LiVeR: Lightweight Vehicle Detection and Classification in Real-Time

CHANDRA SHEKHAR, Computer Science, Indian Institute of Technology Bhubaneswar, Bhubaneswar, India
JAGNYASHINI DEBADARSHINI, Computer Science, Indian Institute of Technology Bhubaneswar, Bhubaneswar, India
SUDIPTA SAHA, Computer Science, Indian Institute of Technology Bhubaneswar, Bhubaneswar, India

Detection of vehicles and their classification is a significant component of wide-area monitoring and surveillance, as well as Intelligent-Transportation. Existing solutions tend to employ heavy-weight infrastructure and costly equipment, as well as largely depend on constant support from the cloud through round-the-clock internet connectivity, and uninterrupted power supply. Moreover, the existing works mainly concentrate on localized measurement and do not discuss their efficient integration to address the problem over a wide area. For practical use in an outdoor environment, apart from being technically sound and accurate, a solution also needs to be cost-effective, lightweight, easy-to-install, flexible, low-overhead, easily maintainable, as well as, self-sufficient as much as possible. However, fulfilling all these goals together is a challenging task. In this work, we propose an IoT-assisted strategy, LiVeR, to accomplish it. For self-sufficient on-the-fly classification in resource-constrained low-power IoT devices, LiVeR minimizes not only the computational requirements but also the energy consumption, which enables sustained operation in a hostile outdoor environment for a considerably long time solely based on battery power. Through extensive studies based on outdoor measurement and trace-based simulation on empirical data, we demonstrate that LiVeR classifies vehicles of small, medium, and large size with an accuracy of 91.3% up to 98.8%, 92.3% up to 98.5%, and 93.8% up to 98.8%, respectively, for single-lane traffic. We also demonstrate that LiVeR spends only about one-third of the number of RF packets to achieve vehicle detection and classification compared to the state-of-the-art RF-based solution, considerably extending the lifetime of the system.

CCS Concepts: • Computer systems organization → Embedded systems; Redundancy; Robotics; • Networks → Network reliability.

Additional Key Words and Phrases: Vehicle Detection, Vehicle Classification, RF-Assisted, Synchronous Transmission

1 Introduction

Detection and classification of vehicles is a significant component in various monitoring and surveillance-based applications, e.g., monitoring of vehicular movements in an institute-campus or residential areas in a city, study of the behavior of the vehicles for detection and prediction of anomalous situations, automatic control of gates, or several other similar actions related to management and control of traffic. For the successful realization and round-the-clock execution of any such outdoor application over a wide area, apart from being technically sound and accurate, it is also required to be highly self-sufficient. Dependence on the cloud as well as stable power supply not only makes a solution relatively weak w.r.t. the adverse outdoor environment but also prone to various possible targeted cyber-attacks. As per the recent trend, a solution for vehicle detection and classification needs...
to make minimum use of the cloud and should be able to sustain over battery power for a long time. However, to fulfill these constraints, a solution mechanism has to be quite simple without compromising detection and classification accuracy, which is quite challenging. In this work, we aim to design and develop a solution that can balance these conflicting requirements.

The existing works for vehicle detection and classification can be broadly classified into three categories: (a) Camera-based, (b) Special sensor-based and (c) RF-based.

**Camera-based solutions:** A major fraction of the works are based on image/video-camera [5]. They exploit computer vision to accomplish the goal. These works need to employ a set of costly equipment over the roadside and assume the availability of a constant supply of power to operate [31]. They also use sophisticated ML/AI tools for image/video-based detection of the vehicles and their classification, which require either the presence of powerful computing facilities on the spot (i.e., roadside) or sufficient network bandwidth for dynamic assistance from the remote cloud servers [3]. Cloud assistance has been a quite common resource in most such solutions. However, for various significant reasons, edge-based solutions are much preferred compared to these cloud-dependent systems [17].

**Sensor-based solutions:** A set of works are based on special sensors, e.g., loop-detectors [20], accelerometer [28], magnetometer [48], infrared [33], acoustic [40] and several others. However, the use of these special sensors also needs special roadside installation. Moreover, they also need close proximity to the vehicles for the sensors to operate correctly. Such constraints restrict the generic use of these solutions in practice. In addition, none of these existing solutions discuss either how to minimize assistance from the cloud or any possible optimization so that the solution can operate on battery power for a long time.

**RF-based solutions:** RF has also been exploited for the detection and classification of vehicles in many works [2, 44, 50, 52, 53]. Most of these RF-based solutions try to exploit the fact that the passage of vehicles across a communication link can impair RF-based communication because of the attenuation of the signal in proportion to the size of the vehicles. The model used for RF-based detection and classification is, hence, ideally quite simple, where RF waves, transmitted from a station on one side of a road, are received and analyzed by another station on the other side of the road. Such a model, apart from being low-cost, requires *neither any special installation/infrastructure nor very close proximity to the vehicles to operate correctly*. In addition, RF-based solutions inherently can preserve the privacy of vehicle users too. Accomplishing these properties together is quite hard to get in other approaches.

**Problems with the RF-based solutions:** Almost all the existing RF-based solutions exploit the degradation of the physical layer parameters when the communication is intersected by vehicles. In particular, they measure the impairment of the signal received at the receiver through the RSSI values and carry out the necessary analysis of the attenuation to understand the presence and type of the vehicle. The dependence on the raw physical layer properties of the communication brings forth certain immediate problems in these approaches. Although the base model seems to be quite simple, it requires adopting a complex deployment/installation procedure with more external control for accurate detection (e.g., see [2, 44]). In addition, it also leads to a quite high degree of communication between the transmitter and the receiver, which makes these solutions quite energy-hungry. For example, one very recent work in this direction, [44], although exhibits quite high accuracy, needs to exploit a continuous flow of RF packets, which quickly drains the battery of the devices.

*Energy-preservation, although a very important issue for the sustained operation of a system in an outdoor environment, has not been addressed in any of the prior works. In an RF-based strategy, to accomplish both energy preservation as well as accurate detection and classification, the number of RF-packets is required to be dynamically controlled through a clever communication protocol. In particular, the protocol should be able to systematically adapt itself and accordingly control the flow of the packets so that, while minimizing the number of packets, it also ensures that enough logs are recorded through the collision of the RF packets and the vehicles for accurate detection and classification. In this work, we propose an IoT-based Lightweight Vehicle Detection and Classification*
strategy that operates in Real-Time (LiVeR) with minimized use of the RF-units of the IoT-devices. In contrast to the existing RF-based works [1, 39, 44] LiVeR carries out the job in a more systematic way. In particular, to make the collision among the vehicles and the RF-packets happen in a very organized form, LiVeR exploits Synchronous-Transmission (ST) [56] based communication between a pair of IoT-devices. The work [42] provides a very preliminary study of the proposed ST-based framework. The use of ST enables LiVeR to keep the detection and classification mechanism quite compact, simple, and self-adaptive so that the main tasks can be executed on-the-fly on the resource-constrained devices.

The main contributions from the work can be summarized as follows.

- We propose an IoT-based low-cost self-adaptive and low-power strategy, LiVeR for vehicle detection and classification over a wide area in real-time. LiVeR is quite self-sufficient and does not need support from either any special infrastructure or cloud/fog setting.
- LiVeR uses lightweight ML models in resource-constrained IoT-edge-devices for on-the-fly detection as well as classification of the vehicles into three classes based on their size, namely, small, medium and large over single-lane traffic.
- LiVeR is implemented in Contiki OS for TelosB devices and extensively studied through emulation, as well as outdoor experiments over multiple different places in a city and an academic institute campus.
- LiVeR exhibits a detection accuracy of 99.8% with quite a decent classification accuracy of 91.3% upto 98.3%, 92.3% upto 98.5%, and 93.8% upto 98.8% for small, medium, and large size vehicles, respectively.
- Self-adaptive nature of LiVeR enables it to carry out accurate detection and classification spending only about a one-third number of RF packets compared to the state-of-the-art RF-based solution. It is demonstrated that minimized use of RF-packets enables LiVeR to sustain over battery power for a substantially longer period of time compared to the state-of-the-art strategies.

**Structure of the paper:** The rest of the paper is organized as follows. Sec. 2 lists the expected features of a lightweight vehicle detection and classification solution and subsequently describes how well the existing works meet them. Sec. 3 summarizes the basic idea and then briefly describes the necessary background on ST and corresponding communication protocols. Sec. 4 describes the details of the design of LiVeR. Sec. 5 provides a theoretical study of the basic mechanisms. Sec. 6 details the experimental setting for outdoor measurement studies. Sec. 7 details the lightweight ML-based strategy to carry out the detection and classification of the vehicles in the edge devices. Sec. 8 provides a detailed report of the evaluation-based study of LiVeR. Finally, Sec.

| Abbreviation | Definition                  | Abbreviation | Definition                  |
|--------------|-----------------------------|--------------|-----------------------------|
| ST           | Synchronous Transmission    | LT           | Latency                     |
| AT           | Asynchronous Transmission   | RXCT         | Packet Reception Count      |
| SCPPhase     | Synchronization and Control Phase | RO       | Radio-on                  |
| DCPhase      | Detection and Classification Phase | R      | Reliability               |
| iPhase       | Interaction Phase           | HC           | Hop-count                  |
| GP           | Glossy Period               | TXP          | Transmission-Power         |
| GD           | Glossy Duration             | PG           | Plain Glossy               |
| NTX          | Number of Transmission      | MPG          | Modified Plain Glossy      |
| RSSI         | Received Signal Strength Indicator | DMPG    | Dynamic Modified Plain Glossy |
| LQI          | Link Quality Index          | CP           | Contact Period             |
| TC           | Timeout-count               | GT           | Ground Truth               |
| FN           | Farwading Nodes             | MP           | Measurement Point          |

Table 1. List of abbreviations used in the paper.
9 depicts a simulation-based study of the wide-area aspects of LiVeR along with a detailed evaluation study of practical applications of LiVeR based on outdoor experiments. Table 1 provides the list of abbreviations and their definitions used in the paper.

2 Related Works
Apart from accurate detection and classification of the vehicles, a solution to be practically useful and perform as expected in outdoor environments should also bear a set of additional features. In the following, in Sec. 2.1 we first provide a comprehensive summary of the expected features of a solution. Subsequently, in Sec. 2.2 we discuss the existing approaches for vehicle detection and classification w.r.t. the availability of these features.

2.1 Expected features
- **Wide-area operation**: A monitoring/surveillance application directly involves dynamic status updates over multiple different locations at the same time. Therefore, the base solution for vehicle detection and classification should be easily extendable over a large area as well as effectively interface with the upper layers in the stack. However, fruitful and easy extension to wide-area operation also brings forth the following requirements.
  1. **Cost-effective design**: Cost per unit/module for detection and classification should be as low as possible so that a large area, e.g., a significant part of a city, a campus, etc., can be effectively covered with many such modules installed at regular intervals.
  2. **Coordinated measurement**: Fruitful realization of wide-area coverage needs support for coordination among multiple base modules installed at different locations. However, such coordination should be quite low-overhead.

- **Outdoor operation**: Compared to an indoor setting, outdoor environments can be highly diverse and adverse. There can be various types of adversities owing to different possible reasons, which include heavy rain, storms, cyclones, earthquakes, and several other similar issues. A solution that is targeted to be deployed outdoors, must consider these issues in advance and satisfy the following requirements.
  1. **Easy Installation**: Easy and quick installation is a highly demanding feature for a system to be practically useful in diverse outdoor settings, especially when it requires covering a wide area. It is highly preferable for a solution to avoid involvement with or dependence on any special infrastructure.
  2. **Quick Re-installation**: Systems targeted for outdoor environments are highly prone to sudden damage due to various natural reasons. Therefore, a solution aimed at outdoor operation should be able to be quickly re-installed. The cost per unit and development complexity should also be quite low so that the modules can be quickly procured/developed as needed.
  3. **Flexibility**: It should be possible for a solution to work at any place/location as needed. It should be equally efficient at any time of the day. Installation should not depend on specific environmental conditions either.
  4. **Maintenance**: Apart from the initial design, development, and deployment of a solution, the issue of maintenance is also very crucial for its sustainability in outdoor settings.
  5. **Privacy**: As per regulations, no data should be collected that has the potential to breach the privacy of the users. To meet the goal, a solution should aim to collect only anonymous data and should be able to carry out the task based on the same.

- **Self-sufficiency**: For successful and sustained outdoor operation, a solution should be as self-sufficient as possible. However, the existing solutions are mostly dependent on both active assistance from the cloud as well as uninterrupted power supply.
(1) **Minimized/No Dependence on Cloud**: Cloud-computation-based solutions are quite common in almost all domains. However, they are mostly centralized in nature, and hence, suffer from the issues related to single-point-of-failure. They bear quite a higher chance of getting victimized by targeted attacks, which may even cause severe disruption of the cloud connectivity itself. Therefore, as a recent trend, especially an outdoor-based solution covering a wide area should reduce the dependence on the cloud as much as possible.

(2) **Battery-based Operation**: The availability of an uninterrupted power supply is a common assumption. However, for a solution to seamlessly work in an outdoor setting despite all sorts of adversities, it should also be self-sufficient w.r.t. the power supply. In particular, in places where there is a lack of consistent power supply or where it is hard to make the necessary arrangements for the same, a solution should be able to work solely based on the limited battery power available.

Many of these constraints described above highly conflict with each other. For example, a solution to be able to accurately detect and classify the vehicles may need to employ more analysis of the recorded data and may depend on high-end cloud servers so that outdoor implementation remains simpler. Similarly, accurate detection and maintaining anonymity may also demand the collection of more sophisticated data per vehicle, which would require complex settings as well as an adequate supply of power. Balancing all the above-mentioned requirements together is possible only when the basic logic behind a solution is very simple and it gets implemented in a very simple but efficient way by minimizing the computing requirements so that the tasks can be executed fully by the edge-devices, as well as it can also live on battery power for a long time.

### 2.2 Existing works

Most of the existing solutions fail to meet a significant fraction of the features described above. A comprehensive summary is provided below. We refer to the three types of solutions, camera-based, sensors-based, and RF-based, as briefed in Sec. 1.

