Fuel Rate Prediction for Heavy-Duty Trucks

Liangkai Liu*, Wei Li†, Dawei Wang‡, Yi Wu§, Ruigang Yang† *Fellow, IEEE* and Weisong Shi¶ *Fellow, IEEE*
*Wayne State University
†Inceptio
‡The University of Hong Kong
§Nanjing University of Posts and Telecommunications
¶University of Delaware

***Abstract*—Fuel cost contributes significantly to the high operation cost of heavy-duty trucks. Developing fuel rate prediction models is the cornerstone of fuel consumption optimization approaches for heavy-duty trucks. However, limited by accurate features directly related to the truck's fuel consumption, state-of-the-art models show poor performance and are rarely deployed in practice. In this paper, we use the truck's engine management system (EMS) and Instant Fuel Meter (IFM) to collect a three-month dataset during the period of December 2019 to June 2020. Seven prediction models, including linear regression, polynomial regression, MLP, CNN, LSTM, CNN-LSTM, and AutoML, are investigated and evaluated to predict real-time fuel rate. The evaluation results show that the EMS and IFM dataset help to improve the coefficient of determination of traditional linear/polynomial models from 0.87 to 0.96, while learning-based approach AutoML improves the coefficient of determination to attain 0.99. Besides, we explore the actual deployment of fuel rate prediction with transfer learning and path planning for autonomous driving.**

I. Introduction

Autonomous trucking techniques have attracted massive attention from academia and industry due to their efficiency in fuel consumption. Fuel consumption accounts for a significant percentage of trucks' total operating costs [1], [2]. Based on the total cost of ownership (TCO) report for the heavy-duty trucks from logistics companies in China [1], the fuel cost accounted for 30 percent while the labor cost accounted for 20 percent. Compared with trucks driven by a human, the Level-3 autonomous driving truck saves a 0.5 to 1 driver budget with 60K to 150K RMB per year, while the Level-3 truck's price is 150K to 200K RMB higher. Optimizing fuel consumption for heavy-duty trucks has become a fundamental challenge for improving the trucking economy.

A critical step for optimizing fuel consumption is building accurate fuel prediction models for heavy-duty trucks [3], [4]. The VECTO model is used in Europe for calculating standardized energy consumption and CO2 emissions from Heavy-Duty Trucks (HDTs) for certification purposes [5]. The results of the fuel rate is average value under different torque, throttle, and engine speed, which has error tolerance of 7.5% [6]. However, in order to save fuel with the truck's planning and control, we need the fuel rate predicted in fine-grain (millisecond level) under any torque, engine speed, and throttle with high accuracy. Generally, there are two major benefits of having a fine-grained and accurate fuel rate prediction model. Firstly, it's necessary for intelligent fleet management of human driven trucks [7]. The fuel rate prediction model could provide a baseline for fuel consumption of each trip, which guides the truck drivers to save fuel while driving. Secondly, the fuel rate prediction model provides more accurate info for the planning module of an autonomous driving truck. Currently, the planning module relies on the engine fuel map for fuel optimization. However, the engine fuel map is coarse-grained and cannot provide accurate fuel rate. An accurate and fine-grained fuel rate prediction model helps to provide driving behavior guidelines, path planning, and vehicle controlling.

Lots of efforts have been made for fuel rate modeling such as VT-Micro, comprehensive modal emission model (CMEM), Virginia Tech comprehensive power-based fuel consumption model (VT-CPFM), etc. Generally, these models can be divided into two main categories: *vehicle dynamic and combustion-based*, and *learning-based*. Among the *vehicle dynamics and combustion-based* approaches, CMEM models the fuel rate as a polynomial function of engine speed and vehicle power while VT-Micro models the fuel rate as a polynomial function of vehicle speed and accelerations [8], [9], [10]. VT-CPFM is proposed to model the fuel rate prediction as a second-order polynomial function of vehicle power to avoid the bang-bang control problem and make the prediction more practical [3], [4]. TuSimple has adopted VT-CPFM into the planning module of autonomous driving trucks, which shows up to 10 percent fuel saving [11]. Similarly, a fourth-order polynomial regression of engine torque is used for fuel rate prediction [12]. *Combustion-based* engine model studies the relationship between the power and the gear position, the average speed, the efficiency of the gearbox, the air-drag coefficients, and the engine parameters. Boris et al. present how to optimize the operating conditions of a model internal combustion engine to obtain maximal efficiency [13]. Imed et al. leveraged the mean attractive force along with the Willan's internal combustion engine model to estimate the fuel efficiency given in liters per 100 kilometers [14]. However, due to incomplete combustion of the gas, the vehicle's equivalent fuel consumption and carbon emissions are usually 50 to 127% higher than the average value [15], [16]. For *learning-based* approaches, Elnaz et al. proposed to predict fuel consumption using artificial neural networks (ANN) based on cyclic activities like loading time, loaded haulage time, and so on [17]. Rahimi et al. applied ANN to the prediction

[1]https://en.inceptio.ai/

of fuel consumption of tractors [18]. Wysocki et al. applied polynomial regression and ANN models to predict the fuel consumption of heavy duty trucks [19]. Perrotta et al. applied vector machine (SVM), random forest (RF), and artificial neural network (ANN) to the fuel rate prediction by leveraging the sensors on trucks [20]. However, traditional polynomial regression and multilayer perceptron -based approaches cannot persist the knowledge learned from previous results like long-short term memory (LSTM). Fang et al. proposed to leverage neural networks to create a fine-grained fuel consumption prediction by leveraging Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) [21], [22], [23]. Generally, coefficient of determination (R^2) is used to show the performance of regression models like fuel rate prediction. For state-of-the-art approaches, the highest R^2 of fuel rate prediction is still less than 0.9.

