	**0.97648**	NaV	x	x	**0.98592**
normalization)	Car	0.95	197		Car	0.99	205	
	Bus	0.33	3		Bus	0.77	9	
Random forest	NaV	0.99	693	**0.97760**	NaV	x	x	**0.98592**
(ED +	Car	0.95	197		Car	0.99	205	
normalization)	Bus	0.50	3		Bus	0.77	9	
Matrix profile 1	NaV	0.93	660	**0.89303**	NaV	x	x	**0.98333**
	Car	0.77	169		Car	0.99	176	
	Bus	0.00	3		Bus	0.67	4	
Matrix profile 2	NaV	0.64	660	**0.56370**	NaV	0.00	x	**0.98889**
	Car	0.45	169		Car	1.00	176	
	Bus	0.00	3		Bus	0.67	4	

bypassing the baseline algorithms accuracy with similar vehicle instances.

C. CORRENSSTRAβE

The measurements at the Corrensstraße on the 19. 24. and 25. of November in 2020 were chronologically the first ones and had different problems during data collection and analysis. Overall only 320 cars and 35 busses were counted due to a low traffic density. For the analysis, the number of instances are split into three different measurements resulting in a very low number of vehicles per measurement, making the results not very meaningful. Therefore, only the results of the first measurement on 19.11.2020 are shown in Table 6. The first problem, which occurred, was a signal loss in the 2.4 GHz frequency. The Wi-Fi disconnected multiple times for a brief period of time. Another problem was the location of the setup, which was next to a bus station. This resulted in a huge variation of duration that the busses needed to drive by the system. The results of the three measurements also exhibit better counting capability in the 5 GHz frequency.

TABLE 6. Accuracy results of the different classification methods for the measurement at the 19.11.2020 at the Corrensstraße.

Correnssstraße	2.4GHz				5GHz			
19.11.2020	vehicle class	f-score	support	overall accuracy	vehicle class	f-score	support	overall accuracy
kNN	NaV	0.91	33	**0.85366**	NaV	0.75	10	**0.66667**
	Car	0.44	6		Car	0.50	8	
	Bus	0.80	2		Bus	x	x	
kNN (normalization)	NaV	0.91	33	**0.82926**	NaV	67	10	**0.61111**
	Car	0.40	6		Car	0.53	8	
	Bus	0.00	2		Bus	x	x	
kNN (ED + normalization)	NaV	0.93	33	**0.87805**	NaV	0.67	10	**0.61111**
	Car	0.44	6		Car	0.53	8	
	Bus	1.00	2		Bus	x	x	
Random forest (ED + normalizattion)	NaV	0.85	33	**0.73170**	NaV	0.70	10	**0.66667**
	Car	0.33	6		Car	62	8	
	Bus	0.00	2		Bus	x	x	
Matrix profile 1	NaV	0.84	25	**0.7**	NaV	0.84	8	**0.78571**
	Car	x	x		Car	0.67	6	
	Bus	0.00	5		Bus	x	x	
Matrix profile 2	NaV	0.79	25	**0.63333**	NaV	0.82	8	**0.71428**
	Car	x	x		Car	0.60	6	
	Bus	0.00	5		Bus	x	x	

TABLE 7. Models' generalizability performances.

Generalizability		
	2.4GHz	5 GHz
	accuracy	accuracy
kNN (normalized + ED)		
Temporal	0.96190	0.98276
Spatial	0.89218	0.88428
Random forest		
Temporal	0.96750	0.97806
Spatial	0.90416	0.88083
Matrix profile 2		
Temporal	0.56456	0.97753
Spatial	0.67069	0.93935

The overall accuracies in all methods were much lower, ranging from 0.47863 to 0.97917.

