Machine Learning Use-Cases in C-ITS Applications

Norman Bereczki and Vilmos Simon

Abstract—**In recent years, the development of Cooperative Intelligent Transportation Systems (C-ITS) have witnessed significant growth thus improving the smart transportation concept. The ground of the new C-ITS applications are machine learning algorithms. The goal of this paper is to give a structured and comprehensive overview of machine learning use-cases in the field of C-ITS. It reviews recent novel studies and solutions on CITS applications that are based on machine learning algorithms. These works are organised based on their operational area, including self-inspection level, inter-vehicle level and infrastructure level. The primary objective of this paper is to demonstrate the potential of artificial intelligence in enhancing C-ITS applications.**

Index Terms—**C-ITS; ITS; V2X; smart city; machine learning; deep learning**

I. Introduction

Transportation has a tremendous impact on everyday life. In the last five years, the number of owned commercial vehicles in Europe has risen by almost 5%, reaching the number of 246 million, putting a strain on the transportation infrastructure, making it highly inefficient [1]. Emissions generated by aviation increased 6.8 times between 1980 and 2015, while the annual distance flown increased by more than 75 times [2]. The number of kilometers traveled on trains increased by 1.5 times between 2000 and 2016 [3]. These data confirm that the use of transport shows a rapidly increasing trend. This leads to overload, unexpected situations, traffic accidents, and makes people spend an extra 27 hours standing in traffic annually [4].

To overcome these issues, a possible solution is designing a new transportation architecture capable of handling the load, however most countries are already too crowded with infrastructure. Intelligent Transportation Systems (ITS) provide potential solutions to increase the safety of commuters and optimize traffic flow, thus reducing traffic congestion and air pollution and increasing the reliability of public transportation. If the listed problems are at least partially solved in a region that has an impact on the quality of life [5].

The development of ITS are supported by industrial and academic resources as well, leading to a significant momentum in the field and making it one of the most researched topics. In recent years, Artificial Intelligence (AI) and sensor technology have undergone great development, which has resulted in an increase in AI-based C-ITS applications.

AI is a powerful tool for processing extensive amounts of gathered data to gain insights into transportation systems. The first use of the terminology "AI" goes back to Dartmouth College in 1956 [6]. They held AI is the process that gives the ability for computers to learn by modelling the learning aspects of the human brain. AI-based applications are a popular and effective way for data-driven applications and used for prediction, anomaly detection, etc.

The goal of this paper is to provide a comprehensive study of novel AI-based applications used in C-ITS. Section III introduces the categorisation of C-ITS applications based on the level of cooperativeness and provides a summary about Machine Learning (ML) using C-ITS related examples. Section IV collects machine learning based applications used in C-ITS and organizes them based on their operational area. Then Section V presents research challenges and unsolved issues related to AI-based C-ITS applications. This review collects novel publications of the field and contributes to our understanding of the impact of machine learning in C-ITS. Although there are survey and review papers on machine learning use-cases in smart transportation, however our aim is to organise these applications based on their operational area, showing how AI contributes in every layer of smart transportation. This novel organisation can help the reader understand how each layer plays an important role in the smart city concept.

We tried to collect articles that were no older than 6 years and had relatively high number of citations. We preferred articles published in high-ranked journals and conferences in the filed of transportation, V2X, and computer science. We aimed at ensuring that the selected articles are recent, well-cited and represent the current state-of-the-art of this field. Using this approach we tried to provide an up-to-date survey of machine learning use-cases in C-ITS applications.

II. Machine Learning

In many scientific fields, such as self-driving cars, vast amounts of data are being generated. This wealth of data enables data-driven approaches that perform well with artificial intelligence-based algorithms.

In recent years, AI and V2X have been extensively studied. The number of AI related publications has increased by over 200% in the last 10 years. The number of filed patents has increased 30 times in the last 7 years, but only the 0.06% of the posted AI related jobs in the U.S. was related to autonomous driving in the last 2 years [7].

AI is often defined as the technique that gives the ability for computers to mimic the human thinking. Machine Learning (ML) is a subcategory of AI. The main essence of ML is not only to let machines use logic but also to give them the ability

The authors are with the Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Budapest, Hungary
(E-mail: {bnorman, svilmos}@hit.bme.hu)

to learn from experience, thereby improving themselves for a better performance. This is typically done by minimizing a loss function defined as a function of input parameters. Learning is a generic process of modifying an existing knowledge base to better adapt to a situation.

The term ”machine learning” covers a wide range of algorithms and has became a popular word in today’s computer science. Most data-driven problems require the ability to recognize complex and abstract connections in the data, which machine learning is highly successful at. This section provides a comprehensive summary of machine learning use-cases and highlights how AI can contribute to various problem categories.

A. Generic Machine Learning Workflow

There is a generic workflow for implementing machine learning models. This section provides a brief summary of these steps and shows examples of how they can be interpreted for C-ITS applications.

1) **Data collecting**: The success of a model highly depends on the amount and quality of the data it is trained on. Althnian et al. [8] and Prusa et al. [9] published a paper which leads to the conclusion that the accuracy of a machine learning model highly depends on the size of the training dataset. Thus, data is highly valuable for fields such as social advertising, recommendation systems, fraud detection.
   C-ITS application can rely on several data sources. Vehicles can share their state, such as speed and position. An RSU (Road Side Unit) can provide information about the infrastructure. Several data collecting units use computer vision, which is a field of AI, that can derive visual information.
2) **Feature selection**: Collected data may contain features that contribute little or none to the target variable one would like to predict / classify on. The goal is to find the important features that are highly relevant and remove the features that are not. This process improves the accuracy of the model, reduces computation time, and prevents the model from overfitting. In a comprehensive literature review of feature selection methods, Chandrashekar and Sahin present different algorithms that might be employed for this task [10].
   High accuracy predictions and proper incident detection, such as an accident caused traffic congestion, are not possible without identifying the correct features. For example, the breaking intensity of a vehicle changes radically when an accident occurs in front of it, but inner temperature is not affected by it. It is essential to build the model on the right features to achieve good performance.
3) **Initialising the algorithm**: Each and every machine learning algorithm solves different problems. At implementation, attention must be paid for the amount of data, data labeling, cleaning, gap-filling, and other preprocessing tasks. Chosen hyperparameters of an algorithm have a great impact on its performance, so optimisation plays a key role here. As Feurer et al. state in their paper about hyperparameter optimisation, it is a fundamental part of the accurate result, or as the Conclusion states: “The devil is in the details” [11].
   Goals of our applications can differ. The algorithm must be selected according to its goal, and the amount and quality of data gathered. A bus delay prediction system may be based on time-series analysis and prediction, but a vehicle identifier algorithm require completely different machine learning algorithms.
4) **Training of the algorithm**: The model needs to be trained on the specific training data from which it can learn. During the training process, the goal is to minimize the problem specific loss function iteratively to improve performance. Just a subset of the collected data is used during the training process. It is important to split the data into train-test-validation sets.
   A clustering algorithm can also be applied on several datasets. It can be utilised, for example, to detect dangerous drivers, but also to detect traffic congestion events.
5) **Evaluation of the model**: After the learning process, the model’s performance is evaluated on a test dataset that differs from the training set. Performance of the model is often measured by computing various performance metrics, such as accuracy (ratio of correct predictions to the number of all the predictions) , precision (ratio of correctly identified cases to the number of all the identified cases), recall (ratio of correctly identified cases to the number of actual positive cases), F1-score (harmonic mean of precision and recall), ROC (curve of true positive rate against fals positive rate at every classification thresholds), AUC (area under the ROC curve). Hossin et al. give a summary about commonly used metrics [12].
   Evaluation of newly adapted C-ITS applications is often enhanced by professionals who monitor the traffic and validate the results of the model compared to the real traffic characteristics.
6) **Implementing feedback from results**: The complex analysis of various accuracy metrics can provide information about the type of errors that the algorithm has made. (Whether it results, for example, high recall but low precision.) This information can tune the model or use dataset balancing metrics.
   Before a new application gets implemented in live transport system, it is preceded by a great amount of simulation to test every possible scenario. A new, adaptive traffic light controlling system, for example, might be dangerous to implement before they fully test it because it can cause serious consequences, such as accidents.

The target variable, or label is a discrete or continuous value that is assigned to an entity. Machine learning algorithms can be categorised based on the occurrence of the target variable. The following sections provide details, use-cases and common algorithms for the four main categories:

B. Supervised Learning

Supervised learning is a type of ML where a target variable is assigned to every entity in the dataset. The main goal is to approximate the value of the label based on other features. There are two main types of problems in supervised learning:

Regression is a statistical approach. Regression uses a function created from the combination of features in order to estimate the value of the continuous target variable. Regression can be used to model the relationship between different variables that affect the transportation. Regression based models can be used to predict the influence of different impacts on the transportation, such as weather conditions to traffic speed or travel time.

Classification is the process of arranging data points into pre-defined groups based on their features. The target variable is discrete and represents a group. Based on the number of classes, two categories are distinguished: OCC (One-class classification) which is used to predict whether the data is in one specified class and the Multiclass classification (MCC) which classifies the data into 3 or more classes. MCC is often solved with multiple applications of OCC. When labeled data is available, anomaly detection problems can be approached as an unbalanced classification problem. Commonly used algorithms include various types of decision trees, multi-layer perceptron, regression trees, logistic regression, support vector machine, etc. [13].

C. Unsupervised Learning

The primary difference between supervised and unsupervised learning is the absence of the target variable. In unsupervised learning, there is no explicitly given label on the data that can be used as the target of the prediction. In most cases, the goal is to find subsets with similar attributes and perform an action on them.

Clustering based algorithms find structure in the dataset, thus creating groups in the data containing similar data points. Hyperparameters of clustering algorithms include the number of clusters, similarity metric, etc. Compared to supervised learning problems, the miss of the target variable indicates that there are no optimal values for these hyperparameters. The result given by a clustering algorithm highly depends on these hyperparameters, but the best result is not easily defined because there is no target variable we can measure the accuracy on. Clustering algorithms can be hierarchical or partitional. Hierarchical clustering results clusters within clusters, while there is no hierarchical relationship between the clusters at partitional clustering. Common algorithms are k-Means, DBSCAN [14].

Anomaly Detection

The goal of anomaly detection is to find data points that do not conform to the dataset's expected behaviour. Anomaly (or as it is often called, outlier) detection is often highly specific to a given domain. Anomaly detection is approached in several ways: as classification problem, clustering problem, nearest-neighbour search, evaluation of different statistics or information theoretic and spectral metrics[15]. Anomaly detection can help to filter noise or identify malicious activities. Commonly used ML technologies are AutoEncoder Neural Networks, Isolation Forest, etc [16].

Dimension reduction is a process that reduces data from a high dimensional space to a lower dimensional latent space using mathematical operations (e.g., different projections) with minimal loss of information. It can be beneficial in several aspects such as reducing the size of the data, improving performance, allowing high-dimensional data to be visualized, and reducing overfitting.

PCA is one of the most popular early statistical technique that is used for dimension reduction. It projects the data into a smaller subspace with the methods of linear algebra. Other popular algorithms are t-SNE, UMAP [17].

D. Deep Learning

Deep learning is a machine learning technique learning representation by examples. It is a technique that is used to try and learn like human beings by modelling the architecture of brain cells.

Neural networks are computational models inspired by the structure and function of the biological neurons in the brain. They consist of interconnected nodes (neurons) that can process and transmit information, allowing it to learn patterns and relationships in data. Using neural networks instead of classical machine learning methods can lead to deeper recognition capability in higher dimensions. These networks mimic the functioning of biological neurons in the brain. A neuron receives an input signal, performs computation on it, then passes the output signal to the next layer. The output of the neuron is the weighted summary of its inputs, passed through an activation function. The main goal of using activation function is to introduce non-linearity into the output, thus allowing the neural network to learn complex, non-linear patterns as well. There are several activation functions, each has its own advantage and the choice depends on the problem's requirements. Commonly used activation functions are [18]:

- *Sigmoid:* The goal of sigmoid function is to map the input into the interval between 0 and 1.
- *ReLU:* Rectified Linear Unit function sets values that are less than 0 to 0 and leaves others unchanged.
- *LeakyReLU:* Similar to ReLU, but for negative inputs, instead of zero it introduces a small slope.
- *ELU:* Similar to LeakyReLU, but it smoothens the curve for negative inputs, thereby improves learning speed and prevents overfitting.
- *Hyperbolic Tangent:* It maps the input to a range between -1 and 1.
- *Swish:* Swish is a relatively new activation function. It is similar to ReLU, but it has a learning parameter and in some cases it can outperform ReLU.

