IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 December 2023, accepted 3 January 2024, date of publication 17 January 2024, date of current version 5 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3355034

SURVEY

Survey on Multi-Task Learning in Smart Transportation

MOHAMMED ALZAHRANI[1], QIANLONG WANG[1], WEIXIAN LIAO[1], XUHUI CHEN[2], (Member, IEEE), AND WEI YU[1], (Senior Member, IEEE)

[1]Department of Computer and Information Sciences, Towson University, Towson, MD 21252, USA
[2]College of Aeronautics and Engineering, Kent State University, Kent, OH 44240, USA

Corresponding author: Qianlong Wang (qwang@towson.edu)

This work was supported in part by the National Science Foundation (NSF) under Grant 2245933.

ABSTRACT Artificial Intelligence (AI) has been widely adopted in numerous fields and enabled various smart systems because of its strong ability to perform tasks, including prediction, event detection, and status estimation, among others. As one of the typical smart systems empowered by AI and Internet of Things (IoT) technologies, the smart transportation system has made dramatic progress for traditional transportation in numerous aspects, including autonomous driving, smart traffic lights, navigation, and traffic forecasting, among others. Deep learning is an essential component to enable such smart systems. Typically, specific deep learning models, e.g., Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN), can be trained on collected transportation data for a particular task. However, traditional deep learning techniques rely on data sufficiency to build an effective model. Additionally, each trained model can only work on one single task. This has limited the efficacy of deep learning techniques in numerous application scenarios. To this end, multi-task learning (MTL) has been studied to train a single model that can work for multiple tasks. This technique effectively allows the learning model to expand to specific tasks with only a limited amount of data affiliated. In the meantime, MTL significantly reduces the training time of each task. The success of MTL requires that there are potential relationships among different tasks. Many tasks in the smart transportation system are related. For instance, traffic speed and vehicle volume estimations for each road are highly correlated. Based on this, research on applying MTL in smart transportation systems has been studied recently. This paper reviews the recent efforts to use MTL in smart transportation systems and conducts an extensive survey to provide insights. In particular, we categorize the MTL applications in smart transportation systems into traffic forecasting, traffic sign recognition, vehicle recognition, travel time estimation, road safety estimation, taxi demand prediction, and autonomous driving. Ultimately, we discuss challenges and future research directions in applying MTL in smart transportation systems.

INDEX TERMS Multi-task learning (MTL), smart transportation, deep learning.

I. INTRODUCTION

Smart transportation has been widely studied nowadays, which plays a vital role in dealing with traditional transportation problems by integrating artificial intelligence (AI) technology with transportation systems [1], [2], [3], [4]. In particular, smart transportation has dramatically improved traffic efficiency, security, reliability, and others. There are numerous characteristics of the smart transportation system [5]. First, there are massive amounts of collected data as the system integrates many technologies that could contribute to generating a large volume of data. Second, such data could be heterogeneous since the smart transportation system uses a variety of ways to collect data (e.g., sensing data collected by sensors deployed in vehicles, text data captured from social media, and geo-data captured from maps). Third, the data is generated rapidly since the smart transportation system needs to provide real-time decisions. For instance, the data will be collected continuously to carry out traffic speed prediction. Fourth, the data captured by the smart

The associate editor coordinating the review of this manuscript and approving it for publication was Thomas Canhao Xu.

transportation system needs to be reliable and trustworthy to support applications (e.g., autonomous driving). In those applications, the quality of the data is a crucial factor. Furthermore, smart transportation systems introduce many applications to improve services, including intelligent traffic management, traveler management, autonomous driving, advanced public and rural transportation, commercial vehicle operation and management, and others [4], [6], [7], [8], [9]. In the meantime, a vehicular ad hoc network (VANET) is developed for vehicle communications through a wireless communication network. Such a network has been introduced in smart transportation systems to assist the vehicle in taking proper action [10].

Deep learning is one of the main techniques to enable smart transportation. The deep learning model can be trained via using the collected data in the transportation system to perform specific tasks such as traffic forecasting, traffic light management, traffic sign recognition, etc. [11], [12], [13]. Security and privacy are also considered important factors in the current smart transportation systems [4], [14]. Deep learning is adopted to help enhance the system resilience that deals with malicious attacks and preserves the users' privacy [15]. Specifically, varied deep learning models are adopted in the smart transportation system, including Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Deep Reinforcement Learning (DRL), and Graph Neural Networks, among others [16], [17], [18], [19], which have been applied to various IoT systems. However, traditional deep learning relies highly on the amount of data to build effective learning models. The shortage of data may cause the model to overfit and lead to poor prediction accuracy. On the other hand, the typical deep learning model is only trained for a single task, meaning that the users need to train a new model from scratch for a particular new task, as shown in Fig. 1. This training process can be highly data-driven and time-consuming. Such concerns are the main limitations of applying traditional deep learning to the smart transportation system.

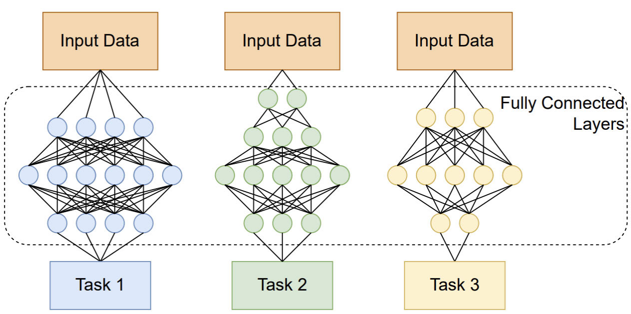


FIGURE 1. **Training tasks in separate models.**

To address the aforementioned issues, Multi-Task Learning (MTL) has been studied, which tends to train a single model to make predictions for different tasks. In MTL, the model can be enhanced by concurrently training related tasks. It can improve the accuracy of the output for the original task, discriminate the related or unrelated features, and help tasks learn from each other, as shown in Fig. 2 [20]. It is worth noting that the MTL shows its effectiveness in numerous applications such as natural language processing, speech recognition, computer vision, and healthcare. In addition, the MTL can reduce overfitting, which is when a model works better for training data than it does with the new data [20].

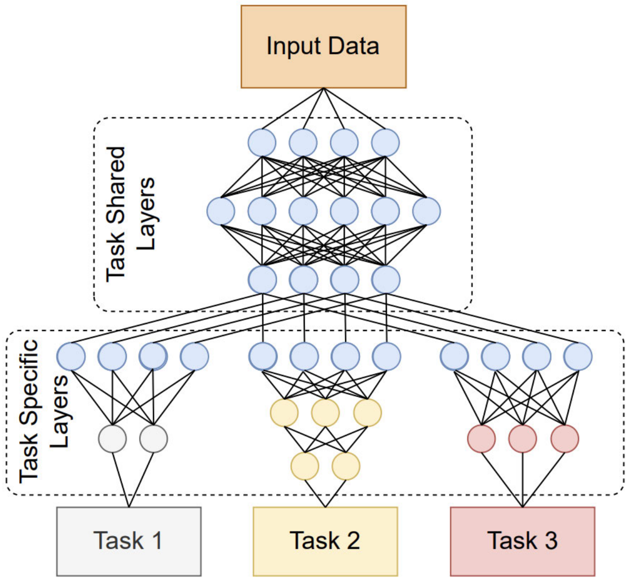


FIGURE 2. **Training related features concurrently in the shared model by MTL and sharing these features among tasks.**

The reason for the MTL being successful is that certain common patterns exist between specific related tasks. For instance, traffic speed estimation and traffic volume estimation could be highly correlated tasks. Thus, the model trained for traffic speed estimation also learns the patterns helpful in estimating the traffic volume. Using MTL, the model trained on one task can be easily applied to another task using only a few data samples. Such a method allows the model to perform the task when a limited amount of data is affiliated. By doing this, we can significantly reduce the training time that the traditional deep learning model requires for each new task. Many IoT systems, such as the smart transportation system, generally require timely decisions.

When it comes to transportation applications, researchers realize that many tasks in the smart transportation system are correlated and fit for the scenario of MTL. A number of research efforts have been studied to apply MTL in smart transportation systems, expanding the learning tasks and enhancing the efficiency of the training process. In particular, compared with regular deep learning models, MTL has further increased the prediction accuracy, training efficiency, model reliability, and scalability in the smart transportation system. Many tasks in the smart transportation system are related, and training them together in one model makes all tasks share important features without missing any important information. There are challenges concerning accuracy if many variables reduce the quality of prediction. MTL has been used in smart transportation systems in ways like traffic and taxi demand prediction and travel time estimation with various objects they have. Also, it is used to distinguish between the multiple types of traffic signs and vehicles. In addition, MTL is used to detect objects to enhance road safety and autonomous driving. MTL has shown its strong capability and is an important and promising tool to increase smart transportation performance and accuracy.

This paper summarizes the existing research efforts on applying MTL in smart transportation systems and conducts a systematic survey. We categorize the applications of MTL in smart transportation systems into the following aspects: traffic forecasting, traffic sign detection, vehicle recognition, travel time estimation, road safety, autonomous driving, and transportation demand prediction. Through the aforementioned aspects, we systematically review and discuss the existing efforts in using MTL to improve prediction and performance via sharing the features associated with learning tasks in smart transportation systems. We also highlight the benefits of MTL in terms of training efficiency, scalability, and model robustness and discuss its limitations concerning performance, security, and privacy. Finally, we outline future research directions that require further study concerning task similarity determination, feature selection, data distribution, and robust and privacy-preserving learning.

The remaining paper is organized as follows. In Section II, we give the overview of MTL. In Section III, we conduct a comprehensive review of the existing research efforts on the use of MTL in the smart transportation system from different aspects. In Section IV, we discuss the challenges and outline future research directions of MTL in the smart transportation system. Finally, we conclude the paper in Section V.

II. OVERVIEW OF MULTI-TASK LEARNING

MTL is useful for training multiple tasks with related features and extracting all the information among tasks to enhance the models' knowledge. There are two methods used to train the associated features in MTL. One of them is the hard parameter that shares the hidden layers in the neural network among tasks as it is shown in Fig. 2. The features or representations will be trained in the shared layers, and they will be shared between the tasks. This is considered the most common method in MTL. The other one is the soft parameter in which each task has its model, as it is shown in Fig. 3. The models in soft parameters are connected and will share the features. The hard shared parameter is better than the soft shared parameter because it can reduce overfitting. Also, hard shared parameters reduce the training time and storage cost.

However, the tasks have to be related to each other to increase the performance quality. Using related tasks as auxiliary tasks to the main task will improve performance. Still, sometimes the associated tasks are not labeled, so the adversarial loss will be used to solve this issue. In addition,

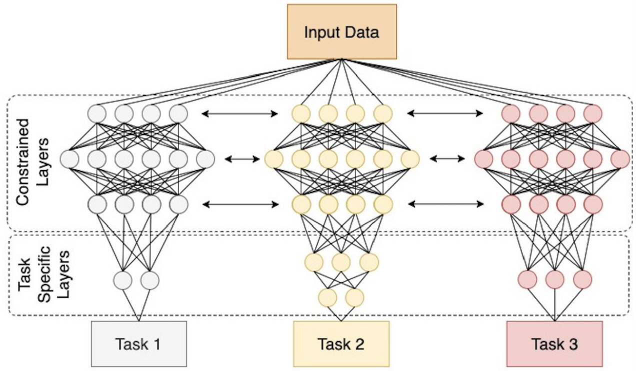


FIGURE 3. Soft parameter sharing for MTL.

auxiliary tasks can be used to concentrate on training specific parts, which is called focus attention, like in images, for example. Also, the loss function, which computes the errors between the algorithm output and the target value, is used to evaluate the performance.

MTL is typically set up as supervised learning. There are certain vital factors for MTL performance. Three typical ones are feature similarity, parameter suitability, and data quality [21]. First, depending on the similarity of the features across different tasks, the model will take advantage of that and learn according to the similarity of the tasks' features. Second, focusing on the parameters of each task, the parameter has to be suitable for each task to make the learning more efficient. Third, data quality is important so that the informative data could help the target tasks to learn more insight knowledge from other tasks.

In addition to supervised learning, unsupervised learning methods are used to improve MTL performance. For instance, clustering and dimensionality reduction are two unsupervised learning techniques that are widely used to improve data quality. Clustering is used to group similar data instances from multiple datasets without knowing what they represent. For instance, in some existing studies, clustering helps MTL to organize related tasks and makes these tasks more efficient for MTL by facilitating the sharing of knowledge between them [22]. Dimensionality reduction can be used to simplify the data to enhance the quality and efficiency of large amounts of data. It keeps the crucial features of the original data [23]. Additionally, autoencoder has become one viable unsupervised learning method to enhance the data representations of the model, improving supervised learning performance by making the data more meaningful and facilitating task training [24]. In addition to supervised and unsupervised learning, online learning is adopted in some MTL models to help train sequential data and continually update the learning model based on newly available data [25].

DNN trains the inputs with different weights and extracts the outputs (e.g., input the road path images to predict the travel time). DNN capacity is considered more than the traditional neural network [3], [19], [26]. Also, RNN are used more for sequential data like the historical data from the Global Positioning System (GPS). Moreover, CNN is used when there are spatial dependencies, like in GPS trace. Graph Convolutional Networks (GCN) is the semi-supervised network representing the CNN variants on a graph [27].

III. FOCUSED TOPIC AREAS

In smart transportation systems, many related features and representations must be trained together by MTL to achieve accurate predictions and better performance because crucial information will not be missed. In this section, we illustrate the effectiveness of MTL in increasing the accuracy and performance of various applications in smart transportation systems.

A. MTL FOR TRAFFIC FORECASTING

Determining traffic forecasting precisely is essential in deep learning and is considered the most crucial mission for GPS. It is challenging to increase its accuracy because many traffic variables, such as flow, speed, travel time, etc., need to be determined together. Training those variables individually made the performance quality low because these variables are related. Combining them with MTL can increase the accuracy rate because the models will learn the whole knowledge about the traffic.