Besides, none of the existing approaches provide a fine-grained prediction of the fuel rate and could work in real-time operation. For the VT dataset used in VT-CPFM, the features include the number of gas emissions like CO, HC, NOx, etc., which only have indirect relationships with fuel consumption [3], [4]. For another dataset collected for 468 vehicles in Southern Africa, only features like payload, elevation gain/lost, maximum speed, etc., are recorded at road level [24]. These external factors affects the fuel rate consumption by changing of intrinsic factors like torque, throttle, and engine speed. External factors like slope, elevation, road conditions, etc., are hard to monitor in real-time. In general, the truck's fuel consumption is related to the real-time engine and road status like speed, RPM, torque, etc. Leveraging engine data from the CAN bus helps us to predict fuel rate at millisecond-level with high accuracy. However, existing fuel rate datasets are either unavailable or does not provide these features with high accuracy and frequency for fine-grained fuel rate prediction. Besides, given the unavoidable errors of the fuel rate label from EMS devices, it's challenging to have accurate fuel rate prediction. Therefore, we proposed to monitor these intrinsic factors with both EMS and high accuracy IFM devices to collect a fine-grained fuel rate dataset.

To overcome the challenge of lacking a training dataset for fuel rate prediction, we proposed using the Engine Management System (EMS) and an Instant Fuel Meter (IFM) to collect the training dataset. We choose an off-the-shelf CAN bus parser to read fuel consumption from truck's EMS with 10mL measurement resolution and $6.3 \pm 0.2\%$ error [25]. Regarding the IFM, which provides more accurate fuel measurement, we use Onosokki's FP-2140H [2] fuel meter with 0.1mL measurement resolution and $\pm 0.2\%$ error. Both EMS and IFM provide frequent features, including torque, engine speed, brake status, road, etc. Figure 1 shows the truck, EMS, and IFM devices used in this study. In this paper, six groups of features are selected based on the dataset collected with EMS and IFM. We propose seven learning-based fuel rate prediction models: linear regression, polynomial regression, MLP, CNN, LSTM, CNN-LSTM, and automated machine learning (AutoML). The coefficient of determinism (R^2) reflect the proportion of the variation in the dependent variable that is predictable from the independent variable(s) [26]. From the experimental results, new models using AutoML attain R^2 with 0.975, which outperforms all the state-of-the-art models, including mathematical modeling-based and ANN-based modeling. Besides, we apply transfer learning in the fuel rate prediction to transfer knowledge from trained fuel rate models to decrease training costs. Moreover, we integrate the fuel rate prediction and path planning to show its application in autonomous driving trucking. In summary, this paper makes the following contributions:

- The EMS and IFM dataset with high accuracy and frequency features for fine-grained fuel rate prediction is collected. We have open-sourced the dataset to the research community at https://github.com/Torreskai0722/FEAD.
- LSTM, CNN-LSTM, and AutoML improve the R^2 of fuel rate prediction to 0.99, outperforming the state-of-the-art approaches to a great extent.
- Integrating fuel rate prediction modeling with transfer learning, planning, and control algorithms helps deploy and decrease fuel consumption for autonomous driving trucks.

The rest of the paper is organized as follows. Section II presents the dataset descriptions. Section III discusses the models used in this paper for fuel rate prediction. Sections IV presents the performance and application of the proposed fuel rate prediction models. Section V concludes the paper.

II. Background and Dataset Description

In this section, we start with the definition of the fuel rate prediction problem. Followed by two datasets, we collected in real trucks: EMS and IFM. Since the actual fuel consumption of a specific truck is highly affected by the truck driver's driving behavior and road conditions, we propose to train the fuel rate prediction model with engine data collected on the same truck and same road. We define a road from one city to another city as a route. Therefore, we represent the engine data in route granularity and propose a route matching algorithm to merge the same truck's data on the same route.

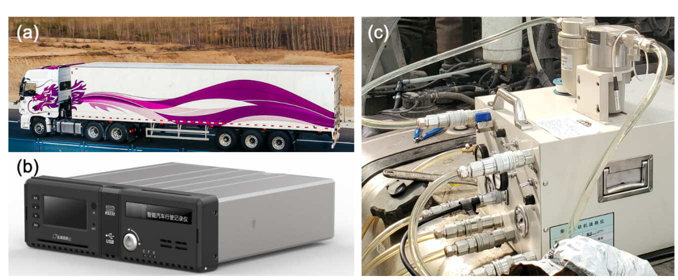


Fig. 1. Collecting hardware of our heavy-duty truck dataset. (a) Dataset collecting truck, (b) EMS equipment, (c) IFM equipment.

[2]https://www.onosokki.co.jp/HP-WK/products/keisoku/vehicle/fp_series.html

A. Fuel Rate Prediction

Fuel rate modeling is one of the essential tasks for energy-efficient driving. Heavy-duty trucks with fuel rate prediction models have proved to cut fuel consumption by at least 10 percent [11]. Since fuel consumption makes up a considerable portion of heavy-duty truck's operating costs, an accurate fuel prediction model can decrease costs a lot.

Fuel rate prediction is a problem that is defined to learn the engine behavior in fuel rate consumption. Figure 2 shows a simplified pipeline of the autonomous driving truck system. We can observe that the fuel rate prediction model's input is engine data while the output is the fuel rate, which can be formulated in Equations 1 and 2. In Equation 2, x_0^t, x_1^t, ..., x_n^t represents features in engine data while $\hat{y}^{t_i}$ and y^{t_i} represent the predicted and actual fuel rate at time t_i. For the regression problem, mean squared error (MSE) is used to define the loss of predicted fuel rate. Therefore, the objective of the regression problem is to get the minimum loss. f represents the fuel rate prediction model, which can be linear, polynomial, deep neural network-based functions, etc. For the LSTM-based model, which takes the history engine data as the input to predict fuel rate, the model's restriction is updated as Equation 3, where the engine date within time t and $t-w$ will be used to predict the fuel rate at the time t. w is defined as the lookback length or window size.