D. TEMPORAL AND SPATIAL GENERALIZABILITY

The results of the model generalizability are shown in Table 7. Only the kNN (ED+normalization) and the matrix profile 2 models were tested for generalizability at this point, because they led to the best results overall (Tables 2 to 6). The temporal generalizability was tested by extracting the peaks from the first measurement at the Austermannstraße and using them as the training data for the classifications. Then the peaks from the second measurement at the Austermannstraße were extracted and used as testing data. For the kNN and random forest algorithm, both frequencies could get a classification accuracy between 0.96190 and 0.96750 for 2.4 GHz and 0.97806 to 0.98276 for 5 GHz. The enhanced matrix profiling approach was able to score an overall accuracy of 0.97753 using the higher frequency. For spatial generalizability testing, data from Mendelstraße was used, and the peaks from Austermannstraße were tested. A significant drop in accuracy ranging from 6% to 10% is noticeable for the two machine learning approaches. The enhanced matrix profiling method, however, was able to score 0.97753 accuracy.

V. DISCUSSION

The research questions are now revisited, before we discuss implications and limitations.

A. REVISITING THE RESEARCH QUESTIONS

1) HOW TO AUTOMATIZE THE COUNTING OF VEHICLES ON THE ROAD BASED ON DEVIATIONS OF WIRELESS SIGNALS?

The results of the peak extraction algorithms show that peaks, which do not correspond to a vehicle type, are counted and passed to the classification algorithm. Because the NaVs (i.e., false positives) are also passed to the classification methods, it is desirable to avoid them as much as possible. The median-based approach suggested and implemented in this work reduced the amount of irrelevant peaks extracted in the three scenarios both for the 2.4 GHz and 5 GHz signals by several orders of magnitude (Table 1). A median-based peak extraction is thus more effective for vehicle counting than a recovery-based approach and is an important step towards automated Wi-Fi-based vehicle counting. In particular, it is more robust to noise, and this means greater efficiency for the Wi-Fi based traffic monitoring approach as a whole.

2) WHAT ARE THE RESPECTIVE MERITS OF DIFFERENT TYPES OF SIGNALS (5GHZ VS 2.4GHZ) FOR DETECTION AND CLASSIFICATION ENDEAVOURS?

The results have shown that the higher Wi-Fi frequency (5 GHz) was superior to the 2.4 GHz, improving the overall amount of counted vehicles as well as the results of the classification algorithms. Furthermore, the higher frequency shows greater robustness in urban areas. For the detection of multiple vehicles passing the system simultaneously, the preliminary results indicate that this was not possible. However, the low amount of occurrences do not allow a final conclusion in that regard. The advantages of the 2.4 GHz is to have provided slightly better accuracy results, when it comes to the spatial generalizability of two models (kNN and random forest).

3) HOW TO AUTOMATICALLY CLASSIFY THE TYPE OF VEHICLES CONSIDERING THE SHIFT OF PATTERNS IN WIRELESS SIGNALS?

The first lesson learned from Tables 2 to 6 is that the use of kNN along with normalization and Euclidean distance as a feature performed best for vehicle classification using the 2.4 GHz signal. This technique consistently performed best across scenarios. For the classification using the 5 GHz signal, the results are more inconclusive as no method consistently provided the highest accuracy value across scenario. kNN techniques seem to have a slight advantage, but the random forest approach has provided largely comparable results. A second lesson from the tables is that distance between the sensors seem to matters. The accuracy results obtained at the Austermanstrasse, which is a one-lane street, are typically about 10% higher than the results in the other scenarios. This suggests that the shorter the distance, the better the results, but the impact of distance on the accuracy results is a matter that needs to be more systematically investigated in future work. A surprising observation from the tables, though, is that accuracy values of classifiers may differ by some order of magnitude (about 5-7%) in a given scenario (i.e., Mendelstrasse). It is unclear why this has been the case, given that the two measurements were only 24 hours apart, with relatively similar conditions (see Section III-A). This too, needs a more systematic investigation in future work. A third lesson, is that matrix profiling techniques have some potential, especially using in conjunction with the 5GHz frequencies. How to make them more robust across scenarios and signal frequencies is an interesting issue for further work.