**Cost-effectiveness**: There have been immense advancements in the design and development of lightweight object-detection strategies, e.g., YOLOv7 [51] and YOLACT++ [4], which are being used for efficient application of computer vision in vehicle detection and classification. These works [4, 5, 13, 24, 36, 51] use costly image capturing equipment for recording the vehicles. Some of the recent works [6, 7, 27, 54] also exploit sophisticated stereo-camera for capturing 3D-vision to achieve superior performance. Moreover, at minimum, they need a GPU system for analysis of the image-data on-the-fly or a costly high-end cloud service. Sensor-assisted solutions also mostly need to depend on the installation of costly-infrastructure [26] for the appropriate functioning of the sensors. In principle, RF-based solutions ideally do not need any such costly equipment [2, 18, 23]. However, the way RF has been employed so far involves a high cost per unit for the necessary setup to achieve controlled measurement of the physical layer parameters (e.g., RSS) [44, 49, 52, 53]. The use of ML/AI-based on-the-fly analysis of the collected data, either by cloud or high-end computing facility, also adds to the overall cost. This results in high-cost algorithms for their analysis. In contrast, in our current work, we use low-power RF communication between off-the-shelf low-cost IoT-devices and in-device recording, as well as lightweight analysis of the data, which by default minimizes the cost per unit.

**Flexibility/Infrastructure**: Unlike a friendly indoor environment, outdoor space is considered to be quite hostile and dynamic. Thus, flexibility and easy re-installation support for a solution are very important. Camera-based solutions need special installation of the costly cameras and their proper maintenance too. Sensor-assisted solutions also need specialized installation of the sensors. For instance, intrusive sensors such as acoustic-sensor [40], light-sensor [46], magnetometer [47, 48], loop detector [20], piezoelectric sensor [35], vibration [37, 45] etc. first need to get appropriately installed on the pavement surfaces. High installation costs, traffic disruption during installation, maintenance, etc. are some of the disadvantages of this approach. In contrast, in our current
work, we use off-the-shelf battery-operated small-size IoT-devices to serve the purpose. The IoT-devices can be installed at any suitable location beside the road, e.g., roadside trees/lamp-posts, etc. It does not need any specialized installation support and, hence, is highly portable.

**Energy-requirement/Power-supply:** In general, all the existing works assume the availability of an adequate and uninterrupted source of power (e.g., wall-power) for carrying out the job [5, 13, 24]. In contrast, our design exploits low-power IoT-systems and, hence, largely relaxes this requirement. In particular, our compact and efficient communication protocol enables the solution to sustain itself for a substantial amount of time solely based on battery power.

Moreover, the system can be appropriately tuned to run over battery power for a much longer duration.

**Cloud assistance/Computation-cost:** The camera-based solutions [4, 5, 13, 24, 33, 36, 51] need to run complex algorithms in the back-end for processing the image and video data for which constant cloud connectivity is inevitable. Frequent communication with the cloud with large amounts of data creates a serious bottleneck in the mass application of such solutions over a large area. To address this common issue, the recent design trend is to minimize the interaction between the cloud and the edge [55]. The IoT-assisted approach proposed in this work uses very simple strategies and ensures that the low-power small-sized IoT devices can carry out all the required analysis tasks without explicit support from the cloud.

**Wide-area synchronized measurement:** Existing works deal with the detection and classification of the vehicles at a single place [2, 52]. However, in the context of a smart-city/intelligent-transportation, the solution needs to be replicated at multiple locations, and the measurements collected from these locations need to be time-correlated for a system-wide behavioral study. None of the existing camera-based, sensor-based, or even RF-based solutions talks about this issue. Our proposed IoT-based framework, apart from being cost-effective and easy to use, takes care of these vital issues, which are largely missing in the existing works. Intrinsically, our proposed framework does not depend on the conventional 3G/4G/LTE network. Rather, it is designed as an ST-based standalone IoT system that connects all its base units using low-power communication technology and supports seamless time-correlation among them.

**System-wide analysis:** The primary goal of wide-area detection and classification is to enable on-the-fly behavioral study, anomaly detection, and quick decision-making in traffic management and control. Most of the

Table 2. Summary of the existing works on vehicle detection and classification.
existing solutions depend on the computation and analysis of the raw data with the help of a cloud server. However, conveying data to the cloud at every epoch from every measurement unit installed over a large area naturally consumes a huge bandwidth, causing network congestion as well as performance degradation [2, 49, 52]. The proposed IoT-based solution in our current work, in contrast, by the virtue of the existing rich IoT-protocol-base [15, 21], can carry out in-network data processing to support on-the-fly end-to-end collaborative decision-making in real-time without any active help from the cloud.

**Privacy:** With the increasing use of technology, the privacy of users has become one of the prime concerns. Camera-based solutions are prone to breaches of privacy and hence are not suitable for all applications [19]. BLE/WiFi-based solutions can also work in a privacy-preserving fashion but need a prior coupling between vehicles and the detection/classification system, e.g., the transmission of any special signal from the vehicles for their detection and classification [30]. Similarly, sensor-based solutions are also mostly not prone to any such breach of privacy [28, 45]. However, the requirement for special infrastructure and dependence on the cloud for necessary analysis make their ubiquitous and widespread use difficult. In our current work, we fundamentally use a low-power RF transceiver as a special type of sensor that directly categorizes a vehicle under a class without the requirement of any further detailing of the actual vehicle and hence maintains privacy.

**Discussion:** RF-based solutions tend to meet many of the expected features which are quite hard by the other two possible approaches. However, the existing works exploit RF in a mostly brute-force way. The role of an efficient protocol for supervising the whole process is extremely important to balance all the goals together, which is missing in the existing works. The approach taken in [44] is the closest to that of ours. It ([44]) uses an AT-based protocol similar to token-ring [14] where six active devices exchange RF-packets across the road at a very high rate. The protocol measures the RSSI values at the receiving devices and checks for any attenuation due to the passage of the vehicles. SVM is used for on-the-fly detection and classification. However, the use of uncontrolled communication among the active devices makes this strategy highly energy-hungry. Moreover, the solution acts blindly, i.e., it does not adapt based on the traffic and, hence, constantly drains the energy from the low-power devices at a very high rate. In addition, the work is quite ignorant about the issue of efficient wide-area coverage. In contrast, our current work LiVeR, demonstrates an adaptive mechanism that achieves the goal of spending approximately one-third of the RF packets compared to the work [44] without compromising accuracy. Section 8 provides a detailed comparison.

Table 2 summarises the existing works w.r.t. the expected features (in the columns). The advantages of a solution are highlighted by a gray background.

### 3 Basic Idea and Background

Several works have reported studies regarding the hindrance in low-power IoT protocols caused by vehicles passing over the roads [39, 43]. This special type of sensitivity of the low-power communication protocols motivates us to use them as a cost-effective, lightweight, and privacy-preserving sensor for the detection and classification of vehicles. However, at the same time, the possibility of severe disturbances caused by the vehicles brings forth a significant challenge in developing a full system solely using low-power communication technology. A solution in this direction, hence, needs to first establish a framework that can allow itself to degrade in a very controlled way as per the passage of the vehicles. There are two primary challenges in effectively using low-power RF-based protocols to systematically serve the purpose, as detailed below.

1. **Base principle:** Degradation of the quality of a low-power RF-link is used as the foundation of detection and classification. Although the base mechanism happens to be quite easy, at the same time, because of the possibility of easy attenuation of low-power RF packets, it is quite challenging to use it for developing a complete system with support for end-to-end coordination. Existing solutions, therefore, use RF in a controlled setting or depend on other technology to develop the complete system, e.g., cloud assistance.
Limited contact period and energy-consumption: A low-power RF technology to satisfactorily detect and classify vehicles need a sufficient number of packets to collide with a vehicle and subsequently analyze the quality of these receptions on-the-fly. However, due to the mobility of the vehicles, the time for which a single vehicle can obstruct a communication link is sufficiently low, and it is quite hard to achieve the goal without any controlled setting/specialized arrangements. The existing state-of-the-art RF-based solutions employ a high number of packets to detect even a single vehicle, which makes them highly energy-hungry. Moreover, detection directly based on a variation of attenuation also demands cloud-based processing.

Our Approach: To balance between the two objectives as described above, in this work, we aim to employ a robust low-power communication protocol. We aim to design the protocol in such a way that it can regularly switch between the following two modes of operation:

1. Sensing-mode: Where it works in a vulnerable state so that it fails to function and degrades its quality of operation in a controlled way when the corresponding RF packets interfere with the vehicles. It also carefully records its degradation while interfering with the vehicles.

2. Wrapping-mode: Here the protocol works with all its features. Its robust and fault-resilient setting not only wraps the sensing mode but also allows multiple sensing units to communicate with one another and realize a wide-area system. The detection and classification are done through in-device analysis of the logs collected during the sensing mode.

In particular, we aim to design a low-power communication protocol that can function as a sensor for the detection and classification of vehicles. Furthermore, to make the solution simpler, in contrast to the existing RF-based works, which use the raw physical layer parameters, e.g., RSSI/LQI, we move up in the network stack and aim to employ the MAC layer (Data Link Layer) parameters to characterize the degradation.

Under traditional Asynchronous-Transmission (AT)-based low-power communication protocols the events take place mostly in an ad hoc and uncoordinated way. Therefore, inherently, it is hard to employ an AT-based mechanism to meet the above-mentioned requirements while keeping it lightweight, low-energy consuming, real-time, and supporting end-to-end coordination over a wide area. To fulfill the goals, in this work, we aim to build an ST-based framework. In the following, we first provide a brief background on AT and ST and describe the specific ST-based protocols that we use as a foundation in our work.

3.1 AT vs ST-based strategies

Most of the traditional protocols in WSN/IoT are based on AT. The nodes/devices in these protocols behave in a highly independent fashion. However, energy preservation is one of the primary issues in any such protocol. RF-transceiver happens to be the most energy-hungry unit in a device. For sustained low-power operation, therefore, any protocol tries to duty-cycle (periodic ON and OFF) their RF-transceiver (radio). In addition, to minimize the use of the RF units, the nodes in an AT-based protocol try to understand each others’ duty cycles, and subsequently make necessary mutual adjustments so that one can easily communicate with the other nodes as necessary by the application. For instance, in the well-known data-collection protocol, CTP [16] a source node syncs its duty cycle with its immediate parent node (i.e., the node that is assigned the responsibility to forward the data from a certain set of source nodes towards the sink node). However, due to such special pair-wise tying up, during any temporal link disturbances caused by vehicular obstacles, when the quality (e.g., PRR) of a certain set of links degrades, it affects the performance of the overall system. In other words, although the communication hindrance happens in a specific direction (or location) affecting only a specific set of links at a time, the AT-based protocols fail to take advantage of the omnidirectional connectivity of the underlying wireless communication medium to prevent performance degradation at the system level.

Synchronous-Transmission (ST): In contrast, ST-based strategies are organized differently. Broadcast-based nature of any wireless communication makes it inherently vulnerable to cascaded collisions, which is a common
problem under AT-based protocols. However, ST-based strategies, employ an in-network time-synchronization strategy that enables them to exploit special physical layer phenomena, e.g., Constructive-Interference (CI) or Capture-Effect (CE)\cite{22} to avoid collisions among multiple RF packets \cite{56}. In particular, when a number of RF packets attempt to approach a single receiver at the same time, an ST-based strategy cleverly exploits the special capability of the underlying radio to accept the strongest packet and discard the others on-the-fly. Although the basic concept of ST was invented earlier, the protocol Glossy \cite{15} for the first time showed how ST can be exploited in a systematic way for the purpose of system-wide flooding. Appropriate use of synchronized transmissions makes the one-to-many dissemination protocols such as Glossy, RoF \cite{25}, and BlueFlood \cite{32} extremely robust in nature \cite{15, 32}. In particular, logically it is quite hard for even a set of temporal/localized and uncorrelated link disturbances caused by the vehicles to affect the performance of an ST-based flooding protocol.

In this work, we use the protocol Glossy as the foundation. It is briefly described below.

**Glossy**: The pioneering protocol Glossy \cite{15} shows how to efficiently achieve quick and reliable one-to-all dissemination, i.e., flooding under ST. Glossy starts with a single node (initiator) broadcasting its data packet. The packet is received at the same time by all the first-hop neighbors of the initiator. Through appropriate time adjustment, Glossy arranges the re-transmissions of the received packets from each of these nodes in such a way that it results in CI/CE in the second-hop neighbors of the initiator which enables them to successfully decode the received packets. The process cascades this way until it covers the full network. Systematic exploitation of CI/CE, thus, enables Glossy, (in general, an ST-based flooding protocol), to successfully exploit the strength of the redundant but synchronized transmissions of the packets from multiple different source nodes. This also makes ST inherently much more resilient to the degradation of the links caused by vehicular obstructions or any such temporal and localized communication hindrance compared to the AT-based strategies.

Glossy is executed in sensing mode and wrapping mode to achieve our goal in this work. Glossy is meant for a network-wide dissemination of data. However, in sensing mode, we use it between a pair of IoT-devices (referred to as a Measurement-Point (MP), described later) installed by the two sides of a road. The whole concept is hence, centered around the controlled measurement of the characteristics of the degradation of the performance of Glossy due to dynamic obstructions caused by the vehicles when they pass through an MP (see Fig. 8i). A similar approach has been adopted in the work \cite{41} for lightweight measurement of traffic volume where the target is only the detection of the vehicles. However, the current work aims to classify the vehicles too. Thus, the sensitivity of Glossy w.r.t to different categories of vehicles is important. To let Glossy get affected by vehicular disturbances, we reduce the transmission power level appropriately in such a way that it works normally when there is no traffic but degrades gracefully while fruitfully capturing the category of a vehicle passing through an MP.