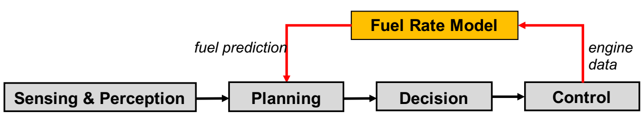


Fig. 2. A typical pipeline for energy-efficient autonomous driving.

$$\min \quad loss = \frac{1}{N}\sum_{i=1}^{N}(y^{t_i} - \hat{y}^{t_i})^2 \tag{1}$$

$$where \quad \hat{y}^t = f(x_0^t, x_1^t, ..., x_n^t) \tag{2}$$

$$\hat{y}^t = f(x_0^t, ..., x_n^t, x_0^{t-1}, ..., x_n^{t-1}, ..., x_0^{t-w}, ..., x_n^{t-w}) \tag{3}$$

The truck's planning module could develop an energy-efficient trajectory with no safety sacrifices with an accurate fuel rate prediction. Generally, a big challenge in designing and implementing the fuel rate model is getting vast amounts of high-accuracy training data. Generally, the road conditions, slope, vehicle's gross weight, wind, latitude, etc., are all external factors that affect the fuel rate. In fact, these external factors contribute to the changing of intrinsic factors, including torque, throttle, engine speed, etc., which affects the fuel rate consumption directly. Since we are aiming at integrating the fuel rate prediction model into the truck's path planning, we need the fuel rate prediction model to work in fine-grained and real-time. External factors like slope, elevation, road conditions, etc., are hard to monitor in real-time. Leveraging engine data from the CAN bus helps us to predict fuel rate at millisecond-level with high accuracy. Therefore, we proposed to leverage the intrinsic features measured from the truck's EMS and an external IFM to collect months of training data from actual trucks to train the fuel rate prediction model.

TABLE I
THE COMPARISON OF EMS AND IFM DEVICES.

	EMS	IFM
Number of collected features	20	57
Frequency (Hz)	10	20
Fuel rate error	6.3±0.2%	±0.2%
Cost (RMB)	1,000	100,000

Both EMS and IFM devices collect engine status data from the engine management units through CAN bus. One key difference is that the fuel rate in EMS dataset is directly read from the CAN bus while that of the IFM dataset is from a flow rate device. Both datasets are used in training fuel rate prediction models. The comparison of these two devices are shown in Table 1. We can find that they have different types of features, frequency, fuel rate accuracy, and the device cost. Since the error of the IFM device is less than 0.2 percent, we take fuel rate from the IFM device as the ground truth for training the fuel rate prediction model. Training on the ground truth helps us to understand the capability of DNN models for fuel rate prediction. However, the IFM device is too big and expensive to deploy on a truck for daily operation. EMS devices are suitable for deployment on any heavy-duty trucks. Therefore, we also train the fuel rate prediction model using EMS devices to evaluate the performance of the fuel rate prediction model in practical deployment. In practical deployment, the predicted fuel rate based on EMS devices will be calibrated based on the errors between EMS and IFM labels.

B. EMS Data

EMS is composed of various sensors, relays, actuators, etc., to make the engine work properly. It can provide the engine's status information, including engine speed, torque, throttle, fuel rate, etc., for every 100 milliseconds. Table II shows selected features we get from the EMS device (x7*) as well as the IFM device (x8*).

In addition to the truck's engine status, the collected dataset provides context information like the truck's location (province, city, longitude, latitude, etc.), road level (freeway, highway, urban, etc.), and truck's information like vehicle ID, model, etc. Two months of EMS data covers 29 trucks with 30 million rows of data. One month of IFM data on one truck contains 872.844 rows of data. Figure 3(a) is an example of the trajectory of one truck. The detailed descriptions of the EMS and IFM dataset are shown in Table III. *Time* means the date when the EMS dataset is collected. *Trucks* means the number of trucks collected in the dataset. *Rows* means the number of data instances in the dataset.

We calculate each feature's correlation index with the fuel rate to obtain related features for model training. All the EMS and location features are analyzed with the label *x7001* in Table II. Based on the correlation analysis results in Table IV, the most correlated features are throttle (x7006), torque (x704F),

TABLE II

DATA DESCRIPTIONS IN EMS AND IFM DATASET.

ID	Description	Unit
x7000	engine speed	km / h
x7001	fuel rate	litre / hour
x7002	total fuel consumption	litre
x7003	brake	1: on , 0: off
x7004	EMS mileage	km
x7005	total mileage	mile
x7006	throttle	%
x7007	temperature of coolant	C
x006C	GPS speed	km / h
x704F	torque	%
x716D	break position	%
x8000	engine speed	km / h
x8001	instant flow rate	litre / hour
x8002	engine fuel rate	litre / hour
x8003	retarder torque	%
x8004	engine torque	%
x8005	engine torque loss	%
x8006	current gear position	-1, 0, 2, 4, ..., 12
x8007	vehicle speed	m / s
x8008	clutch slip rate	%
x8009	vehicle weight	kg
x8010	longitudinal acceleration	m / s^2
x8011	lateral acceleration	m / s^2

TABLE III

DESCRIPTIONS OF EMS AND IFM DATASET.

Name	Time	Trucks	Rows	Features
EMS dataset-1	12/2019	9	10,273,969	EMS engine data, latitude, longitude, triggertime, city, road level, etc.
EMS dataset-2	04/2020	29	26,145,539	
IFM dataset	06/2020	1	872,844	IFM engine data

and engine speed (x7000), so we choose at least two of them to compare the impact of each feature to fuel rate, as shown in Table V.

C. IFM Data

To explore the model's upper bound for fuel rate prediction, we use another fuel rate meter device to collect the ground truth of fuel consumption. IFM device collects fuel rate with an error lower than 0.2 percent for every 50 milliseconds.