Fourth, the choice of the location matters. Overall, the Correnstrasse led to lower accuracy values, compared to other scenarios. There were two possible reasons for this. The first one, already mentioned, was the fact that the system was installed next to a bus station. Therefore the busses that stopped at the station were picked up by the system during acceleration. In contrast to the busses that did not stop at the station, the duration of the peaks was a lot longer, making classification very difficult. Second, this scenario was located in a more populated area than the previous ones. Thus, there was a higher number of devices using the 2.4 GHz frequency, leading to the signal being possibly more disturbed (there were even signal losses during the data collection process for the 2.4 GHz channel).

At last, it must be mentioned that the scope of all observations made in this subsection is limited to the classification of cars and busses. There were too few instances of multiple vehicles passing at the same time in the data collected to offer solid conclusions of the merits of the techniques on 2 cars. The best vehicle classification accuracies in the scenarios ranged from 83.4% to 99.8% (2.4 GHz) and from 78.6% to 100% (5 GHz). The received signal strength was used in this study to derive the amplitudes used as features (see Figure 4), but CSI provides more information than received signal strength (see [40]). It is thus likely that using CSI as the basis for the classification could improve the accuracies even further, but this remains to be tested empirically.

4) TO WHICH EXTENT DO MODELS LEARNED FROM A ROAD ENVIRONMENT CAN BE APPLIED TO OTHER ROAD ENVIRONMENTS?

When a trained model on one scenario was used to classify data from the same scenario at a different point in time, the accuracy values were comparable (Table 7). Thus, model reuse *within* scenarios is possible and sensible. However, when a trained model on one scenario was used to classify peaks from another scenario, the accuracy of the method was significantly reduced. This suggests that reuse of models across different scenarios comes with the cost of accuracy, or put differently that each model needs to be trained separately for every different road scenario. Even the enhanced matrix profiling method was able to score 0.97753 accuracy, which would be a reasonable classification accuracy for a traffic monitoring system. However, it had the downside of not offering consistent results in the internal validity, which is also necessary for generalizability.

B. IMPLICATIONS

Overall, the proposed traffic monitoring system fulfills many desirable requirements. The desirable characteristics mentioned in Barbagli *et al.* [6] are the capability of large-scale deployment, being passive and operating at low power, being cheap, easy to install and maintain. A large-scale deployment was not tested in this work; nonetheless, a few comments can be made about regarding this. The main hurdle for a large-scale traffic monitoring deployment is the high cost of an individual monitoring device. The proposed system being low cost is a first step in the direction of a large-scale deployment. The fact that the proposed approach is using Wi-Fi signals and not interfering with the traffic, as well as having the possibility to detect the traffic automatically, makes it a passive system. The power consumption of the setup was not quantified in this work; however, it was able to operate during multiple hours with a car and motorcycle battery. The amount of power consumption can therefore be compared to the approach of Kochláň *et al.* [17] who indicated that their system is operating at low power due to the fact that a car battery is enough to satisfy its energy needs. Furthermore, the system was installed in under one hour for each measurement without stopping the traffic, demonstrating how easy it is to install. In addition, Wi-Fi signals are not influenced by the day-night cycles and operate in many different weather conditions. The accuracies obtained for the temporal generalizability suggest that in the absence of more recent models, models trained on previous data may still provide useful results. Models trained on two lanes, two directions data can also be used for one-lane, one direction streets, but at the cost of a decrease in accuracy.