**Physical-layer (PHY) parameters** such as RSSI, and LQI have been used in several prior works \cite{18, 23, 38, 43} to characterize the RF communication during the passage of the vehicles. Most of these works show controlled measurements of these PHY parameters analyzed using costly devices/settings. To carry out the same job in a lightweight and easier manner, as well as analyze the measurement data directly in the resource-constrained off-the-shelf IoT-devices, in this work, we capture the Data Link Layer (DLL) parameters instead of the raw PHY parameters. Although disturbances in the DLL parameters are fundamentally an artifact of the same in the PHY parameters, the higher layer in the network stack provides natural filtering of various complex effects which in turn makes the process of analysis simpler, lightweight, and quicker. The DLL performance metrics which we use as a set of features in this study are detailed next.

**NTX**: To ensure complete network coverage, every node in Glossy repeats the transmission of a packet a pre-decided number of times which is known as NTX, where each transmission is followed by a reception. NTX is a quite common parameter used in most of the existing ST-based protocols. We use NTX as one of the prime design parameters.
Reliability (R): Reliability of a Glossy instance is considered to be 100% in the case at least one of the NTX transmission attempts from the initiator reaches the receiver. Due to the inherent robustness of the protocol, the value of R is found to be mostly 100%. However, it may also go down to 0 when the vehicular adversity is so serious that none of the NTX packets becomes able to reach the receiver. We do not use this metric directly for detection/classification purposes. We use it for analyzing the robustness of the protocols.

Timeout-Count (TC): A Glossy initiator automatically makes multiple attempts to send a packet to the receiver in case it does not get any response back from it. TC refers to the total number of times the initiator tries before it gets successful. TC is ideally 1 without the presence of any vehicle and increases depending on the degree of the obstruction.

Latency (LT): It refers to the time taken by a receiver (non-initiator) to successfully receive the data transmitted by the initiator. Note that for dissemination purposes, the reception of at least one packet out of NTX is enough for the receiver. In the absence of any vehicle, LT is, thus, supposed to be very minimal. However, due to the failures in the reception of the packets by the receiver due to vehicular obstruction, LT is supposed to rise.

Reception-Count (RXCT): In Glossy, an initiator transmits the NTX number of packets. Hence, in an off-road setting, a receiver is supposed to receive all of these packets. However, due to vehicular obstruction, some of the packets may get lost, and therefore, due to repeated attempts all the transmissions might not get completed by the time $G_D$. Thus, the number of packets received on the receiver side, referred to as Reception-Count (RXCT), may be lesser than NTX due to vehicular obstructions. Thus, based on the degree of obstruction, RXCT is also supposed to vary.

Radio-On time (RO): RO is measured as the time taken to complete all of the NTX numbers of packet transmissions by a node. In Glossy every node (except the initiator) transmits only after a reception. If a packet gets lost it may be re-transmitted stretching the completion of the communication process within $G_D$ or the process itself may fail to convey all NTX packets because of limited $G_D$. To capture all sorts of such variations caused by vehicular obstruction we include RO also in the set of features. Note that RO captures the variation of the time span where the RXCT captures the number of packets that could successfully reach the receiver (after repeated attempts if necessary and within a limited time).

While we mainly focus on the DLL metrics, the two primary PHY metrics RSSI and LQI are also captured for preliminary study. Under our proposed framework we successfully capture the discrete values of RSSI and LQI from each packet communicated to the receiver. We calculate the average and standard deviation of these RSSI and LQI values over all the packets during DCPhase. We refer to these as $RSSI_{-avg}$, $RSSI_{-stddev}$, $LQI_{-avg}$, and $LQI_{-stddev}$.

4 Design
The design of LiVeR is motivated based on the following three goals.

- The system has to be robust, lightweight, and support cost-effective deployment over a wide area. It is supposed to function based on low-power communication, despite possible communication hindrances caused by vehicular obstructions.
- It is supposed to support sufficient degradation of a set of pre-decided low-power communication links in a controlled way so that the characteristics of the degradation can be recorded and later used for vehicle detection and classification.
- For minimized energy consumption in low-power IoT devices, the system should also minimize the number of RF packets to be sacrificed for detection and classification, despite the highly limited period of contact with the moving vehicles.
Overview: While in the second goal, a number of communication links are to be sacrificed for collisions with the vehicles, the first goal demands minute and implicit (without any additional infrastructure) recording of the degradation of the functioning of the protocol too, for on-the-fly analysis to detect and classify the vehicles. To meet these conflicting goals, we employ an ST-based top-down approach. Unlike other existing RF-based works, we first think about the design of a platform that can support a robust time-correlated operation over a wide area. For the first goal, we use the protocol Glossy in wrapping mode. For the second goal, we use a variant of the protocol Glossy in sensing mode. In particular, we inscribe an independent small communication module that allows itself to degrade in a controlled way during the passage of the vehicles. For sufficient sensitivity, we appropriately lower the transmission power level. The whole process repeats periodically and keeps carrying out the measurements.

Setting: Fig. 1 shows a schematic of the deployment of IoT devices over a possible target area. A device is deployed in either of the two forms: (a) Forwarding-Node (denoted by FN-i in the figure) or (b) A specially designed unit called Measurement-Point (MP) (denoted by MP-i in the figure). An MP is composed of two low-power WSN/IoT devices installed at the two opposite sides of a road. Each of these devices is referred to as Measurement Units (MU). Fig. 5i(c) magnifies a single MP while Fig. 5i(a) shows the placement of two MPs over a road. The two devices in an MP are referred to as the initiator and the receiver. All the MPs and FNs together form a connected IoT system.

The overall process runs as a periodic repetition of the following three phases:

1. Synchronization and Control Phase (SCPhase): Here, Glossy runs over the full system for time-synchronization and sharing of necessary control information. The robustness of ST-based Glossy ensures that the platform keeps on running despite random failures of the links at various locations of the network due to even severe vehicular obstructions.

2. Detection and Classification Phase (DCPhase): The next phase is dedicated to sensing through RF packets. Here, a variant of Glossy is executed in low-power mode. The pattern of the degradation is captured in the devices themselves and analyzed on-the-fly through a lightweight ML strategy for detection and classification.

3. Interaction-Phase (IPhase): Finally, in the third step the data collected from multiple MPs are either disseminated, shared among each other, or collected at a designated sink node for system-wide coordination and decision-making. This stage can be exploited to generate a combined view from multiple MPs to make
fine-grained classifications. However, this stage is purely meant for the target application and, hence, is an optional one, i.e., it may not be required to get executed at every period.

4.1 SCPhase

The major issue in SCPhase is a system-wide time-synchronization which is achieved by an instance of Glossy. The algorithm of Glossy is highly scalable. The time it takes depends on the diameter of the network instead of the exact number of nodes. Apart from that, owing to the modular design of the proposed solution SCPhase can be also implemented based on recently invented advanced solutions [29]. However, the proposed solution being an ST-based one, periodic execution of the SCPhase is an important issue. The following two parameters come into the picture in this context.

**Glossy-Period (GP):** The three-phase process runs in a *periodic fashion* with a defined period and defined duration for each of the three phases. Fig. 5i(b) pictorially depicts this arrangement in the form of a time diagram. GP is the period set for the process (i.e., one instance of both SCPhase and DCPhase, along with one instance of IPhase when needed). GP is set only for the SCPhase and is decided very carefully, as explained in Sec. 6.3. The value of GP partially determines both the energy consumption in each device as well as the value of GD, i.e., *Glossy-Duration* as discussed below.

**Glossy-Duration (GD):** It is known as the part of the period where the actual data communication takes place. By virtue of ST, the execution of Glossy is completed very quickly. For instance, to cover a 100-node 5-hop ZigBee network, Glossy takes about 10–20 ms for a complete network-wide time-synchronization as well as one-to-all dissemination of small data. Thus, in SCPhase, GD is decided based on the size of the full network (we refer to this as GD₅). In DCPhase the duration of the protocol is set based on the expected/planned nature of the interaction of the protocol with the vehicles passing through an MP. We experiment with three possible variations as described below.

4.2 DCPhase

The primary intention is to let the vehicles pass through the installed MPs and collide with the RF packets during DCPhase in such a way that can bring enough variations in the metrics (discussed in the last section) which can be exploited through lightweight ML strategy for vehicle detection and classification. We focus on the following four metrics: TC, LT, RO, and RXCT. For example, the metric TC reflects how many times the initiator had to retry to make a successful communication with the receiver. Therefore, the expectation is to have the values of TC be quite proportional to the size of the vehicles for easy classification. However, to meet this intuition, there are two primary bottlenecks as discussed below.

**Lack of persistent communication:** Let us assume that DCPhase runs a normal instance of Glossy in low *Transmission-Power (TXP)* mode with two nodes (MP, i.e., an initiator and a receiver). Since there is no control over the dynamics that govern the collision of a Glossy packet with a vehicle, it is always good to have a sufficient number of such collision instances, even for a single vehicle. To make this happen, the transmissions from the initiator node have to be persistent, i.e., the initiator should keep sending a sufficient number of packets in a short interval even in case it detects that the last packet was lost. However, note that the original design of Glossy is meant for efficient one-to-all dissemination of data over a network. To support safe multi-hop operation, such persistent behavior is implemented only for the very first packet transmitted from the initiator. After the transmission of the first packet, when nothing is heard back, from an initiator’s perspective, there can be two possibilities.

- **First packet got lost:** The packet from the initiator got lost, and none of the first-hop nodes, i.e., none of the nodes that are directly in touch with the initiator, could hear anything from the initiator, and hence, they did not reply.
• **Reply to the first packet got lost**: None of the reply packets from the first-hop nodes could reach the initiator.

To cope with the first possibility, the initiator can *safely re-transmit the packet* as soon as a certain safe time has passed. However, under the second possibility, the initiator already lost time-synchronization because of lack of packet reception, and hence, *should not re-transmit the packet*. Unfortunately, when the initiator finds that there is no reply to its initial transmission, it cannot distinguish these two possibilities on its own. Therefore, it needs to wait for a safe period called *initiator-time-out* after which it can re-transmit the same packet. Moreover, once the first packet successfully goes through, the same mechanism cannot be repeated for the other packets. When we directly use the plain Glossy protocol for intra-MU communication in an MP in the DCPhase, these restrictions ultimately reduce the frequency of the transmission from the initiator and, hence, proportionately reduce the chance of the collision of the RF packets with the vehicles too.

**Solution**: The dynamics of Glossy in a two-node setting (intra-MP), which we use in the DCPhase of our proposed framework, is different from the dynamics of the protocol in a general multi-hop setting. In particular, in a purely two-node setting, the only reason for no reply for the first packet from the initiator is the loss of the first packet or its reply from the receiver. Hence, each of the NTX transmissions can safely operate in a persistent mode. To systematically realize this and enhance the scope of the collision of the RF packets further (as explained next), we split a Glossy instance into NTX number of persistent Glossy instances each having NTX=1. Each instance is set with a very short value of initiator-time-out and two consecutive instances are separated by a certain small gap (a few milliseconds).

**Interaction between DCPhase and vehicles**: The span of time a vehicle gets for interaction with the packets in a DCPhase is a very important issue and it depends on the size as well as the speed of the vehicles. It has been studied that under low-power communication, the main disturbance happens when a vehicle crosses the line-of-sight (LoS) between the sender and the receiver [39, 43]. Let us define the quantity *Contact-Period* (CP) as the span of time a vehicle takes to cross the LoS in an MP. Thus, it is the responsibility of the DCPhase to increase the scope of the interaction of the communication protocol with vehicles irrespective of different possible values of the CP. It is done as follows.

![Flow diagram showing the sequence of operations periodically taking place in LiVeR.](image)
Algorithm 1 Persistent_Glossy()

Input Parameter: MP_initiator, node_id => initiator flag indicates it is the initiator in the MP, node_id is the id of the node.

Output Parameter: tx_cnt, rx_cnt, lt, ro

1: tx_cnt = 0, rx_cnt = 0, lt = 0, ro = 0, time_start = 0
2: Initialize Timer timer_timeout
3: time_start = TIME_NOW()
4: if my_node_id == MP_initiator then  // Initiator starts here
5: mode = TRANSMIT
6: create_packet (packet, data_to_share)
7: while tx_cnt ≤ 1 do
8: if mode = TRANSMIT then  // Initiator transmits
9: start_transmission (packet)
10: set_timer (timeout_timer, TIME_NOW() + initiator_time_out)  // Initiator sets the timeout
11: mode = RECEIVE
12: tx_cnt ++  // Record transmission count
13: else
14: packet = start_reception()  // Initiator receives packet
15: cancel_timer (timeout_timer)  // Cancel the time when reply received
16: mode = TRANSMIT
17: rx_cnt ++  // Record reception count
18: if rx_cnt = 1 then
19: lt = TIME_NOW() – time_start  // Record latency
20: end if
21: end if
22: end while
23: ro = TIME_NOW() – time_start()
24: else  // Receiver starts here
25: mode = RECEIVE
26: while tx_cnt ≤ NTX do
27: if mode = TRANSMIT then
28: start_transmission (packet)
29: mode = RECEIVE
30: tx_cnt ++
31: else
32: packet = start_reception ()
33: rx_cnt ++
34: if rx_cnt = 1 then
35: lt = TIME_NOW() – time_start
36: end if
37: mode = TRANSMIT
38: end if
39: end while
40: ro = TIME_NOW() – time_start()
41: end if
42: Return tx_cnt, lt, ro, rx_cnt
Algorithm 2 Multiple Persistent Glossy()

**Input Parameter:** Transmission_remaining = 5  
**Output Parameter:** TC, LT, RO, RXCT

1. TC = 0, LT = 0, RO = 0, RXCT = 0
2. if (transmissions_remaining > 0) then
3. set_timer(timer_persistence, TIMER_NOW() + interval)
4. Call Persistent_Glossy()
5. transmissions_remaining --
6. Get txcnt, lt, ro, rxcnt
7. TC = TC + txcnt, LT = LT + lt, RO = RO + rt, RXCT = RXCT + rxcnt
8. transmissions_remaining --
9. end if

Algorithm 3 ISR: Handle_Scheduling_Timer_Expiry()

1. if timer_persistent then
2. if (transmissions_remaining > 0) then  // Refers to the global variable transmissions_remaining
3. set_timer(timer_persistence, TIMER_NOW() + interval)
4. Call Persistent_Glossy()
5. Get txcnt, lt, ro, rxcnt
6. TC = TC + txcnt, LT = LT + lt, RO = RO + rt, RXCT = RXCT + rxcnt
7. transmissions_remaining --
8. end if
9. end if

Algorithm 4 ISR: Handle_Initiator_Timeout_Timer_Expiry()

1. if timer_timeout then
2. mode = TRANSMIT
3. create_packet (packet, data_to_share)
4. start_transmission (packet)  // Initiator transmits
5. set_timer (timer_timeout, TIME_NOW() + initiator_time_out)  // Initiator sets the timeout
6. mode = RECEIVE
7. tx_cnt ++
8. end if

**Solution:** Splitting a single Glossy instance into multiple instances, as is used for systematically enhancing the frequency of communication between the initiator and the receiver, also provides a way to solve the aforementioned problem. However, along with splitting, the scheduling of shorter instances is also an important issue in this regard. In particular, the individual instances of Glossy should be scheduled in such a way that the vehicles having even the minimum size as well as maximum speed (i.e., the minimum possible CP) should also get at least one scope of interaction with the Glossy packets.