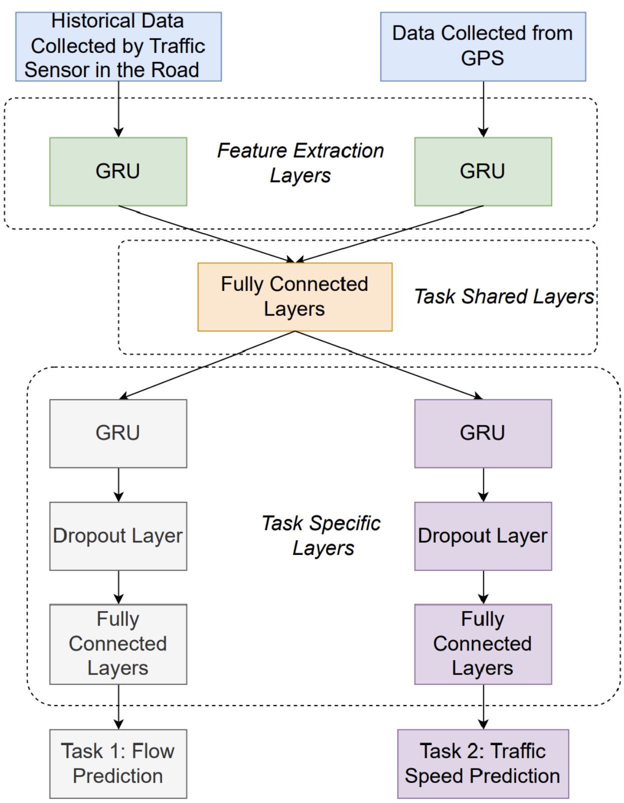


FIGURE 4. MTL based on GRU to predict the traffic flow and speed concurrently.

1) FLOW AND SPEED

The flow and speed are considered necessary tasks for traffic forecasting. Predicting them separately makes traffic forecasting inaccurate because there will be limited knowledge about how fast the cars move according to the traffic. The speed is related to predicting how fast the vehicles move based on the traffic condition, while the flow is associated with predicting the number of cars in the traffic. Buroni et al. confirmed that the MTL could accurately enhance the speed and flow prediction at the road network [28]. They separated the work by using GCN, which represents the road networks, to obtain the spatial features and GRU to obtain the temporal characteristics for the traffic, the flow, and the speed as level one. On the other level, a Fully Neural Network (FNN) and temporal attention block will determine the hidden representations for the related tasks in the first level. Also, Zhang et al. showed the progress of using MTL with GRU to predict the speed and the traffic more than what the Single-Task Learning (STL) could [29]. They noted that STL used the flow and traffic as single tasks, while MTL used both as related tasks, taking advantage of all the information for the tasks. In this way, the result of the speed and traffic forecasting can be improved. Fig. 4 illustrates the use of MTL based on a Gated Recurrent Unit (GRU) to predict both the flow and speed tasks concurrently to enhance traffic forecasting. GRU is considered type of RNN architecture that is used in this method to extract the sequential data that are collected from traffic sensors and GPS. The fully connected layer is used to transfer the features that are extracted to grouped sequential values. There are GRUs for each task as task-specific layer. The dropout layer is used in this method to drop random neurons on the layers to prevent the overfitting of the sequential data before predicting the flow and speed based on a fully connected layer.

Determining the traffic flow precisely helps to enhance traffic forecasting prediction. Using time series information, which is a sequence of data collected over a specific time, for traffic flow independently can lead to missing crucial information, while using them together will exploit every vital information, enhancing the model prediction. For example, Sun used the road links to determine the time series information for the traffic flow concurrently based on both bagging, which is an ensemble learning algorithm used to make better accuracy and prediction by combining many models prediction, and MTL [30]. The experimental results confirmed that MTL could lead to a better effect than STL because the traffic flow information (e.g., the interaction among the drivers and the vehicles based on MTL) will be trained together, and that helps to share similar features among them, which leads to include all important information. Jin et al. showed the traffic flow in the road junction has a relationship with the traffic at the adjacent moment, so using historical data of the time series based on MTL, sigmoid function, and Backpropagation (BP) network, which combines the traffic flow of the sequence of contiguous time interval, could enhance the prediction of the traffic flow [31]. Likewise, Huang et al. explained that the road networks are related to each other. They adopted the Deep Belief Network (DBN) to classify the traffic flow non-prior knowledge features and used MTL to train the time series of traffic flow information of road loops detectors and stations concurrently, thus improving the accuracy of the traffic flow detection [22]. The inductive loop detectors in this study were used to detect the vehicles at intersections or toll collection.

2) ROAD TRAJECTORIES TRAFFIC DETECTION

Predicting the whole road traffic trajectories efficiently helps to increase traffic forecasting accuracy. MTL based on Graph Neural Network (GNN) has its effectiveness in predicting both the traffic trajectories, which is the path of the vehicles, and the interactive behaviors like changing the lane and speed based on time series and Long Short-Term Memory (LSTM) [32]. In addition, regarding Ren et al., MTL could be used to enhance the detection of the road trajectory by training concurrently the moving ratio, which is the driver move or speed, and the road segments based on the spatial-temporal dependencies [33]. They used an encoder for the sequential data and a decoder to determine the moving ratio and the road segments in their proposed method.

3) TRAFFIC SITUATION

MTL also shows its effectiveness with traffic forecasting by using a clustering algorithm that considers every traffic situation and reason, such as rush hours, accidents, or construction, as Deng et al. explained [34]. They linked the MTL in their proposed method to traffic situations instead of linking it to sensors, which improves the efficient traffic forecasting prediction. Furthermore, this study illustrates that training the traffic situation and reason tasks independently could affect traffic prediction accuracy because the training samples are limited. These situations have similar features, and using MTL with lasso regularization to select similar features for the tasks could improve the accuracy by preventing overfitting.

4) TRAFFIC TIME

Traffic time in different cities has a relationship because the work, school days, or the holidays, for example, usually are at the same time in different cities. Sharing the trained knowledge among other cities using MTL and combining both temporal and spatial is helpful for traffic forecasting based on Zhang et al. [35]. They used the cities as multiple tasks because many factors in different cities could make the traffic related to each other. The traffic peaks according to work days and holidays in different cities are similar, so sharing this knowledge between cities using the MTL is capable of enhancing accuracy. Also, one city's temporal correlations and spatial dependencies could affect the traffic flow in different cities. 3D convolution kernels are used in this method to find the spatial and temporal dependencies.

Many attributes in the road network are related to each other. Using them individually reduced the accuracy of the prediction of the traffic. Using MTL can lead to tremendous results in traffic forecasting prediction by considering road attributes (speed, traffic flow, travel time, traffic time, traffic situation, traffic trajectory, etc.). MTL has been used with traffic forecasting, showing its ability to increase the reliability of predicting traffic and its attributes and make it more efficient for GPS users.

B. MTL FOR TRAFFIC SIGN

Traffic signs have various purposes (e.g., speed limits, mandatory signs) and many attributes (e.g., colors, shapes). Distinguishing between these traffic signs for autonomous vehicles is considered a challenging problem. Training the traffic sign tasks individually can make progress in differentiating between the traffic signs for autonomous cars. However, it still takes a long time and needs to be more accurate. Training all these tasks together using MTL leads to better prediction of the different kinds of signs because the models can have better knowledge of traffic signs, further increasing the ability to differentiate between traffic signs precisely. Also, optimizing traffic light detection based on MTL is crucial for enhancing the control of autonomous vehicles with traffic signals.

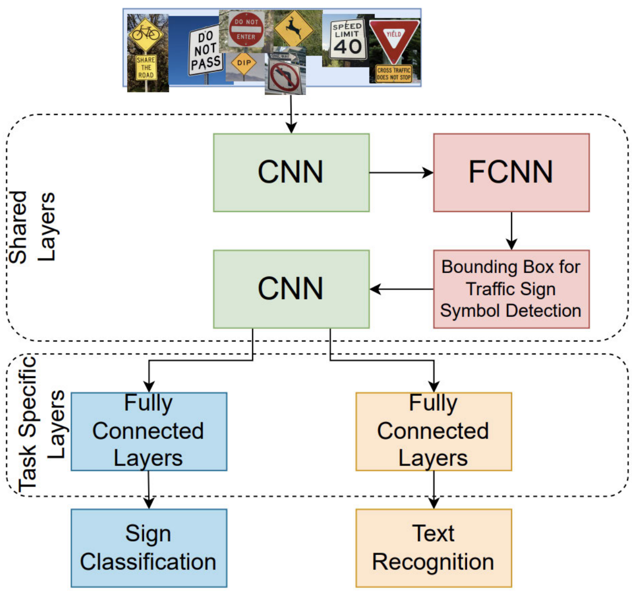


FIGURE 5. Signs classification and texts recognition of traffic signs based on MTL.

1) TRAFFIC SIGNS DETECTION

Traffic signs have different purposes, such as prohibition, speed limits, and mandatory signs with various attributes such as colors and shapes. Lu et al. improved the detection of different signs by dividing the traffic signs into six subsets in a tree structure according to their type and the similarity among the signs [36]. Then, they used MTL to learn

TABLE 1. MTL with traffic forecasting prediction.

Year	Authors	Contribution
2021	BURONI *et al.* [28]	Predicting traffic flow and speed concurrently based on MTL to improve traffic forecasting.
2020	ZHANG *et al.* [29]	Using MTL with GRU to predict the speed and traffic increase the traffic forecasting performance
2021	Li *et al.* [32]	Predicting traffic trajectories and road interactive behaviors based on MTL and GNN improve the traffic prediction.
2017	Deng *et al.* [34]	Combining traffic situations and reasons based on MTL to improve the traffic prediction efficiency.
2021	Zhang *et al.* [35]	Considering the relationships between traffic time of the cities based on MTL improves traffic forecasting prediction.
2009	Sun [30]	Combining the time series of the road traffic flow information based on MTL to enhance traffic forecasting.
2008	Jin *et al.* [31]	Combining the relationship of traffic flow of a road with an adjacent road based on MTL to enhance the traffic forecasting prediction.
2014	Huang [22]	Predicting concurrently the traffic flow time series of road loop detectors and station data based on MTL and DBN to increase the traffic forecasting accuracy.

these tasks concurrently. Their proposed method confirmed that MTL based on altering direction method of multipliers (ADMM), which is used for complex features optimization, could improve the performance of traffic signs prediction. Using MTL can also assist in reducing the processing time for autonomous cars to detect traffic signs and surrounding vehicles, and recognize lane segmentation, concurrently as Bui et al. showed [37]. Their proposed method extracted the features of traffic signs, lane segmentation, and surrounding cars tasks through a single encoder. Also, they used traffic signs and object detection in one decoder and lane line segmentation in another decoder based on MTL, increasing the detection efficiency performance for these tasks. As shown in Fig. 5, MTL can be used for concurrently sign classification and text recognition of traffic signs. A bounding box detects extracted traffic sign symbol features in the images based on a Fully Connected Neural Network (FCNN). The fully connected layer for each task will be used to transfer the extracted features into grouped representations to recognize the text and classify the traffic signs.

Moreover, traffic signs have different types of signs such as symbols and text data (the letter in the traffic sign), and both of them need to be detected accurately to increase the traffic sign classification. Using them together improves traffic sign detection based on Luo et al. study [38]. Luo et al. studied that traffic signs detection performance could be improved by using Maximally Stable Extremal Regions (MSERs) to extract traffic sign regions of interest (ROI) and MTL based on CNN to detect the traffic sign texts and symbols images and the street views. In addition, Qian et al. explained that MTL effectiveness is not only in the traffic sign detection but also in detecting the text in the traffic sign [39]. Their proposed method used RGB images (red, green, and blue) with input images and connected component analysis (CAA) to group the related pixels. Additionally, they employed MTL based on CNN and Rectified Linear Units (ReLU) to improve the accuracy of features and image detection for the traffic sign and the text.

2) TRAFFIC LIGHT DETECTION

The traffic light detection for autonomous vehicles was improved by using MTL based on encoder and decoder [40]. The study found that MTL based on the Conditional Imitation Learning (CIL) framework could lead to the improved efficiency of traffic light detection. It trains and learns together road scenes (based on depth estimation and semantic segmentation) and traffic lights (yellow, green, red), as subtasks with driving control, as a primary task, based on RGB camera images. The driving control is adopted in this method to predict the steering, throttle, and brake of the autonomous car.

The various features of traffic signs can complicate the detection of these signs. Before, there were many issues of detecting the signs by autonomous vehicles because of the limitations of the knowledge. On the other hand, MTL is capable of sharing the features of various traffic signs with each other, improving the detection percentage performance. Taking into account different attributes of the traffic sign (prohibition, speed limit, mandatory sign, traffic light, etc.), and additional traffic sign features (e.g., color, shape) based on MTL could increase the knowledge among tasks. In this way, the traffic signs prediction performance can be improved while the low processing time of the traffic signs detection can be maintained.

C. MTL FOR VEHICLE RECOGNITION

The vehicles have different features, and they share similar features (color, shape, model, etc.). Training these tasks individually could affect the accuracy of vehicle recognition because of the limitations of the model. In contrast, training them together by MTL can improve the model knowledge and lead to improved recognition detection accuracy. Fig. 6 illustrates a typical way to use MTL to predict concurrently the vehicle type, color, and license plate. The vehicle image features are extracted by using CNN, and fully connected layers are used to transfer the vehicle features into grouped pixels. Each task has a dedicated task-specific layer based on

TABLE 2. **MTL with the traffic sign detection.**

Year	Authors	Contribution
2022	Bui *et al.* [37]	Predicting the traffic signs, lane segmentation, and surrounding cars concurrently based on MTL to decrease the detection processing time.
2017	Lu *et al.* [37]	Dividing the traffic signs into different subsets based on their type and training them by MTL to enhance the traffic sign detection.
2018	Luo *et al.* [38]	Predicting the text and symbol traffic sign features based on MTL to improve the traffic sign detection accuracy.
2020	Qian *et al.* [39]	Training both the traffic sign and the text in the traffic sign based on MTL to enhance the detection accuracy of the sign.
2021	Ishihara *et al.* [40]	Training road scene data (based on depth estimation, semantic segmentation), traffic light data (yellow, green, red) and driving control concurrently to improve the traffic light prediction.

fully connected layers architecture to predict the output of the type of vehicle, the color of the vehicle, and the license plate number. For the license plate number task, Single Shot MultiBox Detector (SSD) is used to detect the license plate by using the bounding box. Connected Component Analysis (CCA) is leveraged to group the pixels of the license plate characters.