E. Other categories

1) Semi-Supervised Learning: Labeling data is often difficult, and labels are not always available for the dataset. Semi-supervised learning combines both supervised and unsupervised learning by using large amounts of unlabeled data, together with a significantly smaller amount of labeled data.

This approach is often used when trying to predict cases for which there are no examples in the training set, thus is a good tool for anomaly detection, malicious activity detection and disaster forecasting.

2) Reinforcement Learning: The goal of reinforcement learning is to learn a specific behavior by taking actions and receiving feedback in the form of positive or negative reinforcement. The model learns complex action sequences by trying different combinations and improving itself based on the received feedback. One of the most common type of reinforcement learning is Q-learning, where a table or a function is used to determine the optimal action in a given state.

III. Cooperative Intelligent Transportation Systems (C-ITS)

A big breakthrough in the development of ITS has occurred when new communication capabilities of vehicles and the infrastructure became possible. V2X (Vehicle-to-X or Vehicle-to-everything) is a generic term referring to the usage of communication capabilities of vehicles and transportation infrastructure elements, based on standardized architectures and protocols. V2X communication technology enables vehicles, roads, and pedestrians to link up and exchange information about their environment and state. This communication results in improved transportation efficiency and safety. Through V2X, vehicles can share their measures with other participants, facilitating new cooperative behaviors. V2X based applications can improve various aspects of transportation. Its main goal is to make transportation more efficient and safer. The elements of transportation can have static (number of lanes, structure of intersections, speed-limit, etc.) and dynamic information (position of vehicles, traffic load, etc.). In C-ITS, information exchange through V2X enables static and dynamic information to be shared among ITS elements. The sharing of information from other ITS also allows for cooperative behavior to improve ITS to C-ITS. A common approach is to use cloud-based services that enables vehicles to exchange information. In their work, Chen et al. [19] propose a cloud-based traffic control system. They provide a platform to support cross-sector information sharing. Their system can provide vehicle's weather data, can alert them about traffic congestions and emergency vehicles. Vieria et al. [20] presents a roadside and cloud architecture. Their system has been deployed and is used on different motorways in Portugal. The system also offers traffic visualization and event reporting. The system is not dependent of the underlying communication technologies, thereby it is easy to implement.

1) Radio Access Technologies: C-ITS applications have strong QoS requirements. With the use of V2X, the development of reliable, low-latency radio access technologies (RATs) also became important. Nowadays two RAT competes each other: **802.11p** for dedicated short-range communication (DSCR) and **New Radio 5G V2X (NR V2X)** for cellular V2X (C-V2X). 802.11p is an IEEE 802.11 standard, also known as Wireless Access in Vehicular Environments. It was introduced specifically for V2X communication. It operates in 5.9 GHz and provides low-latency, reliable communication. The Next-Generation Wireless Access for Vehicular Environments (802.11bd) protocol is a more recent standard that builds on 802.11p. It is more flexible, has improved performance and uses both 5.9 GHz and 60 GhZ bands. 5G is the 5th generation of wireless communication technology. 5G can transmit large amount of data quickly, thus enabling vehicle to share a vast amount of information with low latency. 5G network is also highly reliable and has built-in security features that are also important. Both DSCR and C-V2X are promising technologies that can provide reliable, low end-to-end latency communication. C-V2X has the potential to bring additional benefits, such as better coverage for V2X, reducing infrastructure deployment costs and increased deployment flexibility. Nevertheless, 802.11p technology is ready to be used, and several applications have been developed using it, as this standard has been on the market for longer. Europe and the United States, the 802.11p standard is more widely used, while China's V2X developments are based on the NR-V2X RAT [21] [22].

DOI: 10.36244/ICJ.2022.4.6

2) GeoNetworking protocol [23]: GeoNetworking is a protocol in the network layer which makes location-based message forwarding possible. It transports messages over IEEE 802.11p in GeoNetworking packets. It provides services for top layer protocols. There are 3 routing ways that are shown in Fig. 1 below. The GeoUnicast messages are sent from a node to one receiver. The GeoBroadcast message can be concentrated in a physical area (rectangle, square) and the topologically scoped broadcast messages are sent in the environment of the sender node. It allows messages to be received only by nodes for which the information is useful.

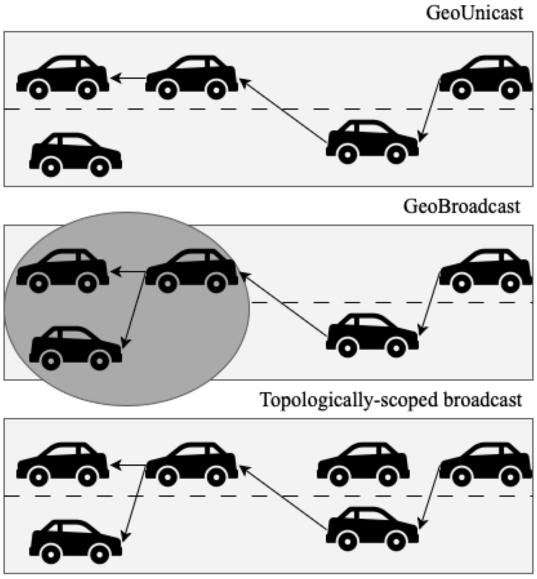

Fig. 1. Package forwarding in GeoNetworking

3) Facility layer protocols: Entities on or near the road can communicate in a wide range of situations. A roadside unit, a traffic light, or a car that travels 130 km/h has to communicate different information to other participants thus several protocols are utilized use in V2X. Communication using unified protocols is important, so standardization is a key aspect of successful communication. This section highlights a few of the most common protocols that C-ITS applications use.

- **CAM** - Cooperative Awareness Message [24]:
  Contains state information (velocity, vehicle attributes, etc.) about the represented entity. Messages sent with CAM are containerized. They are sent in 1-10Hz periods, providing up-to-date information. (The proper sending frequency plays an important role in efficiently using the communication channel. A vehicle travelling at 130 km/h will send a message every 3 meters, if the sending frequency is 10 Hz. In a traffic jam, the vehicle may not even move 1 meter in a second and using 10 Hz sending period in this scenario would be unnecessary.)
- **DENM** - Decentralized Environmental Notification Message [25]:
  Used for sharing incident information (accidents, road construction, etc.) on the road. It can provide information about incidents with constant or static positions. DENM messages can be sent from a vehicle by OBUs (On Board Unit) and from the infrastructure by RSUs (Road Side Unit). DENM contains information about the incident, such as its starting time, its estimated ending time, and describes the triggering event.
- **CPM** - Cooperative Perception Message [26]:
  Allows vehicles to share observations of their local environment based on their sensor data. The messages are sent with 1-10Hz frequency, similar to CAM. CPM often carries the data used for self-driving systems. CPM is a novel development.
- **MAPEM, SPaTEM** - MAP Extended Message [27]:
  Messages from the infrastructure to the ITSs. The MAPEM and SPaTEM describe the lanes and the sequences at crossroads.
- **SREM** - Signal Request Extended Message [28]:
  The vehicle can request modifications in the traffic lights sequence at a crossroad. The request is prioritized on a scale from 1 to 14. It is especially useful for ambulance cars or fire trucks so they can request priority in the traffic.
- **VAM** - Vulnerable Road User Awareness Message [29]:
  Used for describing the position of participants that are not vehicles, such as pedestrians or cyclists. The vehicles can be alarmed about them, thus avoiding unexpected situations and preventing accidents.

A. Dynamic messages in C-ITS

Transportation can have static and dynamic properties. Static information is, for example, the degree of a curve or a position of a crossroad. These properties are permanent and do not change in time. The constant change in the traffic situations generates dynamic data as well. The position of a vehicle, the occupancy of the infrastructure, average speed, velocity, etc. are always changing. Based on these kinds of dynamically changing data, we can classify the evolution of V2X applications into the following categories:

*1) Day 1 - Share sensed information: **Driving Awareness**:* Shared information is valid only in the local environment of the sender. Vehicles can be aware of their environment, and the most important parameters of those travelling nearby. The vehicle receives and processes messages coming from other vehicles in its communication range. (The size of a vehicle's range is changing based on conditions such as travelling speed, infrastructural grid.). Day 1 applications are simple and do not interfere directly with driving. A good example is Cooperative Adaptive Cruise Control (C-ACC). The vehicle accelerates and slows down based on the received messages. (It's important to understand that the basic ACC is performed by the vehicle's local sensors, while the C-ACC relies on data provided by other vehicles.) RWW (Road Works Warning) can inform the driver about ongoing road maintenance. Collision avoidance works by the alert coming from another vehicle.

Day 1 applications are based on the communication capabilities of the participants. It relies on them being capable of sharing their status, thus alerting one another.

2) Day 1.5 - Share sensed information extended to multimodal applications: This category includes multi-modal applications that are based on shared sensing. These applications rely on shared data and improve mostly the utilization of the infrastructure. These applications can be divided into 6 main groups:

- Parking: Provide information about free parking spots, thus minimizing the unnecessary traffic and environmental load.
- Intelligent Routing: Information providing services which can help electric cars to find the optimal charger regarding to the traffic and the destination. These services can implement adaptive routing.
- Transporting: Help manage and support the transport vehicles in cities.
- Safety: Improves the safety of pedestrians and cyclists. Can function as a recommendation system for entities with limited visibility.
- Collision: Can reduce the possibility of a collision in critical situations, such as overtaking or in a turn with a bad viewing angle. For example, vehicles are able to assist in the decision making whether or not to perform an overtake.
- Direction Alert: Can alert the driver and its environment if a vehicle is going against the good driving direction.

*3) Day 2 - Sharing sensor measurements: **Cooperative Sensing**:* These applications extend the use-cases of Day 1 applications. The main difference compared to Day 1 applications is besides sharing measurements and information about the entity, they also share sensor measurements about their environment. These applications share their sensed environment, thus making other entities able to create a more comprehensive model of their own environment.

The CPM (Collective Perception Message) application proposes a standardized method to share the sensed objects in the vehicle's environment and the free space between these objects. The gathered data can be used as an input for more complicated, sensor-fusion based applications, further improving V2X applications.

4) Day 3, 3+ - Cooperative and synchronized cooperative driving: ***Coordinated maneuvering****:* Maneuvering Coordination Service (MCS) is a Day 3 service, and it lets vehicles share their intentions with others [30]. These applications implement a higher level of self-driving. A fully automated transportation can reduce the number of accidents to 0 and can reach the optimal use of infrastructure both in the aspect of travelling time and pollution.

B. Hierarchical layers of C-ITS Applications

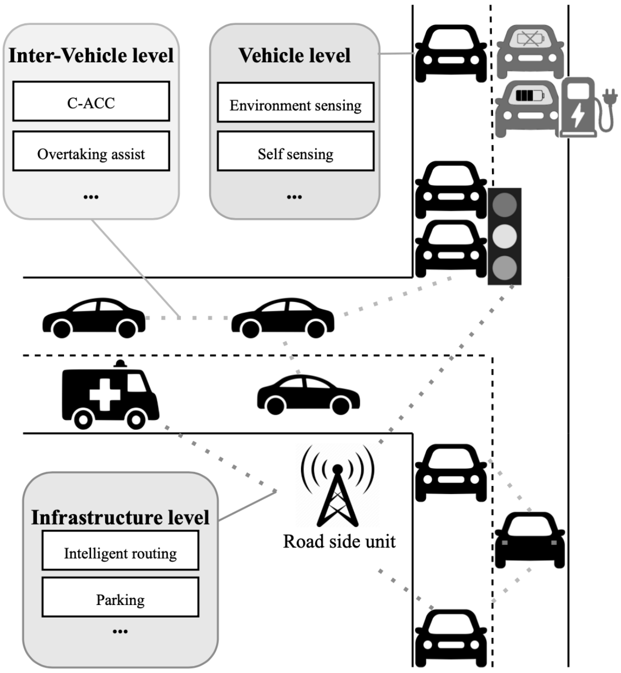

Fig. 2. A model of a smart transportation infrastructure

Fig. 2 shows a model of the smart transportation concept. This paper distinguishes 3 levels of C-ITS applications that are listed in this section. Following sections provide detailed information on:

- **Vehicle level** applications operate on the level of vehicle. They can sense self-states, such as faults, driver tiredness, and the environment of the vehicle, such as object recognition, collision detection.
- **Inter-Vehicle level** applications operate in a group of close vehicles. These applications operate with shared data and can not be realized by a single vehicle.
- **Infrastructure level** applications impact the whole transportation infrastructure.