Based on the aforementioned observations and solution approaches, we design the DCPHase in three possible ways, as described below.

1. **Plain Glossy** (PG): Here we execute a single plain Glossy instance in low-TXP.
(2) **Multiple Persistent Glossy** (MPG): Here, DCPhase is composed as a consecutive NI number of instances, each having NTX=1. They are scheduled inside the SCPhase period (GP) at equal intervals. In other words, the SCPhase period is equally divided into multiple segments based on $t_{CP-min}$ to balance both minimization of energy consumption and enough interaction of the vehicles with persistent Glossy instances.

(3) **Dynamic Multiple Persistent Glossy** (DMPG): Here, the process first starts as MPG. However, when a single Glossy instance collides with a vehicle, based on its reflection in the metrics, the other instances in the same SCPhase are scheduled at a shorter gap ($d_s$) instead of waiting for their original schedule as they were already set in MPG. This dynamic change in the schedule allows more packets to collide with the vehicle and helps in gaining better accuracy in detection and classification.

Algorithm 1 describes Persistent_Glossy, i.e., *persistent version of Glossy* which is used as a unit of operation for both MPG and DMPG. Algorithm 2 shows the details of MPG where Persistent_Glossy is invoked at regular intervals for a certain number of times. PG and DMPG are not shown explicitly through the algorithm since PG is the same as the Glossy protocol while DMPG is quite similar to MPG where the instances of the persistent Glossy are invoked more closely based on the time of the first vehicle detected. Timers are used for setting the initiator timeout and scheduling the instances of the persistent Glossy. Algorithm 3 and 4 demonstrate the interrupt service routines used to handle the timer events. Fig. 2 provides an overall flow diagram of the complete process. Fig. 3 pictorially demonstrates the communication components of all three strategies (PG, MPG and DMPG) through a time diagram.

A small computation step is introduced just after the communication steps inside DCPhase where the extracted features are processed through a lightweight ML-assisted detection and classification mechanism (SVM) as shown in Fig. 2. It is described in detail in Sec. 7.

4.3 **IPhase**

The IPhase section is quite flexible. Various other possible many-to-many or many-to-one data-sharing strategies can be employed in this phase to serve the targeted function. In Sec. 9 we show one case study by integrating Chaos for all-to-all data-sharing among the MPs. In Sec. 9.3 we show a possible practical application of the same.

4.4 **Vehicle Classes**

The low-power communication in DCPhase gets affected differently by different types of vehicles. For instance, the passage of a truck results in more rise in LT, TC, and more drop in RXCT than a car. In general, the disturbances in almost all the metrics vary with the size/volume of the vehicle that collides with the inter-MU communications during the DCPhase. Based on the size of the vehicles we define three classes as follows.

- **Type-S**: These are small-sized vehicles, e.g., vans, autos, etc. Due to their smaller size, the degree of obstruction created by these vehicles is substantially low. However, when the RF packets contain quite low signal strength the effect is still visible, especially when compared with a situation when there is no such disturbance.
- **Type-M**: These are medium-sized vehicles, e.g., SUVs, big-size cars, etc. The higher size of these vehicles helps them to make more impact on the RF packets.
- **Type-L**: These are large-sized vehicles, e.g., trucks, buses, etc. These vehicles have the highest impact on low-power RF transmissions. Packets with even moderate signal strength get completely blocked by these vehicles.

To maintain uniformity in the description, we also consider the presence of no-vehicles as a class denoted by **Type-N**.
5 Theoretical Study

In the following, we try to theoretically understand the feasibility of the proposed solution for detection and classification, its time complexity, and its energy consumption.

5.1 Feasibility Study

We derive a set of constraints that are used as a guideline for scheduling the Glossy instances inside a DCPhase as part of MPG/DMPG. CP, i.e., the time for which DCPhase can interact with the vehicle is one of the key factors. Let us consider $t_{CP-min}$ to be the minimum possible CP. As shown in Fig. 4i, considering two consecutive instances of Glossy with NTX=1 ($N_1$ and $N_2$), each having duration of $t_{dc}$, the gap between them, i.e., $t_{dc-gap}$ should be set in a way so that it meets the following constraint,

$$t_{CP-min} > t_{dc} + t_{dc-gap} + t_{dc}.$$  \hspace{1cm} (1)

A similar constraint can be applied for the gap between SCPhase and the first Glossy instance in DCPhase, i.e., $t_{scdc-gap}$ as,

$$t_{CP-min} > t_{sc} + t_{scdc-gap}.$$  \hspace{1cm} (2)

where $t_{sc}$ is the duration of Glossy in SCPhase.

From Equations 1 and 2, we can conclude,

$$t_{CP-min} > t_{dc} + t_{dc-gap} + t_{dc} \geq t_{sc} + t_{scdc-gap}.$$  \hspace{1cm} (3)

Considering a complete period of SCPhase, all the constraints can be summarized as follows.

---

Fig. 3. Time diagram of the three strategies. (a) PG: The transmissions in DCPhase happen within a single Glossy run. (b) MPG: Instead of a single Glossy instance, a series of Glossy instances (each with NTX=1) are used to uniformly cover the SPhase. (c) DMPG: Same as MPG with more frequent execution of the Glossy instances on demand. For example, Case-1 is the same as MPG, while in Case-2, the vehicle is detected in the start, and hence, all the instances are run sequentially. In Case-3, the vehicle is detected a little later hence, the rest of the Glossy instances only are repeated sequentially.

ACM Trans. Internet Things
Effect of heading angle: Considering possible heading directions, let us assume that a vehicle moves over the road with some heading angle \( \theta \) (as shown in Fig. 4(ii)). For any positive non-zero value of \( \theta \), the contact period of the vehicle will decrease compared to the case when \( \theta = 0 \). However, due to lane restrictions and other traffic rules, \( \theta \) cannot be larger than a certain value. Even a vehicle cannot sustain a heading angle for a long time. Therefore, the corresponding measurements of the metrics captured by DCPhase will not cross their boundaries. Therefore, it will not let the classification system to get confused. In particular, our strategy would support up to a certain heading angle which would be enough for the correct classification of the vehicles.

5.2 Time-Complexity

LiVeR is supposed to carry out on-the-spot detection and classification of the vehicles at the edge devices at every regular interval of GP. Thus, the minimum time taken for a detection/classification will be lower bound by GP. The sole motivation behind the design of LiVeR has been to keep it computationally as simple as possible. There are two places where computation is involved in LiVeR as summarized below.

Communication Step: Inside the communication part of DCPhase, the four metrics (TC, LT, RXCT, and RO) are computed through simple increment and subtraction operations as the RF packets succeed or fail in crossing the road being monitored. Fundamentally, the number of computations in terms of elementary addition/subtraction/increment/decrement operations carried out during the communication step dynamically varies with the type of vehicle it encounters. In other words, the number of such computation steps increases with the size of the vehicle the communication process interferes with. In the best case, for a typical small-size (type-S) vehicle, approx. five numbers, and in the worst case, for a typical large-size vehicle (type-L) approx. 30 numbers of additions/subtractions are executed per GP.

Computation step: Just after the communication step, an SVM is used for recalling purposes (see the flow diagram in Fig. 2) Since we use the one-vs-rest strategy for the SVM, the prediction complexity is \( O(C \times d) \) where \( C \) is the number of classes and \( d \) is the number of features. The value of both \( C \) and \( d \) in our case is 4.

Furthermore, IPhase can be used for fine-grained classification of the vehicles by considering the outcomes of multiple consecutive MPs together. In such cases, the minimum number of MPs to be considered together will define the time complexity as their outcomes will be derived at different GPs. However, such analysis is beyond the scope of this paper.

ACM Trans. Internet Things
5.3 Energy consumption

Energy consumption in a low-power communication-based protocol is directly proportional to the number of RF packets it involves in a unit of operation. In contrast, the design of LiVeR is quite self-adaptive. When there is no traffic, LiVeR runs with a minimum number of RF packets. As the traffic intensity increases, it employs a higher number of packets. All calculations in LiVeR happen in the unit of GP. Let us assume that, based on traffic intensity, a GP encounters a vehicle with probability $\alpha$. Let $V_S$, $V_M$, $V_L$, be the fractions representing the number of Type-S, Type-M, and Type-L vehicles w.r.t. the total number of vehicles $N$ encountered in a given span of time $T_d$. When there is no traffic, every GP is filled up with the NI number of instances of Glossy where each instance employs just one packet transmission between the initiator and the receiver. Let us assume that when a Type-S, Type-M, or Type-L vehicle is encountered, the number of transmissions involved in each such instance is $\gamma_S$, $\gamma_M$, and $\gamma_L$, respectively. Thus, the total number of packets transmitted per GP can be calculated as follows.

$$G = (1 - \alpha) NTX \times NI + \{\alpha V_S NTX \times NI + \beta V_M NTX \times NI + \gamma V_L NTX \times NI\}.$$  \hspace{1cm} (5)

Sec. 8.4 provides a comparative study of the LiVeR and one state-of-the-art strategy w.r.t. energy consumption. Hence, the total number of packets employed for detection and classification in LiVeR within the span of $T_d$, can be calculated as follows.

$$NC = \frac{T_d}{GP} \times G$$  \hspace{1cm} (6)

6 Measurement Study

To understand the feasibility of the proposed methodology, we first carry out rigorous measurement studies under both on-road and off-road conditions. Below, we first describe the experimental setup in detail.
6.1 Experimental Setting
An MP is composed of two MUs which are installed at the two opposite sides of a road, as shown in Fig. 5ii(b). The outer look of an MU is shown in Fig. 5ii(a). We construct it in two ways. Ideally, both the MUs, i.e., the transmitter and the receiver, are constructed by a single off-the-shelf IoT device (we use TelosB) along with a couple of batteries packed inside a water-proof box for uninterrupted outdoor operation (see Fig. 5ii(e)). However, to facilitate easy accessibility of the data from several MPs for further analysis, we use a special arrangement (Fig. 5ii(c)) for the transmitter side MU so that the recorded data directly gets logged in a central machine. Fig. 5ii(d) shows this internal construction. In particular, we connect an IoT-device to an RPi which internally records the data and later communicates the same through a WiFi network to a central server in our lab. An MU is connected to three LEDs which are appropriately programmed to roughly depict the internal operations. The proposed framework is implemented in the Contiki 3.0 operating system. For most of the standalone measurement studies, we use the centralized data-logging facility. We demonstrate a sample application of the proposed vehicle detection and classification mechanism inside an academic institute in Sec. 9.3 where we use the bare-bone simplest possible implementation of the MUs as shown in Fig. 5ii(d).

6.2 Experiments
Extensive measurement experiments are conducted at several places in Bhubaneswar city. The experimental setup is shown in Fig. 5i(c). For each experiment, a video of the whole measurement process is recorded as a Ground-Truth (GT). Fig. 6 plots all four raw DLL-metrics and two PHY-metrics over a seven-minute period of measurement data. For an overall understanding, we also measure the Reliability (R) of the Glossy instance in the DCPhase as shown in Fig. 6(d).

We annotate the collected data based on GT. The readings of all the metrics are found to be quite correlating with the passage of the vehicles through the MP. Several such cases are shown using dashed rectangles in Fig. 6. While LT and TC both are found to be exhibiting a variation with the occurrences of different types of vehicles, RO exhibits mostly a binary pattern: a high value indicating the presence of the vehicles and a low value indicating the absence of vehicles. RXCT behaves in exactly the opposite way. When there is no vehicle, all the NTX transmissions and receptions complete quickly, which makes the RXCT quite high. However, the presence of the vehicles makes RXCT low.

It can be observed that while RSSI-avg hardly shows any correlation with the disturbances, RSSI-stddev demonstrates a good match and also mostly correlates with TC. The same nature is visible in LQI-stddev too. However, from a visual inspection of the difference between the patterns produced by PHY-metrics and DLL-metrics w.r.t GT, it can be clearly understood that PHY-metrics change in a more continuous form while changes in the DLL metrics are more discrete, which happens to be instrumental in simplifying the process of classification in resource-constrained devices.

6.3 Parameter tuning
Appropriate tuning of different vital parameters of the underlying communication protocol is a significant issue in the measurement study as well as collection of the necessary data for real-time analysis. In the following, we study these issues in detail.

Glossy Period (GP): Fig. 7(a) shows the approximate values of the CP for different types of vehicles at different speeds. On the one hand, a larger GP may increase the cases of false negatives as the vehicles may pass through the gaps between instances of DCPhase which results in a fall in detection accuracy as shown in Fig. 7(b). On the other hand, a smaller GP is also not a good solution, as it would increase the energy consumption in low-power devices due to the frequent execution of RF-based communication tasks. It can be seen that the minimum CP
considering all the realistic cases, is about 256 ms (considering 802.15.4-based intra-MP communication). This implies that the period of SCPhase, i.e., GP must be at least 256 ms.