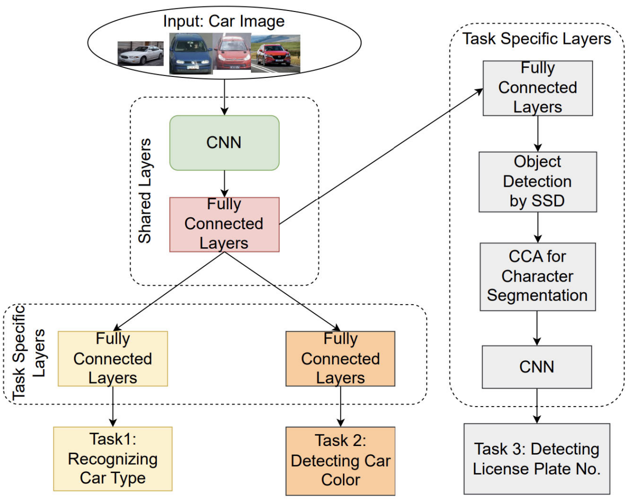


FIGURE 6. **Predicting vehicle's color, type, and license plate together to increase the accuracy for vehicle recognition based on MTL.**

1) SIMILAR VEHICLES DETECTION

Different types of vehicles have similar shapes, making it difficult to recognize their type and model. The use of CNN based on MTL and a large number of car images as input to classify different features and train-related or similar types of vehicles make it efficient to predict both the kind and the year of the cars concurrently. It can achieve a better result than using STL as Avianto et al. explored [41]. This study used the VGG architecture algorithm to extract the vehicle image features. In addition, Sun et al. used MTL based on CNN to solve the similar car appearance by train together the brand and color [42]. Softmax was used in this method to classify the pixel images into different classes, representing the brand and the color of the vehicles. Their results showed training the vehicle's characteristics jointly increases the efficiency of detecting the car. Also, Huo et al. confirmed that using various views of the vehicles, such as the front, back, and side, with the lightning condition of the cars perspective as semantic global attributes using MTL to determine the type of vehicles show an incredible result in the detection performance [43]. Semantic global attributes are known and shared between different classes within the neural network. Region-based Convolutional Neural Network (RCNN) was used in this method to determine the vehicle classes. In addition, Chen et al. showed that vehicle recognition could be more accurate when MTL is used to concurrently classify in detail the type of vehicles as the main task with detecting the vehicle from different angles by 3D bounding box as auxiliary tasks [44].

With the massive amount of data and the similarity of vehicles, searching for a specific kind of vehicle is more complicated. Deep hashing based on DNN, which shows its efficiency for the search for similar images [45], employed ReLU as an activation function to enhance the learning process. The method also utilized GoogLeNet to improve image classification. It implemented MTL to concurrently learn the vehicle's ID, color, and model, effectively addressing the issue of similarity search for images as Liang et al. illustrated [46].

Moreover, using the heatmap and instance segmentation based on MTL to train the attributes of the vehicle (e.g., model, color, orientation) concurrently or pose with different backgrounds to concentrate on the related information shows its effectiveness in recognizing and detecting the differences between the similar cars in different angles [47]. Also, Rajamanoharan et al. illustrated that training the vehicle identification (vehicle ID) alone is not sufficient to detect the vehicles properly because many cars have the same shape [48]. They proposed an efficient method that utilizes MTL based on CNN and FNN to address the issue. Their approach concurrently trains vehicle ID, multi-scale that re-scaled the vehicle ID image to improve the accuracy, and gray-scale image, which concentrates on the vehicle information and orientation. To classify the vehicle ID and orientation, their proposed method employs Softmax.

2) VEHICLE LOGO AND LICENSE PLATES DETECTION

The vehicle logo and license plate detection have been detected powerfully after using MTL. For example, Xia et al.

showed that employing Multi-Layer Perceptrons (MLP) for classifying logos of different vehicles and predicting logo attributes could concurrently be based on MTL enhances the efficiency and performance of vehicle logo detection [49]. The MTL approach is used to predict segmented vehicle logos and logo attributes, such as central symmetry, X-axis, and Y-axis Symmetry, which determine whether the vehicle logo, based on its brand, possesses the presented attributes. In addition, MTL contributes to enhancing the detection accuracy of vehicle license plates based on Goncalves et al.'s study [50]. Their study shows that by performing segmentation to focus on the license plate characters and recognition to detect the characters of the license plates concurrently based on MTL, the performance for detecting low-resolution license plate images can be achieved.

3) AMBIGUOUS VEHICLE APPEARANCE DETECTION

In prior works, independently detected vehicle ROIs have been shown to impact vehicle prediction accuracy, especially if the vehicle's appearance is unclear. Detecting vehicles' ROIs concurrently based on MTL could enhance vehicle detection even if the vehicle's appearance is ambiguous [51]. They used many tasks for the training, such as subcategory, which trains the model in different vehicle views, region overlap, which enhances the measurement of the vehicle detection, bounding-box regression, and category of each training RoI of vehicle attributes based on MTL.

Furthermore, detecting instance segmentation of the vehicle is considered a challenging problem. Training concurrently both vehicle semantic segmentation (which is used to help identify each object around the vehicle's image), and semantic boundary detection (which is used to distinguish between the vehicles at various distances based on MTL) could improve the performance of the vehicle instance segmentation detection [52].

Nowadays, the detection of the vehicle is burdensome because of the similarities of the vehicle's shape, type, colors, logo, etc., and ambiguous vehicle appearance. Sharing the features of the vehicles between the tasks and training them concurrently based on MTL can enhance the knowledge of the vehicle's type and help the models distinguish between the vehicle's types accurately.

D. MTL FOR TRAVEL TIME ESTIMATION

Travel time estimation is an important factor for the GPS to increase the efficiency of estimation arrival time. It has some challenges because many attributes must be included, such as changing the traffic conditions or the driver's preferences, departure time, arrival time, transportation modes, and days. Considering these attributes based on MTL can enhance the detection of estimated arrival time.

1) ROAD HISTORICAL INFORMATION

The previous methods focus more on using road distance information without considering some historical road attributes. For example, Li et al. proposed a technique that enhances the prediction of real-time travel time by taking all road information history, including the distance with traffic lights and turns as auxiliary tasks based on MTL [53]. The road information in their study includes the spatiotemporal dependencies and prior knowledge of the road network (e.g., the source, destination, departure, travel time, and path). The spatial knowledge is represented as a graph that illustrates the regions and the temporal graph that represents the times and days.

2) TRANSPORTATION MODES

The travel estimation time is different from one transportation mode to another. Using the travel time of transportation modes (i.e., buses, riding, bike, or walking) separately could reduce the travel estimation time for each mode to the users' accuracy on the GPS. Nonetheless, incorporating these transportation modes' tasks together based on MTL and the travel time could increase the knowledge of the transportation's travel time. That can further improve the travel estimation time on the GPS devices, according to the transportation modes used by users based on Xu et al. paper [54]. They concurrently combined the travel time with transportation modes recommendation, leveraging historical road segment paths with spatial-temporal dependencies information by considering the week, holidays, and weekend days based on MTL.

3) DRIVERS BEHAVIOR

The travel estimation time can be affected by the driver's behaviors like speeding, sudden braking, and lane change. These behaviors need to be determined to make the travel time prediction more accurate and improve the GPS navigation prediction. For example, Gao et al. illustrated a method that improves the travel time estimation prediction based on the driver behavior and links road data collected by GPS, road network, and sensors that measure the vehicle's motion as inputs. MTL based on LSTM is used in their method to predict jointly the travelers' speed behaviors and the travel time to enhance the travel estimation time [55].

4) SPATIO-TEMPORAL DEPENDENCIES WITH CONTEXTUAL-INFORMATION

The travel estimate time is also considered a challenge because of many elements involved, such as spatial and temporal data and weather or weekdays. Predicting the local path with the entire path concurrently and considering the spatial and temporal dependencies and the other conditions, such as weather or weekdays raises the quality of the accuracy of predicting the travel estimate time [56]. Using both traffic prediction with spatial and temporal graphs and contextual information, which is the relation between roads, based on MTL could enhance the travel estimation time [57]. Yang et al. assured that the temporal and spatial (distance between cars, road segments, speed, traffic, etc.), as well as

TABLE 3. MTL for vehicles recognition accuracy.

Year	Author	Contribution
2022	Avianto *et al.* [41]	Training concurrently the year and the kind of the vehicles to improve the vehicle recognition performance.
2019	Sun *et al.* [42]	Predicting the vehicle's color and brand by MTL to improve the detection of similar vehicles
2016	Huo *et al.* [43]	Increasing the vehicle's type detection by combining different views of the vehicles and considering the lighting conditions of the car's perspective using MTL to improve the car recognition performance.
2020	Chen *et al.* [44]	Classification of the vehicle's type with the vehicle maker and model using MTL to enhance the prediction of vehicles with similar appearance.
2016	Liang *et al.* [46]	Combining learning vehicle ID, color, and model concurrently based on MTL to increase the vehicle recognition accuracy.
2019	Tang *et al.* [47]	Training the vehicle attributes like model, color, and orientation concurrently based on MTL to enhance the vehicle recognition.
2019	Rajamanoharan *et al.* [48]	Training concurrently vehicles ID and the orientation of the vehicles using MTL to increase the vehicle recognition efficiency.
2016	Xia *et al.* [49]	Training various types of vehicle logos and logo attributes using MTL to improve the detection of vehicle's logo.
2019	Goncalves *et al.* [50]	Combining the segmentation of the license plate characters and recognition of the license plate characters concurrently to enhance the license plate detection.
2019	Chu *et al.* [51]	Training different vehicle views, region overlap, bounding-box regression, and category of the vehicle's attributes based on MTL to increase the vehicle recognition performance.
2018	Mou *et al.* [52]	Training concurrently both vehicle semantic segmentation and semantic boundary based on MTL to improve performance of the vehicle instance segmentation detection.

the road segments (e.g., highway or local street), are related to each other and they can affect the travel time [58]. Using MTL based on LSTM and GCN, for dealing with temporal and spatial data, to train them concurrently increases efficiency. In addition, Xu et al. [59] illustrated the use of the LSTM to extract the temporal dependencies for the travel time and traffic speed estimation by using the traffic speed history. Their study showed that MTL based on CNN to train them concurrently demonstrates its progress in the prediction of the future estimate for travel time and traffic.

Dividing the road segments and intersections into separate models affects the travel time estimation. In this direction, Jin et al. showed that MTL increases the performance of time estimation of the whole road [60]. They explained that the relationships of road segments and intersections are extracted by using spatial and temporal graphs. The graphs contain GCN, temporal convolution network (TCN), and GRU to capture spatial and temporal knowledge. Then, the MTL is used to train both the road segments and intersections based on the data on the graphs concurrently.

Travel estimation time for the GPS includes many factors that can affect its performance. Considering elements (weather, intersection, departure time, speed, etc.) based on MTL could enhance the estimation of the arrival time because the model can have complete knowledge about the road conditions, improving travel time estimation.

E. MTL FOR ROAD SAFETY

Object detection on the road is considered a huge challenge, especially for autonomous vehicles. Most previous methods used to detect road risks were collecting the information manually, which had enormous errors. Also, recognizing objects, such as vehicles surrounding the autonomous car, had some obstacles because the related object's features were trained separately in different models, resulting in negative detection performance. Nonetheless, considering all the road features (road objects, curved and straight roads, intersections, traffic signs, etc.) and training them together using MTL could improve car safety on roads. Fig. 7 illustrates the detection of the drivable area and the vehicle surrounding the autonomous car. The images of the drivable area and vehicles will be extracted using an autoencoder to emphasize specific areas (e.g., surrounding vehicles and drivable area) within the image. Feature pyramid network (FPN) is used as task-shared layers to make the extracted features clearer to the network, such as determining the shape and size of the object. Two decoders are used that contain CNNs and FCNNs to group the features on the image for the network. FCNNs are utilized as task-specific layers for each respective task. One FCNN is dedicated to detecting the drivable and non-drivable areas through semantic segmentation. Additionally, another FCNN is employed to perform car detection using bounding boxing.

1) ROAD CONDITIONS PREDICTION

Detecting road conditions is essential in maintaining autonomous vehicles' safety. By using MTL, the accuracy of road detection could be increased. For example, Li et al. showed that MTL could have a significant impact on road condition prediction by proposing a method that concurrently trains both geometric deformations of road segments using a data augmentation algorithm, which determines the shape of straight and curved roads, and segmentation optimization of the road samples based on encoder [61]. These samples

TABLE 4. MTL for travel estimation prediction.