The instant fuel rate monitor captures 57 features in total, and the frequency is 20 Hz. We use this device on one truck and drive on the same path for several days, making this dataset contain the same truck's same routes. The IFM dataset contains two fuel rate labels. One is the same as the EMS dataset, which is reading from the CAN bus (x8002), while the other is from the fuel measure device, which is the IFM label (x8001). We conduct a correlation analysis of selected features with the label to form feature groups. The correlation coefficient (r) results are shown in Table VI. Based on the correlation analysis, we found three features have correlation coefficients higher than 0.4: engine speed (x8000), engine torque (x8004), current gear position (x8006). Four remaining features have positive correlation coefficients: retarder torque (x8003), engine torque loss (x8005), longitudinal acceleration (x8010), lateral acceleration (x8011). So we define three groups: G4 contains three features (x8000, x8004,x8006) and x8005; G5 adds x8010 and x8011 onto G4; G6 contains three features (x8000, x8004,x8006) and x8003. Table VII shows the selection of three groups of features and their labels for the instant fuel meter dataset.

TABLE IV

THE CORRELATION ANALYSIS RESULTS OF SELECTED FEATURES IN THE EMS DATASET WITH X7001 (FUEL RATE).

r	x7000	x7002	x7003	x7004	x7005	x7006	x7007	x006C	x704F	x716D
x7001	**0.530**	-0.005	-0.218	0.089	-0.025	**0.937**	0.217	0.038	**0.665**	-0.120

TABLE V

EMS DATASET FEATURE GROUPS.

Groups	Features	Label
G1	engine speed (x7000), throttle (x7006)	fuel rate (x7001)
G2	engine speed (x7000), throttle (x7006), torque (x704F)	
G3	engine speed (x7000), torque (x704F)	

TABLE VI

THE CORRELATION ANALYSIS RESULTS OF SELECTED FEATURES IN THE IFM DATASET WITH X8001 (IFM LABEL).

r	x8000	x8003	x8004	x8005	x8006	x8007	x8008	x8009	x8010	x8011
x8001	**0.496**	0.218	**0.949**	0.076	**0.468**	0.005	-0.109	0.001	0.122	0.172

TABLE VII

IFM DATASET DESCRIPTION AND FEATURE GROUPS.

Groups	Features	Label-1	Label-2
G4	engine speed (x8000), engine torque (x8004), engine torque loss (x8005), current gear position (x8006)	instant flow rate (x8001)	engine fuel rate (x8002)
G5	G4 + longitudinal acceleration (x8010) lateral acceleration (x8011)		
G6	engine speed (x8000), retarder torque (x8003), engine torque (x8004), current gear position (x8006)		

D. Route modeling and matching

Generally, the actual fuel consumption of a specific truck is highly affected by the truck driver's driving behavior. Besides, since different routes show different traffic and altitude conditions, the fuel rate consumption pattern of the same truck on different routes can also be different. These observations motivate us to train different models for different trucks and routes. In this paper, we define the road between two cities as a route. Since different drivers drive different trucks, we merge engine data collected on the same truck and the same routes for the training of fuel rate prediction models. All the route-level fuel rate prediction models will be ensembled to construct a complete model for deployment on trucks.

How to uniquely represent the route collected on the same truck becomes the first challenge. We define a *Route* class that contains *header* and *data*, as shown in Figure 4. The *header* includes the *date*, *vvid* for the vehicle's ID, *city*, and *mileage*. The *data* contains *path[]* and *ems[]* sequence the latitude, longitude, and EMS data. Using *Route* class, each route within the EMS dataset can be uniquely represented, and it's possible to extract the same routes data for training.

After representing the route as a uniform expression, how to match the same routes becomes another problem. The GPS data is not updated periodically, which means there are

(a) (b)

Fig. 3. (a) A trajectory of one truck in one day. (b) An example of an EMS samples' GPS data on the map.

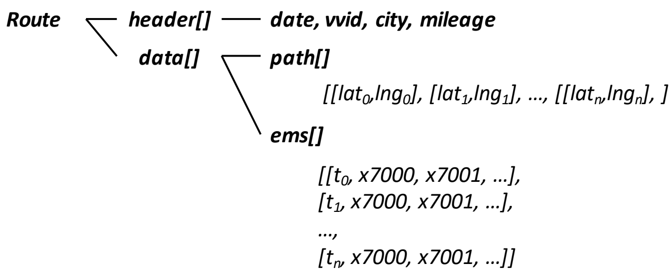


Fig. 4. The code structure of Route class.

random delays in some instances of latitude and longitude data. Figure 3(b) shows an example of the EMS GPS data on the map, in which we can observe aperiodic points and long delays between them. These delays make the route's path attribute has a different length even for the same route. To address this problem and merge the same routes for a specific truck, we propose a sampling-based route matching algorithm that samples a fixed number of points between the route's source and destination and compares their difference. The sampling-based route matching algorithm consists of four main steps:

1: Merge EMS data instance based on city, truck's *vvid*, and *date* to get the route. For each date in the group, calculate the total mileages using the *x7004* between the start and endpoints in *ems*. This step is to extract routes from the EMS dataset to form a routes pool.
2: Compare the start and endpoint of different routes to determine the direction of the route. Compare the mileage of different routes to get the most frequent length of the routes r_1. Since there could be several routes between two cities, the routes that have less than a one-mile difference with r_1 in the same direction will be grouped. There will be two groups: each direction has one group;
3: Each route is represented as an element of *Route*, then we sample the route to get a fixed-length route r'. The sampling interval is calculated by $length/sample$. For every $length/sample$ row, one row of the *path* and *ems* are selected to add into r', where $sample$ is a pre-defined value as 100.
4: Calculate the distance of matrix r'. If the sum of two routes' absolute difference is less than a pre-defined threshold ϵ, these two routes are the same route.

The purpose for sampling is to find the vehicle vid, dates, and timestamps that belong to the same route. When training the fuel rate prediction model, we use all the data from the same route.

III. Prediction Models

In this section, we present the methodologies for building fuel rate prediction models. Generally, the methods can be divided into two categories: mathematical modeling-based and learning-based methods. Mathematical modeling-based methods are usually based on the dynamic models for heavy-duty trucks to connect features like engine speed, torque, etc., with the instant fuel rate [4]. A learning-based approach is purely a data-driven approach that leverages the regressor or neural networks to extract knowledge from the raw data. In this paper, we propose several learning-based methods to build a fuel rate prediction model for heavy-duty trucks, including linear/polynomial regression, multilayer perceptron (MLP), convolutional neural network (CNN), long-short term memory (LSTM), CNN-LSTM, and AutoML.