**Transmission Power** (TXP): Setting an appropriate TXP level plays a very important role in developing a noise-free training data set as well as accurate testing. The detection efficiency of the proposed framework strongly depends on the sensitivity of the DCPhase w.r.t. the passage of the vehicles. Intuitively, a low-strength signal gets more affected due to vehicular obstructions, while a higher-strength signal acquires the ability to completely tolerate the vehicular disturbance. Thus, although low signal strength goes in our favor, setting too
Fig. 8. (a) Variation of reliability and accuracy w.r.t. TXP. (b) Effect of TXP on the metric.

Fig. 8i shows the Reliability of Glossy at different TXP in off-road and on-road settings for inter-MU distances of 8 meters (a) and 12 meters (d). Accuracy in detection of the passage of the vehicles using an SVM classifier trained with data collected at different TXP for inter-MU distances of 8 meters (b) and 12 meters (e). RSSI values at the receiver for inter-MU distances of 8 meters (c) and 12 meters (f). Fig. 8ii shows the effect of TXP on the metric (a) Latency (b) Timeout-Count (c) Reception-count and (d) Radio-On time.

low TXP in the DCPhase may have various undesired effects. In particular, the packet may either be completely abolished or it may result in too low residue signal strength to be successfully detected by the receiver radio. Environmental noise or other CTI-related RF interferences may also be an issue with too low TXP. In other words, on the one hand, setting a low TXP may result in a rise in false positives due to an increase in random packet drops. On the other hand, setting a higher value of TXP may either result in a false negative or even cause a wrong prediction where a vehicle of a specific class is categorized as belonging to some other class of larger size. To get a clear understanding of the dynamics, we repeat the measurement study under different TXPs in both off-road (without traffic) and on-road (over a moderately busy road with approximately 300 vehicles per hour). In general, the target is to make DCPhase sensitive and SCPhase resilient to the passage of vehicles. In the measurement experiments reported in the previous section, we set the TXP in DCPhase and SCPhase as 15 (i.e., low) and 31 (i.e., high), respectively. These values are reported w.r.t. the TXP levels of CC2420 (802.15.4 compatible radio used in TelosB). It allows 32 different TXP levels starting from 0 to 31 where 31 representing the highest power level of 0 dBm.

**Packet Size:** Packet size is also an important issue in the current context. Packets with larger sizes are more sensitive to the passage of vehicles, while packets with smaller sizes may have quite low sensitivity. The work [43] shows a detailed study on this topic. To maintain a balance, we keep a moderate packet size of 30 bytes throughout all our experiments.

The distance between the transmitter and the receivers (i.e., inter-MU distance) in all these experiments is set at 8 meters.

**Off-road vs On-road Study:** All the metrics, reported in Sec. 7.1, are measured over iterations with TXP starting from 3 up to 31. RSSI values at the receiver end are shown in 8(c). To obtain a reference, we also execute the plane Glossy protocol at different TXP in the same setting in both off-road and on-road settings. Fig. 8(a) shows the plot of R obtained from plane-Glossy. In off-road settings, a drastic drop is visible in R below TXP 11. However, under on-road settings, due to the presence of the vehicles, a minimum TXP of 25 is necessary to achieve reliability of Glossy above 99%. We carry out several experiments with the proposed detection strategy in the same setup under different TXP values between 11 and 25. Fig. 8i shows the plots of average LT, TC, RO, and RXCT and their standard deviation over all these experiments in both off-road and on-road. As visible from the results, at TXP above 11 all the metrics are quite stable in off-road settings and do not show any variation. However, all of them are found to be widely varying under on-road settings (as visible through the standard
deviation/error bars in Fig. 8i). In general, the lower the TXP, the wider the variation. It can be inferred from the study that any TXP above 11, but below 25, is quite good for the vehicle-detection purpose.

7 ML-based Detection and Classification in the Edge

The aim of LiVeR is to enable edge devices to detect and classify vehicles on-the-fly without any support from the cloud or any centralized service. This is achieved through a small computation step in the DCPhase just after the communication steps are over. A study of the interaction with the communication protocol with the vehicles depicts that the use of a straightforward threshold-based strategy is not well-suited for the targeted task. Therefore, an ML strategy Support Vector Machine (SVM) which is quite known for its efficient but lightweight operation [44] is employed in the computation step. The flow diagram in Fig. 2 shows the sequence of operations. Communication steps in DCPhase are used for derivation of the metrics which are used as features for the SVM. Sec. 7.2 studies the distribution of the metrics and justifies the use of SVM. Section 7.1 provides the details of various data sets collected for training and evaluation of the SVM-based operation. The detection and classification accuracy are separately evaluated in Sec. 8.1 and Sec. 8.2, respectively.

7.1 Data collection

We carry out measurement experiments with PG, MPG as well as DMPG at multiple different places in the city of Bhubaneswar. Considering the value of $t_{CP-min}$ as 265ms, we set the values of GP, $N_I$, $d_x$ (for MPG) as 256ms, 5, and 16ms, respectively. All four metrics (TC, RO, RXCT, and LT) are used together as a feature set to characterize and label the events in each GP. The datasets and their characteristics are summarized in Table 3a. Headway distribution is a very important characteristic of traffic [8]. It reflects the average spatial gap between the vehicles. We roughly calculate the headway distribution from GT for each data set as shown in Table 3a. The datasets can be categorized into three types. Data sets D1 and D2 are collected in the morning hours when the intensity of traffic (single-lane) is quite low (approx. 120 vehicles/hour) and headway is quite high. Data sets D3, D4, D5, D6, and D7 are collected under moderate intensity and multi-lane traffic (approx. 400 vehicles/hour) with moderate headway. Dataset D8 is collected under multi-lane and high-intensity traffic (approx. 1000 vehicles/hour) with the lowest headway. Datasets D4, D5, and D9 are collected specifically to understand the performance of LiVeR under adverse weather conditions.

| Place            | Vehicle Characteristic | Headway Distribution (Mean) | Time     | Duration | Weather and Light |
|------------------|------------------------|-----------------------------|----------|----------|-------------------|
| City-Area (D1)   | More Type-M and Type-S, rare Type-L | 3min | 7:00 AM | 2 Hour | Sunny and Day |
| College Road (D2)| More Type-M and Type-S, rare Type-L | 4min | 9:00 AM | 5 Hour | Sunny and Day |
| Highway (D3)     | More Type-L and Type-M, moderate Type-S | 10sec | 5:00 PM | 1 Hour | Sunny and Day |
| City-Area (D4)   | All Types of vehicles  | 5sec | 6:00 PM | 1 Hour | Rainy and Night |
| City-Area (D5)   | All Types of vehicles  | 5sec | 7:00 PM | 1 Hour | Rainy and Night |
| City-Area (D6)   | All Types of vehicles  | 2min | 2:00 PM | 1 Hour | Sunny and Day |
| Near to Railway Station (D7) | More Type-M and Type-S, moderate Type-L | 5sec | 7:00 PM | 1 Hour | Winter and Night |
| City-Area (D8)   | More Type-S and Type-M, moderate Type-L | 2sec | 8:00 PM | 1 Hour | Winter and Night |
| City-Area (D9)   | All Types of vehicles  | 3min | 6:00 AM | 2 Hour | Fog and Day |

(a) Description of the experimental setting, the status of the traffic and vehicles during experiments.

(b) Cross-validation of average classification accuracy in the experiment described in Sec. 8.2 over a few different pairs of data sets.

Table 3. Dataset
7.2 Distribution of the metrics
With the help of the GT of the collected data set, we try to understand the variations in the pattern of interactions of the vehicles of different classes on different strategies used in the DCPhase (i.e., PG, MPG, and DMPG). We obtain the distribution of the values of each of the four metrics, i.e., TC, LT, RO, and RXCT, over all the classes as shown in Fig. 9, 10, and 11, for PG, MPG and DMPG, respectively. For a quick comparison, we group the obtained values for each of the different metrics and calculate the number of iterations of DCPhase in the corresponding experiment that satisfies each of these groups for each of the four different classes of vehicles. From Fig. 9, it can be seen that although the distribution of the values approximately correlates with the type of vehicles, there are many cases where the metrics show similar behavior for different classes of vehicles. For instance, the metric TC reflects how many times the initiator had tried to make a successful communication with the receiver. Therefore, ideally, the expectation is to have the values of TC be quite proportional to the size of the vehicles. However, in reality, the distribution shown in Fig. 9 does not reflect it. Rather, the values are found to be more or less spread over the possible range of TC. A similar effect is visible in most of the metrics. It indicates that the automation of the classification process by the devices on-the-fly is not straightforward using PG.

In contrast, the distributions of the metrics for DMPG and MPG exhibit more deterministic behavior with fewer deviations. Fig. 10(a), 10(b), 10(c), and 10(d) show the distribution of TC in MPG for all the four different classes. Fig. 11(a), 11(b), 11(c), and 11(d) depict the same for DMPG. The differences between the same distributions obtained from PG, as shown in Fig. 9(a), 9(b), 9(c), and 9(d) are very clearly visible. In MPG, most of the vehicles get at least one chance to collide with the inter-MU communication, and hence, the ambiguities in the values of the metrics among the classes drastically drop. In the case of DMPG, the ambiguities are even lesser as the vehicles get more chances for collisions, and hence, more instances are obtained to produce a better average. In particular, it can be seen that the metric TC is quite efficient in determining the type of vehicle in DMPG. In MGP the behavior is also almost predictable, while in PG it appears quite difficult due to the higher chances for the vehicles to pass through without colliding. A similar effect can be observed in the distribution of all three other metrics.

Minute study and observations on the distributions of the metrics indicate that a lightweight ML tool like SVM would be quite good enough for carrying out the job of on-the-spot detection and classification of the vehicles on the edge itself. Next, we provide further details on the training of the SVM.

7.3 Training of SVM
The detectability and observability of the proposed networked sensor model largely depend on the appropriate training of the employed ML tool. The SVM used for detection and classification is trained using the data set demonstrated in Table 3a. For training, we use a machine with an Intel i7 processor and 8 GB RAM. To keep the size of the trained model low as well as to have computationally efficient testing, we use a one-vs-rest strategy and a linear kernel for training. The four metrics LT, TC, RXCT, and RO are used as four features. The three categories of vehicles, along with the absence of any vehicle (i.e., Type-N) are used as the target class for training. This results in a quite small size of the trained model (approx. 150 bytes) which we port to TelosB for carrying out the job at the edge itself.

8 Evaluation
This section provides a detailed report on the evaluation studies of the proposed lightweight detection and classification mechanism used in LiVeR. We also report how the proposed strategy works under occlusion in the presence of multiple vehicles with substantially low headway. Finally, we detail the study on the power-consumption of LiVeR in comparison with the state-of-the-art strategy.
8.1 Vehicle Detection

For each of the four metrics (LT, TC, RO, and RXCT), 80% of the collected data are used for training the SVM. The rest of the data are used for recalling/testing. Fig. 8i(b) shows the accuracy (w.r.t. GT) results while recalling at different TXP based on the metric LT. The accuracy is found to be quite high in the range of TXP 11 to 21. Tallying with the collected GT we find that for TXP higher than 21, due to sufficient signal strength, medium- and lower-size vehicles fail to make enough impact.
Fig. 11. Distribution of the metrics TC, LT, RO, and RXCT for vehicles of Type-L, Type-M, Type-S, and Type-N, denoted by L, M, S, and N, respectively, under DMPG.

The appropriate TXP for DCPhase also depends on the inter-MU distance. To understand the dynamics, we repeat all the experiments for an inter-MU distance of 12 meters. RSSI at the receiver for an inter-MU distance of 12 meters is shown in 8i(f). Fig. 8i(d) shows that at least a TXP of 15 is needed for the plane Glossy to function reliably under this setting. Accuracy obtained from SVM (based on LT) (as shown in Fig. 8i(d)) is found to be quite high in the range of TXP 15 to 28. Due to the higher distance between MUs, at TXP beyond 28, medium and smaller vehicles fail to make enough impact.

From all these results, it can be observed that although the exact values differ for different inter-MU distances, the overall pattern remains the same. Moreover, for all other metrics (RO, RXCT, and TC) we also find almost similar behavior. In this work, we primarily focus on single-lane traffic. Hence, for all the rest of the experiments, we set the inter-MU distance at 8 meters. Accordingly, for SCPhase, we set TXP above 25 to keep the system running despite vehicular obstructions, while for DCPhase we set TXP between 11 and 25 for to induce controlled sensitivity.

### 8.2 Vehicle Classification

| Vehicle | Small | Medium | Large | No-vehicle |
|---------|-------|--------|-------|------------|
| Small   | 96.3% | 3.6%   | 0%    | 0.1%       |
| Medium  | 0.98% | 98.5%  | 0.54% | 0%         |
| Large   | 0%    | 1.2%   | 98.8% | 0%         |
| No-vehicle | 0.1% | 0%     | 0%    | 99.9%      |

(a) Detailed classification accuracy when training and testing is done over D1.

| Vehicle | Small | Medium | Large | No-vehicle |
|---------|-------|--------|-------|------------|
| Small   | 91.3% | 8.5%   | 0.1%  | 0.1%       |
| Medium  | 2.1%  | 92.3%  | 5.5%  | 0.1%       |
| Large   | 0.1%  | 6.09%  | 93.8% | 0.01%      |
| No-vehicle | 0.1% | 0%     | 0%    | 99.9%      |

(b) Detailed classification accuracy when training is done over D4 and testing is done over D4.

Table 4. Accuracy

The pattern of the distribution of the metrics obtained from different classes through the measurement studies demonstrates the applicability of the lightweight and comparatively simpler ML tool SVM as the primary means. We use a combination of all four metrics for accurate predictions.
RF Based Works | Detection Accuracy (%) | Classification Accuracy (%)  
---|---|---
Bluetooth Based, Bernas et al. \[2\] | 98 | ? | 97 | ?
WiFi Based, Won et al. \[52\] | 99.4 | ? | 91.1 | 91.1*  
Radar Based, Urazghildiev et al. \[50\] | ? | ? | 85 | ?
Zigbee Based, Benjamin et al. \[44\] | 100 | ? | 95.9 | ?
LiVeR [This Paper] | 99.9 | 84 | 98.4 | 72.5

Table 5. Comparison of detection and classification accuracy with state-of-the-art works. Here, '?' represents the term unknown or not discussed in the literature. The work by Won et al. \[52\], marked with '*', considers rural areas for multi-lane traffic with a very low frequency of vehicles. Thus, in effect, it does not consider overlapping scenarios too.

