



A Review on Machine Learning in Intelligent Transportation Systems Applications

Abul Kalam Azad^{1,2,*}, Travis Atkison¹ and A. F. M. Shahen Shah³

¹Department of Computer Science, University of Alabama, Tuscaloosa, USA

²Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University, Noakhali, Bangladesh

³Electronics and Communication Engineering Department, Yildiz Technical University, Istanbul, Türkiye

Abstract:

With the expansion of transportation networks and the advancement of embedded and communication technologies, research on intelligent transportation systems has gained immense interest from the research community, especially utilizing the potential of machine learning techniques for increasing efficiency, safety, and travel experience. This paper presents a detailed review of intelligent transportation systems applications using machine learning techniques. Initially, 11 popular applications were selected for further study by focusing on four key areas: traffic management, safety management, infotainment and comfort, and autonomous driving. To explore the current trends of each application, 48 recent proposals using machine learning techniques that have gained high attention have been selected by following some selection criteria. The selected proposals have been discussed in detail, focusing on the proposed methods and contributions. After a detailed review, 10 potential issues have been identified and discussed, which could lead to the development of more efficient and optimized intelligent transportation systems solutions. Overall, the review serves as a valuable guide for researchers in identifying the current research trends in popular intelligent transportation systems applications using machine learning, pinpointing the gaps and developing more attractive solutions.

Keywords: Machine learning, ITS, Transportation, Applications, Safety management, Communication technologies.

© 2024 The Author(s). Published by Bentham Open.

This is an open access article distributed under the terms of the Creative Commons Attribution 4.0 International Public License (CC-BY 4.0), a copy of which is available at: <https://creativecommons.org/licenses/by/4.0/legalcode>. This license permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

*Address correspondence to this author at the Department of Computer Science, University of Alabama, Tuscaloosa, USA and Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University, Noakhali, Bangladesh; E-mail: aaazad1@crimson.ua.edu

Cite as: Azad A, Atkison T, Shah A. A Review on Machine Learning in Intelligent Transportation Systems Applications. Open Transp J, 2024; 18: e26671212330496. <http://dx.doi.org/10.2174/0126671212330496240821114216>



CrossMark

Received: June 03, 2024

Revised: July 31, 2024

Accepted: August 07, 2024

Published: September 11, 2024



Send Orders for Reprints to
reprints@benthamscience.net

1. INTRODUCTION

In modern society, transportation systems allow the movement of humans and goods from one place to another by various modes, including road, rail, and air transportation, using different infrastructures, like roads, rails, and waterways. For all these transportation types, the ability to travel long distances at high speed with comfort and safety indicates the progression of modern civilization [1]. Among all, vehicular networks by road are considered the largest transportation systems due to their volume, flexibility, scalability, and ubiquity. Nowadays,

vehicular networks have turned into Intelligent Transportation Systems (ITS) that aim to integrate different innovative services to vehicle users and administrators of vehicle networks in order to enhance traffic safety and traffic efficiency, and offer various value-added services [2].

Many ITS applications have been introduced over the last few years to heighten the comfort, safety, efficiency, and functionality of vehicular networks. In previous studies [3, 4], ITS applications have been categorized into i) road/traffic safety, ii) traffic efficiency, and iii) other

value-added applications. Another classification was proposed in a paper [5] with respect to the type of i) infotainment and comfort, ii) traffic management, iii) road safety, and iv) autonomous driving. According to death meter information [6], the estimated road traffic injury-related death is the eighth leading cause of death worldwide. Improving road safety by introducing many effective and efficient applications, such as traffic accident risk prediction, and road anomaly detection is crucial while ensuring traffic flow efficiency and offering value-added services.

In ITS, vehicles exchange vast amounts of information with other vehicles, human sensors, and infrastructures such as roadside units, clouds, mobile networks, and satellites. Fig. (1) shows different communications in ITS. Most ITS applications utilize this data and the potentiality of Machine Learning (ML) to develop many ITS applications, such as traffic flow prediction, travel time prediction, and smart parking management. Utilizing 5G-

and 6G-enabled networks with ML techniques, the ITS could also improve communications, safety, and decision-making capabilities [7-9]. Over the years, researchers have leveraged machine learning as a powerful tool to develop and improve the effectiveness of various ITS applications. Some researchers have presented surveys on the use of ML and Deep Learning (DL) techniques for those applications [10-15]. However, existing studies in this field typically concentrate on a limited range of applications, including both outdated and recent proposals without discriminating based on their impact, and often overlook critical research gaps that warrant further investigation.

This survey considers most of the important applications covering all application areas, reviews recent proposals that have already gained significant attention from the research community, and discusses the existing issues as a guide for further research. Fig. (2) represents the methodology that has been followed to conduct this research.

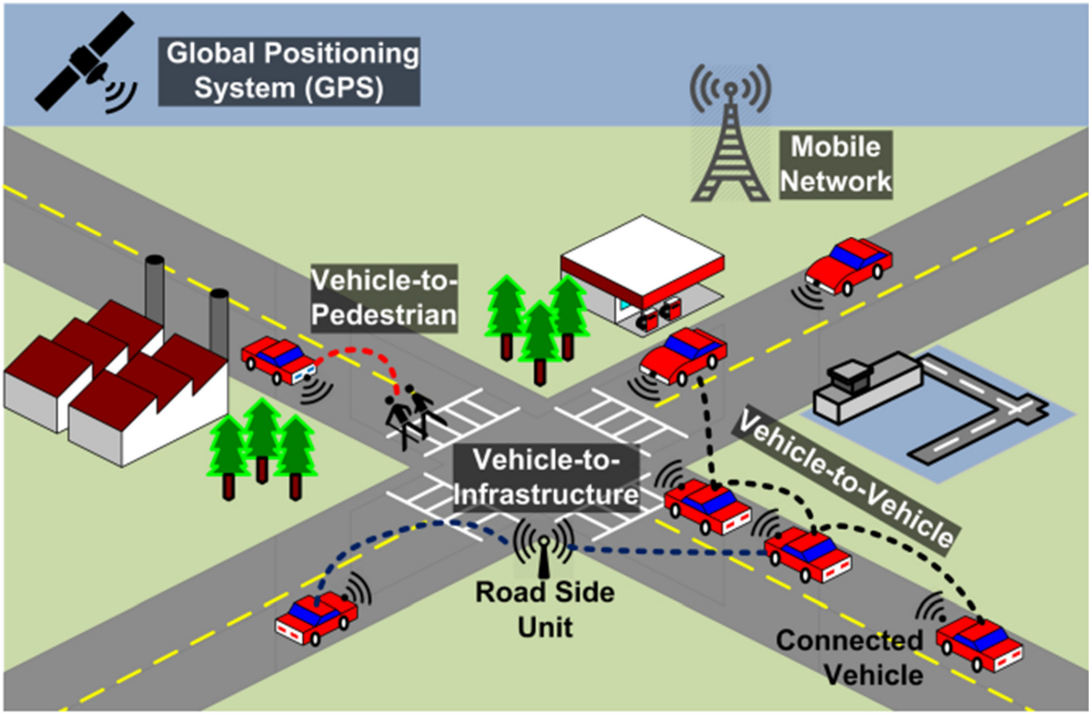


Fig. (1). Different communication systems [5].

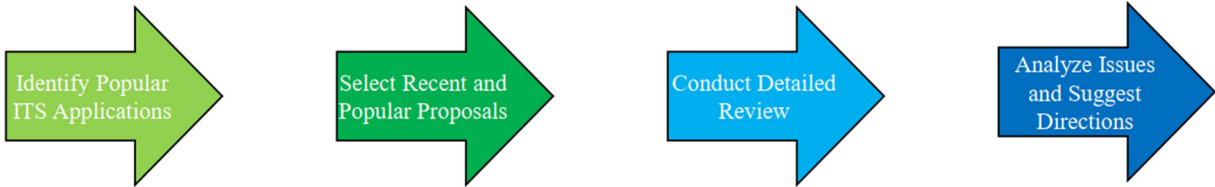


Fig. (2). Research methodology.

The main contributions of this review are as follows: it provides a comprehensive overview of ML, highlights several popular ITS applications, explores recent proposals for each selected application, particularly those leveraging ML techniques that have attracted significant attention of the research community, and after a detailed review, identifies and thoroughly discusses ten critical issues, suggesting directions for future research in the field.

The rest of the paper is organized as follows: section 2 provides an overview of ML with the basic concepts of some popular ML algorithms; section 3 discusses the popular applications and provides an overview of recent proposals that have gained significant interest; open research issues are discussed in section 4, and finally, section 5 concludes the paper.

2. MACHINE LEARNING BACKGROUND

ML is the study of algorithms and the backbone of AI. It enables machines to interpret and learn patterns in data, images, and sound for solving real-life problems. Generally, ML algorithms build models based on sample training data for future prediction and decision-making. Nowadays, ML application has been integrated into almost every sector, including transportation, agriculture, banking, bioinformatics, economics, marketing, computer networks, telecommunications, financial market analysis, medical diagnosis, and robotics. ML algorithms can broadly be classified into four categories: supervised learning, unsupervised learning, semisupervised learning, and reinforcement learning [16, 17].

2.1. Classification of ML Algorithms

2.1.1. Supervised Learning

A supervised learning algorithm develops a mathematical model based on labeled training data. The training data consist of many samples, each having one or more inputs and the desired output. After training the model, the output for new input data that were not present in the training set could be interpreted. Supervised learning algorithms are of two types, classification and regression. The classification algorithm generates discrete value outputs, while the regression algorithm generates continuous value outputs. For instance, housing price prediction is a regression problem, and medical image processing for disease prediction is a classification problem.

2.1.2. Unsupervised Learning

Unsupervised learning algorithms employ unlabeled datasets as test datasets to find meaningful correlations and inferences between data. Cluster-based unsupervised learning algorithms use the unlabeled dataset to draw grouping patterns in the dataset, and association-based algorithms identify the dependency of one data item on another data item. The latter is an effective algorithm to make a profit, especially in the marketing and business sectors. The hidden Markov model is a statistical Markov model that is assumed to be a Markov process X . It is presumed that there exists another process Y that depends on X where the goal is to learn X by observing Y .

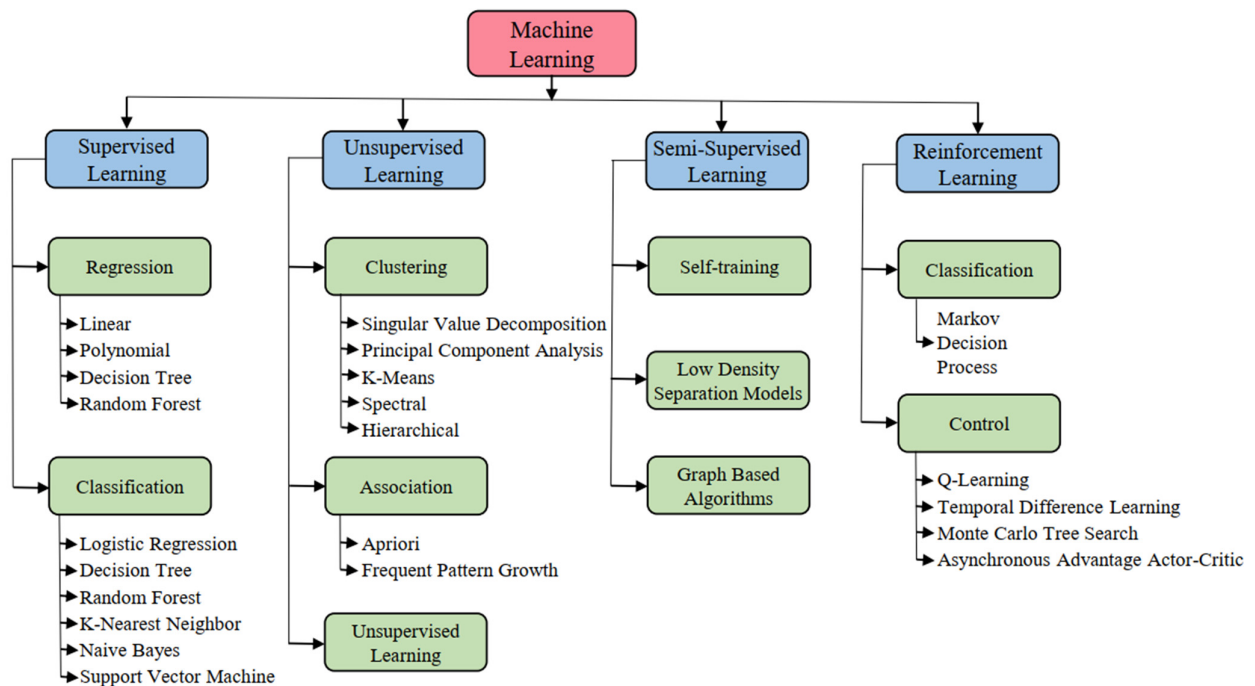


Fig. (3). Classification of machine learning algorithms.