Year	Author	Contribution
2018	Li *et al.* [53]	Including the road information history with spatial and temporal data based on MTL to increase the travel time estimation prediction.
2022	Xu *et al.* [54]	Combining the travel time and transportation modes based on MTL to improve the travel time estimation performance for each mode.
2019	Gao *et al.* [55]	Combining both the travel speed and the driver's behaviors based on MTL to increase the estimation of travel time.
2018	Wang *et al.* [56]	Predicting concurrently the local path with the entire path including the spatial and temporal to enhance the travel time estimation.
2022	Yang *et al.* [58]	Using Spatial, temporal element, and related road segments with MTL to improve travel time prediction.
2021	Xu *et al.* [59]	Predicting jointly the temporal and the spatial knowledge, and the traffic speed history based on MTL to increase the travel time estimation.
2022	Fang *et al.* [57]	Combining the spatio-temporal dependencies and the road contextual information using MTL to increase the travel time estimation accuracy.
2021	Jin *et al.* [60]	Combining road segments and intersections data using MTL to enhance the travel time estimation performance.

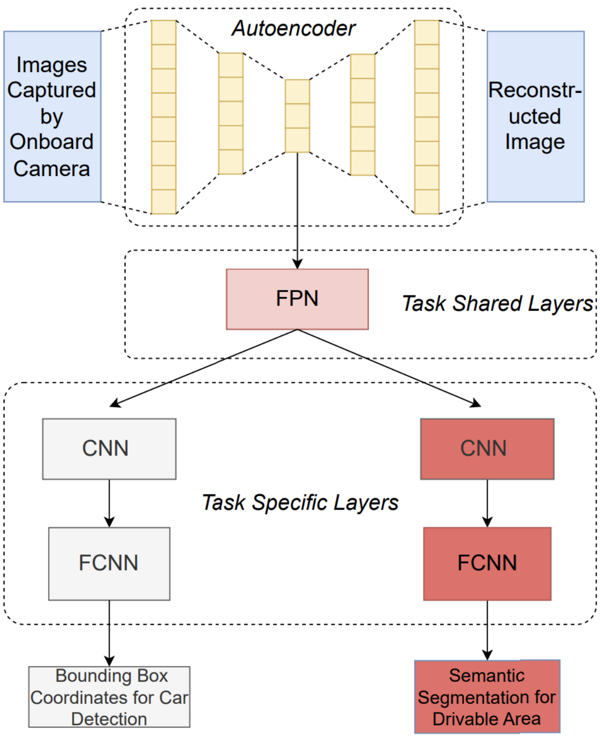


FIGURE 7. The detection of the drivable area and the vehicles that are surrounding autonomous vehicles concurrently based on MTL to enhance the autonomous car driving safety.

are divided into road and background to enable precise detection for any road vision condition based on MTL and ResNet, which solve the pattern complexity by skipping some layers. This combination enhances road detection and safety for autonomous vehicles. Furthermore, predicting road conditions, especially in winter, is an essential factor for car safety. Likewise, Liu et al. showed that MTL could help improve the detection of the road [62]. They train together the road segments as auxiliary tasks to concentrate on the road objects and friction estimation coefficients as the main task. Friction estimation coefficients were used based on datasets that contain images of the road surface. This helps enhance the detection of the road objects in the image. Also, Atrous Spatial Pyramid Pooling (ASPP) was used to improve the accuracy of the semantic segmentation of the image.

2) ROAD ASSESSMENT

Automatic recognition of road conditions is vital for increasing road safety. MTL has improved road safety assessment by enhancing the automatic recognition of road safety rather than relying on manual assessment. MTL makes road assessment automated by concurrently training usRAP, which is used to rate the road conditions, along with auxiliary tasks such as intersection, lane numbers, and road conditions as auxiliary tasks based on Song et al.'s study [63]. Their experiments showed that using MTL with the CNN-based VGG architecture helps enhance the precision of road rating and improve the estimation of some road risk attributes, including roadside hazards and lane width, by utilizing attention layers. Additionally, MTL based on CNN could improve the performance of iRAP, a charity aiming to rate road safety and enhance road assessment, by concurrently training iRAP attributes derived from segments of sequential video input [64]. The attributes of the iRAP methodology are divided into seven categories: road details, observed flow, speed limit, road middle-side objects, roadside objects, intersections, and road user vulnerability factors such as pedestrians.

3) REDUCING CAR ACCIDENTS

MTL has reduced the possibility of car accidents by predicting the risk of traffic accidents. Predicting both fine-grained, which predicts the detailed traffic accidents (e.g., the accident location) and coarse-grained, which predicts the traffic accidents in general (e.g., the accidents rate) concurrently based on MTL has its effectiveness to improve the accidents risk prediction as Wang et al.'s study showed [65]. Based on their study, fine and coarse-grained accidents with GCN and temporal dependencies make progress for traffic risk prediction. Their proposed method used GCN to predict the spatial risk accidents if they are low, medium, or high and to represent the road and the region in a graph. Also, their proposed method could capture the temporal dependencies by LSTM. In addition, autonomous cars have to detect the road carefully to avoid car accidents. To improve road detection, Lee proposed a method that predicts drivable area and lane line simultaneously by classifying pixels in the images using an encoder and decoder based on semantic and instance segmentation, respectively. Additionally, it classifies pixels in the image to determine the road type [66] as shown in Fig. 7.

4) CAR DETECTION OF ROAD USERS

MTL has improved the detection of the people or cyclists who intend to cross the road by combining their actions, such as stopping for traffic, waiting for other cars, or walking, their crossing intents, and the road trajectory [67]. These actions can be extracted by detecting pedestrians from video sequences by using semantic segmentation and 2D pose. This enhancement has improved accuracy in predicting future crossing. Saleh et al. showed progress in predicting the intent of road users from image classification by using a bounding box by training both head and body positions for pedestrians concurrently using MTL and CNN [68].

5) CARS AND DRIVERS BEHAVIOR DETECTION

The autonomous cars' behavioral modes (stopping, intersection crossing, lane following, etc.) shall be considered together. Training all of them by using MTL enhances the performance of the autonomous vehicle and increases the safety of the use of the autonomous vehicles [69]. In addition, identifying and understanding driver behaviors such as sudden braking, speeding, and lane changing are considered challenges. It is a crucial factor for vehicle driving safety. Moreover, MTL enhances the prediction of the driver's behavior based on the driving styles data, including aggressive, calm, and moderate driving. Using GCNNs, which uses fine-grained driving behavior at a detailed level, and semi-supervised learning to classify the driving styles based on MTL, showed its ability to enhance the predicting of the driver's behavior by concurrently training the diver's styles data [70]. That further increases the safety of autonomous vehicles.

6) ROAD OBJECTS DETECTION

Detecting road objects is essential for autonomous vehicles' safety. Detecting the road objects and predicting their distance (pose, size) independently can negatively affect road object detection performance by missing crucial knowledge. This missed knowledge could result in weak prediction of the road objects and their distance, further posing accidents and hurting the autonomous cars' safety. To this end, Chen et al. [71] used MTL based on Cartesian product to perform all possible combinations of both road objects by using bounding box and their distance classes concurrently, and their experiments showed that using both tasks together enhances the accuracy of the road objects detection and distance detection.

7) INSPECTION ACCURACY

MTL plays an essential role in increasing inspection accuracy by automating detection and reducing human mistakes, ultimately improving safety. For example, Gibert et al. demonstrated how MTL could enhance railway inspection by proposing MTL to solve the problem of the segmentation of the railway detectors [72]. By using MTL, they trained multiple railway detectors concurrently, including good, bad, or missing fasteners, and also chips and crumbling concrete ties. Their results showed that MTL could enhance the accuracy of the railway detector inspection. Note that car accidents have risen due to the low quality of detecting risk objects and the surrounding cars. MTL has been used to automate detection and train the different road objects concurrently, resulting in improved detection and progress in road safety.

F. MTL FOR AUTONOMOUS DRIVING

Detection of objects or cars that are surrounding the autonomous vehicles precisely is considered one of the most crucial factors that increases the capability of autonomous vehicle's performance. Training these objects or surrounding cars separately by STL can reduce the efficiency of the autonomous vehicle's detection due to the limitation of the knowledge among the tasks. Nonetheless, sharing information and features between tasks based on MTL increases the accuracy and performance of autonomous vehicles. This concept is shown in Fig. 8, illustrating the prediction of steering angle and the speed control tasks concurrently. Steering angle features are extracted by using CNN, while LSTM is used to capture the sequential data of the speed control task. Fully connected layers are used as shared layers to transfer the pixels of the features into grouped pixel representations. Task-specific layer for each task helps to predict the output values for both the speed and the steering angle.

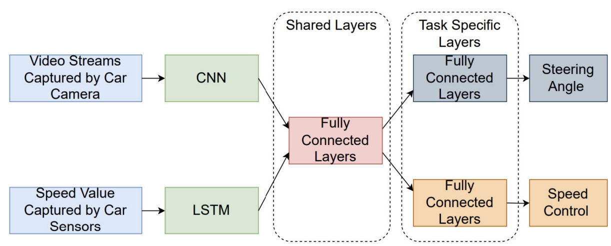


FIGURE 8. Predicting steering angle and speed concurrently using MTL to enhance the autonomous car control.

TABLE 5. MTL for road safety improvement.

Year	Author	Contribution
2021	Li *et al.* [61]	Using geometric deformation of road segments and optimization of the road samples based on MTL to increase the efficiency of road condition detection.
2022	Liu *et al.* [62]	Predicting together road segments and friction estimation of the road based on MTL to increase the road detection performance
2018	Song *et al.* [63]	Using usRAP with auxiliary tasks such as intersection, lane numbers, and road conditions based on MTL to increase the road safety assessment.
2020	Kačan *et al.* [64]	Training concurrently iRAP attributes based on MTL to enhance the rating of the road.
2021	Wang *et al.* [65]	Combining fine and coarse-grained traffic accident data concurrently based on MTL to enhance the future accident prediction.
2021	Lee [66]	Combining the derivable areas and lanes tasks based MTL to increase the driving safety.
2020	Ranga *et al.* [67]	Combining the pedestrian's actions, cross intent, and road trajectories based on MTL to enhance the detection accuracy of pedestrians.
2017	Saleh *et al.* [68]	Combining both head and body positions for the pedestrians concurrently based on MTL to enhance the detection of the intent of crossing the road.
2019	Chowdhuri *et al.* [69]	Training different vehicle behavior modes based on MTL enhances the safety performance of autonomous vehicles.
2016	Chen *et al.* [70]	Predicting fine-grained driving behavior and classifying the driving styles concurrently based MTL enhances the detection of the driver's behavior to improve the vehicle safety.
2018	Chen *et al.* [71]	Combining the objects detection and distance prediction based MTL to enhance the road object detection.
2016	Gibert *et al.* [72]	Combining the railway detectors based on MTL to improve the inspection accuracy.

In addition, Abbas et al. illustrated that combining the autonomous car tasks performs better than predicting them separately [73]. They confirmed that combining the segmentation of brake, steering angle, accelerating of autonomous cars, and estimating the lanes around the vehicle based on MTL could lead to better performance in improving the prediction of these tasks. The different behavior of the tasks in this study was solved by an inverse validation loss weighted scheme based on the normalization scheme. Their designed scheme aims at enhancing the performance of tasks with lower accuracy to establish interrelated task features.

1) SURROUNDING CARS DETECTION

The accuracy of detecting cars surrounding autonomous cars is essential to increase the efficiency of the autonomous car's performance. Including many factors concurrently could enhance the efficiency of detection. Using MTL is effective in concurrently predicting the semantic segmentation for labeling each object in the image, depth estimation for object distance measurement, light detection and ranging (LiDAR) segmentation for distance measurement in different environmental conditions, and bird's eye view to provide a comprehensive view of the environment that around the autonomous vehicles [74]. The imbalanced learning problem, which leads to having a preference for one task more than others, was solved in this method by a loss weighting algorithm.

Furthermore, combining semantic segmentation, boundary prediction to provide instance segmentation of various objects, and object detection by using a fisheye image, which is used to capture a comprehensive view around the vehicle, based on MTL, improves the detection of the objects around the autonomous vehicles [75]. Moreover, the fisheye images show its effectiveness through the use of MTL, which enables concurrently prediction of six tasks: depth estimation, visual odometry to capture the object movement around the car, semantic segmentation, motion segmentation to differentiate between car movement and other objects movement, object detection, and lens soiling detection to detect when the camera vision is not clear, for example, by dirt [76], synergized decoders was used in this method to make the tasks related to each other.

2) UNEXPECTED SITUATIONS DETECTION

There are unexpected situations such as sudden stops or turns that may affect the use of autonomous vehicles. However, considering the future prediction of unforeseen situations with considering the autonomous vehicle's control and position based on MTL enhances the forecast for these situations. For example, Kim et al. proposed a method that shows the effectiveness of including autonomous vehicle's control and position tasks concurrently based on MTL with the unexpected situations prediction task [77]. They divided the tasks into positional and heading angle tasks, which focus on the position and orientation of the vehicle, and longitudinal and lateral tasks, which control the speed and the direction of the moving vehicle.

3) LANE LINE DETECTION

The lane line detection of autonomous vehicles is improved by using MTL. Many tasks have to be considered in lane line detection (e.g., multilabel classification), which classifies the image of the lane line; grid box regression, which detects and locates the lane line; and object mask, which focuses on the line and eliminates the background. For example, Li et al. [78] showed that using multi-label classification, grid box regression, and object mask concurrently based on MTL could increase the quality of the lane line prediction for autonomous cars.

4) AUTONOMOUS CARS DIRECTION

Many factors (steering angle, speed, etc.) affect an autonomous car's direction or motion control. Training them separately can decrease car control performance because they are related. Combining the steering angle and the speed detection of the autonomous car based on MTL enhances car control [79]. Furthermore, Li et al. showed, in their study, the effectiveness of MTL in improving autonomous car's control by predicting concurrently the distance to lane marking, the heading angle distance between the car and lane, and track direction [80]. Moreover, MTL contributed to the growing quality of predicting objects and instance segmentation. Likewise, Chang et al. showed how MTL could enhance the car control decision by using an encoder and decoder based on ShuffleNet and FPN to track and predict the objects of the road and instance segmentation [81].