For each of the data sets shown in Table 3a we first train the SVM using 80% of the labeled data and use the rest for testing purposes. Significantly high accuracy in classification can be seen with data set D1 in Table 4a. Data set D2 also shows quite similar values. We further experiment with the data sets having higher traffic intensity (D3-D7) and find that LiVeR shows quite good performance with an average accuracy of 92.5%. For cross-validation, we train the model using one data set and test it using another data set. We carry out this for several combinations of the data sets as summarized in Table 3b. The rising intensity of traffic can be seen to cause degradation in the performance of LiVeR. Overlapping occurrences of the vehicles are one of the prime reasons which are explicitly studied in the next section (Sec. 8.3).

Systematic use of RF as a sensor for detecting and classifying vehicles enables LiVeR to perform almost without any degradation during even adverse weather conditions or night time which is quite hard for vision-based approaches to accomplish. Results based on data sets D4, D5, and D9 depict this fact as shown in Table 4a and Table 4b.

Table 5 summarizes the performance of the existing RF-based strategies for both detection and classification. The WiFi-based work \[52\] also addresses multi-lane traffic. However, to avoid complexity it deals with only rural traffic which is quite simple compared to the city traffic considered for the testing of LiVeR. Evaluation of LiVeR under multi-lane city traffic is studied in Sec. 8.3.

8.3 Performance in Mixed Setting

Almost all the studies on vehicle detection and classification done so far, especially those based on RF deal with only single-lane traffic \[44\]. None of the works even demonstrate how their solution behaves under multi-lane as well as high-intensity traffic. In particular, the effect of overlap or very low inter-vehicle distance, which is formally described with the help of headway-distribution \[8\], has not been shown in almost any of the existing works.

Low headway can confuse in general any vehicle classification strategy. However, although accurate classification is hard, an approximate idea regarding the type of vehicles or the resulting level of congestion over the road can also make quite good sense and become useful in carrying out behavioral studies of traffic. In the following, we note three issues in LiVeR that enable it to get such an approximate idea in multi-lane traffic.

- **Shading-Effect**: Unlike the existing RF-based works that analyze the attenuation of RSSI/physical layer parameters, LiVeR explicitly uses low-power RF transmissions and hence can exploit the possibility of complete abolition of the full packets too. Therefore, when multiple types of vehicles of different classes overlap or pass together, LiVeR can sense the presence of the vehicle that has the largest size among them. This we refer to as the shading-effect.
- **Degradation-pattern**: when multiple vehicles overlap with each other, although it introduces possible confusion w.r.t. the length of the vehicles, their height still remains unaltered. Therefore, the characteristics

ACM Trans. Internet Things
of the obstructions do not abruptly change. For example, when multiple Type-S vehicles pass in series or even overlap with each other, it would not abruptly alter the pattern of degradation of the inter-MU communication in the DCPhase and hence, the accuracy of the detection and classification w.r.t. the measured metrics would not be affected much. However, confusion arises w.r.t. time, i.e., how to distinguish one vehicle from the other. Fortunately, in LiVer, we use time-slotting. In particular, we reset the analysis in every instance of the GP. Thus, although multiple Type-S vehicles may confuse other existing approaches [2, 44] as a single Type-L vehicle, the inaccuracy of our approach will always be bound by the possibilities that can happen within a single GP.

- **Combined-monitoring**: LiVer can seamlessly combine the outcomes from multiple MPs together through the IPhase. This opens up the avenue for fine-grained classification of the vehicles. As an example, consider a road with three MPs installed sequentially. When multiple vehicles pass through the road due to variations in speed, with a high chance, different MPs would capture different views at different time points as shown in Fig. 12. When the data from all three MPs are shared with each other or collected at the sink node through the IPhase, a simple union along with additional time-stamp-based processing can be employed for deriving a fine-grained picture of the road. We carry out an explicit experiment-based study to further explore this issue which is detailed in Sec. 9.

**Low-headway in camera-based approach**: Note that, low-headway has also been a serious problem due to occlusion in camera-based solutions, despite the use of heavy-weight equipment, computing systems, cloud assistance, and sophisticated computer-vision strategies. Low-light as well as adverse weather conditions, e.g., Fog or rain, have an additional impact on such issues. In contrast, the lightweight and low-cost solution proposed by LiVer performs considerably better even under low headway amidst all such adverse situations. In the following, we provide a quantitative study of the accuracy of LiVer under low headway.

(a) Classification accuracy when training done on high traffic rate data and tested on moderate traffic rate data.

(b) Table showing the time-stamped data and their combined result to derive the complete and accurate picture at the sink.
To study the behavior under inter-vehicle overlap or low headway we introduce three more classes: \textit{L-mix}, \textit{M-mix}, and \textit{S-mix}. These are defined considering the \textit{shading-effect}. For example, the presence of one Type-L vehicle can shade the presence of vehicles of Type-S and Type-M. The class \textit{L-mix}, thus, represents a mixed situation where all types of vehicles may be present with \textit{at least one Type-L vehicle}. \textit{M-mix} represents the mix of Type-S and Type-M vehicles, with \textit{at least one Type-M}, but \textit{no Type-L vehicle}. \textit{S-mix} denotes the cases where \textit{only multiple Type-S vehicles are present}.

**Data-Set:** We collect data-set D8 from multi-lane traffic in a similar way as explained in Sec. 8. The detailed characteristics of D8 are shown in the second last row of Table 3a. We label D8 considering a total of 7 classes (including Type-N) and train an SVM accordingly.

Table 6a summarizes the average accuracy values when training is done on 7 classes over the complete data set D8 while testing is done on the data set D4 for each of the four basic classes. We observe that several cases of Type-S are wrongly classified as \textit{S-mix}. Similarly, some occurrences of Type-M are classified as \textit{M-mix}. We also observe that two overlapping occurrences of Type-M are sometimes classified as Type-L, as well as two overlapping occurrences of Type-S vehicles are sometimes classified as Type-M, as discussed above. The mean accuracy of the overall classification, thus, degrades due to the overlapping occurrences of the vehicles. For a complete understanding, we derive the headway distribution and its average from different segments of our collected data with the help of GT. Fig. 14i shows the accuracy obtained in vehicle classification for different mean-headway. It can be seen that classification accuracy gracefully degrades with lower mean headway, (i.e., under higher traffic intensity). Conversely, the proposed solution would still be able to bring forth an approximate idea regarding the classes of the vehicles under quite low-headway/heavy multi-lane traffic.

### 8.4 Power Measurement Study

A recent RF-based strategy [44] adopts a similar RF-based approach for vehicle detection and classification, as briefed in Sec. 2. However, it fundamentally employs a blind flow of packets with an inter-packet gap of 8 ms which is fixed irrespective of the presence of traffic or even its intensity. The lack of an efficient protocol and dependence on raw RSSI make the strategy quite energy-hungry. In the following, we demonstrate a comparative study in terms of energy consumption and longevity of LiVeR and the work [44] in terms of battery life.

From the collected data set we find that in a single NI in LiVeR for Type-S, Type-M, and Type-L vehicles it results in the transmission of 3, 4, and 6 packets on average. For a generic calculation, we set the values of $\alpha$, $\beta$, and $\gamma$ of Equation 5 accordingly. Considering high traffic over a span of 1 hr, and assuming the value of $x$ to be 0.8, we calculate the total number of packets transmitted under two distinct cases with the values of the fractions $V_N$, $V_M$, $V_L$ as (0.2, 0.3, 0.5) and (0.5, 0.3, and 0.2), referred to as case-1 and case-2, respectively. It can be calculated from Equation 6 that LiVeR employs approx. 167625 packets under case-1 and 155000 packets under case-2, while the work [44] employs, 45,0000 packets under both the cases. Fig. 13i shows the comparison of the number of packets used in LiVeR and the work [44] over a span of 1 hr. In both cases, it can be observed that LiVeR employs approx. one-third number of RF-packets while achieving similar accuracy in the detection and classification of the vehicles.

We further study the consumption of energy by the TelosB devices during the execution of different phases of LiVeR. The study is conducted using \textit{Monsoon power meter}. SCPHase works at TXP of 31, in order to maintain the robustness of the protocol. DCPHase runs under quite a low TXP based on the inter-MU distance. We measure the lifetime of the battery under different values of GP and different TXP for different capacities of the battery. The energy consumption pattern of a single GP in the initiator and the receiver MU are shown in Fig. 13ii(c) and 13ii(d), respectively. The average lifetime of the MUs under different battery capacities as obtained from the assessment made by the power meter is shown in Fig. 13ii(a) and 13ii(b) for two different GP values. It can be
(i) Comparison of the number of RF-packets used for vehicle detection and classification in LiVeR and the strategy studied in the work [44].

(ii) Part (a) and (b) show the battery lifetime in LiVeR for two different GPs. Part (c) and (d) show the pattern of consumption of current in the sender and the receiver, respectively, during the execution of a single iteration of LiVeR (MPG model) implemented in TelosB and Contiki OS.

Fig. 13

(i) Accuracy for different headway

(ii) Simulation scenario

Fig. 14

observed that LiVeR can sustain on battery for at least 145 days under a GP of 256 ms, while it can sustain even for more than one year (500 days) when the GP is set at 2 sec.

9 Wide area monitoring

One of the major aims behind the development of LiVeR is to easily link multiple MPs spread over a wide area and carry out coordinated measurement, which not only helps in fine-grained classification of the vehicles but is also an important issue in solving various practical problems on-the-fly. This section reports our simulation-based case studies on this issue. Subsequently, we also demonstrate an outdoor experiment-based study of a practical application of LiVeR.

9.1 Simulation based study

We use the Contiki network simulator Cooja [34] for all these wide-area simulation-based experiments. Two simulation scenarios, referred by ES-1 and ES-2 (shown in Fig. 14ii), are created to mimic two different types of physical arrangements of the roads. ES-1 and ES-2 employ a total of 100 TelosB devices forming a 2D-grid-like

ACM Trans. Internet Things
structure over an area of 2000 X 2000 sq. meter and 1000 X 1000 sq. meter, respectively. Inter-MP distance is set as 50 meters and 75 meters, in ES-1 and ES-2, respectively.

We explore the roles of the FNs, and their interactions with the MPs in accomplishing a wide-area operation. Since, under the simulation setting, we do not have the real vehicles to obstruct the communication links, for the DCPhase we use the simplest possible design framework, i.e., PG. In the IPhase we integrate the protocol Chaos [21] for all-to-all data-sharing. The metrics Latency (LT), Radio-on (RO), and Reception-Count (RXCT) as defined in Sec. 3.1 are also used for the evaluation of IPhase. However, instead of TC (i.e., Time-out count) we use Hop-Count as described below.

**Hop-Count:** In an instance of Glossy, the parameter Relay-Count (RC) is used by a node to keep track of the number of times a packet has been transmitted/forwarded [15]. It is both maintained as a variable in the node as well as communicated through all the transmissions. The value of the RC that a node obtains from the reception of the very first Glossy-packet indicates an approximate hop-position of the node w.r.t. the initiator. We use this value as Hop-Count (HC) in this study.

Each experiment is executed for at least 1000 iterations. The metrics are measured in each node and presented as average over all the nodes and iterations along with the standard deviation.

**Effect of number of MPs:** First we experiment with a skeleton network created solely by the MPs (i.e., no FNs). Fig. 15i shows the plots of HC, RXCT, LT, and RO for both SCPhase and DCPhase (in PG) for varying numbers of MPs under ES-1. The higher the no of MPs, the higher the diameter of the network which is reflected in almost linear rise in RO, LT, and HC. RXCT remains the same implying a smooth functioning of the system. Note that the change in the number of MPs affects only the results obtained from the SCPhase, but has no impact on DCPhase as it faces no change while getting executed (DCPhase is inscribed inside SCPhase).

**Effect of FN:** To understand the effect of the FNs we start with a setting with 20 MPs and gradually keep on adding the FNs in different ways. Fig. 15ii shows the average HC, RO, RXCT, and LT in each of these cases in both SCPhase and DCPhase in ES-1. It can be observed that there is almost no effect on the system performance as long as the MPs are installed maintaining a proper inter-MP distance so that the network remains connected. Glossy, being a highly robust protocol, is quite able to sustain even with a skeletal setting of the MPs with no FN.

**IPhase:** Fig. 15iii(a), 15iii(b), 15iii(c), 15iii(d) show the LT and RO in IPhase for varying number of MP and FNs under ES-1 and ES-2, respectively. It can be observed that RO and LT in IPhase almost linearly rise with the number of MPs in both settings while do not show any significant effect with the rise in the number of FNs. Note that only the MPs act as the source of data while FNs only act as forwarding devices.

The framework proposed in LiVeR is quite capable of binding a large number of MPs under the same hood. However, due to the installation of the MPs only over the roads, the inter-MP network may look minimally connected. The experiments described above show that the ST-based framework used in LiVeR in spite of having such a minimally connected structure can carry out its job without support from even a single extra forwarding node (FN).

### 9.2 Experiment-based Case Study-1

It is quite interesting to note that the wide-area monitoring capability of LiVeR provides it with a special capability of combining the results obtained from multiple MPs and using it for fine-grained classification of the vehicles as referred in Sec. 8.3. Let us consider the scenario depicted in Fig. 12. There are three MPs installed over a road. Let us also consider three vehicles, namely one large bus and two cars passing over the road. Due to differences and variations in the speeds of the vehicles, there is a high chance that the MPs will have different views at different time points, as shown in the picture as well as summarized in the table 6b.