2.1.3. Semi-supervised Learning

Semi-supervised learning lies in between supervised and unsupervised learning that deals with both labeled and unlabeled data. Supervised learning faces a problem in hand labeling the data, especially when the dataset is large. Unsupervised learning algorithm application is restrictive. Semi-supervised learning techniques address these issues by utilizing unsupervised techniques to group the comparative data. Then, the unlabeled data are labeled using the labeled data. Semi-supervised learning algorithms have been exploited in various business and entertainment applications [18].

2.1.4. Reinforcement Learning

Reinforcement Learning (RL) is a type of ML algorithm that is concerned with teaching intelligent agents from experience to take suitable actions in the application area in order to maximize the accumulative reward [19]. Markov decision process generates a mathematical framework that could decide cases. The results are partly random and partly depend on the decision maker. Control algorithms generally utilize value-based methods to inform the agent as to which actions it should take [20]. Gaming, driverless cars, recommender systems, and many other applications could exploit the concept of RL. Fig. (3) represents the detailed classification of ML algorithms.

2.1.5. Deep Learning

Deep Learning (DL) is a branch of ML that emerges from the concept of Neural Network (NN), where NN is a network of neurons. A biological NN consists of biological

neurons, whereas an ANN consists of artificial neurons for solving AI problems. NN incorporates cognitive science in machines. It is often termed deep learning as it uses artificial brain neurons to solve real-world problems.

Fig. (4) illustrates the basic architecture of an NN with a hidden layer [21, 22]. NN consists of an input layer, a hidden layer, and an output layer, where all inputs are modified by weight and summed together. A positive weight reflects reward, and a negative reflects penalty. An NN has multiple hidden layers to learn complex patterns from the data, called DL. Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-term Memory (LSTM), and Autoencoder (AE) are some popular DL algorithms.

2.2. Brief Description of Some Popular ML Algorithms

Many types of ML and DL algorithms exist. Some of the popular algorithms used in ITS applications are briefly discussed below.

2.2.1. Logistic Regression

Logistic Regression (LR) is a simple and efficient supervised ML algorithm for solving various classification and regression problems [23]. Fundamentally, LR is a regression algorithm. It uses a statistical method to predict the probability (between 0 and 1) of the dependent variable utilizing the linear combination of a set of independent variables using logistic functions. The output value of LR between 0 and 1 exhibits the probability of an occurring event. In LR, when a decision threshold is selected, it turns into a classification algorithm. The

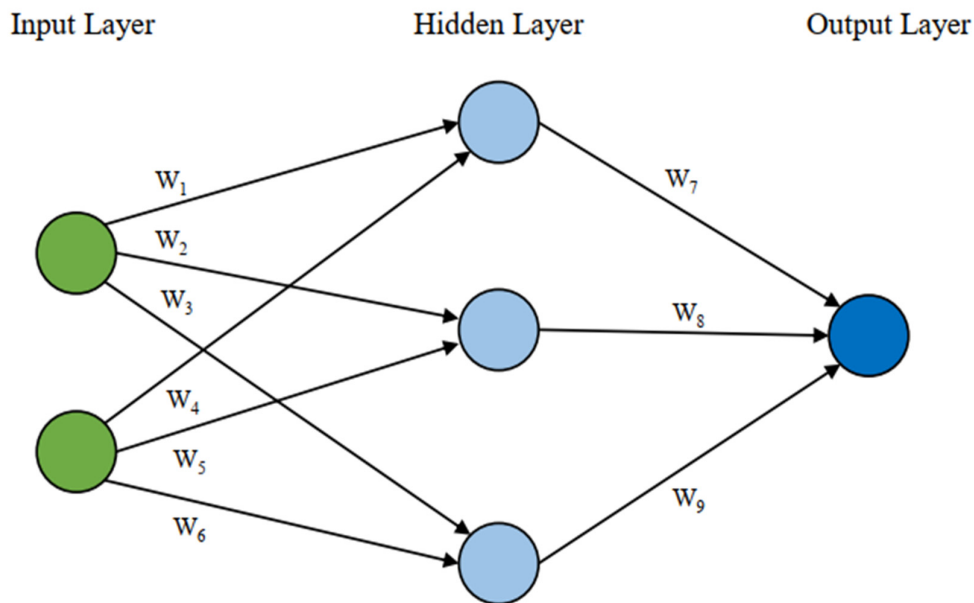


Fig. (4). Basic architecture of a neural network.

sigmoid function determines the maximum likelihood to determine the class. The LR classification could be binary (0 or 1; yes or no), multinomial (3 or more classes without ordering), and ordinal (3 or more classes with ordering). The LR equation for the binary classification problem is as follows:

$$P = \frac{1}{1 + e^{-(a+bX)}} \quad (1)$$

Where, P is the probability of the dependent variable (0 or 1), 'a' is the bias term, 'b' is the coefficient vector, and X is the vector independent variable (input features).

2.2.2. Decision Tree

Decision Tree (DT) [24] is a simple and widely used supervised ML algorithm that can also be used for classification and regression problems. It analyzes data in tree-like structures that have multiple levels of nodes. The topmost level is called the root node (usually the best feature), and the below levels are called child nodes. A node could be a parent node for its immediate below-level nodes and a child node for its immediate upper-level nodes. However, the lowest-level nodes that have no children are called leaf nodes. In DT, nodes represent the decision point for data characteristics, branches represent the possible outcome, and leaves represent the final class or decision value for the data sample. DT is constructed by recursively selecting the attribute with the highest information gain (using entropy) and splitting the dataset accordingly. This process continues until reaching leaf nodes, where the final classification or predicted value is assigned. For the evaluation outcome for a data sample (consisting of some attributes), based on the values of the respective attributes, DT branches from root to leaf nodes for the decision. In DT, the obtained results are easy to interpret.

2.2.3. Random Forest

Random Forest (RF) is also called an ensemble of decision trees where multiple DTs (forests) are used for predicting class (by voting) in classification problems and values in regression (by mean) problems. RF aggregates multiple decisions to form a single decision. Usually, more trees offer more accuracy. In the training stage, a set of trees (forest) is created for the training dataset. During the class prediction for a data sample, each DT of the forest forecasts the class label for that sample, and the majority forecast class is considered as the final decision for that data sample with the expectation of better accuracy [25].

2.2.4. Support Vector Machine

Support Vector Machine (SVM) is a very popular supervised machine learning algorithm used for classification and regression tasks. SVM effectively handles high dimensional data and is very useful when the data is not linearly separable. Thus, researchers use it in their machine-learning tasks. Even with the advancement of ML algorithms, SVM still continues to be one of the best

algorithms that can compete with the newer generation of supervised learning algorithms. Various domains, including image classification, text classification, and bioinformatics, use SVM to predict the result [26].

2.2.5. K-Nearest Neighbor

The K-Nearest Neighbor (KNN) is another supervised ML algorithm used for classification and regression problems. KNN works by calculating the distances (*e.g.*, Euclidian distance) between the new data point and all data points in the training set. Then, for the reclassification task, new data points are assigned to the most common class among the k-nearest neighbors, and for the regression task, the average value of k-nearest neighbors is calculated and that value is assigned to the new data point. KNN basically does not create the model, but memorizes the training dataset. KNN can also be used for handling missing values in data by calculating the similarity among different neighbors [27, 28].

2.2.6. Convolutional Neural Network

Convolutional Neural Network (CNN) [29] is a DL model mostly used to extract features from images or videos. The images are represented as grid-like structures, such as vectors. Its name came from its layer, the convolutional layer. This layer is used to apply filters to the input data to detect some special features within the data. Due to its working nature, CNN is very effective for image processing, object detection, facial recognition, optical character recognition, and other computer vision tasks. The other layers are pooling layers and fully connected layers.

2.2.7. You Only Look Once

You Only Look Once (YOLO) [30] is based on CNN architecture. YOLO's ability to simultaneously identify and locate multiple objects in images has made it a popular choice for recent research articles. YOLO does not use a two-step approach like other models, but it uses only one step to perform detection. YOLO divides the image into a grid and predicts each grid cell. Due to its popularity and efficiency, many researchers are actively working to improve the performance of YOLO. Thus, several versions of YOLO can be found, such as YOLOv1, YOLOv2, and YOLOv3. It is widely used in real-time object detection applications, including surveillance, autonomous vehicles, and image analysis.

2.2.8. Recurrent Neural Network

A Recurrent Neural Network (RNN) [31] is a type of neural network. It is designed for processing sequential data. RNN differs from other traditional feedforward neural networks in its recurrent loop, which works as a memory to the model. This memory distinguishes RNN from other non-memory-based neural networks. RNN is suitable for various tasks, like sequential data, NLP, speech recognition, and time series prediction. Apart from its effectiveness, RNN has been reported to show some inefficiencies, as it lacks long-term memory. These limitations motivate further development of other RNN-based neural networks, such as Gated Recurrent Unit (GRU) and LSTM.

2.2.9. Long Short-term Memory

Long Short-term Memory (LSTM) [31] is another version of RNN that has a memory cell and some gates, including input, forget, and output gates. These gates are used to regulate the flow of information through the cell. It is also used for determining as to which information to forget or which information to remember. Due to its ability to remember selectively, LSTM is mainly used in NLP, speech recognition, time series analysis, *etc.*

2.2.10. Autoencoder

Autoencoder is a popular unsupervised learning technique. It utilizes the neural network concept and performs encoding. It then reconstructs the input by decoding. For these tasks, it uses two components, the encoder and the decoder [32]. Fig. (5) shows a simple autoencoder with six input/output neurons and three hidden layers. The middle layer is also called the latent layer or bottleneck layer. The input layer takes unlabeled inputs. The encoder trains the inputs to generate the compressed knowledge of the input, often called “code”, which is stored in the latent layer. On the other hand, the decoder tries to reconstruct the original input from that code. The difference between the reconstructed input at the output layer and the original input is called loss or reconstruction error. The network can be trained to minimize this loss.

3. REVIEW ON ML-BASED ITS APPLICATIONS

Over the last few years, ITS has become safer and more efficient and has added comfort with the introduction of different ITS applications. Most of these applications utilize the strength of different ML and DL algorithms.

According to a previous work [5], ITS applications could be classified into four types: i) traffic management, which aims to improve the management of traffic flow for

efficient transportation and offering navigation service to drivers; ii) safety management, in which different measures are taken to minimize accidents that could harm vehicles, drivers, and pedestrians; iii) infotainment and comfort, which focuses on enhancing the drivers' experience by offering different value-added services; and iv) autonomous driving, which focuses on the automation of vehicle driving without human engagement. For effective review, we have selected 11 popular applications across four aforementioned types.

Based on the tasks performed, ITS applications could be classified as i) perception tasks that aim to detect, extract, and identify patterns to understand and provide services; ii) prediction tasks that utilize real-time or historical data to predict future states; and iii) management tasks that focus on providing guidelines to vehicles/drivers to achieve different goals, such as improving traffic flow, enhancing safety, and reducing environmental impact [33]. Table 1 classifies the selected applications into different types of tasks.