A. Linear and Polynomial Regression

Linear and polynomial regressions are methods that are widely used in fuel rate prediction. For example, the Comprehensive Modal Emission Model (CMEM) represents the fuel rate as a linear formula of engine speed and truck power [8]. Besides, researchers proposed to use a two-dimensional 4th-order polynomial to model the fuel rate [12]. Moreover, the Virginia Tech Comprehensive Power-based Fuel Consumption Model (VT-CPFM) takes a bottom-up approach, representing the resistance force and truck power as a polynomial of speed and acceleration [3]. Then the fuel rate is formulated as a polynomial of the truck power. The basis of these polynomial regressions is to model the fuel rate as a polynomial combination of features like RPM, truck speed, acceleration, etc. However, these models' performance is usually bounded by the features and polynomial orders. We implement linear regression (LR) and a 4th-order polynomial regression (PR) with the collected dataset as the baseline to the deep learning-based approaches.

B. Deep Learning-based Regression

Deep neural networks (DNN) have been widely deployed in a variety of applications and scenarios, including CNN for image classification and semantic segmentation [22], [27], Recurrent Neural Networks (RNN) for speech recognition and language modeling [28], [29], and Generative Adversarial Networks (GAN) for generating datasets [30], [31]. The reason for DNN's popularity is its excellent performance in many applications and its ability to learn from raw, heterogeneous, and noisy data, which is suitable for fuel rate prediction [32], [33], [34].

Since the fuel rate prediction is modeled as a regression problem and tries to map features from raw driving data with labels, deep neural networks is expected to outperform traditional linear and polynomial-based approaches. To compare the regression performance in fuel rate prediction, we

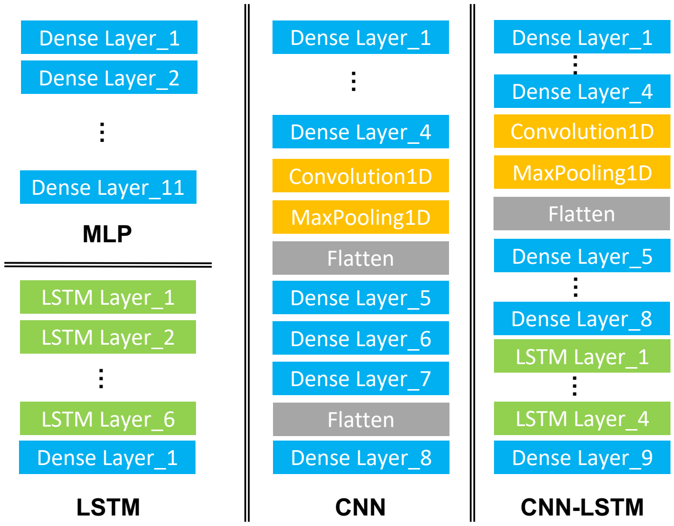


Fig. 5. The model structure of MLP, CNN, LSTM, and CNN-LSTM.

proposed several DNN-based regressions for fuel rate prediction, including MLP, CNN, LSTM, and CNN-LSTM. Figure 5 shows the structures for four deep learning-based models. MLP has the simplest neural network structure with hidden layers fully connected and a non-linear activation function for output. From figure 5, the MLP model consists of eleven dense layers. Each layer has hundreds of neurons which are fully connected with former and next layers. As one of the most successful deep neural networks, CNN performs better in extracting features from raw pixel data for applications like object detection and classification. In this paper, we use low-dimensional convolutional kernels to extract features from the raw data (batch samples of engine status data). The CNN model is start with four fully-connected dense layer, then a convolution layer is used to extract feature maps, followed by a max polling layer, which is to down sample the feature map based on maximum values. Then the feature maps will be flattened and apply to several layers of fully connected dense layers for regression output.

Unlike traditional neural networks that do not store the knowledge learned from previous layers, LSTM introduces loops to make neural networks' learning persistent. The prediction of the truck's engine fuel rate is a time sequence problem, which means the current engine state and the expected engine state determine the fuel rate. For example, currently the truck is running on a flat road and it will start to run uphill. Then the truck will need more torque to run uphill, the delta of current torque and expected torque will affect the fuel rate. LSTM is a recurrent neural network which takes a sequence of history and current EMS/IFM data as well as previous fuel rate as input to predict the current fuel rate. This design makes LSTM suitable for applications like time series analysis. The LSTM model consists of 6 LSTM layers followed by a dense layer for regression output. The input to the LSTM layer is a window of samples contains current and history engine status data as shown in Equation 3. CNN-LSTM is proposed to combine

TABLE VIII
DATA DESCRIPTIONS IN VT DATASET.

Feature	CO2, CO, HC, NOx	velocity	fuel	engine	elevation	phase
Unit	g/s	mph	g/s	rpm	m	N/A

both CNN's strength in feature extraction and LSTM's strength on time series analysis [35]. It can be found in Figure 5 that CNN-LSTM is a stack of the CNN model with the LSTM model.

C. *AutoML-based Regression*

The model's structure is fixed for both linear/polynomial and deep learning-based approaches, and the training process is to find the best parameters. The searching for parameters and model structures usually require expertise in machine learning. How to efficiently search for the most appropriate model structures and parameters become a big challenge. AutoML tackles this challenge by enabling the machine to find the best model for given inputs and labels [36]. In addition, AutoML ensembles several trained weak regressors to build a strong regressor. Each regressor usually consists of three parts: data preprocessor is for data encoding, impulation, and rescaling; feature preprocessor eliminates features corresponding to zero-valued model coefficients; regressor takes the features to predict the output [37]. AutoML explores more combinations and possibilities of algorithms, it usually finds better models than human beings. In this paper, we implement an AutoML-based regression based on *auto-sklearn* for fuel rate prediction to show its performance compared with human-designed DNNs [37].