However, by virtue of the IPhase when the data from the MPs get disseminated among each other or collected at a dedicated sink node, doing a simple analysis based on time-stamp can bring forth a better picture of the...
Fig. 15. Figure 15i shows the effect of the number of MP on the performance of the system. Parts (a), (b), (c), (d) show the plot of LT, RO, HC, and RXCT, in ES-1 for different numbers of MPs. Figure 15ii shows the effect of different numbers of FNs in ES-1. Figure 15iii shows the effect of different numbers of MPs and FNs over the performance of IPhase. FN1: 0, FN2: 3 (L), FN3: 12 (L), FN4: 6 (L, R), FN5: 12 (L, R), FN: 48. L and R indicate the number of FNs taken from the left side and right side of the main road in the emulation setting.

Experimental Study: To evaluate the proposed strategy we carried out an experiment with three MPs installed over a moderately busy road and one sink node in the junction point (endpoint). By using the protocol Chaos in the IPhase we collect data in the sink node. The experiment is conducted for 30 min. The average headway was found to be 20 sec which indicates a setting where several instances were found where vehicles overlapped with each other. An SVM was trained using data sets D1 and D2 (see Table 3a) and it is used for on-the-spot testing in the current experiment. Fig. 16 shows the results. Due to very low headway, the accuracy obtained in the individual MPs is found to be comparatively lower, as is also found for other similar data sets (see Table 3a). However, it can be observed that the predictions at the sink node obtained based on a time-stamp-based analysis and union of the outcomes from the three MPs are up to 20% more accurate than the predictions obtained at the individual MPs for the classification of the vehicles.
9.3 Experiment-based Case study-2

Experiment-based studies reported in Sec. 6 show some failures in LiVeR under high traffic where there are overlaps among the vehicles. However, the performance of LiVeR is much better in single-lane settings where overlapping cases are negligible. In general, the proposed strategy can be quite effective in privacy-preserving detection and classification of vehicles in places where the traffic is low or moderate. For example, understanding the dynamics of the vehicles moving inside a large campus area or inside a certain locality, or residential area inside a city, can be quite crucial for automated and continuous detection of anomalous situation, flow analysis, automatic controls of the gates based on inflow and outflow, automated maintenance of vehicular statistics, and several other similar objectives. Keeping such applications in mind, we carry out experimental studies to explore how LiVeR performs in real standalone applications inside the campus of IIT Bhubaneswar. We build a setup consisting of 12 MPs near the main gate and its surrounding area inside the campus. Fig. 17(a) shows the installation of a single MP while Fig. 17(b) shows the placement of all the 12 MPs over the target area. In the following, we discuss our studies based on the measurements obtained from these MPs.

**Independent Measurement:** MP1 is installed near the main entrance to the campus of IIT Bhubaneswar. It’s installed in a way so that all inbound traffic (i.e., all vehicles that are entering the campus) go through MP1. We carry out the recording using MP1 continuously for a period of 5 hours. Due to installation near the main entry, each and every vehicle is made to enter one by one through the MP. All the measurements done in MP1 are communicated and logged in a central server through the attached RPi (see the device setting in Fig. 5ii). Video recording of the experiment is done to collect GT. A comparison of the on-the-spot classification of the vehicles by MP1 and the same obtained through the GT is shown in Fig. 18i. Due to sequential entries of the vehicles with sufficient headway, MP1 is found to be quite accurately detecting and classifying almost all the vehicles that enter the campus.

Very little public transport (buses) is allowed to enter the campus. Thus, the number of Type-L vehicles inside the campus is comparatively low. It includes school buses, local conveyances (traveller), and a few trucks due to the ongoing construction works. However, trucks are not usually allowed much in the morning hours due to the movement of the students. During our experiments, therefore, we found the number of Type-L vehicles detected to be quite low. Hourly measurement data of MP1 is also shown in Fig. 18i(b). It can be seen that during the morning hours, inbound traffic (vehicles entering through MP1) is quite high, and it decreases gradually with time. Total and hourly measurement data of all the MPs are shown in Fig. 18ii(a) and 18ii(b), respectively. However, note that it is not feasible to record the video (for GT) simultaneously at each of the MPs. The accuracy of these measured data, therefore, can not be checked directly, as is done for MP1. Fig. 18ii shows only the number...
of vehicles detected by the MPs and their hourly decomposition. To indirectly assess the accuracy of all the MPs we exploit the correlated measurements over different MPs throughout the experiment, as explained next.

**Coordinated-monitoring:** For automated data collection and coordinated learning by the MPs, we activate the IPhase of the protocol where we use the protocol Chaos in all-to-all data-sharing mode as studied in Sec. 9. The area selected for the experiment is fundamentally a region that connects three different important places of the campus, namely, the *academic area*, *administrative building*, and *the residential area*. The directions of these places are shown in Fig. 17(b). From the placement of the MPs, it can be observed that if any vehicle passes through MP1, it is supposed to be detected by MP7, followed by either MP9, in case it goes towards the academic area, or by MP6, in case it goes towards the administrative building or by MP4, in case it goes towards the residential area. Some possible paths emerging from MP1 are shown in Table 7a. Similarly, if something is detected to be passing through MP3, the same vehicle should be also detected by MP2 if it goes outside the campus, or by MP7 and MP9 in case it enters the academic area, or by MP6 in case it goes towards the administrative building. If some vehicle is detected by MP8, it may be subsequently detected also by MP6 or MP4 in case it goes towards the administrative building or outside the campus, respectively. From the above-mentioned trajectories, it can be understood that local measurements obtained from different MPs should have an approximate correlation with each other. In this context, we study the following two types of correlations.

| Origin | Path | Equation | Remarks |
|--------|------|----------|---------|
| MP1    | MP1 → MP7 → MP9 → MP10 → MP11(RES) | MP1 + MP5 + MP7 + MP9 = MP2 + MP4 + MP6 + MP9 | Kirchoff’s law for traffic |
| MP1    | MP1 → MP7 → MP9 → MP12(Other academic area) | MP10 + MP11 | Same direction flow |
| MP1    | MP1 → MP7 → MP6 → MP4 (Admin area) | MP1 = MP4 + MP6 + MP9 | For inflow analysis (Frequent analysis) |
| MP1    | MP1 → MP7 → MP4(Residential) | MP9 = MP10 + MP12 | To check flow at department level |

(a) Possible paths through MP1.

(b) Possible equations to summarize the flow of vehicles through the MPs.

Table 7

- **Temporal correlation:** When a specific MP, say MPi detects a vehicle, its quite well understood that after a certain time interval, say $\delta_1$, $\delta_2$, $\delta_3$ a series of other MPs, say MPj1, MPj2, MPj3, respectively, should be also able to detect the same vehicle. However, it depends upon the path of vehicles and the placement of MPs. As per our deployment, when MP7 detects a vehicle, at time $t$, either of MP6 or MP4 should detect the same vehicle at $t+\delta_1$ or $t+\delta_2$, respectively, based on the inter-MP distance and the velocity of the vehicle. When there is a single vehicle traveling through these MPs, such rules are quite applicable and the MPs can collaboratively decide whether there is any anomaly or abnormality such as a vehicle taking unexpected time to cover the distance between two MPs, etc. However, in reality, there are multiple vehicles that can appear at any point in time at any of the entry points MP1, MP3, MP12, or MP10 which would make this process complicated. Inside a campus still, the situation remains under control and a good many conclusions can be derived. However, because of the complexity of the process and since it has its own importance we do not deep dive into these issues in the current paper.

- **Accumulated correlations:** Temporal correlation although easier to understand, however, is quite hard to directly demonstrate through the results. To reflect the same, we first study the possible paths and the possible trajectories of the vehicles over the target area as shown in Fig. 17(b). We isolate a few possible junction points and for each of them, we use the nearest suitable MPs to calculate the inflow and the outflow towards and outwards the junction, respectively, which are finally used to compose the LHS and the RHS of the equations as shown in Table 7b. For example, under normal conditions, over time the sum of the number of vehicles sensed by MP1, MP3, MP5, and MP8 should be similar to the sum of the vehicles sensed by MP2, MP4, MP6, and MP9 together. This fundamentally mimics the *Kirchoff’s law of current* in the context of traffic as expressed by the equation shown in the first row of Table 7b.
LiVeR: Lightweight Vehicle Detection and Classification in Real-Time

Types of Vehicles

| Type-S | Type-M | Type-L |
|--------|--------|--------|
| No. of Vehicles | 0 | 10 | 20 |

Real

Classified

(a) (b)

Real

Classified

Time

9AM 10AM 11AM 12PM 1PM

No. of Vehicles

0 5 10 15 20

Type-L

Time interval (30 sec)

1 2 3 4 5 6 7 8 9 10 11

0 1 2 3 4 5

MP1+MP3+MP5+MP8

MP2+MP4+MP6+MP9

Time interval (30 mins)

1 2 3 4 5 6

Number of Vehicles

0 20 40 60 80

MP1+MP3+MP5+MP8

MP2+MP4+MP6+MP9

Fig. 18. Application related results

(i) Classification accuracy at MP1.

(ii) No. of vehicles detected by different MP over the full period of the experiment (a) and hour-wise (b).

(iii) Comparison of the LHS and the RHS of the equation shown in Table 7b first row, at the interval of 30 sec (a), and 30 min (b).

Results: Fig. 18iii shows the comparison between the RHS and LHS of the equations shown in Table 7b. It especially demonstrates how the comparisons evolve through accumulation over different time spans. For a vehicle to move across all the MPs associated with a given equation takes a finite amount of time. For a bigger junction, e.g., for the equation shown in the first row of Table 7b, involves several possible trajectories, and hence, takes comparatively longer time for the vehicles to cover all the components associated with the inflow and outflow of the junction. Therefore, it can be observed that for accumulation even over 30 seconds or below the match between the LHS and RHS of the equation is quite poor (see Fig. 18iii(a)). However, for accumulation beyond 5 mins, the match is found to substantially improve. With an accumulation of 30 mins or more, an almost perfect match between LHS and RHS can be observed. These results provide enough evidence that the detection at each and every MP has been as accurate as expected.

Discussion: The capability of LiVeR in the detection and counting of vehicles is quite evident from the experimental results, as explained above. From the dynamic information obtained from different MPs, various useful inferences can also be drawn. For example, MP6 and MP5 in Fig. 17(b) yield an approximate inflow and outflow w.r.t. the admin area, respectively. Thus, the values of MP5 and MP6 at any point in time can give an approximate idea regarding the number of vehicles that are currently inside the admin area. Several similar inferences can be drawn based on different MPs and their combinations. ST-based framework enables LiVeR to efficiently use the shared ISM frequency band through lightweight strategies [9, 11] for in-parallel measurements in multiple nearby MPs. Moreover, in the IPhase LiVeR can exploit advanced ST-based data-sharing strategies [10, 12] for efficient inter-MP collaboration. LiVeR can exploit these facilities to verify the cumulative behavior of the vehicles. In general, it can be fed with a set of predefined equations. For the detection or identification of different possible anomalous situations, LiVeR can keep checking the overall status through the evaluation of the equations at regular intervals. Study of such applications is one of our future goals in this direction.
10 Conclusion

Vehicle detection and classification are important components of a smart traffic system. Unfortunately, all the existing solutions involve costly, specialized, and complex strategies, and mostly even ignore the issue of wide-area coverage. In this work, we introduce LiVeR, a low-cost, flexible, and easy-to-install IoT-based strategy for real-time detection and classification of vehicles. LiVeR utilizes an RF-based low-power communication protocol. Unlike the existing RF-based works that directly employ raw physical layer parameters, LiVeR exploits the parameters of the data-link layer to achieve the goal. In particular, it tries to systematically capture the logs generated during the protocol in action and use them to understand how the protocol fails due to collisions of the RF packets with the vehicles. It keeps the computational need for detection and classification quite low so that the process can be executed in real-time in resource-constrained IoT-devices without any support from the cloud. In addition, the design of LiVeR is quite self-adaptive which enables the IoT-devices to conserve energy as much as possible and sustain solely based on battery power for a substantial time period. Through extensive outdoor measurement studies, we show that LiVeR achieves up to 99.8% accuracy in detection while up to 98.4% accuracy in the classification of the vehicles in three standard classes based on the size of the vehicles under single-lane traffic. We also show that LiVeR spends about one-third of the number of RF packets compared to the state-of-the-art RF-based vehicle detection and classification strategy.