Many other applications also contribute to the development of intelligent and smart transportation systems. Some include smart traffic light systems, lane detection, obstacle warning, traffic sign detection, commercial vehicle administration, public transportation planning, infrastructure management, obstacle detection, blind spot information system, *etc.*

To emphasize the recent trends in ITS application research using ML/DL techniques, this study has collected related articles from the Google Scholar database published over the last 5 years (from 2019) and achieved high attention from the research community. A brief introduction of selected popular applications and a detailed overview of various proposals for those applications are presented below.

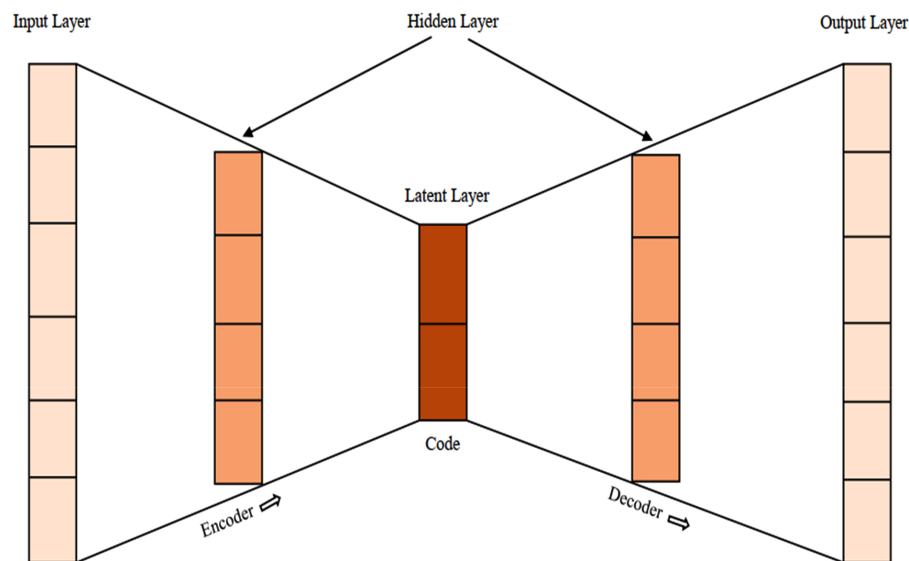


Fig. (5). An autoencoder with one input/output layer and three hidden layers.

Table 1. Different ITS application types along with their tasks.

SL.	Application type	ITS Application	Tasks
1	Traffic management	Traffic flow prediction	Prediction
2		Traffic congestion prediction	Prediction
3		Traffic speed prediction	Prediction
4		Travel time prediction	Prediction
5		Traffic signal control	Management
6		Smart parking management	Perception
7		Automatic tolling	Perception and management
8	Safety management	Traffic accident risk prediction	Prediction
9		Road anomaly detection	Perception
10	Infotainment and comfort	Remote vehicle Diagnostic and maintenance	Perception and prediction
11	Autonomous driving	Autonomous driving	Mixed

3.1. Traffic Management Applications

3.1.1. Traffic Flow Prediction

Traffic flow prediction plays a crucial role in estimating the number of vehicles passing through a specific road segment or region within a fixed time interval in the future (*e.g.*, 10 minutes, 20 minutes). It contributes to efficient transportation management and could support other applications, like navigation services, route planning, and traffic control. Traffic flow prediction is very crucial for successful ITS. Different ML and DL algorithms are used for traffic flow prediction [10, 34-36].

In 2019, a hybrid approach using GCN (Graph Convolutional Network) and LSTM model, where GCN was used to determine the spatial traffic flow relationship between different stations, was employed to extract the temporal features of traffic flow [37]. It also utilized attention mechanisms for final traffic flow prediction. Another approach to spatial-temporal feature-based traffic flow prediction has been presented in another study [38]. The Spatio-temporal Feature Selection Algorithm (STFSA) extracts features from actual traffic data and is then used by CNN for short-term traffic flow prediction. LSTM and Sparse Autoencoder (SAE)-based traffic flow prediction model was utilized in another work [39], employing feature engineering techniques, to extract important features and compressing the big traffic dataset before training the proposed LSTM-SAE hybrid model. Performance evaluation showed that the proposed model could achieve an average prediction accuracy of 97.7%.

In another study [40], before applying LSTM for traffic flow prediction, Chen *et al.* first applied different data denoising schemes to counter various unexpected interferences that hinder accurate traffic flow prediction. They then tested their approach on three different datasets and three denoising schemes. Finally, they concluded that LSTM + EEMD (Ensemble Empirical Model Decomposition) achieved the highest accuracy. Another article was carried out in 2021 [41] on short-term traffic flow prediction by combining 1-dimensional CNN (1DCNN), LSTM, and attention mechanism, hence called the DCNN-LSTM attention model. In this work, 1DCNN was used to extract spatial features from road traffic and

supply them to LSTM as input to generate time features and feed those as input to the regression prediction layer for prediction result calculation. The attention mechanism was integrated to give more attention to the important features and obtain a better model. The experiment revealed that the model performed better when giving attention to the weather factor.

3.1.2. Traffic Congestion Prediction

Traffic flow prediction and traffic congestion prediction may look the same, but they are not. While the former aims to predict the number of vehicles that would pass in a specific area/road segment in the near future, the latter goes a step further and aims to predict the occurrence of congestion that may slow down or gridlock a road considering the number of vehicles, road capacity, weather conditions, accidents, or other incidents. Traffic congestion prediction focuses on forecasting the traffic congestion conditions at the target location and time using different statistical and ML approaches so that the driver can make a plan for an efficient route, traffic management can allocate resources efficiently, and a plan can be implemented for improving transportation systems. Again, early prediction of traffic congestion helps traffic management systems to take early action to avoid or minimize the impact of traffic congestion [13, 42].

Zhang *et al.* proposed [43] an autoencoder-based Deep Congestion Prediction Network (DCPN) model in 2019 that learned the temporal correlation of transportation networks using the created Seattle Area Traffic Congestion Status (SATCS) datasets. The model has been found to outperform the existing models for predicting traffic congestion. A backpropagation-based traffic congestion point prediction model [44] aimed to enhance the traveler's comfort and make better transportation decisions. The time series modeling of the proposed approach achieved excellent results.

In 2020, an efficient and inexpensive data acquisition scheme was introduced [45]. Then, a hybrid approach by combining CNN, LSTM, and transpose CNN was proposed to extract spatial and temporal information from the collected input images and predict traffic congestion. The evaluation showed that the proposed approach

outperformed two existing Deep Neural Network (DNN) models. In 2021, an LSTM-based traffic congestion prediction approach was presented [46]. It was reported to use the vehicle speed data generated from traffic sensors between two sites. The model also predicted the congestion propagation within the 5-minute period and achieved 84-95% accuracy for different road layouts.

3.1.3. Traffic Speed Prediction

Traffic speed prediction aims to estimate the traffic speed at a certain time utilizing historical and real-time data collection and applying different statistical and machine learning approaches. It impacts drivers' route planning, analyses traffic and congestion conditions on the road, and facilitates planning by transportation planners to improve the transportation system. For example, Google Maps also use traffic speed prediction to estimate travel distance and suggest routes to drivers [10, 13, 47].

For traffic speed prediction, a work [48] used an autoencoder for spatial-temporal feature extraction from heterogeneous data sources. The features were then fused using the deep feature fusion method. Then, the traffic speed prediction model was developed using ANN, Support Vector Regression (SVR), regression tree, and KNN. Finally, the results were compared and it was concluded that the deep feature fusion method with SVR achieved the best result. In 2019, an LSTM-based traffic speed prediction model was proposed [49], where road networks were divided into critical paths, and for each path, multiple bidirectional LSTM layers were stacked together to capture spatial-temporal features fed to fully connected layers. The model for each path was then provided as an ensemble for traffic speed prediction over the network. The model also offered interpretability features by explaining the meaning of hidden features.

Sequential Graph Neural Network (SeqGNN) [50] combines a sequence-to-sequence model and GNN where road segments are considered as edges. The connectivity between road segments is considered as node, and both the input and output are represented as graph sequences. The proposed traffic speed prediction model was tested using real-world datasets. Bratsas *et al.* [51] compared the performance of different ML algorithms, such as RF, SVR, Multi-level Perceptron (MLP), and Multiple Linear Regression (MLR), using collected road network datasets in three different scenarios. Performance analysis showed that under stable conditions, SVR performed the best, and with large circumstance variations, the MLP model performed best. In 2023, an urban traffic speed prediction using the concepts of input data fusion (combining the traffic and weather dataset) and LSTM was presented [52]. The experiments showed that the proposed model using input data fusion outperformed the model using traffic data only.

3.1.4. Travel Time Prediction

Travel time prediction application estimates the travel time for traveling from one location to another. The prediction model usually uses historical data. However, it

could also consider the current traffic behavior to improve the prediction accuracy. So, a travel time prediction model could be developed utilizing the traffic speed prediction model and it could also estimate the travel time using historical travel time profiles as well. The application could improve user experience by efficiently planning the tour and optimizing traffic flow in the traffic management system [10, 13].

In 2019, Ran *et al.* [53] found that existing LSTM-based travel time prediction models do not use the departure time for model development. So, they proposed a tree-structured LSTM-based travel time prediction model that employed the departure time by integrating the attention mechanism concept with each output layer of the LSTM unit. AdaGrad method was used to train the proposed model using Highways England's provided dataset and the evaluation proved the effectiveness of the proposed approach compared to existing LSTM and other baseline models. Que *et al.* [54] proposed a travel time prediction approach using the GRU algorithm at the prediction layer. In this work, they divided the entire trajectory into multiple segments and utilized a large-scale real dataset, "Porto", as historical trajectory data. Then, they extracted the trajectory data for the spatial road segment and set the start time of each segment of the trajectory. The work also utilized the velocity feature to represent the adjacent segment structures and road network topology. Performance analysis proved the proposed method to have a Mean Absolute Percentage Error (MAPE) of only 0.070% and perform well compared to the existing methods, like Stacked Autoencoder (SAE)-based approach.

Petersen *et al.* introduced a bus travel time prediction model using convolutional LSTM [55]. They utilized the non-static spatial-temporal correlation in the urban bus network, and experiments were performed using Copenhagen's public transport dataset, Movia, having 1.2 million travel time observations. The performance of the proposed approach was compared with the baseline model, including the Google traffic model for different times of day. In 2020, He *et al.* introduced an approach [56] for bus travel time prediction. The method consisted of training and prediction stages. In the training stage, bus lines were portioned into segments, and for each segment, travel time patterns were extracted from real-world bus travel data that involved 30 bus services. After that, the data for each segment was considered as a cluster, and for each cluster, the LSTM model was trained to calculate the bus riding time. Then, the time of each segment (between a source to destination) was predicted and combined with the waiting time (using historical bus arrival timing records) to estimate the bus travel time. However, the approach could be further improved by implementing a model to predict the waiting time as well. A multistage integrated feature learning-based approach using deep learning for travel time prediction was also studied [57]. It incorporated external weather and the fastest route data for extensive feature engineering, and K-means algorithms were applied to boost the feature

space. Moreover, a deep-stacked autoencoder technique was utilized to represent features in a lower dimension to decrease overfitting and increase performance. Then, a deep MLP was trained to predict travel time. The findings demonstrated that the model outperformed with a Mean Absolute Error (MAE) of approximately 200 seconds and showed the capability to capture general traffic dynamics, although it may fail under significant rare events, such as heavy snow.

3.1.5. Traffic Signal Control

Traffic signal control, also called traffic light control, directs to optimize the traffic flow at intersections so that the vehicle has to stay short of time at the intersections. Traffic signal control applications could be pre-timed or actuated, where the pre-timed traffic signal control could use a fixed timing plan and actuated signal control could use real-time sensor data to adjust the timing plan with the cost of complexity. Proper implementation of traffic signal control could contribute to safe and efficient traffic flow, while minimizing congestion [10].