C. LIMITATIONS

The analysis is subject to multiple limitations as a result of a dependency chain. First of all, the peak extraction is limited to the actual raw data that is collected by the system. If no signal change is recorded on either frequency although a vehicle passed the system, a peak extraction is not possible. Furthermore, the classification is limited to the outcomes of the peak extraction. Two extraction methods were compared during the work. Both methods have different parameters, which have to be tuned for each scenario. One input feature for the machine learning approach is the duration that a vehicle spends between the two system's units. This means it is strongly influenced by the driver's behavior. This became clear when some vehicles drove very slowly at the Austermannstraße because the driver saw the traffic signs as well as at the Corrensstraße where some busses had to accelerate after stopping at a bus station. It follows that the vehicles' speed and length has to be taken into account. During this work all streets had a speed limit of 50km/h. Data visualization during the analysis showed that some short cars as well as cars exceeding the speed limit were not picked up by the system. Increasing the temporal resolution of the data collection could help mitigate this issue. The results also showed the limits of using the 2.4 GHz frequency. In total, the lower frequency picked up less cars and had problems with signal loss, which could be traced back to the amount of devices using it in a city. Also, the number of vehicles passing through the streets used in the scenario is arguably relatively small, and could be increased through data collection in non-residential areas. Finally, the manual data annotation, which was labor intensive, limited the amount of vehicles that could be used during the data analysis. This also led to an under representation of certain vehicle types.

VI. CONCLUSION AND FUTURE WORK

This work has deployed a low-cost, Wi-Fi-based system for traffic data monitoring in three different urban scenarios. It then introduced and thoroughly evaluated a new algorithm for the automatic counting of vehicles (car, busses). The work also evaluated the performance of different models for the classification of vehicles (car, busses). Using normalization and three features (amplitude, duration, Euclidean distance) have helped to improve the accuracy over the baseline used in previous work. Our results also suggest that a trained model can be used to classify data collected about the same road, but at a different point in time.

There are a few directions for future work that can be mentioned. First, as mentioned above, it would be interesting to investigate the impact of spatial distance on the accuracy results more systematically. Also, the impact of noise (e.g., competing signals on the 2.4 GHz frequency) could be looked into more closely. In addition, the study outcomes can be well utilized in consideration with work of [41], improving public transport systems in cities. Developing over the work of [42], the present study could be extended to evaluate the speed of the vehicles on the road. Finally, the matrix profiling technique has shown promise. The density-based clustering algorithm DBSCAN was able to detect clusters of votes for individual cars but had problems combining the votes for busses. In future work, a method for deciding on a cluster's class needs to be implemented. Furthermore, the problem that multiple clusters are detected within the time window of a longer vehicle presents interesting challenges for future work.

REFERENCES

[1] A. Stevanovic, J. Stevanovic, K. Zhang, and S. Batterman, "Optimizing traffic control to reduce fuel consumption and vehicular emissions: Integrated approach with VISSIM, CMEM, and VISGAOST," *Transp. Res. Rec.*, vol. 2128, no. 1, pp. 105–113, 2009.

[2] P. E. K. Fiedler and P. H. T. Zannin, "Evaluation of noise pollution in urban traffic hubs—Noise maps and measurements," *Environ. Impact Assess. Rev.*, vol. 51, pp. 1–9, Feb. 2015.

[3] J. Khan, M. Ketzel, K. Kakosimos, M. Sørensen, and S. S. Jensen, "Road traffic air and noise pollution exposure assessment—A review of tools and techniques," *Sci. Total Environ.*, vol. 634, pp. 661–676, Sep. 2018.

[4] K. Wiegand, *Challenges of the Day-to-Day Operation of a Traffic Monitoring Program*, Georgia Dept. Transp., Atlanta, GA, USA, 2016, p. 25.

[5] 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.

[6] B. Barbagli, G. Manes, R. Facchini, and A. Manes, "Acoustic sensor network for vehicle traffic monitoring," in *Proc. 1st Int. Conf. Adv. Veh. Syst. Technol. Appl.*, 2012, pp. 24–29.

[7] D. Asiain and D. Antolín, "LoRa-based traffic flow detection for smart-road," *Sensors*, vol. 21, no. 2, p. 338, 2021.

[8] A. Chourasia, B. R. Tamma, and A. A. Franklin, "Wi-Fi based road traffic monitoring system with channel hopping functionality," in *Proc. Int. Conf. COMmun. Syst. NETw. (COMSNETS)*, 2021, pp. 680–684.