5) PEDESTRIANS DETECTION

Pedestrian detection from far distances is considered a challenging issue for autonomous cars. Nonetheless, the quality of the pedestrian detection has been improved by concurrently using Region Proposal Network(RPN), which is used for objects' location detection on the images [82], to find semantic segmentation of the pedestrians. Region-based Fully Convolutional Network (RFCN), which detects objects within an entire image [83], refines the instance segmentation that is in RPN by using a bounding box based on MTL [84].

One of the most crucial things in the autonomous car is detecting objects accurately. Having these objects in different models will reduce the accuracy performance because of a lack of knowledge among these tasks. Sharing this knowledge in one model using MTL could improve the detection of objects accurately and the control of autonomous cars.

G. MTL FOR TRANSPORTATION DEMAND PREDICTION

Demand transportation prediction has some problems because of the need for more prediction of the distance between the transportation and the users, which is considered one of the most crucial factors for both users and drivers. Nowadays, many factors could affect the prediction of the distance accuracy, such as traffic, and can cause longer waiting time for users. Taking into account the different road features together by MTL can be helpful to address this issue and lead to tremendous prediction results.

1) TAXI DEMAND PREDICTION

Considering that the existing study did not concentrate on the relationship among the region's data, which negatively affects the accuracy of passengers' taxi demand, and features such as weather and holidays were not carefully considered, Bai et al. showed a study that helps to increase the accuracy by using the historical information of passengers' demand according to regions and using CNN to obtain similar spatial and LSTM for temporal relationships and focus more on the holidays and weather features to be part of the MTL [85]. Also, the passenger demand is either high or low estimated.

There is a relationship in the taxi demand prediction between pickup and drop off the passengers. For example, Zhang et al. noted that training both of these tasks together based on MTL-based CNN and LSTM, which identifies the spatial and temporal dependencies for these tasks, is critical to have a better prediction of the passenger's taxi request and reducing the passengers' wait time [86]. Also, Kuang et al. assured that using LSTM with MTL and 3D CNN to find spatiotemporal features and having pick up and drop off as related tasks by taking the weather, days, and transportation conditions into account could increase the accuracy of predicting the taxi demand for passengers, driver, or taxi demand applications [87]. Likewise, Wu et al. illustrated that using both spatial and temporal graph attention networks (GTA) together to find the relationship among the road regions to capture the taxi pick up and drop off information and train them concurrently with MTL make improvement better than using only the spatial dependencies [88].

Combining many road zones with considering the temporal and spatial dependencies could improve taxi demand prediction. For example, Luo et al. showed that dividing the road zones and training them concurrently based on MTL and spatial-temporal dependencies of the zones have the efficacy of the taxi demand prediction [89]. The relationships of the temporal and spatial dependencies among the zones in their study were acknowledged by nonlinear Granger causality analysis, which measures the relationship between different time series data based on LSTM.

Furthermore, using GCN to find the spatial dependencies by taxi trips on the network and finding the temporal dependencies based on LSTM and then train them using MTL could lead to better taxi demand prediction as Chen et al. explained [90]. They illustrated that the spatial dependencies in their method are local and global, which are the relationships of taxi departure or arrival flow of the adjoining and disconnected roads, respectively.

Moreover, there are various service modes like shared or unshared taxi for the taxi demand platforms (Uber, Lyft, etc.), which are highly related. For example, Ke et al. proposed Multi-Graph Convolutional (MGC) to capture different taxi demand services for taxi rides and then use MTL to train together the knowledge captured by MGC [91]. Also, they used Multi-Linear Relationship (MLR) and Regularized Cross-Task (RCT) learning, which are MTL structures, to determine how the knowledge is shared among tasks.

Considering the original distance (OD) in taxi demand prediction is vital as it provides more detailed information about the trip of the taxi. Using the OD with the road zones of the riding requests based on MTL concurrently is considered necessary to predict the taxi demands of future trips in a short time [92]. The collection of the relationship of the spatial and temporal dependencies in this study for the road zone uses a Mixture-Model Graph Convolutional based on GCN. Also, Wang et al. showed that training the OD trips between

TABLE 6. MTL for autonomous vehicles.

Year	Author	Contribution
2021	Abbas *et al.* [73]	Combining brake, steering angle, accelerating of the autonomous cars, and estimating the road lanes tasks based on MTL to enhance the prediction performance of these tasks.
2022	Natan *et al.* [74]	Concurrently predicting the semantic segmentation, depth estimation, light detection, and ranging (LiDAR) segmentation by using fish eyes view to enhance the detection of surrounding objects for autonomous cars.
2019	Arsenali *et al.* [75]	Combining semantic segmentation, boundary prediction, and object detection to improve the detection of surroundings around the autonomous cars.
2020	Kim *et al.* [77]	Enhancing predicting unexpected situations for the autonomous cars by including the car control task and car position task with unexpected situations task based on MTL.
2021	Li *et al.* [78]	Improving lane line detection by using together multilabel classification, grid box regression, and object mask based on MTL.
2018	Yang *et al.* [79]	Improving the car control and direction by combining steering angle and the speed detection of the autonomous car based on MTL.
2019	Li *et al.* [80]	Improving the autonomous car control by predicting concurrently the distance to lane marking, the heading angle distance between the car and lane, and tracking the direction.
2021	Chang *et al.* [81]	Increasing control decision of the autonomous car by predicting the tracking of the road objects and instance segmentation based on MTL.
2022	Zhou *et al.* [84]	Improving pedestrians detection by concurrently identifying semantic segmentation of the pedestrians and refining the instance segmentation based on MTL.

different areas and the spatial information concurrently based on MTL could improve the prediction of taxi demands [93].

2) PUBLIC TRANSPORTATION DEMAND PREDICTION

Public transportation has related temporal and spatial dependencies. Predicting different modes (e.g., busses, light trails, ferries) individually will negatively affect the transportation demand prediction because important information could be missed while predicting the accurate distance of each mode for the user. Using the knowledge from station-intensive transportation modes such as buses with station-sparse transportation modes (train, light rail, and ferry) concurrently based on MTL could increase the transportation demand accuracy for each mode as Li et al. illustrated [94]. LSTM was leveraged to capture the temporal information in their method based on a Memory-Augmented Recurrent Network (MARN), to retrieve the historical data for the modes.

In addition, considering the relationship between the spatial and temporal in transportation modes such as subway or taxi apps is essential in the transportation demand prediction to make it more accurate. For example, Liang et al. illustrated that train both spatial dependencies, which is captured by using the relationship of the intra-modal relation graph and inter-modal relation graph, and temporal dependencies, which are captured by using historical time series data, based on MTL, could increase the prediction accuracy of transportation demands [95]. The inter-modal relation graph points to the relationship of the spatial dependencies in the same transportation mode. In contrast, the inter-modal relation graph points to the relationship of the spatial dependencies between the different transportation modes.

Furthermore, time series is considered an essential part of smart transportation as it could affect the travel demand estimation in public transportation. It shows its efficiency when used with MTL because it takes advantage of all the time series information in the travel demand. Chidlovskii showed the preference of using MTL over STL to train the time series in public transportation concurrently by measuring the passengers riding public transportation over a specific period of time [96]. In their proposed method, many factors that could affect the travel demand (weather, weekdays, and holidays) are considered. Dynamic time warping was also leveraged in their method to capture the similarity or the correlation among time series. Also, support vector regression was used in their method to select appropriate predictions for the dependent time series values by finding appropriate weights for the time series data.

Factors (pickup, drop-off, road networks, spatiotemporal dependencies, etc.) shall be included in one or shared models to make the transportation demand prediction accurate. Having the knowledge and sharing it among these factors by using MTL showed progress in the demand prediction accuracy in many areas, and it could reduce the user's waiting time as well.

IV. DISCUSSION AND FUTURE CHALLENGES

Through the literature, we have indicated that using MTL can enhance prediction and performance by sharing the tasks' features. In this section, we discuss the significant benefits and limitations of using MTL in smart transportation systems and possible challenges that require further study.

A. MAIN BENEFITS OF MTL

1) TRAINING EFFICIENCY

MTL has better training capability because it uses a pre-trained model instead of training it from scratch. Normally, in MTL, a new task can be trained on a model that is pre-trained by other tasks. Since such a model is already trained by other related tasks, it can extract informative features that might be useful for the new task. To maintain the quality accuracy of MTL training performance, many factors have to be considered, such as fine-tuning, sharing knowledge

TABLE 7. MTL for transportation demand prediction.

Year	Author	Contribution
2019	Bai *et al.* [85]	Enhancing the taxi demand prediction accuracy by using historical taxi demand information of the regions and their similar spatio-temporal information based on MTL.
2022	Zhang *et al.* [86]	Reducing the taxi demand wait time by combining the pickup and drop off data based on MTL and LSTM.
2019	Kuang *et al.* [87]	Increasing the accuracy of predicting the taxi demand based on MTL and LSTM by combining pick-up and drop-off tasks, including spatio-temporal information.
2020	Wu *et al.* [88]	Improving the taxi demand prediction by using both spatial and temporal information of the road regions and combining the taxi pick up and drop off information based on MTL.
2021	Luo *et al.* [89]	Improving the taxi demand performance by training the road zones concurrently based on MTL and spatio-temporal information of each zone.
2020	Chen *et al.* [90]	Improving the taxi demand prediction by combining both temporal and spatial information based on MTL.
2021	Ke *et al.* [91]	Improving the taxi demand prediction by training the service modes concurrently based on MTL.
2021	Feng *et al.* [92]	Enhancing the taxi demand prediction performance by concurrently including the original distance and road zones of the user's requests based on MTL.
2019	Wang *et al.* [93]	Enhancing the taxi demand prediction performance by concurrently utilizing the original distance and spatial information of taxi trips.
2020	Li *et al.* [94]	Using the knowledge of station-intensive transportation modes with station-sparse transportation modes concurrently based on MTL increases the transportation demand accuracy.
2022	Liang *et al.* [95]	Training both spatial dependencies and temporal dependencies based on MTL to increase the transportation demand prediction.
2017	Chidlovskii *et al.* [96]	Training concurrently the number of passengers and the travel time in the public transportation based on MTL to increase the efficiency of time series in travel demand estimation.

among tasks in an independent model, and learning shared representations [97], [98]. Fine-tuning is the process of adjusting a pre-trained model to make it more suitable for a specific task or dataset. Additionally, sharing knowledge between tasks in multiple models allows them to leverage valuable information across different tasks. Learning shared representations involves training the model to enhance its understanding of common features among various tasks. This leads to improved performance in handling shared features among tasks and enhances the ability to train on multiple tasks concurrently. In this way, the pre-trained model can be easily adjusted to the new task and substantially reduce the training time for the new task. In addition, through the studies, MTL can not only make the training more efficient but also increase prediction accuracy and furthermore reduce overfitting.

2) SCALABILITY

Compared with using STL to deal with each task, using MTL is more practical, especially when the size of data and number of tasks are large. The reasons are as follows. First, the overhead of training a model for each task might be large and acceptable. Having the STL model trained from scratch could take a lot of time and use more computational resources. Inversely, MTL tries to adopt a pre-trained model for new tasks; hence, making learning new tasks much more efficient. Second, MTL might perform better than STL with informative data and high model capacity. With the good design of the MTL model (model structure, number of parameters, etc.), the model can learn informative patterns for different tasks. Based on these factors, utilizing MTL in large-scale data and tasks is efficient and effective.

Regarding the smart transportation system, MTL has enhanced applications' scalability, such as in autonomous cars. Many related tasks need to be detected accurately for autonomous vehicles, including car detection, sign, light recognition, etc., and having a massive number of datasets for these tasks can improve the detection. Also, it increases the ability to distinguish between these tasks, making the prediction faster and decreasing detection errors.

Fig. 9 demonstrates the training time efficiency of MTL compared to STL across different methods. The datasets are used to predict the user demand prediction for taxis in multiple zones. It is evident that leveraging users' demands in various areas as different tasks based on MTL, which includes pick-up time and information as well as drop-off time and information, leads to better training time performance and prediction accuracy. A comparison of different methods shows that multi-task learning temporal convolutional neural network spatiotemporal dynamic time warping (MTL-TCNN (ST-DTW)), MTL-ConvLSTM, and MTL-LSTM achieved 3.16, 5.60, and 5.39 training hours, respectively. In contrast, single-task learning temporal convolutional neural network (STL-TCNN), STL-ConvLSTM, and STL-LSTM achieved 3.70, 6.22, and 6.18 training hours, respectively [99]. These results demonstrate that methods based on MTL outperformed STL concerning training times. Furthermore, this study illustrates that the MTL model has better scalability than STL. It included a large collected dataset, which contains 63 zones, about the users' taxi demand, and it showed MTL

TCNN (STDTW) has better performance with a considerable dataset compared to the other methods, including STL methods.

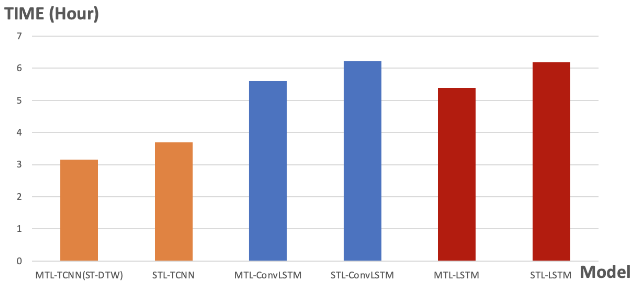


FIGURE 9. Methods based on MTL show better training efficiency and lower training times compared to STL methods.

3) MODEL ROBUSTNESS

MTL increases the strength of the model by training multiple tasks concurrently, which can result in knowledge improvement for the main and auxiliary tasks. This could make the model to have more data efficiency and make better decisions. Sharing information among tasks can also enhance the model's robustness by making the model provide more information. Furthermore, MTL makes the model more adaptive to new data and data changes by making regularization in the model, which preserves the model's performance if new data are acquired and prevents overfitting.