In 2019, Genders and Razavi [58] proposed an N-step asynchronous Q-learning algorithm with two hidden layer ANN reinforcement learning approaches for the agent to develop an adaptive traffic signal control. The effectiveness of agent performance was tested using the developed dynamic and stochastic rush hour simulation and it was proven that the proposed approach performed better than linear Q-learning, and the traditional loop detector actuated traffic signal control method with the reduction of mean total delay by 40%. However, compared to the traditional actuated approach, the proposed technique exhibited some extra delays for left-turning vehicles. A fully scalable and robust decentralized Multi-agent RL (MARL) approach using Advantage Actor Critic (A2C) was proposed [59] to overcome the scalability issue of centralized RL for adaptive traffic signal control. The effectiveness of the proposed approach was improved by reducing the local agents' learning difficulty and increasing observability. Performance analysis performed using both synthetic traffic grid and real word traffic network of Monaco city proved the effectiveness of the approach against both independent Q-learning and Advantage Actor Critic (A2C) approach.

In 2020, Joo *et al.* [60] presented a Q-learning-based Traffic Signal Control (TSC) system to maximize the number of vehicles crossing the intersection by considering the standard deviation of queue length and throughput as the main parameters. Performance analysis in a 4-way intersection proved that the proposed method encountered shorter queue length and waiting time than other Q-learning-based TSC systems. The strength of the work was that it could adapt to different intersection structures. However, it could improve further by sharing information among nearby intersections instead of using information from one intersection. A value-based meta RL approach named MetaLight has also been presented earlier [61]. It eliminated the need for learning from scratch that most existing approaches demand, and hence

sped up the learning process in new scenarios by utilizing the learned knowledge from existing scenarios. Moreover, the proposed approach improved the performance of the existing RL model FRAP by optimizing the model structure and improving the updating scheme. The proposed approach used four real-world datasets to prove its effectiveness in quick adaptation in new scenarios with consistent performance.

A deep RL-based traffic signal control system for large-scale networks to overcome the problem of multi-intersection traffic signal control was proposed by Chen *et al.* in 2020 [62]. The work tackled existing challenges, like scalability and signal coordination. The model proposed was evaluated in a large-scale real-world scenario with 2510 traffic lights in Manhattan, New York City. However, the performance could be further improved by adopting a more appropriate design of coordination and cooperation among neighboring intersections.

3.1.6. Smart Parking Management

Finding a parking spot in many locations is a tedious and time-consuming task. Smart parking management systems contribute to the development of smart cities. The application allows vehicle users to find free parking spots effectively and optimize parking space usage. It also allows users to reserve the available parking spot in advance, which saves the available parking spot searching time and car emissions. In addition, the system also contributes to minimizing drivers' stress, fuel consumption, and delay, while finding a parking space and improving road traffic conditions as well [11, 14, 63].

In 2020, Saharan *et al.* proposed an ML-based approach [64] to predict parking occupancy in Seattle city using street parking data. The occupancy-driven ML approach was utilized to predict the parking pricing for the next vehicles. It benefits both the parking authority and those parking based on the parking demands. In a high occupancy situation, it favors parking owners, and in a low occupancy scenario, it favors the individuals parking. The model has been tested using four ML algorithms, including Linear (LIN), DT, NN, and RF, where RF has been found to perform best with 99.01% accuracy. Ali *et al.* proposed a smart parking prediction system using a deep LSTM algorithm and integrated the concepts of IoT, cloud computing, and sensor networks [65]. They implemented the model using the Birmingham parking sensors dataset. Three experiments were performed considering the location, days of the week, and hours of the day, proving its effectiveness compared to the state-of-the-art methods.

Awan *et al.* [66] performed a comparative study on parking space availability prediction using different ML/DL algorithms, like MLP, KNN, DT, RF, and Santanders' parking dataset. The model could recommend the top K parking spots to those parking based on the distance between the cars' current position and parking spots. After performance analysis, they concluded simple algorithms, like DT, RF, and KNN, to outperform complex algorithms, like MLP, for prediction accuracy.

In 2022, the Car Parking Space Prediction (CPSP) scheme was introduced [67] using Deep Extreme Learning Machine (DELm) techniques. The proposed approach achieved the highest precision rate of 91.25%. However, it only achieved 60% training and 40% testing and validation accuracy. Another car parking space prediction approach was presented in another study [68]. Here, the prediction model was optimized by integrating the concept of IoT and an ensemble-based prediction model. The Birmingham parking dataset was used to test the model, and Bagging Regression (BR) was used as the base model for the ensemble approach. Performance evaluation showed the proposed scheme to have only 0.06% Mean Absolute Error (MAE) and outperform the existing approach by over 6.6%, but with less complexity.

3.1.7. Automatic Tolling

Automatic tolling applications allow vehicle users to pay the toll electronically. It is also called the Electronic Toll Collection (ETC) system. In ETC, drivers could pay the toll without stopping at the toll collection point, such as toll roads, bridges, and tunnels, but the vehicles are charged by being identified using Radio-Frequency Identification (RFID) tags or license plate recognition techniques. The application has been found to contribute to reducing congestion and cost, while increasing safety and efficiency [69].

In 2019, a Multitask License Plate detection and Recognition (MTLPR) approach using CNN with the aim of better accuracy and less computational complexity was proposed [70]. The approach was tested on the Chinese City Parking Dataset (CCPD); it initially detected the license plate and then recognized the information on the license plate using Convolutional Recurrent Neural Network (CRNN) and Connectionist Temporal Classification (CTC)-based models. The experiment results showed that the lightweight model outperformed other methods in terms of detection and recognition speed and achieved up to 98% precision.

Tourani *et al.* presented an automated and robust model for Iranian license plate detection and recognition techniques using the YOLOv3 technique [71]. The method involved License Plate Detection (LPD) and Character Recognition (CR), serving as a unified application to be implemented in real time with high accuracy. For model development and testing, image data (including both color and grayscale images) were collected from the installed surveillance system for different distances, shooting angles, brightness, and resolutions. The experiment results showed that the model could achieve 95.05% accuracy and perform the task within 119.73 milliseconds, proving its effectiveness for real-world applications.

In 2021, a study involved the development of an Automatic Number Plate Recognition (ANPR) system using DL techniques [72]. The system was trained using the YOLO object detection algorithm and the ImageAI framework to optimize and improve the efficiency of ANPR. It consisted of vehicle detection, license plate localization, and character recognition. It has been trained

using NVIDIA Jetson Nano kit and Python software for car detection using the Stanford car dataset. Then, number plate localization and optical character recognition were performed. The result analysis showed that the proposed DL model achieved an accuracy of 98.5% for car detection, 97% for number plate localization, and 96.7% for optical character recognition. However, in bright illuminated environments, the YOLO object detection algorithm performed poorly than the ImageAI framework.

In 2021, Laroca *et al.* [73] proposed an Automatic License Plate Recognition (ALPR) system using YOLO-based CNN models at all stages using eight publicly available datasets from five different regions of the world, reflecting its robustness in various conditions. They manually performed annotation by labeling 38351 bounding boxes on 6239 images for the vehicle's position, License Plates (LPs), and characters across all datasets where the annotation was missing. The approach consisted of stages, like vehicle detection, LP detection and layout classification, and LP recognition. For the LP detection and layout classification stage, a unified approach by combining LP detection and layout classification was used. The integration of layout classifications, such as American, Brazilian, and Taiwanese, significantly improved the LP recognition results. It achieved an average end-to-end LP recognition accuracy of 96.9% by considering eight public datasets from five different regions under various conditions that performed better than most existing works. The approach performed better than commercial systems using four datasets, and the results were comparable for other datasets. The approach achieved an impressive number of frames processing per second even if there were four vehicles in a frame, proving its ability to perform well in real-time applications. Table 2 summarizes the popular traffic management application proposals.

3.2. Safety Management Applications

3.2.1. Traffic Accident Risk Prediction

In 2023, about 1.26 million people died of road injuries worldwide [6]. The traffic accident risk prediction model estimates the likelihood of traffic accidents that could happen in different road segments and times. Such prediction is essential for taking preventive measures to minimize the risk and save lives. The prediction system can notify about the risk to the drivers so that they can concentrate on the road, avoid mistakes, and minimize the risk of accidents [34].

Theofilatos *et al.* performed a study in 2019 [74] to compare the performance of ML and DL-based approaches for real-time crash occurrence prediction in different weather conditions. For the experiment, they chose KNN, NB, DT, and SVM for the ML model, and shallow NN (with 1 layer), and DNN (with 4 layers) for the DL model. The prediction models were developed using traffic and weather data (generated from the Attica Tollway in Greece). The performance of the models was compared in terms of prediction accuracy, sensitivity (true positive rate), specificity (true negative rate), and AUC. Among all

models, NB and DT achieved the highest accuracy (72.15%), but their achieved sensitivity score was low (between 0.225 and 0.324). Considering the scores of all metrics, the DNN model outperformed the other models with an accuracy of 68.95%, sensitivity of 0.521, specificity of 0.77, and AUC of 0.641. The study revealed the potentiality of DNN for traffic accident risk prediction.

In 2020, Shi *et al.* [75] presented a model for Autonomous Vehicles (AVs) risk prediction with the addition of behavior assessment utilizing the end-to-end Automated ML (AutoML) method. The AutoML framework integrates three main components, like unsupervised risk identification, feature learning (using XGBoost), and model auto-tuning (by Bayesian optimization). The proposed AutoML approach could distinguish safe-risk conditions

with 95% accuracy. Overall, the study has demonstrated the effectiveness of AutoML for AV risk prediction. A Rear-end Collision Prediction Mechanism (RCPM) was introduced by Wang *et al.* [76]. The model used the CNN and Next Generation Simulation (NGSIM) trajectory dataset for model development to predict the rear-end collisions in real-time. The class imbalance issue of the dataset was handled by genetic theory-based approaches. The proposed model's performance was compared to Honda, Berkeley, and MLP NN-based algorithms. It outperformed the existing models for accuracy, precision, recall, and ROC. Performance analysis proved that RCPM could provide early warning of rear-end collisions with less than 0.1 s average prediction delay, making it suitable for real-time applications.

Table 2. An overview of traffic management ITS applications.