[9] S. P. Hoogendoorn, H. J. Van Zuylen, M. Schreuder, B. Gorte, and G. Vosselman, "Microscopic traffic data collection by remote sensing," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1855, no. 1, pp. 121–128, Jan. 2003.

[10] M. Schreuder, S. P. Hoogendoorn, H. J. Van Zulyen, B. Gorte, and G. Vosselman, "Traffic data collection from aerial imagery," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Shanghai, China, 2003, pp. 779–784.

[11] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Enhancing security and privacy in traffic-monitoring systems," *IEEE Pervasive Comput.*, vol. 5, no. 4, pp. 38–46, Oct.–Dec. 2006.

[12] C. Nanthawichit, T. Nakatsuji, and H. Suzuki, "Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1855, no. 1, pp. 49–59, Jan. 2003.

[13] A. Ryan, C. Ai, M. Knodler, and C. Fitzpatrick, "Evaluation of small unmanned aerial system highway volume and speed-sensing applications," *IET Intell. Transp. Syst.*, vol. 15, no. 1, pp. 84–97, 2021.

[14] S. Byun, I.-K. Shin, J. Moon, J. Kang, and S.-I. Choi, "Road traffic monitoring from UAV images using deep learning networks," *Remote Sens.*, vol. 13, no. 20, p. 4027, 2021.

[15] J. F. Forren and D. Jaarsma, "Traffic monitoring by tire noise," in *Proc. Conf. Intell. Transp. Syst.*, Nov. 1997, pp. 177–182.

[16] R. Sen, "Different sensing modalities for traffic monitoring in developing regions," Ph.D. thesis, Dept. Comput. Sci., Indian Inst. Tech., Bombay, India, 2014, p. 212.

[17] M. Kochláň, M. Hodoň, L. Čechovič, J. Kapitulík, and M. Jurečka, "WSN for traffic monitoring using raspberry Pi board," in *Proc. IEEE Federated Conf. Comput. Sci. Inf. Syst.*, 2014, pp. 1023–1026.

[18] H. B. Tulay and C. E. Koksal, "Road traffic monitoring using DSRC signals," Dec. 2020, *arXiv:2012.13448*.

[19] A. Czyżewski, J. Kotus, and G. Szwoch, "Estimating traffic intensity employing passive acoustic radar and enhanced microwave doppler radar sensor," *Remote Sens.*, vol. 12, no. 1, p. 110, Jan. 2020.

[20] S. Zhang, M. Won, and S. H. Son, "WiTraffic: Non-intrusive vehicle classification using WiFi: Poster abstract," in *Proc. 14th ACM Conf. Embedded Netw. Sens. Syst. CD-ROM*, 2016, pp. 358–359.

[21] M. Won, S. Sahu, and K.-J. Park, "DeepWiTraffic: Low cost WiFi-based traffic monitoring system using deep learning," in *Proc. IEEE 16th Int. Conf. Mobile Ad Hoc Sens. Syst. (MASS)*, 2019, pp. 476–484.

[22] M. Homchan and C. Aswakul, "Testing of vehicular traffic monitoring technique by using WiFi packet measurement in software-defined wireless mesh network," in *Proc. Int. Symp. Electr. Electron. Inf. Eng.*, 2021, pp. 190–194.

[23] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN model-based approach in classification," in *On the Move to Meaningful Internet Systems: CoopIS, DOA, and ODBASE* (Lecture Notes in Computer Science), R. Meersman, Z. Tari, and D. C. Schmidt, Eds. Heidelberg, Germany: Springer, 2003, pp. 986–996.

[24] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, Feb. 2009.

[25] M. M. Islam, H. Iqbal, M. R. Haque, and M. K. Hasan, "Prediction of breast cancer using support vector machine and K-Nearest neighbors," in *Proc. IEEE Region 10 Humanitarian Technol. Conf. (R10-HTC)*, Dhaka, Bangladesh, Dec. 2017, pp. 226–229.