IV. Evaluation

In this section, we start with the experiment setup and metrics for evaluation. The evaluation of the fuel rate prediction is present in three datasets: EMS dataset, IFM dataset, and VT dataset. Seven models are trained for fuel rate prediction: linear regression (LR), polynomial regression (PR), MLP, CNN, LSTM, CNN-LSTM, and AutoML. Next, we discuss the transfer learning of fuel rate modeling between trucks and days. Finally, we evaluate the application of integrating fuel rate modeling with planning and control.

A. *Experiment Setup*

Hardware and software setup. The experiments are conducted on an Intel Fog Node, which has 8 Intel(R) Xeon(R) CPU E3-1275 v5 @ 3.60GHz and 32 GB DDR4 memory with 34.1 GB/s bandwidth. The platform is installed with Ubuntu 18.04. The software includes sklearn 0.24.2, auto-sklearn 0.12.7, tensorflow 1.14.0, numpy 1.18.1, etc. We leverage sklearn and tensorflow for linear/polynomial regression, MLP, CNN, LSTM and CNN-LSTM-based fuel rate prediction, while auto-sklearn is used for AutoML-based fuel rate prediction.

Baseline. The VT dataset is used as a complementary dataset for evaluation [4]. As a popular fuel rate model, VT-CPFM

TABLE IX
THE CORRELATION ANALYSIS RESULTS ON THE VT DATASET.

r	CO2	CO	HC	NOx	velocity	engine	elevation	phase
fuel rate	**0.88**	0.47	0.57	0.91	0.4	**0.46**	0.18	-0.11

takes a bottom-up approach, representing the resistance force and truck power as a polynomial of speed and acceleration [3]. To better evaluate the proposed learning-based approaches' performance, we choose VT-CPFM as a baseline and compare it with learning-based approaches on the VT dataset. The VT dataset is collected on three trucks, including 128,355 rows of data. Table VIII shows the features and units in the VT dataset. Table IX shows the correlation analysis results. We select six features whose correlation coefficients with the fuel rate are larger than 0.4, including CO2, CO, HC, NOx, vel, and engine.

Cross validation. To evaluate these learning-based models' performance on various datasets, all the training models follow the same procedure. First, we use standard normalization, which standardizes features by removing the mean and scaling to unit variance. The effect of features with higher values on the fuel rate is decreased. Next, we use cross-validation with five folds to divide the training and testing data [38]. As is shown in Figure 6, we divide all the dataset into average five parts. Then choose one of the five parts as testing dataset while the remaining parts as training dataset to train the model. Besides, ten percent of the training data is used to validate the training loss to save the best model. So five models will be trained and validated. We calculated the average R^2 and accuracy of five models as the result. For LSTM, the input to the model contains a look-back window, which means some history data is also used for training. After training, the average R^2 and accuracy of five folds are calculated.

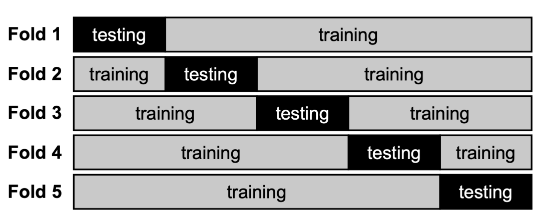


Fig. 6. Illustration of five fold cross validation.

Metrics. The problem of fuel rate prediction can be formulated as a regression problem, which means the model is trying to learn the mapping between the input features and the fuel rate. As is shown in Section II, MSE is used as the loss function, and the training process is to find the minimum MSE until convergence. Therefore, we use the coefficient of determinism (R^2) to evaluate the fuel rate prediction model's performance. As shown in Equation 4, R^2 is defined as one minus the sum of predicted errors' square divided by errors' square when using the mean value as the predicted value. y^{t_i} and $\hat{y}^{t_i}$ represent the actual and predicted fuel rate, respectively, while $\bar{y}$ represents the mean value of actual fuel rate during time t_i and t_N. We can find that R^2 represents the distance of the predicted value with the actual and average values. To understand the regression performance from another perspective, we define the accuracy of the fuel rate prediction. From Equation 5, the accuracy is defined as one minus the absolute error's median value divided by the mean value.

TABLE X
THE RESULTS ON VT DATASET.

Metrics	LR	VT-CPFM	MLP	CNN	LSTM	CNN-LSTM
R^2	0.8704	0.8177	0.9172	0.9164	0.8985	**0.9260**
Accuracy	0.8483	0.9196	0.9256	**0.9352**	0.8333	0.9318

TABLE XI
NUMBER OF PARAMETERS FOR SIX MODELS USED ON THE VT DATASET.

Model	LR	VT-CPFM	MLP	CNN	LSTM	CNN-LSTM
Parameters	7	462	71,501	44,977	394,651	**1,687,677**

1) Coefficient of Determination:

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \left(y^{t_i} - \hat{y}^{t_i}\right)^2}{\sum_{i=1}^{N} \left(y^{t_i} - \bar{y}\right)^2} \tag{4}$$

2) Accuracy:

$$Accuracy = 1 - \frac{median\left\{\left|y^{t_1} - \hat{y}^{t_1}\right|, ..., \left|y^{t_N} - \hat{y}^{t_N}\right|\right\}}{\bar{y}} \tag{5}$$

B. Comparison with VT-CPFM

As a baseline, we implement VT-CPFM as a 5th order polynomial regression of the input. The results for 5-fold average R^2 and $accuracy$ are shown in Table X. We find that learning-based models show better performance than VT-CPFM in both R^2 (0.8177) and $accuracy$ (0.9196). CNN-LSTM has the highest R^2 with 0.926, while CNN has the best $accuracy$ with 0.9352. The comparison of the VT dataset shows that learning-based approaches are more efficient than the VT-CPFM. One explanation for it is the number of parameters for training models. Table XI shows the numbers of parameters for different fuel rate prediction models. For VT-CPFM, which takes six features with 5th order polynomial regression, the total number of model parameters is 462. However, the number of parameters in CNN-LSTM models is 1.69 million. It can be found that the number of parameters in DNN models is around 97x to 3653x that of the polynomial regression (VT-CPFM) model. This huge number of parameters make the DNN-based model extract more knowledge from the raw engine data.