In order to ensure the success of smart transportation, it is crucial for the different layers to heavily rely on each other. The flowchart of Fig. 3 is an example of a C-ITS application that uses all abstraction layers demonstrated in this paper.

When a vehicle recognizes an ambulance vehicle, but it is unable directly notify every other vehicle, or an infrastructural element (it may be out of the range of any RSU). Therefore, it sends an alert to the vehicles that are on the same road. If one of these vehicles reaches a Road Side Unit (RSU), it can notify the TCC (Traffic Control Center). The C ITS-S (Central Intelligent Transportation System Station) then gives priority to the ambulance by interrupting the traffic light sequence, making the way free for it. Then the infrastructure notifies vehicles in the area so they can prepare for the arriving ambulance car. This application is indisputably important in terms of healthcare services. According to the publication of Russel G., there are on average 10 minutes of delay caused by traffic congestion in Alabama for emergency medical services. Even this small amount of time can play a critical role in the hospitalisation of patients [31] .

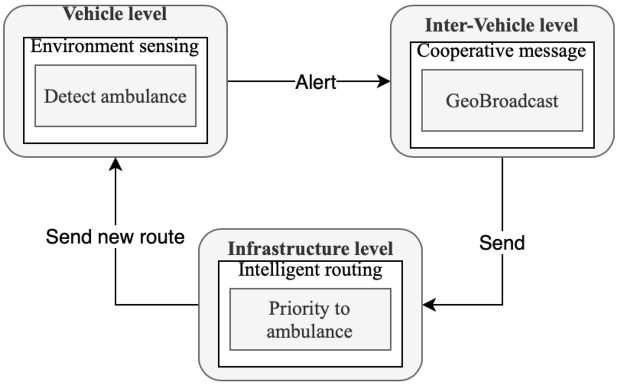

Fig. 3. A simple C-ITS use-case in the infrastructure, presented in Fig. 2.

The mentioned example demonstrates how different layers collaborate to implement a complex smart transportation based service. The infrastructural layer is responsible for determining the new routes for the vehicles, but detecting the ambulance vehicle is out of its responsibilities. (The detection of the priority vehicle is based on the vehicle level in this example Fig. 3. In other use-cases, the ambulance may notify the infrastructure.) It is important to notice that every layer must satisfy its own responsibility in order to implement this service. Applications in each layer have their own role and responsibility and they communicate with applications on other layers. This highlights the importance of our categorization. This architectural approach is frequently used in complex systems, such as the OSI model prevalent in computer networks.

IV. C-ITS APPLICATIONS

Section II briefly summarised how machine learning is used to solve a wide variety of problems. The following sections review novel C-ITS applications that are based on machine learning for every hierarchical layer. The listed articles that proposes a new solution are listed in Table I.

A. Vehicle level

Vehicle level applications operate within the vehicle. The primary object of these is to measure the condition of ITS and the current state of its environment. ITS measure and gather information, which they then process and share the results with other participants. This layer is fundamental in every category of C-ITS applications that are demonstrated in Section III. However applications in this layer do not directly play role in the cooperativeness of C-ITS applications, they often rely on the results of vehicle level applications. This paper classifies vehicle-level applications into 3 categories: machine-level sensing, driver-level sensing, and environment-level sensing.

1) Machine-level: The recent development of sensor technology has enabled to collect information about the mechanical status of the vehicle. The goal is to get a clear view of the vehicle's condition. These applications can contribute to detect vehicle fault. Based on these applications the infrastructure can be notified if a vehicle has improper or dangerous conditions. Other drivers can be alerted about a vehicle with faulty brake or one that has broken down and stopped in the middle of the road. By sharing information about the failure, other vehicles can be prepared for the unexpected event, thus preventing unwanted situations and increasing road safety.

The issue of **Maintenance prediction** has received considerable attention.

Thanks to the recent sensorisation trend in the car industry, it has become possible to measure the state of several subsystems in a car. The primary object of this is to notify the owner when maintenance is required. There are determined maintenance periods that can be scheduled based on given intervals, such as the age of the part or the mileage. Condition-based maintenance needs to be conducted based on the condition of the given part of the vehicle. While most parts, such as the timing belt and the clutch, have planned, given maintenance periods, unexpected faults can happen and result in expensive maintenance. To overcome this situation, faults can be recognized in their early stages before they cause fatal problems. This way the owner of the vehicle can be alerted to repair the vehicle before the fault happens. Using predictive maintenance notifications, the owner can prevent bigger faults and can save a considerable amounts of money on costly repairs. Maintenance prediction helps to optimise expenses, increase the reliability of the vehicle and minimise downtime. These optimizations are very important in the transportation sector. Implementing predictive maintenance reduces the number of broken cars, thus avoiding possible traffic congestion caused by unexpectedly stopped vehicles. It can also improve safety by identifying potential problems before they become critical and prevent accidents. Predictive maintenance can help reduce fuel consumption as well. If a vehicle's engine or transmission system is not operating at peak performance due to a faulty component, it may require more fuel to travel the distance. Burning more fuel releases more gases causing pollution, leads to higher travelling costs and inhaling more gas has negative health effect. Predictive maintenance has been the subject of many survey papers [32], [33], [34], [35].

Braking system is one of the most important parts of a vehicle, its proper functioning is crucial for the safety, and therefore, braking system fault detection has been studied extensively. A vehicle with a faulty braking system can brake more slowly or not at all. This leads to dangerous situations. Other vehicles can be real-time alerted about the arrival of a dangerous vehicle with brake problems. This work observes braking systems based on vibration monitoring, thermal imaging and oil particle analysis. This allows prepared drivers to handle the situation with appropriate caution, which increases road safety.

An extensive summary of machine learning based break fault diagnosis has been carried out by Manghai [36].

Ryan M. et al. [37] showed a digital twin simulation (DTS) for generating training data for brake fault detection systems. DTS is a virtual replica, a simulation of a physical system and allows researchers for testing and optimizing it.

Joshuva et al. [38] have introduced a classification-based system to identify brake faults. In their work data was directly collected from a Ford EcoBoost model's brake vibration measured by an accelerometer paired with an integrated charging amplifier and a signal conditioning module. They compared C4.5 and Logistic Model Tree (LMT) algorithms with 7 selected features. C4.5 is a decision tree based classification algorithm [39]. LMT builds a tree of logistic regression models to predict the class-probability for a new data point. It produced 92.75% accuracy at early fault detection [38] . In his further publication he proposed a random tree classifier that can produce 98.5% accuracy on the same dataset measured on the Ford [40].

One of the most accurate algorithm is developed by Zhang et al. [41]. Their fault detection system performed 100% accuracy and it is based on a modified convolutional neural network (CNN) and a support vector machine (SVM). CNNs are widely used in AI based applications, they are deep learning models that can learn spatial hierarchies of features from the high dimensional input data using convolutional layers. SVM is a commonly used ML algorithm to classify data in high-dimensional feature spaces. It finds the optimal hyperplane to separate the different data classes. Their study uses the disc brake simulation bench of China University of Mining and Technology as data source. This simulation contains the raw signal of friction factor and friction surface temperature of several types of breaking discs. Their work uses both statically monitored object measurements, such as the friction of the braking system parts, and the temperature of braking surfaces and uses dynamically extracted features as well.

Monitoring braking systems is not only useful for fault detection. Braking intensity can highly influence the success of a race vehicle. Federico et al. [42] studied the optimization of carbon brakes in MotoGP. The study revealed that braking efficiency is influenced by three primary factors: disc temperature, the relative speed between the disc and pads, and the amount of applied normal force. To identify the relationship between the braking efficiency and the parameters, SVM, decision trees and neural network models were used. The analysis has been proceeded on simulation generated data and concluded that the increasing temperature of a braking system can reduce intensively the braking power [42].

Suspension systems play an important role in the dynamics and handling of a vehicle. They are designed to reduce the vibration of the vehicle caused by the road. Fault in the suspension system can reduce the vehicle's controllability. A vehicle that is difficult to control is also a danger to other road users. By detecting the fault, other drivers on the road can be alerted to the arrival of a dangerous vehicle. The fault of the suspension system can be detected by the use of clustering based algorithms [43]. This data-driven solution detects faults without explicitly outpointed fault features using unsupervised clustering to detect anomalous clusters based on the spring coefficient of the observed suspension systems [43]. The proposed method uses dimension reduction to determine whether a new fault occurred and the Fisher discriminant analysis to classify the data points into fault categories [43].

2) Driver-level: The behaviour and driving style of drivers play crucial role in transportation safety. A careless, dangerous driver can pose significant risk to others on the road. The goal of driver-level applications is to monitor the actions and behavior of the driver. Based on these, C-ITS services can alert drivers real-time when a dangerous driver occurs and prepare them to handle the dangerous driver with increased attention.

Sleepy driving, also known as driving while drowsy, is a highly dangerous but common condition among drivers [44]. According to Kalsi [45], sleepiness is the most significant risk factor for drivers. When drivers are tired, their reaction times can increase by more than two-fold. For instance, at a speed of 50km/h, if the aware reaction time is one second between sensing and acting, drowsiness can add an extra 14 meters before a driver can react. Moreover, Vogelpohl [46] claims that drivers in autonomous vehicles tend to lose focus faster and and take longer to regain their concentration. Alerting the driver to take a break reduces the risk coming from unfocused driving. In recent years, most research on detecting sleepiness has emphasized the use of machine learning techniques. Drowsiness detection methodologies use sensors to monitor a driver's cognitive state to detect signs of fatigue and drowsiness. It can also be integrated with other C-ITS systems, such as traffic management and emergency services, to provide real-time alerts to other road users and authorities about the presence of an unaware driver on the road. This can help prevent accidents and improve road safety for all users.

One way to detect drowsy driving is by analyzing the facial expressions of drivers. In Nietjet's work [47] a Raspberry Pi based hardware detects the tiredness of facial expressions. The system considers the eye closure ratio as the main indicator of drowsiness and detects it using a Haar cascade classifier. It is common used in face detection applications. The main essence of Haar cascade classifier is to detect and identify parts of an image that are relevant to the face (such as edges, lines).

The usage of Haar cascade classifier occurs also in the work of Byrnes and Sturton [48]. The used data has been directly provided by 23 recruited participants. The work shows that there is a change in the eye behaviour when the observed person feels drowsy (eye closure is higher, blinking becomes more frequent).

Jabbar et al. [49] developed a lightweight system that can detect drowsiness by analyzing the face of drivers and can reach up to 83% accuracy in finding drowsiness. This method also relies on the state of the driver's eye, identified by the Viola-Jones algorithm. Then a CNN extracts the features and a SoftMax layer classifies the samples into binary (drowsy/awake) categories. The Viola-Jones algorithm is a widely used object detection method. Its basic principle involves a cascade of simple classifiers, which are combined to form a more complex classifier. Each simple classifier is constructed using a Haar-like feature, which is a rectangular window that measures the difference in intensity between adjacent sub-regions of an image. The training data used by Jabbar et al. is from the National Tsing Hua University (NTHU) Driver Drowsiness Detection Dataset [50] which contains 4157 images from 4 different persons with open and closed eyes.

Transfer learning (VGG-16, VGG-19) and fully designed neural network based approach has been introduced by Hashemi et al. [50]. Transfer learning is a technique where a pre-trained model is used as a starting point for the domain specific training. This allows the model to rely on pre-existing knowledge letting it learn more quickly and effectively. VGG-16 and VGG-19 are two deep convolutional neural network models. They are widely used in computer vision tasks because they perform well on image classification and object detection tasks [51]. Their work uses the ZJU Eyeblink Database and with another 4 models added by the author and results in a 98.15% accuracy.

Guo and Markoni [52] proposed a concept that detects drowsiness real-time based on CNN and LSTM. With the use of LSTM the proposed work uses not just a picture frame at a time to detect drowsiness, but frame sequences, thus analyzing the change of the face. LSTM (Long Short-Term Memory) is a recurrent neural network that can model sequential data. It uses memory cell and captures long-term dependencies in the data. The system has been tested on the public drowsy driver dataset from ACCV 2016 competition. It contains images of people sitting in front of a simulated steering wheel and acting like they are driving. The proposed Time Skip Combination Long-Short Term Memory has outperformed most models with a 84.85% accuracy. Other methodologies are summarised here: [53]. **Driver identification** and profiling has a great amount of literature. The driver can be identified in 2 minutes of urban area driving with a 85% success probability based on their pedal handling [54]. Yang et al. [55] proposed a deep learning-based solution that can identify drivers in 10 seconds with 83% accuracy by observing several patterns of measures in the car, such as the following distance, the usage of the turn indicator, acceleration, etc. In their work they used a driving pool containing 51 drivers. Furthermore a single turn can identify a driver from 64 different drivers with more than a 50% accuracy [56]. These publications support the fact that our driving style is unique, and drivers can be identified, but the scalability of these works has not been investigated.