References

[1] Ravneet Bajwa, Ram Rajagopal, Pravin Varaiya, and Robert Kavaler. 2011. In-pavement wireless sensor network for vehicle classification. In Proceedings of the 10th ACM/IEEE IPSN (2011), 85–96.
[2] Marcin Bernas, Bartłomiej Placzek, and Wojciech Korski. 2018. Wireless Network with Bluetooth Low Energy Beacons for Vehicle Detection and Classification. In Computer Networks. Springer International Publishing, Cham, 429–444.
[3] Salim Bitam and Abdelhamid Mellouk. 2012. ITS-cloud: Cloud computing for Intelligent transportation system. In Proceedings of IEEE GLOBECOM. 2054–2059. https://doi.org/10.1109/GLOCOM.2012.6503418
[4] Daniel Bolya, Chong Zhou, Fanxi Xiao, and Yong Jae Lee. 2022. YOLACT++ Better Real-Time Instance Segmentation. In IEEE Transactions on Pattern Analysis and Machine Intelligence 44, 2 (2022), 1108–1121. https://doi.org/10.1109/TPAMI.2020.3014297
[5] Z. Chen, T. Ellis, and S. A. Velastin. 2012. Vehicle detection, tracking and classification in urban traffic. In Proceedings of 2012 15th International IEEE Conference on Intelligent Transportation Systems. 951–956. https://doi.org/10.1109/ITSC.2012.6338852
[6] Marcelo Contreras, Neel P. Bhatt, and Ehsan Hashemi. 2023. A Stereo Visual Odometry Framework with Augmented Perception for Dynamic Urban Environments. In Proceedings of 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC). 4094–4099. https://doi.org/10.1109/ITSC57777.2023.10421981
[7] Marcelo Contreras, Aayush Jain, Neel P Bhatt, Arunava Banerjee, and Ehsan Hashemi. 2024. A survey on 3D object detection in real time for autonomous driving. In Frontiers in Robotics and AI 11 (2024), 1212070.
[8] Richard J. Cowan. 1975. Useful headway models. In Transportation Research 9, 6 (1975), 371–375. https://doi.org/10.1016/0041-1647(75)90008-8
[9] Jagnyashini Debardarshini and Sudipta Saha. 2023. Fine-grained frequencies meet synchronous transmission. In Computer Communications 203 (2023), 99–118. https://doi.org/10.1016/j.comcom.2023.02.026
[10] Jagnyashini Debardarshini and Sudipta Saha. 2023. SyncCast: Real-Time Self-Adaptive and Fault-Tolerant All-to-All Data-Sharing in IoT. In Proceedings of 2023 IEEE 83rd International Conference on Network Protocols (ICNP). 1–11. https://doi.org/10.1109/ICNP59255.2023.10355628
[11] Jagnyashini Debardarshini, Sudipta Saha, Olaf Landsiedel, and Mun Choon Chan. 2020. Start-of-Frame Delimiters (SFDs) for Simultaneous Intra-Group One-to-All Dissemination. In Proceedings of 2020 IEEE 45th Conference on Local Computer Networks (LCN). 100–111. https://doi.org/10.1109/LCN48667.2020.9314842
[12] Jagnyashini Debardarshini, Madhav Tummala, Sudipta Saha, Olaf Landsiedel, and Mun Choon Chan. 2023. TimeCast: Real-Time Many-to-Many Data-Sharing in Low-Power Wireless Distributed Systems. In IEEE Systems Journal 17, 4 (2023), 5726–5737. https://doi.org/10.1109/JSYST.2023.3292892
[13] Zhen Dong, Yuwei Wu, Mingtao Fei, and Yunde Jia. 2015. Vehicle Type Classification Using a Semisupervised Convolutional Neural Network. In IEEE Access. 3 (2015), 2247–2256. https://doi.org/10.1109/TITS.2015.2402438
[14] M. Ergen, Duke Lee, Raja Sengupta, and P. Varaiya. 2004. WTRP - wireless token ring protocol. In IEEE Transactions on Vehicular Technology 53, 6 (2004), 1863–1881. https://doi.org/10.1109/TVT.2004.836928
[15] Federico Ferrari, Marco Zimmerling, Lothar Thiele, and Olga Saukh. 2011. Efficient network flooding and time synchronization with glossy. In Proceedings of ACM/IEEE IPSN. 73–84.
[16] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. 2009. Collection Tree Protocol. In Proceedings of the 7th ACM SenSys (Berkeley, California) (SenSys ’09). Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/1644038.1644040

[17] Vini Gupta and Swades De. 2021. An Energy-Efficient Edge Computing Framework for Decentralized Sensing in WSN-Assisted IoT. In IEEE Transactions on Wireless Communications 20, 8 (2021), 4811–4827. https://doi.org/10.1109/TWC.2021.3062568

[18] M. Haferkamp, M. Al-Askary, D. Dorn, B. Sliwa, L. Habel, M. Schreckenberg, and C. Wietfeld. 2017. Radio-Based Traffic Flow Detection and Vehicle Classification for Future Smart Cities. In Proceedings of IEEE 85th VTC Spring, 1–5.

[19] Dalton Haln, Arslan Munir, and Vahid Behzadan. 2021. Security and Privacy Issues in Intelligent Transportation Systems: Classification and Challenges. In IEEE Intelligent Transportation Systems Magazine 13, 1 (2021), 181–196. https://doi.org/10.1109/MITS.2019.2898973

[20] S. Jeng and L. Chu. 2014. A high-definition traffic performance monitoring system with the Inductive Loop Detector signature technology. In Proceedings of 17th IEEE ITSC. 1820–1825.

[21] Olaf Landsiedel, Federico Ferrari, and Marco Zimmerling. 2013. Chaos: Versatile and efficient all-to-all data sharing and in-network processing at scale. In Proceedings of ACM SenSys. 1.

[22] K. Leentvaar and J. Flint. 1976. The Capture Effect in FM Receivers. In IEEE Tc 24, 5 (1976), 531–539. https://doi.org/10.1109/T.COM.1976.1093327

[23] X. Li and J. Wu. 2016. A new method and verification of vehicles detection based on RSSI variation. In Proceedings of 10th IcST. 1–6.

[24] Mingpei Liang, Xinyu Huang, Chung-Hao Chen, Xin Chen, and Alade Tokuta. 2015. Counting and Classification of Highway Vehicles by Regression Analysis. In IEEE T-ITS 16, 5 (2015), 2878–2888. https://doi.org/10.1109/TITS.2015.2424917

[25] Roman Lim, Reto Da Forno, Felix Sutton, and Lothar Thiele. 2017. Competition: Robust Flooding using Back-to-Back Synchronous Transmissions with Channel-Hopping. In Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks (Uppsala, Sweden) (EWSN ’17). USA, 270–271.

[26] Trista Lin, Hervé Rivano, and Frédéric Le Moué. 2017. A Survey of Smart Parking Solutions. In IEEE T-ITS 18, 12 (2017), 3229–3253. https://doi.org/10.1109/TITS.2017.2685143

[27] Yuhao Liu and Jun Miura. 2021. RDS-SLAM: Real-Time Dynamic SLAM Using Semantic Segmentation Methods. In IEEE Access 9 (2021), 23772–23785. https://doi.org/10.1109/ACCESS.2021.3050617

[28] Wenteng Ma, Daniel Xing, Adam McKee, Ravneet Bajwa, Christopher Flores, Brian Fuller, and Pravin Varaiya. 2014. A Wireless Accelerometer-Based Automatic Vehicle Classification Prototype System. In IEEE T-ITS 15, 1 (2014), 104–111. https://doi.org/10.1109/T-ITS.2013.2273488

[29] Fabian Mager, Andreas Biri, Lothar Thiele, and Marco Zimmerling. 2023. BUTLER: Increasing the Availability of Low-Power Wireless Communication Protocols. In Proceedings of the 2022 International Conference on Embedded Wireless Systems and Networks (, Linz, Austria.) (EWSN ’22). Association for Computing Machinery, New York, NY, USA, 108–119.

[30] Gabriel Michau, Alfredo Nantes, Ashish Bhaskar, Edward Chung, Patrice Abry, and Pierre Borgnat. 2017. Bluetooth Data in an Urban Context: Retrieving Vehicle Trajectories. In IEEE T-ITS 18, 9 (2017), 2377–2386. https://doi.org/10.1109/TITS.2016.2642304

[31] Daniel Minoli, Kazem Sohraby, and Benedict Ochiodotto. 2017. IoT Considerations, Requirements, and Architectures for Smart Buildings—Energy Optimization and Next-Generation Building Management Systems. In IEEE Internet of Things Journal 4, 1 (2017), 269–283. https://doi.org/10.1109/IOTJ.2017.2647887

[32] Beshr Al Nahas, Antonio Escobar-Molero, Jirka Klaue, Simon Duquennoy, and Olaf Landsiedel. 2021. BlueFlood: Concurrent Transmissions for Multi-Hop Bluetooth 5—Modeling and Evaluation. In ACM Trans. Internet Things 2, 4, Article 22 (Jul 2021), 30 pages. https://doi.org/10.1145/3462755

[33] E. Odat, J. S. Shamma, and C. Claudel. 2018. Vehicle Classification and Speed Estimation Using Combined Passive Infrared/Ultrasound Sensors. In IEEE T-ITS 19, 5 (2018), 1593–1606.

[34] Fredrik Osterlund, Adam Dunkels, Joakim Eriksson, Niclas Finne, and Thiemo Voigt. 2006. Cross-Level Sensor Network Simulation with COOJA. In Proceedings of 31st IEEE LCN. 641–648. https://doi.org/10.1109/LCN.2006.322172

[35] Samer Rajab, Ahmad Mayeli, and Hazem Refai. 2014. Vehicle classification and accurate speed calculation using multi-element piezoelectric sensor. In Proceedings of IEEE Intelligent Vehicles Symposium. 894–899. https://doi.org/10.1109/IVS.2014.6856432

[36] Nikunj Bhooshan, Basia Runka, and Elisa Kehl. 2022. Confidence Estimator Design for Dynamic Feature Point Removal in Robot Visual-Inertial Odometry. In Proceedings of IEEE ICNN 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society, 1–6. https://doi.org/10.1109/ICNN49645.2022.9968660

[37] Javier Rivas, Ralf Wunderlich, and Stefan J. Hein. 2017. Road Vibrations as a Source to Detect the Presence and Speed of Vehicles. In IEEE Sensors Journal 17, 2 (2017), 377–385. https://doi.org/10.1109/JSEN.2016.2628858

[38] S. Roy, R. Sen, S. Kulkarni, P. Kulkarni, B. Raman, and L. K. Singh. 2011. Wireless across road: RF based road traffic congestion detection. In Proceedings of Third IEEE COMSNETS. 1–6.

[39] Bijurekha Sen, Abhinav Maurya, Bhaskaran Raman, Rupesh Mehta, Ramakrishnan Kalyanaraman, Nagamanoj Vankadhara, Swaroop Roy, and Prashma Sharma. 2012. Kyun Queue: A Sensor Network System to Monitor Road Traffic Queues. In Proceedings of the 10th ACM SenSys (Toronto, Ontario, Canada) (SenSys ’12). New York, NY, USA, 127–140. https://doi.org/10.1145/2426656.2426670

ACM Trans. Internet Things
[40] Rijuurekha Sen, Bhaskaran Raman, and Prashima Sharma. 2010. Horn-Ok-Please. In Proceedings of the 8th MobiSys (San Francisco, California, USA) (MobiSys ’10). New York, NY, USA, 137–150. https://doi.org/10.1145/1814433.1814449

[41] Chandra Shekhar and Sudipta Saha. 2022. IoT-Assisted Low-Cost Traffic Volume Measurement and Control. In Proceedings of 14th International Conference on COMmunication Systems & NETworkS (COMSNETS). 806–811. https://doi.org/10.1109/COMSNETS53615.2022.9668354

[42] Chandra Shekhar and Sudipta Saha. 2022. An IoT-based Framework for Low-Cost and Light-Weight Vehicle Detection. In Proceedings of 18th International Conference on Distributed Computing in Sensor Systems (DCOSS). 69–71. https://doi.org/10.1109/DCOSS54816.2022.00023

[43] Chandra Shekhar, Sudipta Saha, and Mun Choon Chan. 2021. Mitigating Adversities in Urban IoT-setups: A Sensor Assisted Approach. In Proceedings of IEEE COMSNETS. 413–420. https://doi.org/10.1109/COMSNETSS1098.2021.9352840

[44] Benjamin Sliwa, Nico Piatkowski, and Christian Wietfeld. 2020. The Channel as a Traffic Sensor: Vehicle Detection and Classification Based on Radio Fingerprinting. In IEEE Internet of Things Journal 7, 8 (2020), 7392–7406. https://doi.org/10.1109/JIOT.2020.2983207

[45] M. Stocker, M. Rönkkö, and M. Kolehmainen. 2014. Situational Knowledge Representation for Traffic Observed by a Pavement Vibration Sensor Network. In IEEE T-ITS 15, 4 (2014), 1441–1450.

[46] Z. Sun, G. Bebis, and R. Miller. 2004. On-road vehicle detection using optical sensors: a review. In Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749). 585–590. https://doi.org/10.1109/ITSC.2004.1398966

[47] Hasan Tafish, Walid Balid, and Hazem H. Refai. 2016. Cost effective Vehicle Classification using a single wireless magnetometer. In Proceedings of IWCMC. 194–199. https://doi.org/10.1109/IWCMC.2016.7577056

[48] S. Taghvaeeyan and R. Rajamani. 2014. Portable Roadside Sensors for Vehicle Counting, Classification, and Speed Measurement. In IEEE T-ITS 15, 1 (2014), 73–83.

[49] H. B. Tulay and C. E. Koksal. 2020. Increasing Situational Awareness in Vehicular Networks: Passive Traffic Sensing Based on Machine Learning. In Proceedings of 91st IEEE VTC. 1–7.

[50] Ildar Urazhigildiev, Rolf Ragnarsson, Pierre Riddervolst, Anders Rydberg, Eric Ojefors, Kjell Wallin, Per Enochsson, Magnus Ericson, and Gran Lofqvist. 2007. Vehicle Classification Based on the Radar Measurement of Height Profiles. In IEEE Transactions on Intelligent Transportation Systems 8, 2 (2007), 245–253. https://doi.org/10.1109/TITS.2006.879071

[51] C. Wang, A. Bochkovskiy, and H. Liao. 2023. YOLOv7: Trainable Bag-of-Features Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of 2023 IEEE/CVF CVPR. Los Alamitos, CA, USA, 7464–7475. https://doi.org/10.1109/CVPR52729.2023.00721

[52] M. Won, S. Sahu, and K. Park. 2019. DeepWiTraffic: Low Cost WiFi-Based Traffic Monitoring System Using Deep Learning. In Proceedings of IEEE 16th MASS. 476–484.

[53] M. Won, S. Zhang, and S. H. Son. 2017. WiTraffic: Low-Cost and Non-Intrusive Traffic Monitoring System Using WiFi. In Proceedings of 26th ICCCN. 1–9.

[54] Baosheng Zhang, Xiaoguang Ma, Hong-Jun Ma, and Chunbo Luo. 2024. DynPL-SVO: A Robust Stereo Visual Odometry for Dynamic Scenes. In IEEE Transactions on Instrumentation and Measurement 73 (2024), 1–10. https://doi.org/10.1109/TIM.2023.3348882

[55] Lei Zhao, Jiadai Wang, Jiajia Liu, and Nei Kato. 2019. Optimal Edge Resource Allocation in IoT-Based Smart Cities. In IEEE Network 33, 2 (2019), 30–35. https://doi.org/10.1109/MNET.2019.1800221

[56] Marco Zimmerling, Luca Mottola, and Silvia Santini. 2020. Synchronous Transmissions in Low-Power Wireless: A Survey of Communication Protocols and Network Services. In ACM Comput. Surv. 53, 6, Article 121 (dec 2020), 39 pages. https://doi.org/10.1145/3410139

Received 8 February 2023; revised 17 May 2024; accepted 7 June 2024