Ref.	Year	Algorithm	Contributions	Application
[37]	2019	GCN, LSTM	A hybrid approach combining GCN and LSTM with an attention mechanism to accurately predict traffic flow by considering both spatial and temporal features.	Traffic flow prediction
[38]	2019	CNN	An STFS-CNN model, where STFS extracts spatio-temporal features from actual traffic data and CNN realizes short-term traffic flow prediction.	
[39]	2020	LSTM, SAE	LSTM and SAE-based traffic flow prediction models achieved an average prediction accuracy of 97.7%.	
[40]	2021	LSTM	Applying EEMD for data denoising before using LSTM for traffic flow prediction achieved the highest accuracy across the three datasets.	
[41]	2021	CNN, LSTM	A combination of 1D CNN, LSTM, and an attention mechanism to extract spatial and temporal features for short-term traffic flow prediction achieved improved performance when considering weather factors.	
[43]	2019	Autoencoder	A novel AE-based DCPN model trained on a custom SATCS dataset achieved superior performance compared to existing models.	Traffic congestion prediction
[44]	2019	Backpropagation	A backpropagation-based traffic congestion point prediction model with time series modeling demonstrated excellent performance.	
[45]	2020	CNN, LSTM	An efficient data scheme with a hybrid CNN-LSTM-Transpose CNN model outperformed DNNs for traffic congestion prediction by exploiting spatial and temporal information from images.	
[46]	2021	LSTM	An LSTM-based model that utilized traffic sensor data to predict congestion propagation within 5 minutes with 84-95% accuracy for diverse road layouts demonstrated the effectiveness on different road types.	
[48]	2019	Autoencoder, ANN, SVR, Regression Tree, KNN	An autoencoder combined with deep feature fusion and SVR outperformed ANN, regression tree, and KNN for traffic speed prediction using heterogeneous data sources.	Traffic speed prediction
[49]	2019	LSTM	An interpretable LSTM model predicted traffic speed over a network by capturing spatial-temporal features for each critical path and interpreting hidden features.	
[50]	2020	GNN	Leveraged GNNs and sequence-to-sequence learning to predict traffic speed by representing road segments as edges and connectivity as nodes, achieving effectiveness on real-world data.	
[51]	2020	RF, SVR, MLP, MLR	Compared RF, SVR, MLP, and MLR for traffic speed prediction reveal SVR's dominance in stable conditions and MLP's superiority under high variability.	
[52]	2023	LSTM	Combining traffic and weather data via input data fusion and LSTM for urban traffic speed prediction proves more accurate than using traffic data alone.	
[53]	2019	LSTM	Addressed the limitation of existing LSTM models for travel time prediction and improved by incorporating departure time through a tree structure with an attention mechanism, achieving superior performance.	Travel Time Prediction
[54]	2019	GRU	Leveraged GRU with trajectory segmenting and feature extraction to achieve low MAPE (0.070%) for travel time prediction, outperforming SAE-based approaches.	
[55]	2019	Conv-LSTM	Bus travel time prediction model utilizing non-static spatial-temporal correlation in an urban bus network that outperformed existing models, including Google traffic model in different times of day.	
[56]	2020	LSTM	Predicts bus journey travel time by combining segment-based LSTM models and historical arrival times, but could further improve by predicting waiting time.	
[57]	2020	AE, Deep MLP	Combining extensive feature engineering and DL, the multistage approach achieves high travel time prediction accuracy (MAE 200 seconds) but struggles under rare events like heavy snow.	

(Table 2) contd....

Ref.	Year	Algorithm	Contributions	Application
[58]	2019	Q-learning	Utilizing n-step Q-learning achieved a 40% reduction in total delay compared to traditional traffic signal control methods, but the delay is slightly high for left-turning vehicles.	Traffic Signal Control
[59]	2019	A2C	A fully scalable, robust MARL approach using A2C overcomes the scalability issue of centralized RL for adaptive traffic signal control, outperforming independent Q-learning and A2C approaches.	
[60]	2020	Q-learning	Aims to maximize the number of vehicles passing in intersections. It is flexible to different intersection structures.	
[61]	2020	Meta RL	Eliminates the need for scratch learning in new scenarios by leveraging knowledge from existing scenarios, improving performance and adaptation speed compared to existing RL models like FRAP.	
[62]	2020	Deep RL	A deep RL-based traffic signal control system for large-scale networks, tackling scalability and coordination challenges, but leaving room for improved interintersection cooperation.	
[64]	2020	LIN, DT, NN, RF	Predict parking occupancy and dynamically adjust parking prices based on demand, achieving 99.01% accuracy using RF.	Smart Parking Management
[65]	2020	LSTM	Employs deep LSTM, IoT, cloud computing, and sensor networks for parking space availability prediction using Birmingham parking sensor data, achieving accurate results based on location, day of the week, and time of day.	
[66]	2020	MLP, KNN, DT, RF, Ensemble	Analyzed the performance of different ML/DL models for parking space availability prediction and revealed that shallow algorithms DT, RF, and KNN performed better. It can also recommend the top k available parking spaces.	
[67]	2022	DELM	A CPSP scheme using DELM achieves high precision (91.25%) but suffers from low training (60%) and testing/validation (40%) accuracy.	
[68]	2022	BR	An ensemble prediction approach with IoT integration achieves low MAE (0.06%) for parking space prediction, outperforming existing methods by 6.6% while remaining simpler.	
[70]	2019	CRNN, CTC	A lightweight MTLPR approach using CNN, achieving high accuracy (98%) and outperforming other methods in detection and recognition speed.	Automatic Tolling
[71]	2020	YOLOv3	An automated YOLOv3-based approach for Iranian LP detection and recognition, achieves 95.05% accuracy and real-time performance (119.73ms), proving the effectiveness for real-time applications.	
[72]	2021	YOLO	DL-based ANPR system achieves 97% car, 98% plate, and 90% character recognition, showcasing the potential for traffic management.	
[73]	2021	YOLOv3	YOLO-based ALPR system, achieved >95% recognition rate, surpassing commercial systems and excelling with fixed LP layouts like Brazil and China with 2.1% and 3.6% average improvements.	

Huang *et al.* [77] proposed a crash detection, prediction, and crash risk estimation approach using CNN techniques. They utilized real-world volume, speed, and sensor occupancy datasets. For crash detection, CNN outperformed shallow ML models, like LR, DT, RF, SVM, and KNN. However, for crash prediction, CNN performance was found to be comparable to the shallow algorithms. For crash risk estimation, they used 1-minute, 5-minute, and 10-minute prior data of the crash incident and observed it as hard to estimate the crash risk with 10-minute prior data.

3.2.2. Road Anomaly Detection

Road anomalies, like bumps, potholes, and cracks, could lead to serious consequences, such as vehicle damage, delay, traffic congestion, and even accidents. So, detecting such anomalies and notifying the drivers could contribute to avoiding such potential damage. Road anomaly detection application meets the requirement by monitoring the road surface condition and notifying the drivers when such anomaly is detected. However, road anomaly detection often becomes challenging due to the different and random shapes of potholes, cracks, *etc* [11, 14, 78].

Basavaraju *et al.* [79] proposed an ML approach to assess road surface anomalies using smartphone sensors in 2019. The authors analyzed different supervised machine learning techniques, including SVM, DT, and MLP, to classify road surface conditions using accelerometers, gyroscopes, and GPS data collected from smart-

phones. They compared the performance of models trained with features from all three axes of the sensors to models that have used features from only one axis. The model trained with data from all three axes outperformed the model trained with data from only one axis. To classify the road surface conditions, the authors also applied deep NN (MLP) with and without manual feature extraction that could also classify road conditions effectively. Finally, the proposed model could monitor the road condition for defect classification, present the risk factors to commuters, and inform the respective authorities for maintenance. Varona *et al.* introduced a DL-based approach for road surface monitoring and pothole detection [80]. They discussed the impact of road anomalies on vehicle maintenance and driver safety. Data segmentation and augmentation play a crucial role in increasing the model accuracy. The proposed approach has utilized different DL algorithms, like CNN, LSTM, and reservoir computing, to identify various road surfaces automatically and distinguish different potholes. Overall, CNN performed best for stability event identification and road surface classification. In the experiment, real-world information about routine travels in Tandil, Argentina, was used.

In 2021, Aboah *et al.* [81] proposed a vision-based traffic anomaly detection system using DL (YOLOV5) and DT. In this approach, YOLOV5 served as the backbone for the detection model. It extracted candidate anomalies by video sorting and analyzing traffic camera videos through different steps, like background estimation, road mask

Table 3. A review on safety management in ITS applications.

Ref.	Year	Algorithm	Contributions	Application
[74]	2019	KNN, NB, DT, RF, SVM, DNN	For real-time crash prediction, ML and DL models were compared using the Attica Tollway in Greece dataset. DL model outperformed all, achieving 68.95% accuracy, 0.521 sensitivity, 0.77 specificity, and 0.641 AUC.	Traffic accident risk prediction
[75]	2020	XGBoost	The autoML-based method achieved 91.7% accuracy for risk level prediction and 95% for safe-risk distinction.	
[76]	2020	CNN	CNN-based RCPM was employed for real-time rear-end collision prediction, outperforming existing algorithms in accuracy and recall. Using synthetic oversampling to address data imbalance, RCPM delivered early warnings with <0.1s delay, making it suitable for real-time ITS.	
[77]	2020	CNN, LR, DT, RF, SVM, KNN	The CNN-based approach was employed for crash detection, prediction, and risk estimation. While CNN outperformed shallow models for detection, its prediction performance was comparable. Predicting crash risk with 10-minute prior data was difficult, but using 1-minute and 5-minute data showed promise.	
[79]	2019	SVM, DT, MLP	ML approach using smartphone sensors achieved superior performance with features from all axes of the accelerometer, gyroscope, and GPS data compared to using single axes.	Road anomaly detection
[80]	2020	CNN, LSTM	CNN-based methods were used to classify different road surfaces and distinguish potholes with promising accuracy, highlighting the importance of data augmentation and segmentation.	
[81]	2021	YOLOv5, DT	The model achieved promising traffic anomaly detection with an F1 score of 0.8571 and S4 score of 0.568, but struggled with distant anomalies and small objects.	
[82]	2021	Dilated CNN	A dilated CNN network was employed for smart pothole detection, which surpassed traditional CNNs in accuracy and computation cost, offering better resolution, training results, and reduced false positives/negatives.	

generation, and adaptive thresholding. After that, candidate anomalies were passed through DT to detect and analyze the anomalies. The model could also estimate the start and end times of the anomalies. It achieved an F1 score of 0.8571 and an S4 score of 0.5686, indicating its effectiveness in finding anomalies in most videos. However, the approach could detect anomalies near the camera easily, but struggled to detect them in locations being far from the camera or objects being too small. Later, a combination of conditions was simulated, and robust algorithms and an IOU tracker were employed to increase annotation database with long-distance objects and increase the detection accuracy for distance anomalies from the camera.

A work [82] employed a dilated CNN for smart pothole detection. The traditional CNN algorithms convolved and pooled the image to increase the receptive field and reduce computation, but caused a loss of resolution. Later, a combination of the proposed MVGG16 network removed some convolution layer of VGG16 and served as the backbone for faster RCNN using pothole images. It provided better precision and low inference time. The proposed model outperformed the existing CNN-based YoLoV5 models using the ResNet101 backbone and it was able to balance between the pothole detection accuracy and detection speed. A summary of works focusing on the safety

management of ITS applications is presented in Table 3.

3.3. Infotainment and Comfort

3.3.1. Remote Vehicle Diagnostic and Maintenance

Remote vehicle diagnosis focuses on detecting provable vehicle failure or future issues of different subsystems, like fuel, exhaust, cooling, and ignition, by sending sensor data from the vehicle to the server, processing the data to predict or identify faults, and sending notifications to drivers for vehicle maintenance. It also tried to fix the issues without requiring the vehicle to go to the repair shop. It contributed to avoiding frequent maintenance by diagnosing and fixing the faults early [83, 84]. In this way, the application contributed to infotainment and comfort by enhancing the overall driving experience, convenience, and peace of mind for vehicle owners.

In 2020, Gong *et al.* [85] proposed the implementation of ML-based fault classification of in-vehicle Power Transmission Systems (VPTS). The experiment covered different ML algorithms, like DT, SVM, KNN, MLP, and DNN. To improve the efficiency of the ML algorithms, dimensionality reduction was utilized using PCA. For the experiment, a variety of fault signals under 15 fault conditions were collected. Mel-scale Frequency Cepstral Coefficient (MFCC) was used to extract the original signal characteristics from the collected signals. Then, a variety

Table 4. An overview of infotainment and comfort ITS applications.

Ref.	Year	Algorithm	Contributions	Application
[85]	2020	DT, SVM, KNN, MLP, DNN	ML-based approach for VPTS fault diagnosis and classification. PCA-based dimension reduction improved the efficiency of MFCC for extracting signal characteristics. The model proved its effectiveness for VPTS fault diagnosis.	Remote vehicle detection and maintenance
[86]	2022	LR, RF, XGB, CatBoost, GNN, RoBERTa	Combining transformer pretraining with GNNs achieved superior accuracy in CloV software escalation prediction, showcasing the effectiveness of DL and graph modeling integration.	

of fault characteristics were used with different ML algorithms to automatically diagnose the faults and classify them. Experiments were implemented using Matlab and Python. Overall, SVM, MLP, and DNN performed best in the Python environment.