Table 8 shows the robustness performance of MTL models compared to STL across different methods. Metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) were used for evaluation. First, methods presented in the table include sLSTM (STL-based) and pmlLSTM (MTL-based) based on [86]. These methods are used to predict taxi demand from diverse datasets in different areas. They used factors like pick-up and drop-off data for a taxi to have better prediction performance. The results indicated that leveraging the relationship between these factors is more effective than training them separately. Furthermore, multi-task federated learning (MT-FL) and single-task federated learning (ST-FL) methods were used to predict the speed in various traffic situations [100]. MT involved training speed with different traffic conditions such as weather, time, and events. Leveraging the speed with these traffic conditions under MTL showed fewer errors than STL across short, medium, and long terms. In addition, MTL stacked autoencoders (MSAE) based on MTL outperformed STL stacked autoencoders (SAE-STL) based on [101]. Dividing the traffic flow of different areas into different tasks and training them concurrently shows its effectiveness in predicting traffic based on MAE and RMSE results. Finally, the multi-task adversarial spatial-temporal network model (MT-ASTN) achieved lower errors than the single-task spatial-temporal network model (ST-STN) in both MAE and RMSE based on [102]. These models were used for predicting crowd flow in the streets across different cities, utilizing data collected from two datasets–one from bikes and the other from taxis. MT-ASTN based on MTL used the crowd flows and the flow original distance concurrently exploiting shared knowledge between them. Based on the results, MTL achieved better results in different methods and conditions. That proves the robustness of MTL models than STL.

TABLE 8. The MTL model shows its strength and robustness across different methods and conditions.

<table>
<tr><th rowspan="2">Methods</th><th colspan="8">Metrics</th></tr>
<tr><th colspan="4">MAE</th><th colspan="4">RMSE</th></tr>
<tr><td rowspan="3">SLSTM based on STL [86]</td><td colspan="2">pick-up</td><td colspan="2">drop-off</td><td colspan="2">pick-up</td><td colspan="2">drop-off</td></tr>
<tr><td>Dataset 1</td><td>Dataset 2</td><td>Dataset 1</td><td>Dataset 2</td><td>Dataset1</td><td>Dataset2</td><td>Dataset1</td><td>Dataset2</td></tr>
<tr><td>6.359</td><td>16.529</td><td>5.630</td><td>14.941</td><td>9.671</td><td>26.813</td><td>8.248</td><td>22.984</td></tr>
<tr><td>PmlLSTM based on MTL [86]</td><td>6.202</td><td>16.176</td><td>5.533</td><td>14.499</td><td>9.399</td><td>26.311</td><td>8.130</td><td>22.411</td></tr>
<tr><td></td><th colspan="4">AMAE</th><th colspan="4">ARMSE</th></tr>
<tr><td rowspan="2">MT FL [100]</td><td>Short-term</td><td>Mid-term</td><td colspan="2">Long-term</td><td>Short-term</td><td>Mid-term</td><td colspan="2">Long-term</td></tr>
<tr><td>0.737</td><td>1.378</td><td colspan="2">2.077</td><td>1.186</td><td>2.695</td><td colspan="2">4.118</td></tr>
<tr><td>ST FL [100]</td><td>0.894</td><td>1.827</td><td colspan="2">2.951</td><td>1.401</td><td>3.307</td><td colspan="2">5.144</td></tr>
<tr><td></td><th colspan="4">MAE</th><th colspan="4">RMSE</th></tr>
<tr><td>SAE-STL [101]</td><td colspan="4">34.56</td><td colspan="4">49.00</td></tr>
<tr><td>MSAE based on MTL [101]</td><td colspan="4">31.72</td><td colspan="4">45.16</td></tr>
<tr><td></td><th colspan="4">MAE</th><th colspan="4">RMSE</th></tr>
<tr><td rowspan="3">ST-STN based on STL [102]</td><td colspan="2">Crowd Flow</td><td colspan="2">Flow OD</td><td colspan="2">Crowd Flow</td><td colspan="2">Flow OD</td></tr>
<tr><td>Dataset 1</td><td>Dataset 2</td><td>Dataset 1</td><td>Dataset 2</td><td>Dataset 1</td><td>Dataset 2</td><td>Dataset 1</td><td>Dataset 2</td></tr>
<tr><td>1.562</td><td>7.604</td><td>0.015</td><td>0.077</td><td>3.103</td><td>15.699</td><td>0.112</td><td>0.131</td></tr>
<tr><td>MT-ASTN based on MTL [102]</td><td>1.413</td><td>6.417</td><td>0.011</td><td>0.030</td><td>2.995</td><td>12.299</td><td>0.074</td><td>0.087</td></tr>
</table>

B. LIMITATIONS

1) PERFORMANCE

As aforementioned, the MTL could increase the performance on each task. However, it can only be achieved under certain conditions. First, high task similarity is one of the important premises. The task similarity depicts whether it is fit to apply MTL to the tasks. The high task similarity reflects that there might be more common patterns existing among tasks. Hence, a well-trained MTL can perform all tasks well. Applying MTL to tasks with low task similarity could cause the model to miss the crucial patterns for specific tasks and further affect the model's performance.

However, analyzing the task similarity and grouping related tasks together are challenging in MTL. Based on our findings, estimating task similarity is still an open research area yet to be thoroughly studied. Most existing methods still have limitations, e.g., high complexity and time-consuming, which makes it hard to apply in real smart transportation applications.

Second, the data is crucial for MTL to reach effective performance. The informative and clean data enable the MTL to capture the patterns among data more effectively and reduce overfitting. The data collected for different tasks might be noisy and imbalanced regarding the smart transportation system. Furthermore, considering the distributed manner of current smart transportation applications, e.g., collected by IoT devices, the data might be multimodality and heterogeneous, e.g., non-independent, identically distributed (non-i.i.d.) data. Such data makes the learning process of MTL more challenging. For instance, in non-i.i.d. data, data distribution can differ from task to task. Thus, balancing the different data distributions to avoid bias toward specific tasks in the result is considered a challenging issue.

In addition, in MTL, data quality becomes more critical. High-quality data may contain more common patterns for different tasks. Such common patterns are crucial for improving MTL's performance. In some situations, data may have more task-specific patterns than common patterns. Such data can make learning of one task affect the performance of other tasks, hindering the achievement of optimal performance. Also, adapting the model to the various data types/modalities between tasks presents a substantial challenge. The transfer of knowledge between tasks depends on their relationship, and the diversity of tasks complicates this knowledge transfer in MTL.

Furthermore, each task may focus on different goals. Balancing the diverse goals of various tasks and mitigating interference among tasks is considered a complex process. In general, multimodality and heterogeneous data are still some of the biggest challenges that need to be overcome. Considering these issues, it is vital to preprocess the data and select informative and independent features before feeding it into the model. This is one of the significant obstacles to applying MTL in smart transportation systems.

2) SECURITY AND PRIVACY

In addition to performance, security, and privacy are concerns of applying MTL in smart transportation systems. Particularly, most data in smart transportation systems might be collected by IoT devices. Such devices might be smart edge devices, roadside units, or personal devices, e.g., cell phones or PCs. Some of this data might be private and sensitive. The data owner may not want to share the data with others. Considering this, how to effectively preserve data privacy while training MTL in smart transportation systems becomes important. To enable the training performance, the typical MTL will inevitably cause the privacy leakage of data. This is also a major obstacle and shortcoming of applying MTL in smart transportation systems. To effectively train models without leaking data privacy, federated learning has been proposed and has become popular. Combining federated learning and MTL is promising to enable effective learning performance without privacy leakage.

On the other hand, regarding the distributed nature of smart transportation platforms, various attacks and malicious information might exist. For instance, the data with deliberately designed noise inserted, i.e., poisoned data, could mislead the model's prediction. This is a critical issue, especially regarding smart transportation applications, e.g., autonomous driving. Such poisoned data can cause the model to fail on car or traffic light detection, leading to critical traffic incidents. Additionally, there are many other potential attacks in specific applications. For instance, the attacker could adopt certain eavesdropping tools to steal or infer private information or identities of different parties in the system. Such issues are also major obstacles to applying MTL in smart applications.

C. FUTURE CHALLENGES

In smart transportation systems, various tasks must be trained together (e.g., weather, speed, traffic, and weekdays for estimating travel time), and many factors must be considered simultaneously. These factors are considered a challenge in MTL because there are many tasks, features, and datasets. We now describe some challenges.

1) TASK SIMILARITY

Task similarity is one of the critical factors in MTL performance. The usability of MTL could be influenced or compromised by the similarity and correlations of tasks. The reason MTL can be widely used nowadays is that some tasks are intuitively highly correlated. For example, flow and speed tasks are related to traffic forecasting in a smart transportation system because they can affect each other. Road sign detection and lane detection are considered related, where they both may use the image or video data collected by car cameras. Meanwhile, it has been demonstrated in many works that applying MTL to these tasks can effectively improve training efficiency. However, it is still challenging to develop evaluation tools or metrics to learn the task similarity effectively. Such evaluations could help us better understand if MTL fits the scenarios before applying it. Furthermore, if fit, it may help us learn the number of MTL models and their capacities that may be needed to learn the tasks. In this way, the utility of MTL can be further improved.

Currently, the research on task similarity is not fully explored. One approach is identifying related tasks learned together over the MTL network [103]. This approach trains the tasks and evaluates their inter-task affinity, which measures how well the tasks work together during training. It then groups tasks with high inter-task affinity, because they work well together, to make the training process more efficient and effective. Because there are many tasks, this method can be costly and degrade performance, especially if there is a change in any task over time. Also, higher-order approximations (HOA) are used to predict the performance of MTL networks and determine the related tasks to be trained concurrently. However, the accuracy of the network performance prediction for multiple tasks might not be perfect because of HOA's limited consideration of task complexity and non-linearity [104]. Moreover, model agnostic meta-learning (MAML) helps to improve grouping-related tasks by enabling the model to generalize knowledge from one task to another, thus equipping the model to handle related tasks more effectively. However, it suffers from extensive memory usage and long training time [105].

2) FEATURES SELECTION

Selecting relevant features for each task is a critical part that may affect the MTL model performance. Choosing irrelevant features could make learning the model more complicated as it can cause overfitting and noise in the training data set and makes it complex to learn pattern recognition [106], which shows the relationship between trained data sets. That effect can affect the prediction accuracy of the new data. For example, in taxi demand prediction, features such as the weather and holidays have to be considered in historical information on the passengers' taxi demand to increase the prediction accuracy.

Many methods were proposed for feature selection (e.g., $L1$ regularization, mutual information, and principal component analysis) [107]. Those methods progress with the feature selection and help select relevant features for each task. The proposed methods still have some challenges, such as choosing redundant features because each task requires different features [108]. Selecting various features for different tasks is challenging because there is a large number of data, which can affect the model's performance in learning accurate features.

3) DATA DISTRIBUTION

Data distribution needs to be carefully addressed in MTL, as the data may be distributed differently among tasks, leading to imbalanced data and negatively affecting the model's performance. Imbalanced data can make the model biased to the task with more or better data than other tasks. That can negatively affect the performance of the other tasks because that can lead to overfitting or underfitting, which can decrease the capability of generating new data. Many methods are used to address the data imbalance, showing advances like oversampling and under-sampling, but there are still some challenges like overfitting and missing important information [109]. For example, for vehicle recognition, there are many tasks involved in detecting the appearance of cars such as car type, color, and license plate number. For each of these tasks, the data might be non-i.i.d. For instance, sedan and SUV might be more common car types than others in the dataset. In this case, the training of the model may lead the model to focus more on sedans and SUVs over other car types. Therefore, the accuracy might be compromised.

4) ROBUST AND PRIVACY-PRESERVING LEARNING

Preserving data privacy in smart transportation systems based on MTL in a cost-effective manner is a challenging issue. It is considered one of the critical issues, especially with MTL, because the tasks share a vast amount of features that can lead to sensitive information leakage. For example, in traffic forecasting, sharing the users' data from GPS between tasks (e.g., predicting the navigation and traffic) can lead to exposing privately sensitive information for the users (e.g., users' location and movements). Many proposed methods and techniques exist to solve data privacy-preservation problems, including access control, data encryption, differential privacy, and others. Although these techniques have protected data privacy, it is still challenging to protect data privacy in MTL, especially when it comes to distributed MTL applications. The traditional way of protecting privacy in distributed learning is to aggregate gradients of local models instead of collecting data or models from each user. However, it is found that it may still leak sensitive information by using certain inferring algorithms. Considering this, the security guarantee of the existing methods might not be reliable and could be further improved and verified.

In addition to the privacy challenge, protecting the MTL models from malicious attacks is a challenging problem. Sharing vast amounts of information from multiple tasks in MTL makes it an obstacle because it can increase the possibility of the attackers finding vulnerable data, which can be exploited [110]. The complexity of the MTL model's design makes identifying the security risks more challenging. This is due to the concurrent operation of multiple components, which requires comprehensive detection through various areas to detect vulnerability. Data sanitization is one method used against attacks, but data poisoning remains an issue even with the method [111], noting that data sanitization filters the training data before training them to defend against the attacks. Also, that study indicated that data poisoning attack strategies could avoid these defenses. One is putting the poisoned data in a sensitive location in the model. The other one is making a constraint attack designed to prevent the attack's detection.

5) OTHERS

There are many challenges with using MTL with smart transportation systems. One of them is dealing with heterogeneous data, which is different features from various sources. Designing shared representation for the tasks is difficult because each task has a different data distribution. That also makes communication between multiple tasks challenging because the data features or formats of the tasks are different. Another challenge is real-time detection, which is an important part of smart transportation systems. Many objects need to be detected quickly (e.g., traffic signs for autonomous vehicles), which requires significant computational power to deal with large data sets. Moreover, there are many unexpected events in smart transportation systems (e.g., weather conditions, accidents, or road risk objects), and having a framework model, hardware device sensors, and allocating resources that accommodate these events and provide accurate predictions is considered a challenge. In addition, hardware overhead is considered a challenge. Smart transportation with MTL requires more computational resources and memory, which can further increase the hardware tool cost.