Once a driver is identified a user profile can be assigned to them. This profile carries information about their driving characteristics. Recently there has been an increasing interest in **user behaviour characteristic identification** [57], [58]. It can enable real-time monitoring of driver behavior and provide warnings to the driver and other road users in case of dangerous situations. It can also allow for more accurate risk assessment, which can lead to fairer and more personalized insurance pricing.

The goal of a well cited article written by Shahverdy et al. [59] is to classify drivers into 5 driving style category (safe, drunk, drowsy, inattentive, aggressive) based on driving features, such as acceleration, gravity, throttle, speed, and used revolutions per minute. The problem is addressed in this work as an image classification task. A 2D CNN performs the classification based on visualisation, which is constructed by the help of the measured driving signals. The data was gathered by observing 3 drivers who imitate the above listed different driving styles.

Ferreira et al. [60] shows that the data for a user behaviour monitoring system can be collected from smartphone sensors. In this work the data is collected from 4 different vehicles using smartphones in the cars. The goal of this study is to identify maneuvers such as aggressive lane changing, turning, braking, accelerating and normal events. The gyroscope and the accelerometer sensors of the phone provided the data. They have compared SVM, RF, MLP, and BN models. Bayesian Network (BN) model is a probabilistic model that represents a set of variables and their dependencies using a directed acyclic graph which is used for decision making and prediction. The best performing model is Random Forest with 80% accuracy.

Lowering insurance fees for drivers who comply with traffic rules can serve as a good motivation to encourage normal driving. This approach is beneficial for both drivers and insurance companies since aggressive driving can cause accidents and create additional expenses.

AutoCoach is an intelligent agent developed by Marafie et al. [61] that implements a pervasive system to improve driver's behaviour. The system classifies drivers based on their driving behaviour and calculates a risk score based on the actions they have performed. A Support Vector Machine is used in this work to classify actions into 3 risk category. The training dataset is collected from a Toyota and BMW car. If the risk score crosses a defined threshold, the algorithm classifies the driver's behaviour as a bad one. The agent provides feedback on the current and the historical driving style.

There is extensive research on Usage Based Insurance (UBI) services as well. Yin and Chen have developed a framework that calculates the driving risk [62]. A common approach to implement UBI is to identify the driver behaviour characteristic and based on that provide an insurance pricing model [63], [64].

3) Environment-level: Environment-level applications measure the current state of the environment through sensors and use the obtained data to gain insights. ITS utilize this data to provide details about their environment to others. C-ITS applications share the information extracted from the raw data measured by the sensors in each system, rather than the data itself. This section aims to identify machine learning use-cases related to environment sensing.

Vehicle detection is an important part of environment sensing. In his study, Abbas [65] provides a comprehensive review of vehicle detection techniques. Vehicle detection is often the base of overtaking assistance, congestion prediction C-ITS applications.

Vehicle detection from Unmanned Aerial Vehicle (UAV) is studied widely due to its potential applications, such as traffic management, vehicle tracking, etc. The detection must be done precisely and fast. A detailed review from Bouguettaya et al. [66] shows how AutoEncoders, Generative Adversarial Networks, Recurrent Neural Networks and Convolutional Neural Networks are used for vehicle detection performed by UAVs.

The license plate is a unique identifier of a vehicle. Many solutions have been proposed for recognizing license plates. Fast and precise vehicle identification makes it possible to measure accurate travel time on public roads. This can contribute to several applications, such as traffic congestion detection or dangerous vehicle tracking.

In his literature review Tang [67] shows, how novel works perform at License Plate Recognition.

Motorcycle is a quick and compact way of travelling in cities with high traffic load. However, wearing proper protection is indispensable. Dasgupta et al. [68] introduced a CNN-based algorithm that first identifies a motorcycle, then crops the head of the driver (and passenger) and detects **helmet usage** with 96.3% accuracy. ITS can detect motorcyclists driving irresponsibly without helmets and alert the relevant authorities. This will increase the regulation of the wearing of protective equipment, resulting in safer driving.

Forward Collision Warning (FCW) is an important and valuable application that has received a lot of attention in research. According to statistics, the use of FCW reduces front-to-rear crash rates by 27% and front-to-rear injury crash rates by 20% [69]. FCW was introduced first in 2000 by Mercedes-Benz, but today, almost every car manufacturer offers it as an optional extra.

Autonomous Emergency Braking Systems (AEB) are triggered by FCW. When the vehicle detects the possibility of a collision, it executes an emergency braking maneuver to prevent the crash. In the U.S. the front-to-rear-crash rate has been reduced by 50% and front-to-rear injury crash rate by 56% thanks to the usage of FCW and AEB % [69]. Tawfeek and El-Basyouny [70] developed a novel FCW method and compared it with 6 existing FCW algorithms. This comparison showed that the proposed work can overperform every existing algorithm on every speed level. In their work, they defined a custom warning distance function and selected the important parameters to use in the function using linear regression. These warnings can be propagated on the road and can prepare the following vehicles for sudden braking.

Different lanes allow different maneuvers. A dashed lines allows vehicles to perform overtaking and lane changing, while a continuous white line prohibits such actions. FCW must consider the lane type and recognize incoming and oncoming traffic. Overtaking assisting C-ITS applications also need to be able to recognise the traffic lanes in order to support or reject overtaking. Song et al. [71] approach this problem using stereo vision processing. Stereo vision processing extracts information from 3D visual data. Using 3D data provides added value to the system, as it makes it possible to extract relative distance between the cameras and the object, as well as to correct the perspective distortion. They present a CNN based FCW system that classifies current and adjacent lanes with 100% accuracy [71].

The study by Tang et al. [72] is one of the first publications to review lane detection systems.

They claim that camera or lidar sensor-based lane detection can achieve up to 96% accuracy. This result was achieved by Zou et al. [73]. They combined a CNN with an LSTM to detect lanes in a continuous driving scenario. They evaluated the developed model on TuSimple dataset that contains almost 4000 image sequences of driving.

Traffic signs are the most common form of regulation in traffic. They indicate what actions are prohibited, allowed, or limited to drivers. However, distracted drivers may overlook traffic signs, leading to dangerous situations. If a vehicle can recognize traffic signs, it can continuously display them to the driver or, in autonomous cases, the vehicle can follow the rules indicated by traffic signs. Vehicles can also share recognized traffic signs to inform, help or alert others.

The first review on **Traffic Sign Recognition** (TSR) was published by De la Escalera et al. in 2003 [74]. It is noticeable that back in that time they claimed the number of possible solutions was highly limited. They investigated a NN based solution that achieved 80% accuracy in average on several signs. The significant improvement in neural network technologies radically increased the performance of TSR. Novel methodologies can achieve performances up to 95% and can reach even 100%. This shows how the spread of machine learning contributes to the efficiency of ITS applications.

Stallkamp et al. [75] surveyed a human focus group to classify traffic signs and concluded that the average human performance was 98.8% accurate. This is close to perfect, but NN-based solutions can clearly outperform human capabilities.

Rajendran et al. [76] proposed a Region Based CNN (R-CNN) method that can reach up to 99.7% accuracy on the German Traffic Sign Detection Benchmark (GTSRB) dataset. It contains 43 classes of traffic signs on more than 50.000 images [77]. R-CNNs are a subclass of CNNs. They localize and classify images by dividing them into regions and extract features from them separately.

Li and Wang [78] proposed another R-CNN solution using YOLOv3 that reaches comparable (99.6%) performance on the same dataset. YOLOv3 is a pre-trained CNN architecture, that is often used for transfer learning. This proves that CNN is a very effective and successful architecture for TSR.

B. Inter-Vehicle level

This Section presents how machine learning can contribute to applications that are used inter-vehicle. These applications rely on V2V communications, and the base of them is the capability of information exchanging. The cooperativeness of applications from Section IV-A comes from sharing the measured information. Solutions provided by this section are based on the communication, the sent information and the formed networks among vehicles. The goal of these applications is to implement inter-vehicle communication and to facilitate warnings and recommendations to drivers based on collective information about a network of vehicles.

VANET is an important subcategory of Mobile Ad Hoc Networks (MANET) and is commonly used in the network layer of V2V. It provides ad hoc network solution to vehicles for ubiquitous connectivity on the road with OBUs and RSUs. VANET is a practical solution to implement V2V communication because it can differ and really flexible in numerous aspects: size, number of vehicles, speed, etc. The characteristic of VANET is highly dynamic in terms of their topology, because it has to implement high mobility, dynamic topology, frequent connection and disconnections [79]. It is an useful solution, however it suffers from many security threats: information integrity, modulation problems, traffic overload, etc. Machine learning based solutions are to provide solution to these. The following sections present machine learning use-cases in V2V communication and in VANET environments. The content of this section is summarized in Table II.

A comprehensive and detailed summary of machine learning use-cases in vehicular networks can be found in the survey conducted by Tang et al. [80].

1) Network security level: Ensuring the security of V2V and VANETs is fundamental. Although vehicle networks are protected by several authentication and encryption measures, fraudulent activities can still occur due to vulnerabilities in the system. To detect and prevent such attacks, machine learning methodologies are implemented to detect abnormalities and predict hacker attacks. Ensuring the continuous security of VANETs is crucial because applications rely on the data transmitted through them. Using corrupted data leads to misbehaving, causing an inefficient, moreover, dangerous transportation. VANET attacks can occur in different aspects of the network [81]:

- Availability: The network is functioning at any time and always available for nodes.
- Authenticity and identification: Network stations must be identified before using it.
- Confidentiality: The transmitted data is not readable for everyone because it can contain confidential information.
- Integrity and data trust: The transmitted data has not been changed after sending it.
- Accountability: Change of transmitted data must be transparent and accountable.

To develop useful, efficient and safe V2V applications, it is fundamental to reduce the risk coming from vulnerability. This section presents novel solutions that can increase the security of V2V. **Availability** corruption of VANETs makes the network unavailable. This is often accomplished through Distributed Denial of Service (DDoS) attack against the network. The goal of DoS is to overload the network's resources, causing delays or complete interruptions in the network's behavior. In their study, Alrehan and Alhaidari [82] provide a comprehensive overview of how machine learning can prevent DDoS attacks. Most of the surveyed works in their article focus on detecting outlying behavior in VANETs using classification algorithms such as SVM.

Kadam and Krovi [83] have developed a system based on a hybrid combination of k-NN and SVM. They evaluated their system on the Kaggle DDoS dataset and achieved up to 93% accuracy. This contains attributes of packets, such as protocol, source and destination IP and port, etc.

Polat et al. [84] use a more complex, deep learning based model to solve this problem on software-defined network based VANETs. They introduced a sparse AutoEncoder and

TABLE I
SUMMARY OF VEHICLE-LEVEL APPLICATIONS

Author	Article	Year	Application	Used ML Algorithm
Manghai et al.	[36]	2017	Brake fault detection	SVM
Joshuva et al.	[38]	2020	Brake fault detection	C4.5, LMT
Joshuva et al.	[40]	2021	Brake fault detection	LMT
Zhang et al.	[41]	2021	Brake fault detection	SVM, CNN
Federico et al.	[42]	2021	MotoGP braking torque estimation	SVM, Decision Tree, NN
Wang	[43]	2014	Suspension fault detection	Discriminant analysis
Nietjet	[47]	2018	Driver tiredness detection	Haar cascade classifier
Byrnes and Sturton	[48]	2018	Driver tiredness detection	Haar cascade classifier
Jabbar et al.	[49]	2020	Driver tiredness detection	Viola-Jones + CNN
Hashemi et al.	[50]	2020	Driver tiredness detection	VGG-16, VGG-19
Guo and Markoni	[52]	2019	Driver tiredness detection	CNN+LSTM
Remeli et al.	[54]	2019	Driver identification from driving style	CNN+LSTM
Yang et al.	[55]	2021	Driver identification from driving style	CNN
Shahverdy	[59]	2020	Driving style classification	2d CNN
Ferreira et al.	[60]	2017	Driving style identification from smartphone sensor	SVM, RF, MLP, BN
Marafie et al.	[61]	2019	Driving behaviour management	SVM
Yin and Chen	[62]	2018	Usage-based insurance	AdaBoost
Dasgupta et al.	[68]	2019	Motorcycle helmet detection	CNN
Tawfeek and El-Basyouny	[70]	2018	Forward collision warning	Regression analysis
Song et al.	[71]	2018	Lane detection and classification	CNN
Zou et al.	[73]	2019	Lane detection	CNN+LSTM
Rajendran et al.	[76]	2019	Traffic sign recognition	YOLOv3
Li and Wang	[78]	2018	Traffic sign recognition	YOLOv3

SoftMax classifier-based neural network and evaluated it on simulated traffic flow data. Their model achieved 96.9% accuracy in detecting DDoS attacks.