Observation 1: Deep learning-based approaches show better regression performance in fuel rate prediction than traditional polynomial regression, mainly due to the huge number of parameters (around 97 to 3653 times).

C. EMS Dataset Results

We apply the same route matching algorithm on the EMS dataset to get a specific truck's data to train models, including LR, PR, MLP, CNN, LSTM, and CNN-LSTM. Three feature groups of the EMS dataset are chosen for training. The results for R^2 and accuracy of all three groups are shown in Table XII. The window size in LSTM is set as 10. We can observe from

TABLE XII

RESULTS FOR LEARNING-BASED APPROACHES WITH ONE TRUCK'S SAME ROUTE ON EMS DATASET.

Groups	Metrics	LR	PR	MLP	CNN	LSTM	CNN-LSTM
G1	R^2	0.8681	0.9215	0.9135	**0.9222**	0.8788	0.9145
	accuracy	0.8768	0.9608	0.9578	0.9557	0.8860	**0.9618**
G2	R^2	0.8682	0.9222	0.9131	0.9223	**0.9523**	0.9146
	accuracy	0.8758	0.9595	0.9467	0.9583	0.9340	**0.9619**
G3	R^2	0.4948	0.5622	0.5445	0.5631	**0.9474**	0.5427
	accuracy	0.5413	0.6777	0.6342	0.6333	**0.8886**	0.6808

TABLE XIII

PERFORMANCE OF CNN-BASED FUEL RATE MODEL BEFORE AND AFTER ROUTE MATCHING.

vid - 9964	*row*	*CNN - R2*	*CNN - Accuracy*	*Training Time (s)*
after route matching	84,334	**0.9224**	**0.9655**	400
before route matching	652,937	0.9081	0.9474	**3,800**

the results that G2 shows better performance than other groups. As we have seen in Table IV, throttle and torque have much higher correlation coefficient with the fuel rate than engine speed. In fact, the torque is correlated with the fuel rate because the higher the torque output, the higher the fueling level. Besides, to achieve a certain torque, the in-cylinder mixture needs a certain amount of fuel, which is controlled by the throttle position. Therefore, fuel rate is highly related to torque and throttle. LSTM shows better performance in G2 and G3 than all the other models, especially in G3 where all the other models have R^2 less than 0.57. The reason is that LSTM takes current as well as history data as input to train the fuel rate prediction model. LSTM could extract and maintain history information. The results of G3 mean that the history of torque information makes a big impact on the fuel rate. By looking into the results of G2 and G3, we can find that the current throttle data makes an even bigger impact on fuel rate than the history of torque information, that's why the LSTM model in G2 achieves the highest R^2.

To show the effectiveness of the route matching algorithm, we apply it to EMS dataset-1. There are a total of 148 routes in EMS dataset-1. There are 126 routes matched classified as the same routes. Besides, to show the effectiveness of route matching for fuel rate prediction model, we also show the results of CNN-based fuel rate prediction model before and after route matching in Table XIII. The experiments are done on the same truck with vehicle ID (*vid*) ending with 9964. We can find that the amount of input rows is reduced by 8 times after route matching. Besides, the R^2 has been increased from 0.9081 to 0.9224, while the *accuracy* has increased from 0.9474 to 0.9655. These improvements show the effectiveness of the route matching algorithm. In addition, we selected three trucks in the EMS dataset to show the performance of G2. These three trucks are driven by different drivers on different routes. We apply route matching algorithms to each truck's driving data to merge the same routes for fuel rate prediction model training. Table XIV show the results of R^2 and *accuracy*. We can find that although there is some difference between different vehicles, the best R^2 is larger than 0.95 while the best *accuracy* is larger than 0.96 for each vehicle.

TABLE XIV

RESULTS FOR LEARNING-BASED APPROACHES FOR THREE TRUCKS IN EMS DATASET.

Groups	Metrics	LR	PR	MLP	CNN	LSTM	CNN-LSTM
vid - 89B7	R^2	0.8646	0.9065	0.9106	**0.9502**	0.9481	0.9105
	accuracy	0.8203	0.9492	0.9490	**0.9766**	0.9147	0.9343
vid - 9964	R^2	0.8682	0.9222	0.9131	0.9223	**0.9523**	0.9146
	accuracy	0.8758	0.9595	0.9467	0.9583	0.9340	**0.9619**
vid - F069	R^2	0.9652	0.9943	0.7951	**0.9944**	0.9580	**0.9944**
	accuracy	0.9166	0.9845	0.7818	0.9843	0.8705	**0.9863**

TABLE XV

R^2 OF LSTM WITH DIFFERENT LOOKBACK WINDOW SIZE

LSTM - window	5	10	15	20	25	30
G2	0.8832	0.9523	0.9538	**0.9567**	0.9552	0.9559

To show the performance of LSTM more deeply, we change the window size of the input for training and get the R^2 for them in Table XV. Since EMS data is generated every 100ms, we set the window size as 5, 10, 15, 20, 25, and 30, respectively. The longest window covers history EMS data for 3 seconds. As we can observe from Table XV, when the window size increases, the R^2 of LSTM first increases significantly from 0.88 to 0.95, then it converges. The reason is that when the window size is not enough to contain all the history information, the increase will help the LSTM model to predict the fuel rate. However, the R^2 would not further increase after a certain point because the added history data is not necessary anymore. On the other hand, a larger windows size means less real-time fuel rate prediction because the LSTM model needs to take all the needed historical data as input. Given these two findings, we choose 20 as the window size, which means to predict fuel rate every 2 seconds.