In 2022, an approach called Software Escalation Prediction (SEP) was proposed [86] that aimed to predict software escalation in the Cognitive IoV (CIoV) using DL techniques. The model combined pretraining mechanisms of transformers with software upgrade-related events to dynamically model software sequence activities. It also used Graph Neural Networks (GNNs). The GNN could capture the complex life activity rule of software and integrate the characteristics of the software. For the experiment, a software activity log dataset was used, and the proposed SEP performance was compared with LR, RF, XGBoost, Catboost, GNN, and RoBERTa. The proposed SEP outperformed all models, including RoBERTa methods with a 6-8% improvement. Some effective solutions for infotainment and comfort ITS applications are presented in Table 4.

3.4. Autonomous Driving

Autonomous driving or self-driving is the advancement of ITS and it has the potential to revolutionize transportation systems. Autonomous driving aims to improve safe and effective automatic driving by analyzing the environment and making effective decisions and control strategies. The main challenge of autonomous driving is to detect and interpret the surroundings, such as pedestrians, other vehicles, road signs, traffic lights, and road conditions, and take action accordingly. It uses a combination of different technologies, including ML, AI, sensors, cameras, and radar, making it able to move from one location to another safely considering the surroundings without or with limited human involvement [11, 13, 87, 88].

In 2019 [89], an automated lane-changing approach for making high-level decisions in dynamic and uncertain traffic scenarios was presented. For the experiment, a simulation environment was created by emulating various challenges, such as uncertainty in driver's behaviors, and considering the trade-off between safety and agility. The deep RL-based agent performed significantly better compared to a heuristic-based safe lane change algorithm called MOBIL. It also contributed to minimizing the estimated arrival time. Overall, the proposed approach performed consistently in a variety of uncertain and noisy traffic environments.

Wei *et al.* [90] introduced an improvised version of CNN-based visual object detection in Advanced Driving Assistance Systems (ADAS) using deconvolution and fusion of CNN feature maps at the first stage to handle large object scale variations. Then, soft Non-maximal Suppression (NMS) was applied to address the challenge of object occlusion. Finally, the authors set anchor boxes based on aspect ratio statistics for better object matching

and localization. Experiments performed using the KITTI dataset demonstrated the effectiveness of the proposed approach, with good detection performance improvement over the baseline MS-CNN model. Chen *et al.* [91] designed a specific input representation using a bird-view image to reduce the complexity of the problem and used visual encoding to capture low-dimensional latent states. They implemented it to

capture low-dimensional latent states. They implemented three state-of-the-art model-free deep RL algorithms (DDQN, TD3, and SAC) in their framework and applied them to a challenging roundabout task in a driving simulator. The results showed their method to be significantly better than the baseline approach in both scenarios, without and with dense surrounding vehicles. Overall, the SAC algorithm exhibited the best performance, achieving high success rates in entering and passing through the roundabout.

In 2020, an intelligent path planning scheme for autonomous vehicle platoons using deep reinforcement learning on the network edge was proposed [92] to improve the driving efficiency and fuel consumption of autonomous vehicular platoons. First, a system model was developed for the platoon, and then a joint optimization problem was formulated considering the task deadline and fuel consumption. A path determination strategy was then designed using deep reinforcement learning. The evaluation of the proposed greedy algorithm combined with the Q-learning approach through simulation showed that the proposed approach could significantly reduce the fuel consumption of the vehicle platoon, while ensuring the task deadlines. However, further research needs to evaluate the approach in an unknown dynamic environment.

Autonomous braking is very crucial for autonomous vehicles. Fu *et al.* presented [93] a deep RL-based autonomous braking approach through the selection of a precise decision-making strategy for ensuring driving safety in an emergency. The proposed approach focused on three factors: i) effectively learning the driving strategy through detailed analysis of lane changing and braking process, ii) determining the optimal strategy for autonomous braking by analyzing different brake moments, degree of accidents, and passengers' comfort using deep RL, and iii) improving the efficiency of the optimal strategy; for continuous control tasks, Deterministic Policy Gradient (DPG), an Actor-critic (AC) algorithm, has been reported to be utilized. Through simulations, the effectiveness of the proposed autonomous braking decision-making strategy was analyzed and validated in terms of learning effectiveness, decision-making accuracy, and driving safety. Table 5 summarizes the different proposals for autonomous driving applications.

4. OPEN RESEARCH ISSUES

There exist a number of issues that need further research for effective solutions. Some important ones are discussed below.

Table 5. An overview of autonomous driving applications.

Ref.	Year	Algorithm	Contributions	Application
[89]	2019	Deep RL	A deep RL-based approach performed significantly better in noisy highway environments compared to heuristic methods, making safe lane changes while minimizing the estimated time of arrival.	Autonomous driving
[90]	2019	CNN	The approach improved CNN-based object detection for ADAS with 1) deconvolution and feature fusion for large object scales, 2) soft non-maximal suppression for occlusion, and 3) aspect ratio-based anchor boxes, achieving performance gains over baseline MS-CNN on the KITTI dataset.	
[91]	2019	DDQN, TD3, SAC	A bird's-eye view input with visual encoding for roundabout navigation in driving simulators achieved superior performance with the SAC algorithm compared to baseline, even in dense traffic.	
[92]	2020	Deep RL	Deep RL-based path planning for autonomous vehicle platoons reduced fuel consumption.	
[93]	2020	DDPG	This approach employed a multiobjective reward function and DDPG algorithm, leading to improved driving safety and effective and accurate decision-making in emergency situations.	

4.1. Lack of Publicly Available Benchmark Datasets

Most existing works rely on simulated traffic datasets and small road segments of an area or city often impact ML models' performance as it might not work well for different roads or areas. Moreover, models trained using synthetic datasets often perform poorly in real environments. So, more focus is needed to create publicly available datasets covering various traffic scenarios in diverse geographic locations.

4.2. Integration of Multimodal Data

Real-world traffic is highly correlated with various factors, like social media, weather information, traffic sensors, *etc.* A few existing works have utilized data from different sources limiting the model to understanding actual traffic dynamics. Heterogeneous data fusion could contribute to the performance improvement of ITS applications.

4.3. Adaptive Learning Algorithm

Most existing models use historical and fixed data having a lack of flexibility to adapt to changing situations, like road closure, accidents, or sudden weather changes that would not perform satisfactorily in changing situations. Developing adaptive algorithms by incorporating real-time traffic data with changing situations for better performance is crucial. Exploring online learning techniques to update the model with new data could contribute to this area.

4.4. Privacy-preserving Data Sharing

Since traffic data could usually contain sensitive user information that often hinders data collection and effective model development, proper privacy-preserving data collection and processing methods, like homomorphic encryption and differential privacy, need to be incorporated to ensure privacy-preserved data sharing.

4.5. Multimodal Travel Prediction

In real traffic scenarios, various modes of transportation, like buses, cars, and trucks, co-exist, and their interactions are evident. Existing applications mainly consider single vehicles, like cars or buses, that fail to capture the actual traffic scenarios and could lead to improper traffic management suggestions and inaccurate predictions, especially in urban areas.

4.6. Model Scalability and Generalization

Existing models often exhibit promising performance in selected scenarios with used datasets, but they may not work well in diverse scenarios with various data. So, more potential research could be performed on developing scalable and generalized models using data from a wide range of scenarios.

4.7. Model Robustness and Resilience

Existing proposals have mainly focused on training and testing for a specific scenario and with specific datasets, but they have not been tested extensively in different challenging events and unexpected scenarios to prove their robustness and resilience. Data augmentation and adversarial testing could contribute to robust and resilient model development.

4.8. Real-time Implementation

For several applications, including traffic congestion prediction, traffic signal control, and traffic accident risk prediction, real-time prediction is crucial. However, most proposals do not analyze the latency and computational requirements, which must be evaluated to ensure their effectiveness in providing real-time services. Utilizing efficient ML algorithms and optimized model architecture could contribute to this situation.

4.9. Cooperation among Applications

More effective, efficient, and intelligent applications could develop by cooperation and data sharing among different applications. For example, traffic flow, traffic congestion, and travel time prediction applications could cooperate, and traffic accident risk prediction and road anomaly detection could cooperate to develop better solutions. Further research on multimodel and ensemble learning could enhance the effect of cooperation among different applications.

4.10. Interpretability and Explainability

In general, ML models are like black boxes that often make it difficult to realize how the output is generated. Interpretability and explainability of the ML model for ITS applications are crucial to increase the trust and transparency of the model's outcome. They also contribute to detecting the model's bias and errors that could allow developers to address and improve the model's performance further.

CONCLUSION

This paper has presented how ITS application research has been revolutionized by leveraging the effectiveness of ML and DL algorithms. It has first identified 11 popular ITS applications across four areas: traffic management, safety management, infotainment and comfort, and autonomous driving. Then, for each ITS application, recent ML-based potential proposals have been analyzed that have gained high attention from the research community. Reviewing the high-impact articles has revealed the significant potential and impact of the DL algorithm for effective solutions. For some applications, hybrid models have exhibited superior performance by effectively extracting spatial and temporal features. However, due to the lack of publicly available benchmark datasets, most of the models have been trained and tested using datasets collected from a specific region or city, and it is uncertain whether these models have performed expectedly for other locations. After a detailed review, the discussion of some open issues has been accommodated that requires further research for improved predictive capabilities, decision making, and optimized operational efficient ITS solutions.

Finally, the review has offered valuable insights into tracking recent trends in ITS application research using ML and DL algorithms. The open research issues have provided directions for future research and development of more efficient, safe, and user-friendly ITS applications using ML techniques that could reshape future mobility.

AUTHORS' CONTRIBUTION

Abul Kalam Azad and Travis Atkison contributed to the study's conception and design; Abul Kalam Azad collected and reviewed the papers and prepared the original draft; Abul Kalam Azad, Travis Atkison, and A. F. M. Shahen Shah performed writing, review, and editing; Travis Atkison supervised the study. All authors have reviewed and approved the final version of the manuscript.

LIST OF ABBREVIATIONS

ITS	= Intelligent Transportation Systems
ML	= Machine Learning
DL	= Deep Learning
DT	= Decision Tree
RF	= Random Forest
SVM	= Support Vector Machine
KNN	= K-Nearest Neighbor
CNN	= Convolutional Neural Network
YOLO	= You Only Look Once
GRU	= Gated Recurrent Unit
STFSA	= Spatio-temporal Feature Selection Algorithm
SAE	= Sparse Autoencoder
DCPN	= Deep Congestion Prediction Network
SATCS	= Seattle Area Traffic Congestion Status

CONSENT FOR PUBLICATION

Not applicable.

FUNDING

None.