When a corrupted node in the network refuses to forward packages, thus blocking the communication is called a blackhole attack. This can disrupt the operation of the network by making it unavailable for applications.

Acharya and Oluoch [85] demonstrate a method in which the combination of statistical approaches and SVM can help identify corrupted nodes and detect blackhole attacks. Their system finds blackhole attacks with over 98% accuracy in VANETs. Similarly, Pandey and Singh [86] use SVM to detect nodes performing blackhole attack with 95% accuracy by observing their energy consumption. An unsupervised, clustering based anomaly detection is performed in the work of Farahani [87]. In this work k-NN model is implemented to identify fraud clusters in the latent space of nodes with high confidence.

Providing authenticity and identity is an important task in VANETs. To operate correctly, data correctness must be guaranteed. Attacks against authenticity can harm transmitted data correctness, causing congestion or danger in traffic. A common and dangerous form of attack is location spoofing.

In their study, So et al. [88] use k-NN and SVM to classify nodes that perform attacks against the network. They evaluate their approach on the Vehicular Reference Misbehavior (VeReMi) dataset, a labeled simulated dataset containing several attack types [89]. This approach achieves 81.6% accuracy in finding malicious nodes.

Sharma et al. [90] propose a machine learning based framework that can classify several types of position falsification: fixed position transmitting, added offset to actual position or transmitting random positions. The framework was also evaluated on the VeReMi dataset, and performed with 98.9% accuracy

Sybil attacks allow hackers to show non-existing nodes to overrule decisions by faking the state of the architecture. The proposed work of Hammi et al.[91] uses a resource testing approach that is built on the fact that each node has limited computational resource. If a node presents multiple entities, their computation testing will indicate that. For the classification they used LR, SVM, RF, NN. The proposed classification method can recognize several sybil scenarios and reaches 86% accuracy on a simulated dataset.

Kamel et al. [92] use a different approach based on OBU's and RSU's misbehaviour reports. The proposed system is distributed into 3 phases: general misbehaviour type detection to detect the type of misbehavior, pseudonym linking to link pseudonyms coming from the same vehicle, sybil type detection to detect the type of sybil attack. These phases rely on AutoEncoders, RNNs, and MLP. The experimental data is simulated on the Luxembourg SUMO Traffic scenario validated by the University of Luxembourg. The proposed complex system performs 95.6% accuracy.

Confidentiality harming attacks compromise the privacy of nodes by making private messages accessible to hackers. Eavesdropping is a passive attack wherein hackers intercept information by listening in on VANET communications without permission. Detecting eavesdropping attacks can be challenging due to their passive nature.

Rahal et al. [93] introduced a process for botnet detection in VANETs that monitors vehicle and in-vehicle activity to detect DDoS attacks and Eavesdropping. The proposed work uses ML based algorithms (SVM, k-NN) and performs 99.4% accuracy at finding botnets.

Hackers compromising **integrity and data trust** modify packets in transit, thus altering their content before they are received by the intended recipient. A common attack form is broadcasting again messages that have already been broadcasted to manipulate the state of the network to gain benefits. These attacks are called replay attacks.

Fan et al. [94] showed that SVM can accurately detect

TABLE II
SUMMARY OF INTER-VEHICLE LEVEL APPLICATIONS

Author	Article	Year	Application	Used ML Algorithm
Kadam and Krovi	[83]	2021	Detect DDoS in VANET	Hybrid k-NN+SVM
Polat et al.	[84]	2020	Detect DDoS on SDN-based VANET	AutoEncoder
Acharya and Oluoch	[85]	2021	Detect blackhole attack	SVM
Pandey and Singh	[86]	2020	Detect blackhole attack	SVM
Farahani	[87]	2021	Detect blackhole attack	k-NN
So et al.	[88]	2018	Node misbehaviour detection in VANET	k-NN, SVM
Sharma et al.	[90]	2021	Location spoofing detection	k-NN, Decision Tree, RF
Hammi et al.	[91]	2022	Sybil attack detection	LR, SVM, RF, NN
Kamel et al.	[92]	2019	Sybil attack detection	AutoEncoder, RNN, MLP
Rahal et al.	[93]	2022	Botnet detection	SVM, k-NN
Fan et al.	[94]	2016	Routing replay attack detection	SVM
Ye et al.	[95]	2019	Resource allocation	Q-learning
Gao et al.	[96]	2019	Resource allocation	NN
Morocho et al.	[97]	2020	Extend wireless reachability in V2X	Encoder-Decoder NN
Moreira et al.	[98]	2020	QoS predictability in V2X communcation	RF
Mo et al.	[99]	2018	Overtaking assistant	Decision tree

replay attacks on a simulated communication scenario with 99.95% accuracy by analyzing the MAC Layer information of packets sent in network.

Accountability is mostly provided by implementing encryption technologies, using machine learning is not a common approach to provide the accountability of VANETs.

2) Network efficiency level: As discussed in Section IV-B1, VANETs have a dynamic nature, which makes effective resource allocation critical for ensuring availability. Machine learning can play a vital role in supporting resource allocation tasks.

The paper of Ye et al. [95] is one of the most cited publications on the topic. They present a novel deep reinforcement learning (Q-learning) based method for decentralized resource allocation. They claim that their algorithm allows each node and link to learn how to make resource allocation decision locally without global information effectively. The proposed algorithm results approximately 5% higher probability of satisfied V2V links than the defined baseline and increases the Sum Rate of V2I links by 3.5 times compared to random resource allocation method.

Gao et al. [96] approach this problem differently. They proposed a supervised learning deep neural network (DNN) based resource allocation method. To forward packages, hopping strategy is based on the shortest hop distance between nodes.

Morocho et al. [97] introduce a new, deep encoder-decoder and reinforcement learning based strategy for extending the reachability of multi-hop communications by up to 66.7%. The experiments in these works were tested on end-to-end IEEE 802.11p simulation.

Applications using V2X technology have stringent Quality of Service (QoS) requirements. Each application requires different network QoS. If netwok parameters do not satisfy the requirements of the application, its proper working is not guaranteed. The proposed work of Moreira et al. [98] proposes a machine learning based method to predict the QoS of a V2V network. In the proposed work, several supervised algorithms are evaluated, but the best performing is Random Forest that reaches up to 85% accuracy.

3) Application level: Applications that operate between a group of vehicle to perform synchronised actions are the base of cooperative driving. The aim is to enable vehicles to make decisions collectively, rather than independently, often by using summarized individual measurements. In this section, we cover the most common cooperative driving applications.

Cooperative Adaptive Cruise Control (C-ACC) is an application that allows vehicles to follow each other cooperatively. Using C-ACC gives the ability for vehicles to perform platooning, that is commonly referred as “driving together”. By continuously communicating with each other, vehicles can reduce the distance between them, which would not be possible without cooperation due to human reaction delay. The minimum following interval is typically 0.5 sec, compared to non-cooperative ACC where it is 1.6 sec [100]. Using C-ACC causes greater fuel economy, reduces congestion and lower the occupancy of roads [100]. This application mainly uses CAM and DENM messages to communicate hazardous road situations and to share vehicle parameters between convoys [100].

Cooperative overtaking assistant (COA) helps drivers to decide whether to perform overtaking maneuver on roads with bad visibility.

In their work, Mo et al. [99] introduce an overtaking assistant system and based on simulated experiments it increases the safety of overtaking. The overtaking process is built on multiple decision trees.

Traditional overtaking systems rely on processed visual data, but their work takes advantage of V2X. Strunz et al. [101] demonstrate how COA can contribute to platooning on freeways, increasing the average speed by 36%.

Other applications that are based on cooperativity [102]: intersection movement assist, left turn assist, cooperative forward collision avoidance, and emergency electronic brake light.

C. Infrastructure level

Infrastructure level applications operate on the “big picture”. Their goal is to make the whole infrastructure effective, fast and safe. Smart city is the concept where sustainability and efficiency is provided by the support of technology. Smart cities implement solutions for a great variety of their utilities, such as health care, infrastructure, water supply, pollution reducing, and transportation. Smart transportation management systems in smart cities integrate novel C-ITS technologies to improve constructional, management and support operations for ITS. The papers presenting solutions are listed in Table III.

The limited space and continuously increasing number of vehicles in big cities make parking a real challenge. Searching for parking lots, especially during peak hours, can waste a great amount of time and fuel, hold up traffic, and increase emissions. **Parking space** inventory control systems have been developed to provide solution to this difficulty.

Shin et al. [103] propose a neural network based method to enhance the effectiveness of intelligent parking guidance systems in parking areas where the necessary IoT devices are installed. This robust and lightweight method is capable of managing both public and private parking spaces in cities. The proposed work uses the current position of the vehicle and a centralized database about parking lots.

The work of Liu et al. [104] predicts the available parking lots with a neural network based approach. This work reached high accuracy based on the experimental results.

Priya et al. [105] propose an intelligent parking system that calculates the best-fit and optimal parking area based on size. The system extracts the dimensions of the vehicles using image processing CNN. Their work can achieve up to 85% accuracy in extracting proper dimensions.

Yang and Lam [106] survey how the implementation of intelligent parking services can create benefits for drivers.

Traffic congestion often forms in crowded areas. Spending more unexpected time on the road has a bad effect on travelling or transportation. Machine learning can help **predict the delay** of transportation.

Traffic congestion recognition and forecasting can be based on several data sources. Using computer vision and image processing for congestion detection is extensively studied. In their work, Jian et al. [107] show that roadway congestion detection can be done effectively using recordings of UAVs, as the “Electronic Eyes in the Skies”. They use convolutional neural network (CNN) to extract information from the captured images. They evaluated the system on a dataset that contained 8000 images and reached 93.5% accuracy. Bisio et al. [108] have published an extensive review of drone-based traffic monitoring systems. The work of Liu et al. [109] also uses computer vision to detect congestion, but their work relies on large and medium cities’ public security cameras and their algorithm reaches up to 82.5% accuracy. Gatto and Forster [110] use audio data from roads to determine the congestion. They extract 13 MFCC coefficients from the sound files and classify them using Random Forest Classifier. Their work reaches 95% accuracy on average on data collected from YouTube traffic videos.

Leung et al. [111] propose a heterogeneous approach to predict trolley bus delay based on transit and weather data. The presented method takes into account the current weather condition and uses fuzzy-logic based machine learning. Rather than using binary labels as the output for a decision it is capable of reasoning and uses a value between 0 and 1. The case study in this work, evaluated on Toronto Transit Commission database shows that this approach can approximate the trolley bus delay with 3.7 minutes mean prediction error.

Wu et al. [112] presents a method to predict the delay of urban railways. They present a novel Convolution LSTM Encoder-Decoder neural network based approach to predict delay. Their model using pure LSTM network reaches 16.82 sec mean prediction error, that is remarkably accurate.

An extensive amount of studies can be found on airplane arrival time delay prediction [113], [114].

Traffic prediction is also important in the aspect of infrastructure. **Predicting traffic** helps to react proactively to possible events in the future, thus preventing them. Proper and accurate prediction can forecast possible congestion, thus letting the smart routing applications avoid them. Nagy and Simon [115] give a comprehensive and detailed survey about novel traffic prediction methodologies.

Using graph convolution networks (GCN) has gained a big momentum at predicting traffic. Graph neural networks are capable of capturing rich feature of non-euclidean structured data and preserve its spatial structure. The paper written by Zhao et al.[116] is a frequently cited article that shows how real-time traffic forecasting can be performed using temporal GCN (t-GCN) and GCN. Other methodologies also use GCN to predict traffic [117].

RF and ARIMA, -which is a statistical approach-, are commonly used algorithms for traffic prediction, however, complex models, such as GCN tend to overperform it [117].

Other solutions published in the last five years mostly use LSTM and CNN [118].