V. FINAL REMARKS

In this paper, we have discussed the issues in smart transportation systems concerning the accuracy and detection performance and how MTL uses the advantage of training features among related tasks in shared models to solve these problems. After that, we systematically reviewed the MTL's progress in various smart transportation applications, including traffic forecasting, traffic signs, vehicle recognition, travel estimate time, road safety, taxi demand, and autonomous driving. MTL has shown great potential in improving arrival time and taxi demand estimation, detecting the various types of signs and objects, and increasing the ability to control autonomous cars precisely. In addition, we have discussed the MTL's capability to leverage pre-trained models, resulting in improved knowledge for new tasks. We have also demonstrated MTL's effectiveness in handling a large amount of data and enhancing the model's robustness. Some experiment results from the literature have been presented to illustrate that MTL leads to lower errors than STL and provides better robustness against various attacks. Furthermore, we have discussed the future challenges in applying MTL in smart transportation.

ACKNOWLEDGMENT

This material is based upon work in part supported by the National Science Foundation (NSF) under Grant No. 2245933. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, ''A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications,'' *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[2] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, ''Big data analytics in intelligent transportation systems: A survey,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.

[3] W. G. Hatcher and W. Yu, ''A survey of deep learning: Platforms, applications and emerging research trends,'' *IEEE Access*, vol. 6, pp. 24411–24432, 2018.

[4] X. Liu, C. Qian, W. G. Hatcher, H. Xu, W. Liao, and W. Yu, ''Secure Internet of Things (IoT)-based smart-world critical infrastructures: Survey, case study and research opportunities,'' *IEEE Access*, vol. 7, pp. 79523–79544, 2019.

[5] S. M. Khan, M. Rahman, A. Apon, and M. Chowdhury, ''Characteristics of intelligent transportation systems and its relationship with data analytics,'' in *Data Analytics for Intelligent Transportation Systems*. Amsterdam, The Netherlands: Elsevier, 2017, pp. 1–29.

[6] *Smart Transportation Applications*. Accessed: Mar. 9, 2023. [Online]. Available: https://www.geospatialworld.net/blogs/what-is-intelligent-transport-system-and-how-it-works/

[7] S. Mehar, S. Zeadally, G. Rémy, and S. M. Senouci, ''Sustainable transportation management system for a fleet of electric vehicles,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1401–1414, Jun. 2015.

[8] J. Lin, W. Yu, X. Yang, Q. Yang, X. Fu, and W. Zhao, ''A novel dynamic en-route decision real-time route guidance scheme in intelligent transportation systems,'' in *Proc. IEEE 35th Int. Conf. Distrib. Comput. Syst.*, Jun. 2015, pp. 61–72.

[9] J. Lin, W. Yu, X. Yang, P. Zhao, H. Zhang, and W. Zhao, ''An edge computing based public vehicle system for smart transportation,'' *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12635–12651, Nov. 2020.

[10] *Smart Transportation Applications*. Accessed: Mar. 9, 2023. [Online]. Available: https://worksafetci.com/2022/09/what-are-examples-of-intelligent-transportation-systems/

[11] J. Guerrero-Ibañez, J. Contreras-Castillo, and S. Zeadally, ''Deep learning support for intelligent transportation systems,'' *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 3, p. e4169, Mar. 2021.

[12] A. Ferdowsi, U. Challita, and W. Saad, ''Deep learning for reliable mobile edge analytics in intelligent transportation systems: An overview,'' *IEEE Veh. Technol. Mag.*, vol. 14, no. 1, pp. 62–70, Mar. 2019.

[13] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, ''Deep learning for IoT big data and streaming analytics: A survey,'' *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2923–2960, 4th Quart., 2018, doi: 10.1109/COMST.2018.2844341.

[14] X. Yang, J. Lin, W. Yu, P.-M. Moulema, X. Fu, and W. Zhao, ''A novel en-route filtering scheme against false data injection attacks in cyber-physical networked systems,'' *IEEE Trans. Comput.*, vol. 64, no. 1, pp. 4–18, Jan. 2015.

[15] Z. Lv, S. Zhang, and W. Xiu, ''Solving the security problem of intelligent transportation system with deep learning,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4281–4290, Jul. 2021.

[16] A. K. Haghighat, V. Ravichandra-Mouli, P. Chakraborty, Y. Esfandiari, S. Arabi, and A. Sharma, ''Applications of deep learning in intelligent transportation systems,'' *J. Big Data Anal. Transp.*, vol. 2, pp. 115–145, Aug. 2020.

[17] H. Xu, X. Liu, W. Yu, D. Griffith, and N. Golmie, ''Reinforcement learning-based control and networking co-design for industrial Internet of Things,'' *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 885–898, May 2020.

[18] A. A. Musa, A. Hussaini, W. Liao, F. Liang, and W. Yu, ''Deep neural networks for spatial–temporal cyber-physical systems: A survey,'' *Future Internet*, vol. 15, no. 6, p. 199, May 2023. [Online]. Available: https://www.mdpi.com/1999-5903/15/6/199

[19] F. Liang, C. Qian, W. Yu, D. Griffith, and N. Golmie, ''Survey of graph neural networks and applications,'' *Wireless Commun. Mobile Comput.*, vol. 2022, Jul. 2022, Art. no. 9261537.

[20] *Multi-Task Learning*. Accessed: Mar. 9, 2023. [Online]. Available: https://www.ruder.io/multi-task/

[21] Y. Zhang and Q. Yang, ''An overview of multi-task learning,'' *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 30–43, Jan. 2018.

[22] W. Huang, G. Song, H. Hong, and K. Xie, ''Deep architecture for traffic flow prediction: Deep belief networks with multitask learning,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[23] A. Kumar, ''Analysis of unsupervised dimensionality reduction techniques,'' *Comput. Sci. Inf. Syst.*, vol. 6, no. 2, pp. 217–227, 2009.

[24] J. Liu, C. Li, and W. Yang, ''Supervised learning via unsupervised sparse autoencoder,'' *IEEE Access*, vol. 6, pp. 73802–73814, 2018.

[25] S. C. H. Hoi, D. Sahoo, J. Lu, and P. Zhao, ''Online learning: A comprehensive survey,'' *Neurocomputing*, vol. 459, pp. 249–289, Oct. 2021.

[26] 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, Aug. 2020.

[27] T. N. Kipf and M. Welling, ''Semi-supervised classification with graph convolutional networks,'' 2016, *arXiv:1609.02907*.

[28] G. Buroni, B. Lebichot, and G. Bontempi, ''AST-MTL: An attention-based multi-task learning strategy for traffic forecasting,'' *IEEE Access*, vol. 9, pp. 77359–77370, 2021.

[29] K. Zhang, L. Wu, Z. Zhu, and J. Deng, ''A multitask learning model for traffic flow and speed forecasting,'' *IEEE Access*, vol. 8, pp. 80707–80715, 2020.

[30] S. Sun, ''Traffic flow forecasting based on multitask ensemble learning,'' in *Proc. 1st ACM/SIGEVO Summit Genetic Evol. Comput.*, Jun. 2009, pp. 961–964.

[31] F. Jin and S. Sun, ''Neural network multitask learning for traffic flow forecasting,'' in *Proc. IEEE Int. Joint Conf. Neural Netw., IEEE World Congr. Comput. Intell.*, Jun. 2008, pp. 1897–1901.

[32] Z. Li, J. Gong, C. Lu, and Y. Yi, ''Interactive behavior prediction for heterogeneous traffic participants in the urban road: A graph-neural-network-based multitask learning framework,'' *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 3, pp. 1339–1349, Jun. 2021.

[33] H. Ren, S. Ruan, Y. Li, J. Bao, C. Meng, R. Li, and Y. Zheng, ''MTrajRec: Map-constrained trajectory recovery via Seq2Seq multi-task learning,'' in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 1410–1419.

[34] D. Deng, C. Shahabi, U. Demiryurek, and L. Zhu, ''Situation aware multi-task learning for traffic prediction,'' in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, New Orleans, LA, USA, Nov. 2017, pp. 81–90.

[35] Y. Zhang, Y. Yang, W. Zhou, H. Wang, and X. Ouyang, ''Multi-city traffic flow forecasting via multi-task learning,'' *Appl. Intell.*, vol. 51, pp. 1–19, Oct. 2021.

[36] X. Lu, Y. Wang, X. Zhou, Z. Zhang, and Z. Ling, ''Traffic sign recognition via multi-modal tree-structure embedded multi-task learning,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 4, pp. 960–972, Apr. 2017.

[37] K.-P. Bui, H.-L. Ngoc Le, Q.-T. Le, D.-H. Huynh, and V.-H. Tran, ''The design of a multi-task learning system for autonomous vehicles,'' in *Proc. 6th Int. Conf. Green Technol. Sustain. Develop. (GTSD)*, Jul. 2022, pp. 201–207.

[38] H. Luo, Y. Yang, B. Tong, F. Wu, and B. Fan, ''Traffic sign recognition using a multi-task convolutional neural network,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1100–1111, Apr. 2018.

[39] R. Qian, B. Zhang, Y. Yue, Z. Wang, and F. Coenen, ''Robust Chinese traffic sign detection and recognition with deep convolutional neural network,'' in *Proc. 11th Int. Conf. Natural Comput. (ICNC)*, Aug. 2015, pp. 791–796.

[40] K. Ishihara, A. Kanervisto, J. Miura, and V. Hautamäki, ''Multi-task learning with attention for end-to-end autonomous driving,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 2896–2905.

[41] D. Avianto, A. Harjoko, and Afiahayati, ''CNN-based classification for highly similar vehicle model using multi-task learning,'' *J. Imag.*, vol. 8, no. 11, p. 293, Oct. 2022.

[42] J. Sun, C. Jia, and Z. Shi, ''Vehicle attribute recognition algorithm based on multi-task learning,'' in *Proc. IEEE Int. Conf. Smart Internet Things (SmartIoT)*, Aug. 2019, pp. 135–141.

[43] Z. Huo, Y. Xia, and B. Zhang, ''Vehicle type classification and attribute prediction using multi-task RCNN,'' in *Proc. 9th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, Oct. 2016, pp. 564–569.

[44] Q. Chen, W. Liu, and X. Yu, ''A viewpoint aware multi-task learning framework for fine-grained vehicle recognition,'' *IEEE Access*, vol. 8, pp. 171912–171923, 2020.

[45] H. Zhu, M. Long, J. Wang, and Y. Cao, ''Deep hashing network for efficient similarity retrieval,'' in *Proc. AAAI Conf. Artif. Intell.*, 2016, vol. 30, no. 1, p. 2416.

[46] D. Liang, K. Yan, Y. Wang, W. Zeng, Q. Yuan, X. Bao, and Y. Tian, ''Deep hashing with multi-task learning for large-scale instance-level vehicle search,'' in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2017, pp. 192–197.

[47] Z. Tang, M. Naphade, S. Birchfield, J. Tremblay, W. Hodge, R. Kumar, S. Wang, and X. Yang, ''PAMTRI: Pose-aware multi-task learning for vehicle re-identification using highly randomized synthetic data,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 211–220.

[48] G. Rajamanoharan, A. Kanaci, M. Li, and S. Gong, ''Multi-task mutual learning for vehicle re-identification,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 62–70.

[49] Y. Xia, J. Feng, and B. Zhang, ''Vehicle logo recognition and attributes prediction by multi-task learning with CNN,'' in *Proc. 12th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery (ICNC-FSKD)*, Aug. 2016, pp. 668–672.

[50] G. R. Gonçalves, M. A. Diniz, R. Laroca, D. Menotti, and W. R. Schwartz, ''Multi-task learning for low-resolution license plate recognition,'' in *Proc. Iberoamerican Congr. Pattern Recognit. (CIARP)*, Havana, Cuba. Cham, Switzerland: Springer, Oct. 2019, pp. 251–261.

[51] W. Chu, Y. Liu, C. Shen, D. Cai, and X.-S. Hua, ''Multi-task vehicle detection with region-of-interest voting,'' *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 432–441, Jan. 2018.

[52] L. Mou and X. X. Zhu, ''Vehicle instance segmentation from aerial image and video using a multitask learning residual fully convolutional network,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6699–6711, Nov. 2018.

[53] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, ''Multi-task representation learning for travel time estimation,'' in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1695–1704.

[54] S. Xu, R. Zhang, W. Cheng, and J. Xu, ''MTLM: A multi-task learning model for travel time estimation,'' *GeoInformatica*, vol. 26, no. 2, pp. 379–395, Apr. 2022.

[55] R. Gao, X. Guo, F. Sun, L. Dai, J. Zhu, C. Hu, and H. Li, ''Aggressive driving saves more time? Multi-task learning for customized travel time estimation,'' in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1689–1696.

[56] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, ''When will you arrive? Estimating travel time based on deep neural networks,'' in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2018, vol. 32, no. 1, pp. 2–7.

[57] X. Fang, J. Huang, F. Wang, L. Zeng, H. Liang, and H. Wang, ''ConSTGAT: Contextual spatial–temporal graph attention network for travel time estimation at Baidu maps,'' in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 2697–2705.

[58] L. Yang, S. Jiang, and F. Zhang, ''Multitask learning with graph neural network for travel time estimation,'' *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–9, Mar. 2022.

[59] J. Xu, S. Xu, R. Zhou, C. Liu, A. Liu, and L. Zhao, ''TAML: A traffic-aware multi-task learning model for estimating travel time,'' *ACM Trans. Intell. Syst. Technol.*, vol. 12, no. 6, pp. 1–14, Dec. 2021.

[60] G. Jin, H. Yan, F. Li, J. Huang, and Y. Li, ''Spatio-temporal dual graph neural networks for travel time estimation,'' 2021, *arXiv:2105.13591*.