CONFLICT OF INTEREST

The authors declare no conflict of interest, financial or otherwise.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

- [1] "Transportation basics and future", Available from: <https://witan.world.com/article/2020/02/08/transportation/>
- [2] M. Alam, J. Ferreira, and J. Fonseca, *Intelligent Transportation System (ITS), Dependable Vehicular Communications for Improved Road Safety*, Springer: Cham, Switzerland, 2016, pp. 1-17.
[<http://dx.doi.org/10.1007/978-3-319-28183-4>]
- [3] S.H. Ahmed, S.H. Bouk, and D. Kim, "Target RSU Selection with low scanning latency in WiMAX-enabled vehicular networks", *Mob. Netw. Appl.*, vol. 20, no. 2, pp. 239-250, 2009.
[<http://dx.doi.org/10.1007/s11036-015-0578-3>]
- [4] "Intelligent Transport Systems (ITS); Vehicular communications; Basic set of applications; Definitions", Available from: https://www.etsi.org/deliver/etsi_tr/102600_102699/102638/01.01.01_60/tr_102638v010101p.pdf
- [5] E. Hamida, H. Noura, and W. Znaidi, "Security of cooperative intelligent transport systems: Standards, threats analysis and cryptographic countermeasures", *Electronics (Basel)*, vol. 4, no. 3, pp. 380-423, 2015.
[<http://dx.doi.org/10.3390/electronics4030380>]
- [6] "WHO mortality database", Available from: <https://www.who.int/data/data-collection-tools/who-mortality-database> (Accessed April. 29, 2024)
- [7] K. Yu, L. Lin, and M. Alazab, "Deep learning-based traffic safety solution for a mixture of autonomous and manual vehicles in a 5G-enabled intelligent transportation system", *IEEE Trans. Intell. Transp. Syst.*, 2021.
[<http://dx.doi.org/10.1109/TITS.2020.3042504>]
- [8] C. Roy, R. Saha, S. Misra, and K. Dev, "Micro-safe: Microservices-and deep learning-based safety-as-a-service architecture for 6G-enabled intelligent transportation system", *IEEE Trans. Intell. Transp. Syst.*, pp. 1-10, 2022.
[<http://dx.doi.org/10.1109/TITS.2021.3110725>]
- [9] A.F.M. Shahen Shah, "A survey from 1G to 5G including the advent of 6G: Architectures, multiple access techniques, and emerging technologies", *Proc. of IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, 2022, pp. 1117-1123.
[<http://dx.doi.org/10.1109/CCWC54503.2022.9720781>]
- [10] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L.H. Lee, "Enhancing transportation systems via deep learning: A survey", *Transp. Res., Part C Emerg. Technol.*, vol. 99, pp. 144-163, 2019.
[<http://dx.doi.org/10.1016/j.trc.2018.12.004>]
- [11] D. Gangwani, and P. Gangwani, "Applications of machine learning and artificial intelligence in intelligent transportation system: A review", In: *Lecture Notes in Electrical Engineering*, 2021, pp. 203-216.
[http://dx.doi.org/10.1007/978-981-16-3067-5_16]
- [12] R. Santhiya, and C. GeethaPriya, "Machine learning techniques for intelligent transportation systems-An overview", *12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, 2021, pp.

- 1-7.
[<http://dx.doi.org/10.1109/ICCNT51525.2021.9579970>]
- [13] G.L. Huang, A. Zaslavsky, S.W. Loke, A. Abkenar, A. Medvedev, and A. Hassani, "Context-aware machine learning for intelligent transportation systems: A survey", *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 17-36, 2023.
[<http://dx.doi.org/10.1109/TITS.2022.3216462>]
- [14] F. Zantalis, G. Koulouras, S. Karabetsos, and D. Kandris, "A review of machine learning and IoT in smart transportation", *Future Internet*, vol. 11, no. 4, p. 94, 2019.
[<http://dx.doi.org/10.3390/fi11040094>]
- [15] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah, "A survey of deep learning applications to autonomous vehicle control", *IEEE Trans. Intell. Transp. Syst.*, 2020.
[<http://dx.doi.org/10.1109/TITS.2019.2962338>]
- [16] A. Dey, "Machine Learning Algorithms: A review", *Int. J. Comput. Sci. Inf. Technol.*, vol. 7, no. 3, pp. 1174-1179, 2016.
- [17] I.H. Sarker, "Machine learning: Algorithms, real-world applications and research directions", *SN Comput. Sci.*, vol. 2, no. 3, 2021.
[<http://dx.doi.org/10.1007/s42979-021-00592-x>]
- [18] R. Thourani, "Types of semi supervised machine learning algorithms", Available from: <https://www.finsliqblog.com/ai-and-machine-learning/types-of-semi-supervised-algorithms/> (Accessed on: 11-Jun-2021)
- [19] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning", *IEEE Trans. Vehicular Technol.*, vol. 69, no. 12, pp. 14413-14423, 2020.
[<http://dx.doi.org/10.1109/TVT.2020.3034800>]
- [20] R. Bellman, "A markovian decision process", *Indiana Univ. Math. J.*, vol. 6, no. 4, pp. 679-684, 1957.
[<http://dx.doi.org/10.1512/iumj.1957.6.56038>]
- [21] J.J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *Proc. Natl. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554-2558, 1982.
[<http://dx.doi.org/10.1073/pnas.79.8.2554>] [PMID: 6953413]
- [22] C.C. Aggarwal, *Neural networks and deep learning: a textbook*, Springer: New York, 2019.
- [23] E. Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, 2019.
[<http://dx.doi.org/10.1007/978-1-4842-4470-8>]
- [24] A. Navada, A.N. Ansari, S. Patil, and B.A. Sonkamble, "Overview of use of decision tree algorithms in machine learning", *IEEE Control and System Graduate Research Colloquium*, 2011 Shah Alam, Malaysia, 2011, pp. 37-42.
[<http://dx.doi.org/10.1109/ICSGRC.2011.5991826>]
- [25] A. Liaw, and M. Wiener, "Classification and regression by random forest", *R News*, vol. 2, no. 3, pp. 18-22, 2002.
- [26] D.A. Pisner, and D.M. Schnyer, "Support vector machine", In: *Machine Learning Mechelli*, Academic: Cambridge, MA, USA, 2020, pp. 101-121.
[<http://dx.doi.org/10.1016/B978-0-12-815739-8.00006-7>]
- [27] P. Cunningham, and S.J. Delany, "k-Nearest neighbour classifiers", *Multiple Classifier Syst.*, vol. 34, pp. 1-17, 2007.
- [28] M. Steinbach, and P.N. Tan, "KNN: K-nearest neighbors", In: *The Top Ten Algorithms in Data Mining*, CRC Press: Boca Raton, FL, USA, 2009, pp. 165-176.
[<http://dx.doi.org/10.1201/9781420089653.ch8>]
- [29] S. Albawi, T.A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network", *International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, 2017, pp. 1-6.
[<http://dx.doi.org/10.1109/ICEngTechnol.2017.8308186>]
- [30] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments", *Procedia Comput. Sci.*, vol. 199, pp. 1066-1073, 2022.
[<http://dx.doi.org/10.1016/j.procs.2022.01.135>]
- [31] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network", *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.
[<http://dx.doi.org/10.1016/j.physd.2019.132306>]
- [32] D. Bank, N. Koenigstein, and Giryev, "Autoencoders", *arXiv*, 2020.
[<http://dx.doi.org/10.48550/arXiv.2003.05991>]
- [33] T. Yuan, W. Da Rocha Neto, C.E. Rothenberg, K. Obraczka, C. Barakat, and T. Turlatti, "Machine learning for next-generation intelligent transportation systems: A survey", *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 4, p. e4427, 2022.
[<http://dx.doi.org/10.1002/ett.4427>]
- [34] M. Veres, and M. Moussa, "Deep learning for intelligent transportation systems: A survey of emerging trends", *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3152-3168, 2020.
[<http://dx.doi.org/10.1109/TITS.2019.2929020>]
- [35] Y. Lv, Y. Duan, W. Kang, Z. Li, and F-Y. Wang, "Traffic flow prediction with big data: A deep learning approach", *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865-873, 2015.
- [36] B. Medina-Salgado, E. Sánchez-DelaCruz, P. Pozos-Parra, and J. E. Sierra, "Urban traffic flow prediction techniques: A review", *Sustain. Comput. Inform. Syst.*, vol. 35, no. 7, p. 100739, 2022.
[<http://dx.doi.org/10.1016/j.suscom.2022.100739>]
- [37] Z. Li, G. Xiong, Y. Chen, Y. Lv, B. Hu, F. Zhu, and F-Y. Wang, "A hybrid deep learning approach with GCN and LSTM for traffic flow prediction", *IEEE Intell. Transp. Syst. Conf. (ITSC)*, Auckland, New Zealand, 2019, pp. 1929-1933.
[<http://dx.doi.org/10.1109/ITSC.2019.8916778>]
- [38] W.B. Zhang, Y.H. Yu, Y. Qi, F. Shu, and Y.H. Wang, "Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning", *Transportmetrica*, vol. 15, no. 2, pp. 1688-1711, 2019.
- [39] C. Chen, Z. Liu, S. Wan, J. Luan, and Q. Pei, "Traffic flow prediction based on deep learning in internet of vehicles", *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3776-3789, 2021.
[<http://dx.doi.org/10.1109/TITS.2020.3025856>]
- [40] X. Chen, H. Chen, Y. Yang, H. Wu, W. Zhang, J. Zhao, and Y. Xiong, "Traffic flow prediction by an ensemble framework with data denoising and deep learning model", *Physica A*, vol. 565, p. 125574, 2021.
[<http://dx.doi.org/10.1016/j.physa.2020.125574>]
- [41] K. Wang, C. Ma, Y. Qiao, X. Lu, W. Hao, and S. Dong, "A hybrid deep learning model with 1DCNN-LSTM-Attention networks for short-term traffic flow prediction", *Physica A*, vol. 583, p. 126293, 2021.
[<http://dx.doi.org/10.1016/j.physa.2021.126293>]
- [42] M. Bai, Y. Lin, M. Ma, P. Wang, and L. Duan, "PrePCT: Traffic congestion prediction in smart cities with relative position congestion tensor", *Neurocomputing*, vol. 444, pp. 147-157, 2021.
[<http://dx.doi.org/10.1016/j.neucom.2020.08.075>]
- [43] S. Zhang, Y. Yao, J. Hu, Y. Zhao, S. Li, and J. Hu, "Deep autoencoder neural networks for short-term traffic congestion prediction of transportation networks", *Sensors (Basel)*, vol. 19, no. 10, p. 2229, 2019.
[<http://dx.doi.org/10.3390/s19102229>] [PMID: 31091802]
- [44] A. Ata, M.A. Khan, S. Abbas, G. Ahmad, and A. Fatima, "Modelling smart road traffic congestion control system using machine learning techniques", *Neural Netw. World*, vol. 29, no. 2, pp. 99-110, 2019.
[<http://dx.doi.org/10.14311/NNW.2019.29.008>]
- [45] N. Ranjan, S. Bhandari, H.P. Zhao, H. Kim, and P. Khan, "City-wide traffic congestion prediction based on CNN LSTM and transpose CNN", *IEEE Access*, vol. 8, pp. 81606-81620, 2020.
[<http://dx.doi.org/10.1109/ACCESS.2020.2991462>]
- [46] A. Boukerche, and J. Wang, "Machine Learning-based traffic prediction models for intelligent transportation systems", *Comput. Netw.*, vol. 181, p. 107530, 2020.
[<http://dx.doi.org/10.1016/j.comnet.2020.107530>]
- [47] L. Lin, J. Li, F. Chen, J. Ye, and J. Huai, "Road traffic speed prediction: A probabilistic model fusing multi-source data", *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1310-1323, 2018.
[<http://dx.doi.org/10.1109/TKDE.2017.2718525>]