High traffic affects the network layer as well by overloading it. Zhao et al. [119] proposed a deep reinforcement learning-based system that **optimizes network usage** and load for crowd management in smart cities. The proposed system uses deep Q-network model. The experiments are evaluated on a simulated environment from Topology Zoo, NSFCNET in Beijing, China. Their work has succeeded at optimizing network load and service availability.

One area where such possibilities can be useful is emergency management, which is expected to play a crucial role in the evolution of modern cities. Emergency vehicles, including ambulances, fire trucks, police cars, and transit agents’ vehicles, need to be quickly assigned to respond to critical situations as soon as possible.

Vehicle Routing Problem (VRP) is one of the most intensively studied optimization problem for which sufficient models and algorithms have been proposed. Emergency management reduces negative unfortunate or critical events. Travel time of EVs is a critical aspect of health care. Smart routing systems (SRS) can help to reduce this time by optimizing emergency vehicle routing and reducing hospitalization time. The survey of Tassone and Choudhury [120] shows that solving VRP helps to reduce the travelling and response time, and minimize the total cost of hospitalization. It details classical approaches and used algorithms and compares their performance.

Bai et al. [121] let the reader understand how machine learning can contribute to VRP and survey novel ML based

TABLE I
SUMMARY OF INFRASTRUCTURE-LEVEL APPLICATIONS

Author	Article	Year	Application	Used ML Algorithm
Shin et al.	[103]	2018	Intelligent parking guidance system	DNN
Liu et al.	[104]	2020	Parking lot availability	DNN
Priya et al.	[105]	2019	Optimal parking	CNN
Jial et al.	[107]	2019	Congestion detection using UAVs	CNN
Liu et al.	[109]	2020	Congestion detection using public cameras	CNN
Gatto and Forster	[110]	2021	Congestion detection using audio features	RF
Leung et al.	[111]	2020	Trolley bus delay prediction	Fuzzy-logic
Wu et al.	[112]	2019	Urban railway delay prediction	C-LSTM+Encoder-Decoder
Zhao et al.	[116]	2019	Real-time traffic forecasting	t-GCN
Zhao et al.	[119]	2019	Routing management	Q-Learning
Hussein et al.	[122]	2022	Ambulance vehicle routing	DNN
Hussein et al.	[123]	2022	Ambulance vehicle routing	BA-CNN
Nallaperume et al.	[125]	2019	Smart traffic control	Q-learning

methodologies.

Hussein [122] investigates how machine learning can help find the shortest travelling time for ambulance vehicles (AV). In their first ML based approach, they developed a neural network based model that can suggest optimal travelling path based on information about the accident, the injured patients and on position describers. In their simulation, the proposed approach could reduce the travelling distance by 3 times with negligible computing time [122]. In their another publication [123], a Bat Algorithm-Based Convolutional Neural Network (BA-CNN) is proposed. Bat is one of the most powerful optimisation algorithm that is inspired by the echolocation behavior of bats proposed by Yang and Gandomi [124]. The BA-CNN is very effective and can outperform existing methodologies [123].

Nallaperume et al. [125] introduce a novel framework that can implement smart traffic control based on multiple data source, such as social media data, IoT, weather data. The presented framework analyses the emotional status of drivers using social media data. This methodology uses deep reinforcement learning to improve traffic flow, reducing average waiting time at traffic lights by making traffic signal control decisions based on processing real-time data streams.

Smart traffic light control is a fundamental part of infrastructure optimization. Reinforcement learning is a common approach, but different works have different reward definitions. Rewarding scores can be based on reducing queue length [126], waiting time [127], or improving throughput and signal frequency in traffic lights [128].

V. CHALLENGES

Reducing computational time is a major challenge in developing ML-based C-ITS applications that can outperform existing solutions based on classical approaches. Evaluating complex NN architectures, slow data processing, or using ML for problems with high-dimensional state space representation can be time-consuming. In rapidly changing traffic, the response time of these applications must be minimized.

Ensuring that these applications are reliable enough to be implemented in real-world scenarios is also of paramount importance. Even if the application can achieve impressive accuracy, rare misbehavior can cause fatal issues.

If safety is provided, it is truly important to gain confidence and trust from customers. The real implementation of these application is depending on the acceptance and trust, thus the demand of users.

The accuracy of these applications depends largely on the quality and quantity of the input data. Researchers need to use accurate, reliable and correct data, which are difficult to collect and very expensive to buy from various data providers.

However, between all the listed problems, the most pressing is scalability. The applications surveyed in our article operate most of the time in simulated environment, or on a pilot area. For example, forecasting traffic congestion propagation is much more difficult on a complex city architecture.

VI. CONCLUSION

This review presented novel C-ITS use-cases that utilize machine learning. Firstly, a comprehensive introduction to machine learning and C-ITS technologies is provided to help readers understand their potential. Next, the paper presents novel C-ITS applications using machine learning. The categorization of the applications presented in this paper were developed by us, and we believe they enhance the clarity and structure of the paper. The paper shows that C-ITS applications are built on vehicle-level, inter vehicle-level and infrastructure-level applications as well.

The study explains that each hierarchical level has its own responsibility in accomplishing complex solutions, and the cooperation of applications on different levels is indispensable for the implementation of smart transportation. Machine learning has a huge impact on almost every aspect of C-ITS applications and can radically improve the efficiency, speed, and performance of these applications. The article highlights the need for further research and development in this field to overcome challenges such as reducing computational time, gaining customer confidence and solving scalability issues. Using ML based technologies has a huge potential to develop novel C-ITS applications.

We hope that our article will help the reader understand the importance of machine learning in C-ITS applications and give a summary of how this area builds up.

ACKNOWLEDGMENT

The research reported in this paper is part of project no. BME-NVA-02, implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021 funding scheme.

REFERENCES

[1] "Vehicles in use europe 2022 report." ACEA, 01 2022.

[2] D. Lee, D. Fahey, A. Skowron, M. Allen, U. Burkhardt, Q. Chen, S. Doherty, S. Freeman, P. Forster, J. Fuglestvedt, A. Gettelman, R. D. León, L. Lim, M. Lund, R. Millar, B. Owen, J. Penner, G. Pitari, M. Prather, R. Sausen, and L. Wilcox, "The contribution of global aviation to anthropogenic climate forcing for 2000 to 2018," *Atmospheric Environment*, vol. 244, p. 117834, jan, **DOI**: 10.1016/j.atmosenv.2020.117834.

[3] *The Future of Rail*. OECD, feb 2019, **DOI**: 10.1787/9789264312821-en.

[4] P. Lasley, "2021 urban mobility report," 2021.

[5] M. Zuurbier, G. Hoek, M. Oldenwening, V. Lenters, K. Meliefste, P. van den Hazel, and B. Brunekreef, "Commuters' exposure to particulate matter air pollution is affected by mode of transport, fuel type, and route," *Environmental Health Perspectives*, vol. 118, no. 6, pp. 783–789, 2010, **DOI**: 10.1289/ehp.0901622.

[6] S. Dick, "Artificial Intelligence," *Harvard Data Science Review*, vol. 1, no. 1, 07 2019, **DOI**: 10.1162/99608f92.92fe150c.

[7] S. I. for Human-Centered AI, "The ai index 2022 annual report," march 2022.

[8] A. Althnian, D. AlSaeed, H. Al-Baity, A. Samha, A. B. Dris, N. Alzakari, A. Abou Elwafa, and H. Kurdi, "Impact of dataset size on classification performance: an empirical evaluation in the medical domain," *Applied Sciences*, vol. 11, no. 2, p. 796, 2021, **DOI**: 10.3390/app11020796.

[9] J. Prusa, T. M. Khoshgoftaar, and N. Seliya, "The effect of dataset size on training tweet sentiment classifiers," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2015, pp. 96–102, **DOI**: 10.1109/icmla.2015.22.

[10] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[11] M. Feurer and F. Hutter, "Hyperparameter optimization," in *Automated machine learning*. Springer, Cham, 2019, pp. 3–33.

[12] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015, **DOI**: 10.5121/ijdkp.2015.5201.

[13] S. B. Kotsiantis, I. Zaharakis, P. Pintelas et al., "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[14] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on neural networks*, vol. 16, no. 3, pp. 645–678, 2005.

[15] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[16] M. Kiran, C. Wang, G. Papadimitriou, A. Mandal, and E. Deelman, "Detecting anomalous packets in network transfers: investigations using pca, autoencoder and isolation forest in tcp," *Machine Learning*, vol. 109, no. 5, pp. 1127–1143, 2020, **DOI**: 10.1007/s10994-020-05870-y.

[17] M. Sánchez-Rico and J. Alvardo, "Dimensionality reduction techniques as a preliminary step to cluster analysis: A comparison between pca, t-sne and umap," in *9th European Congress of Methodology*, 2020.

[18] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *Towards Data Sci*, vol. 6, no. 12, pp. 310–316, 2017.

[19] L. Chen, A. Habibovic, M. Gråsjö, M. Adebahr, and P. King, "Cloud-based traffic control: a system of systems for accelerating c-its deployment and autonomous vehicle integration," in *Virtual ITS European Congress*, *9-10 November 2020*, 2020.

[20] E. Vieira, J. Almeida, J. Ferreira, T. Dias, A. V. Silva, and L. Moura, "A roadside and cloud-based vehicular communications framework for the provision of c-ITS services," *Information*, vol. 14, no. 3, p. 153, mar 2023, **DOI**: 10.3390/info14030153.

[21] G. Naik, B. Choudhury, and J.-M. Park, "IEEE 802.11bd &amp 5g NR v2x: Evolution of radio access technologies for v2x communications," *IEEE Access*, vol. 7, pp. 70 169–70 184, 2019, **DOI**: 10.1109/access.2019.2919489.

[22] S. Chen, J. Hu, Y. Shi, Y. Peng, J. Fang, R. Zhao, and L. Zhao, "Vehicle-to-everything (v2x) services supported by LTE-based systems and 5g," *IEEE Communications Standards Magazine*, vol. 1, no. 2, pp. 70–76, 2017, **DOI**: 10.1109/mcomstd.2017.1700015.

[23] "En 302 636-4-1 (v1.4.1) – geographical addressing and forwarding for point-to-point and point-to-multipoint communications." ETSI, 11 2019.

[24] "En 302 637-2 (v1.4.1) – media-dependent functionalities for its-g5." ETSI, 04 2019.

[25] "En 302 637-3 (v1.3.1) – specifications of decentralized environmental notification basic service." ETSI, 04 2019.

[26] "Tr 103 562 (v2.1.1) – analysis of the collective perception service (cps); release 2." ETSI, 12 2019.

[27] "Ts 103 301 (v1.2.1) – facilities layer protocols and communication requirements for infrastructure services." ETSI, 08 2018.

[28] "Ts 102 894-2 (v1.3.1) – applications and facilities layer common data dictionary." ETSI, 08 2018.

[29] "Ts 103 300-3 (v2.1.1) – specification of vru awareness basic service; release 2." ETSI, 11 2020.

[30] L. Bokor, "V2x jarmukommunikacios protokollok forgalommenedzsment fokuszu bigdata elemzeshez es ebeavatkozashoz felhasznalhato adatelemei," 2020, accessed: 2022-05-19.

[31] R. Griffin and G. McGwin Jr, "Emergency medical service providers' experiences with traffic congestion," *The Journal of emergency medicine*, vol. 44, no. 2, pp. 398–405, 2013, **DOI**: 10.1016/j.jemermed.2012.01.066.

[32] C. Krupitzer, T. Wagenhals, M. Züfle, V. Lesch, D. Schäfer, A. Mozaffarin, J. Edinger, C. Becker, and S. Kounev, "A survey on predictive maintenance for industry 4.0," *arXiv preprint arXiv:2002.08224*, 2020.

[33] R. Prytz, "Machine learning methods for vehicle predictive maintenance using off-board and on-board data," Ph.D. dissertation, Halmstad University Press, 2014.

[34] K. Miller and A. Dubrawski, "System-level predictive maintenance: review of research literature and gap analysis," *arXiv preprint arXiv:2005.05239*, 2020.

[35] Y. Wen, M. F. Rahman, H. Xu, and T.-L. B. Tseng, "Recent advances and trends of predictive maintenance from data-driven machine prognostics perspective," *Measurement*, vol. 187, p. 110276, 2022, **DOI**: 10.1016/j.measurement.2021.110276.

[36] A. Manghai, R. Jegadeeshwaran, and V. Sugumaran, "Brake fault diagnosis through machine learning approaches–a review," *Structural Durability & Health Monitoring*, vol. 11, no. 1, p. 43, 2017.