[26] Y. Song, J. Huang, D. Zhou, H. Zha, and C. L. Giles, "IKNN: Informative K-nearest neighbor pattern classification," in *Knowledge Discovery in Databases: PKDD* (Lecture Notes in Computer Science), J. N. Kok, J. Koronacki, R. L. de Mantaras, S. Matwin, D. Mladenič, and A. Skowron, Eds. Heidelberg, Germany: Springer, 2007, pp. 248–264.

[27] "6.3. Preprocessing Data—Scikit-learn 0.24.1 Documentation." [Online]. Available: https://scikit-learn.org/stable/modules/preprocessing.html (Accessed: Jul. 31, 2021).

[28] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[29] W. Liu and S. Chawla, "Class confidence weighted kNN algorithms for imbalanced data sets," in *Proc. Pacific–Asia Conf. Knowl. Discov. Data Min.*, 2011, pp. 345–356.

[30] "imblearn.over_sampling.SMOTE—Imbalanced-Learn 0.3.0.dev0 Documentation." [Online]. Available: http://glemaitre.github.io/imbalanced-learn/generated/imblearn.over_sampling.SMOTE.html (Accessed: Jul. 31, 2021).

[31] G. Biau and E. Scornet, "A random forest guided tour," *TEST*, vol. 25, no. 2, pp. 197–227, 2016.

[32] P. Bonissone, J. M. Cadenas, M. C. Garrido, and R. A. Díaz-Valladares, "A fuzzy random forest," *Int. J. Approximate Reason.*, vol. 51, no. 7, pp. 729–747, Sep. 2010.

[33] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How many trees in a random forest?" in *Machine Learning and Data Mining in Pattern Recognition* (Lecture Notes in Computer Science, 7376), D. Hutchison *et al.*, Eds. Heidelberg, Germany: Springer, 2012, pp. 154–168.

[34] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *J. Mach. Learn. Res.*, vol. 15, no. 90, pp. 3133–3181, 2014.

[35] N. Sirikulviriya and S. Sinthupinyo, "Integration of rules from a random forest," in *Proc. Int. Conf. Inf. Electron. Eng.*, vol. 6, 2011, pp. 194–198.

[36] C.-C. M. Yeh *et al.*, "Matrix profile I: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets," in *Proc. IEEE 16th Int. Conf. Data Min. (ICDM)*, Barcelona, Spain, Dec. 2016, pp. 1317–1322.

[37] Y. He, X. Chu, and Y. Wang, "Neighbor profile: Bagging nearest neighbors for unsupervised time series mining," in *Proc. IEEE 36th Int. Conf. Data Eng. (ICDE)*, Dallas, TX, USA, Apr. 2020, pp. 373–384.

[38] "The Matrix Profile—Stumpy 1.8.0 Documentation." [Online]. Available: https://stumpy.readthedocs.io/en/latest/Tutorial_The_Matrix_Profile.html (Accessed: Jul. 31, 2021).

[39] L. Ferguson, "External validity, generalizability, and knowledge utilization," *J. Nurs. Scholarship*, vol. 36, no. 1, pp. 16–22, 2004.

[40] Y. Ma, G. Zhou, and S. Wang, "WiFi sensing with channel state information: A survey," *ACM Comput. Surv.*, vol. 52, no. 3, pp. 1–36, Jul. 2019. [Online]. Available: https://dl.acm.org/doi/10.1145/3310194

[41] R. Zhang *et al.*, "WiFi sensing-based real-time bus tracking and arrival time prediction in urban environments," *IEEE Sensors J.*, vol. 18, no. 11, pp. 4746–4760, Jun. 2018.

[42] S. Li *et al.*, "WiFi-based device-free vehicle speed measurement using fast phase correction MUSIC algorithm," in *Proc. Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber Phys. Soc. Comput. (CPSCom) IEEE Smart Data (SmartData) IEEE Congr. Cybermatics (Cybermatics)*, 2020, pp. 93–99.