- [48] L. Li, X. Qu, J. Zhang, Y. Wang, and B. Ran, "Traffic speed prediction for intelligent transportation system based on a deep feature fusion model", *J. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 605-616, 2019.
[<http://dx.doi.org/10.1080/15472450.2019.1583965>]
- [49] J. Wang, R. Chen, and Z. He, "Traffic speed prediction for urban transportation network: A path based deep learning approach", *Transp. Res., Part C Emerg. Technol.*, vol. 100, pp. 372-385, 2019.
[<http://dx.doi.org/10.1016/j.trc.2019.02.002>]
- [50] Z. Xie, W. Lv, S. Huang, Z. Lu, B. Du, and R. Huang, "Sequential graph neural network for urban road traffic speed prediction", *IEEE Access*, vol. PP, no. 99, p. 1, 2020.
[<http://dx.doi.org/10.1109/ACCESS.2019.2915364>]
- [51] C. Bratsas, K. Koupidis, J.M. Salanova, K. Giannakopoulos, A. Kaloudis, and G. Aifadopoulou, "A comparison of machine learning methods for the prediction of traffic speed in urban places", *Sustainability (Basel)*, vol. 12, no. 1, p. 142, 2019.
[<http://dx.doi.org/10.3390/su12010142>]
- [52] A. Essien, I.S.P. Petrounias, and S. Sampaio, "Improving urban traffic speed prediction using data source fusion and deep learning", *IEEE International Conference on Big Data and Smart Computing (BigComp)*, Kyoto, Japan, 2019, pp. 1-8.
[<http://dx.doi.org/10.1109/BIGCOMP.2019.8679231>]
- [53] X. Ran, Z. Shan, Y. Fang, and C. Lin, "An LSTM-based method with attention mechanism for travel time prediction", *Sensors (Basel)*, vol. 19, no. 4, p. 861, 2019.
[<http://dx.doi.org/10.3390/s19040861>] [PMID: 30791424]
- [54] J. Qiu, L. Du, D. Zhang, S. Su, and Z. Tian, "Nei-TTE: Intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city", *IEEE Trans. Industr. Inform.*, vol. 16, no. 4, pp. 2659-2666, 2020.
[<http://dx.doi.org/10.1109/TII.2019.2943906>]
- [55] N.C. Petersen, F. Rodrigues, and F.C. Pereira, "Multi-output bus travel time prediction with convolutional LSTM neural network", *Expert Syst. Appl.*, vol. 120, pp. 426-435, 2019.
[<http://dx.doi.org/10.1016/j.eswa.2018.11.028>]
- [56] P. He, G. Jiang, S.K. Lam, and Y. Sun, "Learning heterogeneous traffic patterns for travel time prediction of bus journeys", *Inf. Sci.*, vol. 512, pp. 1394-1406, 2020.
[<http://dx.doi.org/10.1016/j.ins.2019.10.073>]
- [57] M. Abdollahi, T. Khaleghi, and K. Yang, "An integrated feature learning approach using deep learning for travel time prediction", *Expert Syst. Appl.*, vol. 139, p. 112864, 2020.
[<http://dx.doi.org/10.1016/j.eswa.2019.112864>]
- [58] W. Genders, and S. Razavi, "Asynchronous n -step Q-learning adaptive traffic signal control", *J. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 319-331, 2019.
[<http://dx.doi.org/10.1080/15472450.2018.1491003>]
- [59] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control", *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1086-1095, 2020.
[<http://dx.doi.org/10.1109/TITS.2019.2901791>]
- [60] H. Joo, S.H. Ahmed, and Y. Lim, "Traffic signal control for smart cities using reinforcement learning", *Comput. Commun.*, vol. 154, pp. 324-330, 2020.
[<http://dx.doi.org/10.1016/j.comcom.2020.03.005>]
- [61] X. Zang, H. Yao, G. Zheng, N. Xu, K. Xu, and Z. Li, "MetaLight: Value-based meta-reinforcement learning for traffic signal control", *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.
[<http://dx.doi.org/10.1609/aaai.v34i01.5467>]
- [62] C. Chen, H. Wei, and Nan. Xu, "Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control", *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.
[<http://dx.doi.org/10.1609/aaai.v34i04.5744>]
- [63] P. Melnyk, S. Djahel, and F. Nait-Abdesselam, "Towards a smart parking management system for smart cities", *IEEE International Smart Cities Conference (ISC2)*, Casablanca, Morocco, 2019, pp. 542-546.
[<http://dx.doi.org/10.1109/ISC246665.2019.9071740>]
- [64] S. Saharan, N. Kumar, and S. Bawa, "An efficient smart parking pricing system for smart city environment: A machine-learning based approach", *Future Gener. Comput. Syst.*, vol. 106, pp. 622-640, 2020.
[<http://dx.doi.org/10.1016/j.future.2020.01.031>]
- [65] G. Ali, T. Ali, M. Irfan, U. Draz, M. Sohail, A. Glowacz, M. Sulowicz, R. Mielnik, Z.B. Faheem, and C. Martis, "IoT based smart parking system using deep long short memory network", *Electronics (Basel)*, vol. 9, no. 10, p. 1696, 2020.
[<http://dx.doi.org/10.3390/electronics9101696>]
- [66] F.M. Awan, Y. Saleem, R. Minerva, and N. Crespi, "A comparative analysis of machine/deep learning models for parking space availability prediction", *Sensors (Basel)*, vol. 20, no. 1, p. 322, 2020.
[<http://dx.doi.org/10.3390/s20010322>] [PMID: 31935953]
- [67] S.Y. Siddiqui, M.A. Khan, S. Abbas, and F. Khan, "Smart occupancy detection for road traffic parking using deep extreme learning machine", *J. King Saud Univ. Comp. Inform. Sci.*, vol. 34, no. 3, 2020.
[<http://dx.doi.org/10.1016/j.jksuci.2020.01.016>]
- [68] S.C.K. Tekouabou, E.A.A. Alaoui, W. Cherif, and H. Silkan, "Improving parking availability prediction in smart cities with IoT and ensemble-based model", *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 3, pp. 687-697, 2022.
[<http://dx.doi.org/10.1016/j.jksuci.2020.01.008>]
- [69] A. Kumawat, and K. Chandramore, "Automatic toll collection system using RFID", *Int. J. Electr. Electron. Res.*, vol. 2, pp. 67-72, 2014.
- [70] W. Wang, J. Yang, M. Chen, and P. Wang, "A light CNN for end-to-end car license plates detection and recognition", *IEEE Access*, vol. 7, pp. 173875-173883, 2019.
[<http://dx.doi.org/10.1109/ACCESS.2019.2956357>]
- [71] A. Tourani, A. Shahbahrami, S. Soroori, S. Khazaei, and C.Y. Suen, "A robust deep learning approach for automatic Iranian vehicle license plate detection and recognition for surveillance systems", *IEEE Access*, vol. 8, pp. 201317-201330, 2020.
[<http://dx.doi.org/10.1109/ACCESS.2020.3035992>]
- [72] V. Ganaprakash, N. Kanthimathi, and N. Saranya, "Automatic number plate recognition using deep learning", *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1084, no. 1, p. 012027, .
[<http://dx.doi.org/10.1088/1757-899X/1084/1/012027>]
- [73] R. Laroca, A.L. Zanlorensi, R.G. Gonçalves, E. Todt, W.R. Schwartz, and D. Menotti, "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector", *arXiv*, 2019.
[<http://dx.doi.org/10.48550/arXiv.1909.01754>]
- [74] A. Theofilatos, C. Chen, and C. Antoniou, "Comparing machine learning and deep learning methods for real-time crash prediction", *Transp. Res. Rec.*, vol. 2673, no. 8, pp. 169-178, 2019.
[<http://dx.doi.org/10.1177/0361198119841571>]
- [75] X. Shi, Y.D. Wong, C. Chai, and M.Z.F. Li, "An automated machine learning (AutoML) method of risk prediction for decision-making of autonomous vehicles", *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7145-7154, 2021.
[<http://dx.doi.org/10.1109/TITS.2020.3002419>]
- [76] X. Wang, J. Liu, T. Qiu, C. Mu, C. Chen, and P. Zhou, "A real-time collision prediction mechanism with deep learning for intelligent transportation system", *IEEE Trans. Vehicular Technol.*, vol. 69, no. 9, pp. 9497-9508, 2020.
[<http://dx.doi.org/10.1109/TVT.2020.3003933>]
- [77] T. Huang, S. Wang, and A. Sharma, "Highway crash detection and risk estimation using deep learning", *Accid. Anal. Prev.*, vol. 135, p. 105392, 2020.
[<http://dx.doi.org/10.1016/j.aap.2019.105392>] [PMID: 31841865]
- [78] J. Dib, K. Sirlantzis, and G. Howells, "A review on negative road anomaly detection methods", *IEEE Access*, vol. 8, pp. 57298-57316, 2020.
[<http://dx.doi.org/10.1109/ACCESS.2020.2982220>]
- [79] A. Basavaraju, J. Du, F. Zhou, and J. Ji, "A machine learning

- approach to road surface anomaly assessment using smartphone sensors", *IEEE Sens. J.*, vol. 20, no. 5, pp. 2635-2647, 2020.
[<http://dx.doi.org/10.1109/JSEN.2019.2952857>]
- [80] B. Varona, A. Monteserin, and A. Teyseyre, "A deep learning approach to automatic road surface monitoring and pothole detection", *Pers. Ubiquitous Comput.*, vol. 24, no. 4, pp. 519-534, 2020.
[<http://dx.doi.org/10.1007/s00779-019-01234-z>]
- [81] A. Aboah, M. Shoman, and V. Mandal, "A vision-based system for traffic anomaly detection using deep learning and decision trees", *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Nashville, TN, USA, 2021, pp. 4202-4207.
[<http://dx.doi.org/10.1109/CVPRW53098.2021.00475>]
- [82] K.R. Ahmed, "Smart pothole detection using deep learning based on dilated convolution", *Sensors*, vol. 21, no. 24, p. 8406, 2021.
[<http://dx.doi.org/10.3390/s21248406>]
- [83] M.A. Rahman, M.A. Rahim, M.M. Rahman, N. Moustafa, I. Razzak, T. Ahmad, and M.N. Patwary, "A secure and intelligent framework for vehicle health monitoring exploiting big-data analytics", *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19727-19742, 2022.
[<http://dx.doi.org/10.1109/TITS.2021.3138255>]
- [84] U. Shafi, A. Safi, A.R. Shahid, S. Ziauddin, and M.Q. Saleem, "Vehicle remote health monitoring and prognostic maintenance system", *J. Adv. Transp.*, vol. 2018, pp. 1-10, 2018.
[<http://dx.doi.org/10.1155/2018/8061514>]
- [85] C-S. Alex Gong, C-H. Simon Su, and K.H. Tseng, "Implementation of machine learning for fault classification on vehicle power transmission system", *IEEE Sens. J.*, vol. 20, no. 24, pp. 15163-15176, 2020.
[<http://dx.doi.org/10.1109/JSEN.2020.3010291>]
- [86] R. Wang, Y. Zhang, G. Fortino, Q. Guan, J. Liu, and J. Song, "Software escalation prediction based on deep learning in the cognitive internet of vehicles", *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25408-25418, 2022.
[<http://dx.doi.org/10.1109/TITS.2022.3140903>]
- [87] A. Faisal, T. Yigitcanlar, M. Kamruzzaman, and G. Currie, "Understanding autonomous vehicles: A systematic literature review on capability, impact, planning and policy", *J. Transp. Land Use*, vol. 12, no. 1, pp. 45-72, 2019.
[<http://dx.doi.org/10.5198/jtlu.2019.1405>]
- [88] A.E.L. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving", *IS&T Int. Symp. Electron. Imaging*, vol. 29, no. 19, pp. 70-76, 2017.
[<http://dx.doi.org/10.2352/ISSN.2470-1173.2017.19.AVM-023>]
- [89] A. Alizadeh, M. Moghadam, Y. Bicer, N.K. Ure, U. Yavas, and C. Kurtulus, "Automated lane change decision making using deep reinforcement learning in dynamic and uncertain highway environment", *IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, 2019, pp. 1399-1404.
[<http://dx.doi.org/10.1109/ITSC.2019.8917192>]
- [90] J. Wei, J. He, Y. Zhou, K. Chen, Z. Tang, and Z. Xiong, "Enhanced object detection with deep convolutional neural networks for advanced driving assistance", *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1572-1583, 2020.
[<http://dx.doi.org/10.1109/TITS.2019.2910643>]
- [91] J. Chen, B. Yuan, and M. Tomizuka, "Model-free deep reinforcement learning for urban autonomous driving", *IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, 2019, pp. 2765-2771.
[<http://dx.doi.org/10.1109/ITSC.2019.8917306>]
- [92] C. Chen, J. Jiang, N. Lv, and S. Li, "An intelligent path planning scheme of autonomous vehicles platoon using deep reinforcement learning on network edge", *IEEE Access*, vol. 8, pp. 99059-99069, 2020.
[<http://dx.doi.org/10.1109/ACCESS.2020.2998015>]
- [93] Y. Fu, C. Li, F.R. Yu, T.H. Luan, and Y. Zhang, "A decision-making strategy for vehicle autonomous braking in emergency via deep reinforcement learning", *IEEE Trans. Vehicular Technol.*, vol. 69, no. 6, pp. 5876-5888, 2020.
[<http://dx.doi.org/10.1109/TVT.2020.2986005>]