[37] R. Magargle, L. Johnson, P. Mandloi, P. Davoudabadi, O. Kesarkar, S. Krishnaswamy, J. Batteh, and A. Pitchaikani, "A simulation-based digital twin for model-driven health monitoring and predictive maintenance of an automotive braking system," in *Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15- 17, 2017*, no. 132. Linko ̈ping University Electronic Press, 2017, pp. 35–46, **DOI**: 10.3384/ecp1713235.

[38] A. Joshuva, S. Anaimuthu, N. Selvaraju, S. J. Muthiya, and M. Subramaniam, "A machine learning approach for vibration signal based fault classification on hydraulic braking system through c4. 5 decision tree classifier and logistic model tree classifier," *SAE Technical Paper, Tech. Rep.*, 2020.

[39] S. Ruggieri, "Efficient c4. 5 [classification algorithm]," *IEEE transac- tions on knowledge and data engineering*, vol. 14, no. 2, pp. 438–444, 2002.

[40] J. Arockia Dhanraj, S. Muthiya, M. Subramaniam, S. Salyan, P. Chaurasiya, A. Gopalan, and S. Anaimuthu, "A comparative study with j48 and random tree classifier for predicting the state of hydraulic braking system through vibration signals," *SAE Tech. Pap,* pp. 1–6, 2021, **DOI**: 10.4271/2021-28-0254.

[41] X. Zhang, M. Zhang, Z. Xiang, and J. Mo, "Research on diagnosis algorithm of mechanical equipment brake friction fault based on mcnn-svm," *Measurement*, vol. 186, p. 110065, 2021, **DOI**: 10.1016/j.measurement.2021.110065.

[42] F. Bonini, G. Manduchi, N. Mancinelli, and A. Martini, "Estimation of the braking torque for motogp class motorcycles with carbon braking systems through machine learning algorithms," in *2021 IEEE International Workshop on Metrology for Automotive (MetroAutomotive)*. IEEE, 2021, pp. 1–6, **DOI**: 10.1109/metroautomotive50197.2021.9502878.

[43] G. Wang and S. Yin, "Data-driven fault diagnosis for an automobile suspension system by using a clustering based method," *Journal of the Franklin Institute*, vol. 351, no. 6, pp. 3231–3244, 2014, DOI: 10.1016/j.jfranklin.2014.03.004.

[44] A. Moradi, S. S. H. Nazari, and K. Rahmani, "Sleepiness and the risk of road traffic accidents: A systematic review and meta-analysis of previous studies," *Transportation research part F: traffic psychology and behaviour*, vol. 65, pp. 620–629, 2019, DOI: 10.1016/j.trf.2018.09.013.

[45] J. Kalsi, T. Tervo, A. Bachour, and M. Partinen, "Sleep versus non-sleep-related fatal road accidents," *Sleep medicine*, vol. 51, pp. 148–152, 2018, DOI: 10.1016/j.sleep.2018.04.017.

[46] T. Vogelpohl, M. Kühn, T. Hummel, and M. Vollrath, "Asleep at the automated wheel—sleepiness and fatigue during highly automated driving," *Accident Analysis & Prevention*, vol. 126, pp. 70–84, 2019, DOI: 10.1016/j.aap.2018.03.013.

[47] N. Nietjet, "A proposed accident preventive model for smart vehicles," vol. 6, pp. 001–010, 02 2018.

[48] A. Byrnes and C. Sturton, "On using drivers' eyes to predict accident-causing drowsiness levels," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2092–2097, DOI: 10.1109/itsc.2018.8569293.

[49] R. Jabbar, M. Shinoy, M. Kharbeche, K. Al-Khalifa, M. Krichen, and K. Barkaoui, "Driver drowsiness detection model using convolutional neural networks techniques for android application," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*. IEEE, 2020, pp. 237–242, DOI: 10.1109/iciot48696.2020.9089484.

[50] M. Hashemi, A. Mirrashid, and A. Beheshti Shirazi, "Driver safety development: Real-time driver drowsiness detection system based on convolutional neural network," *SN Computer Science*, vol. 1, no. 5, pp. 1–10, 2020, DOI: 10.1007/s42979-020-00306-9.

[51] A. Bagaskara and M. Suryanegara, "Evaluation of VGG-16 and VGG-19 deep learning architecture for classifying dementia people," in *2021 4th International Conference of Computer and Informatics Engineering (IC2IE)*. IEEE, sep 2021, DOI: 10.1109/ic2ie53219.2021.9649132.

[52] J.-M. Guo and H. Markoni, "Driver drowsiness detection using hybrid convolutional neural network and long short-term memory," *Multimedia tools and applications*, vol. 78, no. 20, pp. 29059–29087, 2019, DOI: 10.1007/s11042-018-6378-6.

[53] M. Ngxande, J.-R. Tapamo, and M. Burke, "Driver drowsiness detection using behavioral measures and machine learning techniques: A review of state-of-art techniques," *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*, pp. 156–161, 2017, DOI: 10.1109/robomech.2017.8261140.

[54] M. Remeli, S. Lestyán, G. Acs, and G. Biczók, "Automatic driver identification from in-vehicle network logs," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 1150–1157, DOI: 10.1109/itsc.2019.8917354.

[55] J. Yang, R. Zhao, M. Zhu, D. Hallac, J. Sodnik, and J. Leskovec, "Driver2vec: Driver identification from automotive data," *arXiv preprint arXiv:2102.05234*, 2021.

[56] D. Hallac, A. Sharang, R. Stahlmann, A. Lamprecht, M. Huber, M. Roehder, J. Leskovec et al., "Driver identification using automobile sensor data from a single turn," in *2016 IEEE 19th International ConferenceonIntelligentTransportationSystems (ITSC)*. IEEE, 2016, pp. 953–958, DOI: 10.1109/itsc.2016.7795670.

[57] N. Lin, C. Zong, M. Tomizuka, P. Song, Z. Zhang, and G. Li, "An overview on study of identification of driver behavior characteristics for automotive control," *Mathematical Problems in Engineering*, vol. 2014, 2014, DOI: 10.1155/2014/569109.

[58] G. A. M. Meiring and H. C. Myburgh, "A review of intelligent driving style analysis systems and related artificial intelligence algorithms," *Sensors*, vol. 15, no. 12, pp. 30 653–30 682, 2015, DOI: 10.3390/s151229822.

[59] M. Shahverdy, M. Fathy, R. Berangi, and M. Sabokrou, "Driver behavior detection and classification using deep convolutional neural networks," *Expert Systems with Applications*, vol. 149, p. 113240, 2020, DOI: 10.1016/j.eswa.2020.113240.

[60] J. Ferreira, E. Carvalho, B. V. Ferreira, C. de Souza, Y. Suhara, A. Pentland, and G. Pessin, "Driver behavior profiling: An investigation with different smartphone sensors and machine learning," *PLoS one*, vol. 12, no. 4, p. e0174959, 2017, DOI: 10.1371/journal.pone.0174959.

[61] Z. Marafie, K.-J. Lin, D. Wang, H. Lyu, Y. Meng, and T. Ito, "Autocoach: Driving behavior management using intelligent iot services," in *2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA)*. IEEE, 2019, pp. 103–110, DOI: 10.1109/soca.2019.00023.

[62] J.-L. Yin and B.-H. Chen, "An advanced driver risk measurement system for usage-based insurance on big driving data," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 585–594, 2018, DOI: 10.1109/tiv.2018.2874530.

[63] Y. Bian, C. Yang, J. L. Zhao, and L. Liang, "Good drivers pay less: A study of usage-based vehicle insurance models," *Transportation research part A: policy and practice*, vol. 107, pp. 20–34, 2018, DOI: 10.1016/j.tra.2017.10.018.

[64] M. F. Carfora, F. Martinelli, F. Mercaldo, V. Nardone, A. Orlando, A. Santone, and G. Vaglini, "A "pay-how-you-drive" car insurance approach through cluster analysis," *Soft Computing*, vol. 23, no. 9, pp. 2863–2875, 2019, DOI: 10.1007/s00500-018-3274-y.

[65] A. F. Abbas, U. U. Sheikh, F. T. AL-Dhief, and M. N. H. Mohd, "A comprehensive review of vehicle detection using computer vision," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 3, pp. 838–850, 2021, DOI: 10.12928/telkomnika.v19i3.12880.

[66] A. Bouguettaya, H. Zarzour, A. Kechida, and A. M. Taberkit, "Vehicle detection from uav imagery with deep learning: a review," *IEEE Transactions on Neural Networks and Learning Systems*, 2021, DOI: 10.1109/tnnls.2021.3080276.

[67] J. Tang, L. Wan, J. Schooling, P. Zhao, J. Chen, and S. Wei, "Automatic number plate recognition (anpr) in smart cities: A systematic review on technological advancements and application cases," *Cities*, vol. 129, p. 103833, 2022, DOI: 10.1016/j.cities.2022.103833.

[68] M. Dasgupta, O. Bandyopadhyay, and S. Chatterji, "Automated helmet detection for multiple motorcycle riders using cnn," in *2019 IEEE Conference on Information and Communication Technology*. IEEE, 2019, pp. 1–4, DOI: 10.1109/cict48419.2019.9066191.

[69] J. B. Cicchino, "Effectiveness of forward collision warning and autonomous emergency braking systems in reducing front-to-rear crash rates," *Accident Analysis & Prevention*, vol. 99, pp. 142–152, 2017, DOI: 10.1016/j.aap.2016.11.009.

[70] M. H. Tawfeek and K. El-Basyouny, "A perceptual forward collision warning model using naturalistic driving data," *Canadian Journal of Civil Engineering*, vol. 45, no. 10, pp. 899–907, 2018, DOI: 10.1139/cjce-2017-0592.

[71] W. Song, Y. Yang, M. Fu, Y. Li, and M. Wang, "Lane detection and classification for forward collision warning system based on stereo vision," *IEEE Sensors Journal*, vol. 18, no. 12, pp. 5151–5163, 2018, DOI: 10.1109/jsen.2018.2832291.

[72] J. Tang, S. Li, and P. Liu, "A review of lane detection methods based on deep learning," *Pattern Recognition*, vol. 111, p. 107623, 2021, DOI: 10.1016/j.patcog.2020.107623.

[73] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust lane detection from continuous driving scenes using deep neural networks," *IEEE transactions on vehicular technology*, vol. 69, no. 1, pp. 41–54, 2019, DOI: 10.1109/tvt.2019.2949603.

[74] A. De la Escalera, J. M. Armingol, and M. Mata, "Traffic sign recognition and analysis for intelligent vehicles," *Image and vision computing*, vol. 21, no. 3, pp. 247–258, 2003.

[75] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural networks*, vol. 32, pp. 323–332, 2012, DOI: 10.1016/j.neunet.2012.02.016.

[76] S. P. Rajendran, L. Shine, R. Pradeep, and S. Vijayaraghavan, "Real-time traffic sign recognition using yolov3 based detector," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 2019, pp. 1–7, DOI: 10.1109/icccnt45670.2019.8944890.

[77] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: a multi-class classification competition," in *The 2011 international joint conference on neural networks*. IEEE, 2011, pp. 1453–1460.

[78] J. Li and Z. Wang, "Real-time traffic sign recognition based on efficient cnns in the wild," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 975–984, 2018, DOI: 10.1109/tits.2018.2843815.

[79] S. Ftaimi and T. Mazri, "A comparative study of machine learning algorithms for vanet networks," in *Proceedings of the 3rd International Conference on Networking, Information Systems & Security*, 2020, pp. 1–8, **DOI**: 10.1145/3386723.3387829.

[80] F. Tang, B. Mao, N. Kato, and G. Gui, "Comprehensive survey on machine learning in vehicular network: technology, applications and challenges," *IEEE Communications Surveys & Tutorials*, 2021, **DOI**: 10.1109/comst.2021.3089688.

[81] M. N. Mejri, J. Ben-Othman, and M. Hamdi, "Survey on vanet security challenges and possible cryptographic solutions," *Vehicular Communications*, vol. 1, no. 2, pp. 53–66, 2014, **DOI**: 10.1016/j.vehcom.2014.05.001.

[82] A. M. Alrehan and F. A. Alhaidari, "Machine learning techniques to detect ddos attacks on vanet system: a survey," in *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2019, pp. 1–6, **DOI**: 10.1109/cais.2019.8769454.

[83] N. Kadam and R. S. Krovi, "Machine learning approach of hybrid ksvn algorithm to detect ddos attack in vanet," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, 2021, **DOI**: 10.14569/ijacsa.2021.0120782.