[61] K. Li, H. Xiong, D. Yu, J. Liu, Y. Guo, and J. Wang, ''An end-to-end multi-task learning model for drivable road detection via edge refinement and geometric deformation,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8641–8651, Jul. 2022.

[62] F. Liu, Y. Wu, X. Yang, Y. Mo, and Y. Liao, ''Identification of winter road friction coefficient based on multi-task distillation attention network,'' *Pattern Anal. Appl.*, vol. 25, no. 2, pp. 441–449, May 2022.

[63] W. Song, S. Workman, A. Hadzic, X. Zhang, E. Green, M. Chen, R. Souleyrette, and N. Jacobs, ''FARSA: Fully automated roadway safety assessment,'' in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 521–529.

[64] M. Kacan, M. Oršić, S. Šegvic, and M. Ševrovic, ''Multi-task learning for iRAP attribute classification and road safety assessment,'' in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.

[65] S. Wang, J. Zhang, J. Li, H. Miao, and J. Cao, ''Traffic accident risk prediction via multi-view multi-task spatio-temporal networks,'' *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12323–12336, Dec. 2021.

[66] D.-G. Lee, ''Fast drivable areas estimation with multi-task learning for real-time autonomous driving assistant,'' *Appl. Sci.*, vol. 11, no. 22, p. 10713, Nov. 2021.

[67] A. Ranga, F. Giruzzi, J. Bhanushali, E. Wirbel, P. Pérez, T.-H. Vu, and X. Perrotton, ''VRUNet: Multi-task learning model for intent prediction of vulnerable road users,'' 2020, *arXiv:2007.05397*.

[68] K. Saleh, M. Hossny, and S. Nahavandi, ''Early intent prediction of vulnerable road users from visual attributes using multi-task learning network,'' in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 3367–3372.

[69] S. Chowdhuri, T. Pankaj, and K. Zipser, ''MultiNet: Multi-modal multi-task learning for autonomous driving,'' in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1496–1504.

[70] C. Chen, Q. Liu, X. Wang, C. Liao, and D. Zhang, ''Semi-Traj2Graph identifying fine-grained driving style with GPS trajectory data via multi-task learning,'' *IEEE Trans. Big Data*, vol. 8, no. 6, pp. 1550–1565, Dec. 2022.

[71] Y. Chen, D. Zhao, L. Lv, and Q. Zhang, ''Multi-task learning for dangerous object detection in autonomous driving,'' *Inf. Sci.*, vol. 432, pp. 559–571, Mar. 2018.

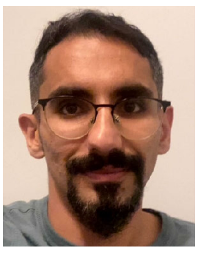
[72] X. Gibert, V. M. Patel, and R. Chellappa, ''Deep multitask learning for railway track inspection,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 1, pp. 153–164, Jan. 2017.

[73] W. Abbas, M. F. Khan, M. Taj, and A. Mahmood, ''Statistically correlated multi-task learning for autonomous driving,'' *Neural Comput. Appl.*, vol. 33, no. 19, pp. 12921–12938, Oct. 2021.

[74] O. Natan and J. Miura, ''Towards compact autonomous driving perception with balanced learning and multi-sensor fusion,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16249–16266, Sep. 2022.

[75] B. Arsenali, P. Viswanath, and J. Novosel, ''RotInvMTL: Rotation invariant MultiNet on fisheye images for autonomous driving applications,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2373–2382.

[76] V. Ravi Kumar, S. Yogamani, H. Rashed, G. Sitsu, C. Witt, I. Leang, S. Milz, and P. Mäder, ''OmniDet: Surround view cameras based multi-task visual perception network for autonomous driving,'' *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2830–2837, Apr. 2021.
[77] I. Kim, H. Lee, J. Lee, E. Lee, and D. Kim, ''Multi-task learning with future states for vision-based autonomous driving,'' in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 4–8.
[78] J. Li, D. Zhang, Y. Ma, and Q. Liu, ''Lane image detection based on convolution neural network multi-task learning,'' *Electronics*, vol. 10, no. 19, p. 2356, Sep. 2021.
[79] Z. Yang, Y. Zhang, J. Yu, J. Cai, and J. Luo, ''End-to-end multi-modal multi-task vehicle control for self-driving cars with visual perceptions,'' in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 2289–2294.
[80] D. Li, D. Zhao, Q. Zhang, and Y. Chen, ''Reinforcement learning and deep learning based lateral control for autonomous driving [application notes],'' *IEEE Comput. Intell. Mag.*, vol. 14, no. 2, pp. 83–98, May 2019.
[81] X. Chang, H. Pan, W. Sun, and H. Gao, ''YolTrack: Multitask learning based real-time multiobject tracking and segmentation for autonomous vehicles,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5323–5333, Dec. 2021.
[82] S. Ren, K. He, R. Girshick, and J. Sun, ''Faster R-CNN: Towards real-time object detection with region proposal networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, p. 2.
[83] J. Dai, Y. Li, K. He, and J. Sun, ''R-FCN: Object detection via region-based fully convolutional networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, p. 2.
[84] C. Zhou, M. Wu, and S.-K. Lam, ''Enhanced multi-task learning architecture for detecting pedestrian at far distance,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15588–15604, Sep. 2022.
[85] L. Bai, L. Yao, S. S. Kanhere, Z. Yang, J. Chu, and X. Wang, ''Passenger demand forecasting with multi-task convolutional recurrent neural networks,'' in *Proc. 23rd Pacific–Asia Conf. Adv. Knowl. Discovery Data Mining (PAKDD)*, Macau, China. Cham, Switzerland: Springer, Apr. 2019, pp. 29–42.
[86] C. Zhang, F. Zhu, X. Wang, L. Sun, H. Tang, and Y. Lv, ''Taxi demand prediction using parallel multi-task learning model,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 794–803, Feb. 2022.
[87] L. Kuang, X. Yan, X. Tan, S. Li, and X. Yang, ''Predicting taxi demand based on 3D convolutional neural network and multi-task learning,'' *Remote Sens.*, vol. 11, no. 11, p. 1265, May 2019.
[88] M. Wu, C. Zhu, and L. Chen, ''Multi-task spatial–temporal graph attention network for taxi demand prediction,'' in *Proc. 5th Int. Conf. Math. Artif. Intell.*, Apr. 2020, pp. 224–228.
[89] H. Luo, J. Cai, K. Zhang, R. Xie, and L. Zheng, ''A multi-task deep learning model for short-term taxi demand forecasting considering spatiotemporal dependences,'' *J. Traffic Transp. Eng. English Ed.*, vol. 8, no. 1, pp. 83–94, Feb. 2021.
[90] Z. Chen, B. Zhao, Y. Wang, Z. Duan, and X. Zhao, ''Multitask learning and GCN-based taxi demand prediction for a traffic road network,'' *Sensors*, vol. 20, no. 13, p. 3776, Jul. 2020.
[91] J. Ke, S. Feng, Z. Zhu, H. Yang, and J. Ye, ''Joint predictions of multi-modal ride-hailing demands: A deep multi-task multi-graph learning-based approach,'' *Transp. Res. C, Emerg. Technol.*, vol. 127, Jun. 2021, Art. no. 103063.
[92] S. Feng, J. Ke, H. Yang, and J. Ye, ''A multi-task matrix factorized graph neural network for co-prediction of zone-based and OD-based ride-hailing demand,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5704–5716, Jun. 2022.
[93] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, ''Origin-destination matrix prediction via graph convolution: A new perspective of passenger demand modeling,'' in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1227–1235.
[94] C. Li, L. Bai, W. Liu, L. Yao, and S. T. Waller, ''Knowledge adaption for demand prediction based on multi-task memory neural network,'' in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 715–724.
[95] Y. Liang, G. Huang, and Z. Zhao, ''Joint demand prediction for multimodal systems: A multi-task multi-relational spatiotemporal graph neural network approach,'' *Transp. Res. C, Emerg. Technol.*, vol. 140, Jul. 2022, Art. no. 103731.
[96] B. Chidlovskii, ''Multi-task learning of time series and its application to the travel demand,'' 2017, *arXiv:1712.08164*.
[97] R. K. Mahabadi, S. Ruder, M. Dehghani, and J. Henderson, ''Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks,'' 2021, *arXiv:2106.04489*.
[98] X. Sun, R. Panda, R. Feris, and K. Saenko, ''AdaShare: Learning what to share for efficient deep multi-task learning,'' *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 8728–8740.
[99] K. Zhang, Z. Liu, and L. Zheng, ''Short-term prediction of passenger demand in multi-zone level: Temporal convolutional neural network with multi-task learning,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1480–1490, Apr. 2020.
[100] T. Zeng, J. Guo, K. J. Kim, K. Parsons, P. Orlik, S. Di Cairano, and W. Saad, ''Multi-task federated learning for traffic prediction and its application to route planning,'' in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jul. 2021, pp. 451–457.
[101] D. Yang, H.-M. Yang, P. Wang, and S.-J. Li, ''MSAE: A multitask learning approach for traffic flow prediction using deep neural network,'' in *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, Jilin, China. Singapore: Springer, Jul. 2020, pp. 153–161.
[102] S. Wang, H. Miao, H. Chen, and Z. Huang, ''Multi-task adversarial spatial–temporal networks for crowd flow prediction,'' in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1555–1564.
[103] C. Fifty, E. Amid, Z. Zhao, T. Yu, R. Anil, and C. Finn, ''Efficiently identifying task groupings for multi-task learning,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 27503–27516.
[104] T. Standley, A. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, ''Which tasks should be learned together in multi-task learning?'' in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 9120–9132.
[105] S. Mahmud and K. H. Lim, ''One-step model agnostic meta-learning using two-phase switching optimization strategy,'' *Neural Comput. Appl.*, vol. 34, no. 16, pp. 13529–13537, Aug. 2022.
[106] G. H. John, R. Kohavi, and K. Pfleger, ''Irrelevant features and the subset selection problem,'' in *Proc. Mach. Learn.* Amsterdam, The Netherlands: Elsevier, 1994, pp. 121–129.
[107] J. F. Barraza, E. L. Droguett, and M. R. Martins, ''Towards interpretable deep learning: A feature selection framework for prognostics and health management using deep neural networks,'' *Sensors*, vol. 21, no. 17, p. 5888, Sep. 2021.
[108] B. Venkatesh and J. Anuradha, ''A review of feature selection and its methods,'' *Cybern. Inf. Technol.*, vol. 19, no. 1, pp. 3–26, Mar. 2019.
[109] P. Branco, L. Torgo, and R. P. Ribeiro, ''A survey of predictive modeling on imbalanced domains,'' *ACM Comput. Surv.*, vol. 49, no. 2, pp. 1–50, Jun. 2017.
[110] M. Zhao, B. An, Y. Yu, S. Liu, and S. Pan, ''Data poisoning attacks on multi-task relationship learning,'' in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 2629–2630.
[111] P. W. Koh, J. Steinhardt, and P. Liang, ''Stronger data poisoning attacks break data sanitization defenses,'' *Mach. Learn.*, vol. 111, no. 1, pp. 1–47, Jan. 2022.



MOHAMMED ALZAHRANI received the B.S. degree in computer information systems from Albaha University, Saudi Arabia, in 2014, and the M.S. degree in information systems technology track from Monmouth University, Long Branch, NJ, USA, in 2021. He is currently pursuing the Ph.D. degree in information technology (major) with Towson University. His research interests include data science, specifically in deep and machine learning and blockchain, specifically in scholarship and education.

QIANLONG WANG received the B.E. degree in electrical engineering from Wuhan University, China, in 2013, the M.E. degree in electrical and computer engineering from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2015, and the Ph.D. degree in computer engineering from Case Western Reserve University, Cleveland, OH, USA, in 2021. He is currently an Assistant Professor with the Department of Computer and Information Sciences, Towson University. His research interests include robust and secure deep learning networks and security and privacy in distributed and decentralized systems, e.g., cyber physical systems (CPS), the Internet of Things (IoT), and related applications, e.g., smart transportation, smart city, and smart healthcare.

WEIXIAN LIAO received the B.S. degree in information engineering from Xidian University, Xi'an, China, in 2012, the M.S. degree in electrical and computer engineering from Mississippi State University, Starkville, MS, USA, in 2015, and the Ph.D. degree in computer engineering from Case Western Reserve University, Cleveland, OH, USA, in 2018. He is currently an Assistant Professor with the Department of Computer and Information Sciences, Towson University, MD, USA. His research interests include cybersecurity and optimization in big data applications, cyber-physical systems, and machine learning.

XUHUI CHEN (Member, IEEE) received the B.E. degree in information engineering from Xidian University, Xi'an, China, in 2012, the M.S. degree in electrical and computer engineering from Mississippi State University, in 2015, and the Ph.D. degree in computer engineering from Case Western Reserve University, in 2019. She is currently an Assistant Professor with the College of Aeronautics and Engineering, Kent State University. Her research interests include machine learning and cybersecurity (security and privacy in distributed machine learning and blockchain technology), with applications in bioinformatics and cyber-physical systems (UAV and power systems).

WEI YU (Senior Member, IEEE) received the B.S. degree in electrical engineering from the Nanjing University of Technology, Nanjing, China, in 1992, the M.S. degree in electrical engineering from Tongji University, Shanghai, China, in 1995, and the Ph.D. degree in computer engineering from Texas A&M University, College Station, TX, USA, in 2008. He is currently a Full Professor with the Department of Computer and Information Sciences, Towson University, MD, USA. His current research interests include cybersecurity and privacy, cyber-physical systems/the Internet of Things, data science, and machine learning driven applications. He was a recipient of the 2014 NSF CAREER Award; the 2015 University System of Maryland (USM) Regents Faculty Award for Excellence in Scholarship, Research, or Creative Activity; and the 2016 USM Wilson H. Elkins Professorship.