[84] H. Polat, M. Turkoglu, and O. Polat, "Deep network approach with stacked sparse autoencoders in detection of ddos attacks on sdn-based vanet," *IET Communications*, vol. 14, no. 22, pp. 4089–4100, 2020, **DOI**: 10.1049/iet-com.2020.0477.

[85] A. Acharya and J. Oluoch, "A dual approach for preventing blackhole attacks in vehicular ad hoc networks using statistical techniques and supervised machine learning," in *2021 IEEE International Conference on Electro Information Technology (EIT)*. IEEE, 2021, pp. 230–235, **DOI**: 10.1109/eit51626.2021.9491885.

[86] S. Pandey and V. Singh, "Blackhole attack detection using machine learning approach on manet," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*. IEEE, 2020, pp. 797–802, **DOI**: 10.1109/icesc48915.2020.9155770.

[87] G. Farahani, "Black hole attack detection using k-nearest neighbor algorithm and reputation calculation in mobile ad hoc networks," *Security and Communication Networks*, vol. 2021, 2021.

[88] S. So, P. Sharma, and J. Petit, "Integrating plausibility checks and machine learning for misbehavior detection in vanet," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 564–571, **DOI**: 10.1109/icmla.2018.00091.

[89] R. W. van der Heijden, T. Lukaseder, and F. Kargl, "VeReMi: A dataset for comparable evaluation of misbehavior detection in VANETs," in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Springer International Publishing, 2018, pp. 318–337, **DOI**: 10.1007/978-3-030-01701-9_18.

[90] A. Sharma and A. Jaekel, "Machine learning approach for detecting location spoofing in vanet," in *2021 International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2021, pp. 1–6, **DOI**: 10.1109/icccn52240.2021.9522170.

[91] B. Hammi, M. Y. Idir, and R. Khatoun, "A machine learning based approach for the detection of sybil attacks in c-its," in *2022 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 2022, pp. 1–4, **DOI**: 10.23919/apnoms56106.2022.9919991.

[92] J. Kamel, F. Haidar, I. B. Jemaa, A. Kaiser, B. Lonc, and P. Urien, "A misbehavior authority system for sybil attack detection in c-its," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2019, pp. 1117–1123, **DOI**: 10.1109/uemcon47517.2019.8993045.

[93] R. Rahal, A. Amara Korba, N. Ghoualmi-Zine, Y. Challal, and M. Y. Ghamri-Doudane, "Antibotv: A multilevel behaviour-based framework for botnets detection in vehicular networks," *Journal of Network and Systems Management*, vol. 30, no. 1, pp. 1–40, 2022, **DOI**: 10.1007/s10922-021-09630-8.

[94] Q. G. Fan, L. Wang, Y. N. Cai, Y. Q. Li, and J. Chen, "Vanet routing replay attack detection research based on svm," in *MATEC Web of Conferences*, vol. 63. EDP Sciences, 2016, p. 05020, **DOI**: 10.1051/matecconf/20166305020.

[95] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for v2v communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3163–3173, 2019, **DOI**: 10.1109/icc.2018.8422586.

[96] J. Gao, M. R. Khandaker, F. Tariq, K.-K. Wong, and R. T. Khan, "Deep neural network based resource allocation for v2x communications," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. IEEE, 2019, pp. 1–5, **DOI**: 10.1109/vtcfall.2019.8891446.

[97] M. E. Morocho-Cayamcela, H. Lee, and W. Lim, "Machine learning to improve multi-hop searching and extended wireless reachability in v2x," *IEEE Communications Letters*, vol. 24, no. 7, pp. 1477–1481, 2020, **DOI**: 10.1109/lcomm.2020.2982887.

[98] D. C. Moreira, I. M. Guerreiro, W. Sun, C. C. Cavalcante, and D. A. Sousa, "Qos predictability in v2x communication with machine learning," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5, **DOI**: 10.1109/vtc2020-spring48590.2020.9129490.

[99] C. Mo, Y. Li, and L. Zheng, "Simulation and analysis on overtaking safety assistance system based on vehicle-to-vehicle communication," *Automotive Innovation*, vol. 1, no. 2, pp. 158–166, 2018, **DOI**: 10.1007/s42154-018-0017-9.

[100] A. Vinel, L. Lan, and N. Lyamin, "Vehicle-to-vehicle communication in c-acc/platooning scenarios," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 192–197, 2015, **DOI**: 10.1109/mcom.2015.7180527.

[101] M. Strunz, J. Heinovski, and F. Dressler, "Coop: V2v-based cooperative overtaking for platoons on freeways," in *ITSC*, 2021, pp. 1090–1097, **DOI**: 10.1109/itsc48978.2021.9565122.

[102] R. Miucic, A. Sheikh, Z. Medenica, and R. Kunde, "V2x applications using collaborative perception," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2018, pp. 1–6, **DOI**: 10.1109/vtcfall.2018.8690818.

[103] J.-H. Shin, H.-B. Jun, and J.-G. Kim, "Dynamic control of intelligent parking guidance using neural network predictive control," *Computers & Industrial Engineering*, vol. 120, pp. 15–30, 2018, **DOI**: 10.1016/j.cie.2018.04.023.

[104] J. Liu, J. Wu, and L. Sun, "Control method of urban intelligent parking guidance system based on internet of things," *Computer Communications*, vol. 153, pp. 279–285, 2020, **DOI**: 10.1016/j.comcom.2020.01.063.

[105] D. B. Priya, V. R. Rao, and C. V. Kumar, "Intelligent parking system," *Int. Journal of Recent Technology and Engineering*, no. Part V, pp. 70–83, 2019.

[106] W. Yang and P. T. Lam, "Evaluation of drivers' benefits accruing from an intelligent parking information system," *Journal of cleaner production*, vol. 231, pp. 783–793, 2019, **DOI**: 10.1016/j.jclepro.2019.05.247.

[107] L. Jian, Z. Li, X. Yang, W. Wu, A. Ahmad, and G. Jeon, "Combining unmanned aerial vehicles with artificial-intelligence technology for traffic-congestion recognition: Electronic eyes in the skies to spot clogged roads," *IEEE Consumer Electronics Magazine*, vol. 8, no. 3, pp. 81–86, may 2019, **DOI**: 10.1109/mce.2019.2892286.

[108] I. Bisio, C. Garibotto, H. Haleem, F. Lavagetto, and A. Sciarrone, "A systematic review of drone based road traffic monitoring system," *IEEE Access*, vol. 10, pp. 101 537–101 555, 2022, **DOI**: 10.1109/access.2022.3207282.

[109] X. Liu, W. Gao, D. Feng, and X. Gao, "Abnormal traffic congestion recognition based on video analysis," in *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, aug 2020, **DOI**: 10.1109/mipr49039.2020.00016.

[110] R. C. Gatto and C. H. Q. Forster, "Audio-based machine learning model for traffic congestion detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7200–7207, nov 2021, **DOI**: 10.1109/tits.2020.3003111.

[111] C. K. Leung, J. D. Elias, S. M. Minuk, A. R. R. de Jesus, and A. Cuzzocrea, "An innovative fuzzy logic-based machine learning algorithm for supporting predictive analytics on big transportation data," in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2020, pp. 1–8, **DOI**: 10.1109/fuzz48607.2020.9177823.

[112] J. Wu, L. Zhou, C. Cai, F. Dong, J. Shen, and G. Sun, "Towards a general prediction system for the primary delay in urban railways," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3482–3487, **DOI**: 10.1109/itsc.2019.8916868.

[113] B. Yu, Z. Guo, S. Asian, H. Wang, and G. Chen, "Flight delay prediction for commercial air transport: A deep learning approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 125, pp. 203–221, 2019, **DOI**: 10.1016/j.tre.2019.03.013.

[114] N. Chakrabarty, T. Kundu, S. Dandapat, A. Sarkar, and D. K. Kole, "Flight arrival delay prediction using gradient boosting classifier," in *Emerging technologies in data mining and information security.* Springer, 2019, pp. 651–659, **DOI**: 10.1007/978-981-13-1498-8_57.

[115] A. M. Nagy and V. Simon, “Survey on traffic prediction in smart cities,” *Pervasive and Mobile Computing*, vol. 50, pp. 148–163, 2018, **doi**: 10.1016/j.pmcj.2018.07.004.

[116] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, “T-gcn: A temporal graph convolutional network for traffic prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2019, **doi**: 10.1109/tits.2019.2935152.

[117] K. Guo, Y. Hu, Z. Qian, H. Liu, K. Zhang, Y. Sun, J. Gao, and B. Yin, “Optimized graph convolution recurrent neural network for traffic prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 1138–1149, 2020, **doi**: 10.1109/tits.2019.2963722.

[118] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. Choudhury, and A. K. Qin, “A survey on modern deep neural network for traffic prediction: Trends, methods and challenges,” *IEEE Transactions on Knowledge and Data Engineering*, 2020, **doi**: 10.1109/tkde.2020.3001195.

[119] L. Zhao, J. Wang, J. Liu, and N. Kato, “Routing for crowd management in smart cities: A deep reinforcement learning perspective,” *IEEE Communications Magazine*, vol. 57, no. 4, pp. 88–93, 2019, **doi**: 10.1109/mcom.2019.1800603.

[120] J. Tassone and S. Choudhury, “A comprehensive survey on the ambulance routing and location problems,” *arXiv preprint arXiv:2001.05288*, 2020.

[121] R. Bai, X. Chen, Z.-L. Chen, T. Cui, S. Gong, W. He, X. Jiang, H. Jin, J. Jin, G. Kendall et al., “Analytics and machine learning in vehicle routing research,” *International Journal of Production Research*, pp. 1–27, 2021, **doi**: 10.1080/00207543.2021.2013566.

[122] T. D. H. Hussein, M. Frikha, S. Ahmed, and J. Rahebi, “Ambulance vehicle routing in smart cities using artificial neural network,” in *2022 6th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. IEEE, 2022, pp. 1–6, **doi**: 10.1109/atsip55956.2022.9805857.

[123] T. Darwassh Hanawy Hussein, M. Frikha, S. Ahmed, and J. Rahebi, “Ba-cnn: Bat algorithm-based convolutional neural network algorithm for ambulance vehicle routing in smart cities,” *Mobile Information Systems*, vol. 2022, 2022, **doi**: 10.1155/2022/7339647.

[124] X.-S. Yang and A. H. Gandomi, “Bat algorithm: a novel approach for global engineering optimization,” *Engineering computations*, 2012, **doi**: 10.1108/02644401211235834.

[125] D. Nallaperuma, R. Nawaratne, T. Bandaragoda, A. Adikari, S. Nguyen, T. Kempitiya, D. De Silva, D. Alahakoon, and D. Pothuhera, “Online incremental machine learning platform for big data-driven smart traffic management,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4679–4690, 2019, **doi**: 10.1109/tits.2019.2924883.

[126] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li, “Colight: Learning network-level cooperation for traffic signal control,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1913–1922.

[127] T. Chu, J. Wang, L. Codecà, and Z. Li, “Multi-agent deep reinforcement learning for large-scale traffic signal control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1086–1095, 2019, **doi**: 10.1109/tits.2019.2901791.

[128] M. Guo, P. Wang, C.-Y. Chan, and S. Askary, “A reinforcement learning approach for intelligent traffic signal control at urban intersections,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 4242–4247, **doi**: 10.1109/itsc.2019.8917268.

Norman Bereczki received his M.Sc. degree from the Budapest University of Technology and Economics (BME) in 2023. During his B.Sc. and M.Sc., he was part of the PARIPA university and industry cooperation programme and now is an active organizer of the PARIPA Alumni. He is currently doing his Ph.D. at the Department of Networked Systems and Services (HIT) in the Multimedia Networks and Services Laboratory (MEDIANETS). His research interests cover artificial intelligence use-cases for optimizing smart city services. He is currently working as a data scientist at Nitrowise Labs.

Vilmos Simon received his PhD from the Budapest University of Technology and Economics (BME) in 2009. Currently he is an Associate Professor at the Department of Networked Systems and Services and Head of the Multimedia Networks and Services Laboratory. He has done research on mobility management, energy efficiency in mobile cellular systems and self-organized mobile networks, recently his research interests include machine learning and data analytics for smart cities and intelligent transportation management systems. He published 60+ papers in international journals and conferences, and acts as a reviewer or organizer for numerous scientific conferences. He leaded the development of the first ITS system in Hungary.