Observation 2: Current throttle and history torque play a significant role in fuel rate prediction. The performance of the LSTM model converges when the window size increases to a certain point.

D. IFM Dataset Results

Compared with the EMS dataset, the frequency of IFM is higher, and the value of the IFM is more accurate. To show the performance of different models under different feature groups and labels, we train several models with different configurations and present R^2 and $accuracy$ for the IFM label and EMS label in Table XVI and Table XVII. For LSTM, the window size is set as 50 (2.5 seconds).

Table XVI shows that the results for R^2 and $accuracy$ vary for different groups and approaches. We can find that AutoML shows the best R^2 and $accuracy$ for all groups on both IFM and EMS label. The details of the AutoML model trained on IFM dataset is shown in Table XXIII, Table XXIV, and Table XXV. We can find that it ensembles several weak regressors with different data preprocessing and feature processing algorithms together to construct a strong regressor. There are two primary reasons why AutoML can achieve better performance than other deep-learning based methods [36], [39]. First, for raw data which contains noises

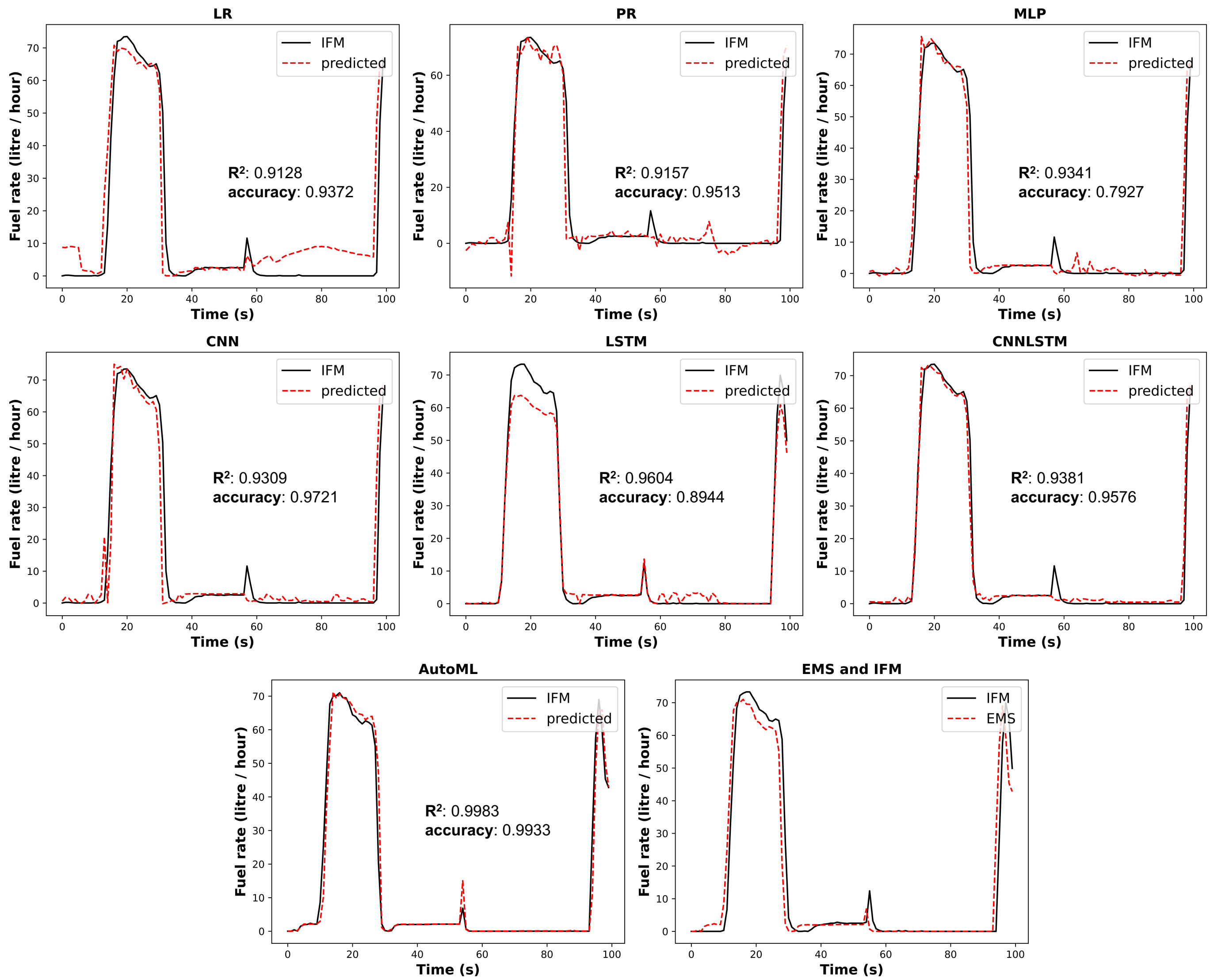


Fig. 7. Time histories of the IFM and model-predicted fuel rate for all the models under group *G5*.

and errors, AutoML performs better than non-expert in feature engineering [39]. Feature engineering is the process of finding the best set of variables and the best data encoding and normalization for input to the model training process, while AutoML can search for more combinations of raw data preprocessing to get the best method [36]. Second, through hyperparameter optimization, meta-learning, and neural network searching, AutoML could search for a wider model space than human beings to get the best model [40], [41], [39].

Figure 7 shows a 200s time histories of IFM collected and predicted fuel rate. We can observe that EMS and IFM fuel rate shows obvious difference. The predicted fuel rate from AutoML aligns exactly with the IFM label. For LSTM, the trend of predicted fuel rate is consistent with IFM label but there is constant difference between them, which makes it has high R^2 but low $accuracy$.

Compared with the learning-based approach on the VT dataset, the highest R^2 is improved from 0.9260 to 0.9983, proving that high-quality EMS data helps the fuel rate prediction. Compared with the EMS dataset with city-level same routes, the IFM with EMS label has improved the highest R^2 from 0.9523 to 0.9983. Because when the training data is fine-tuned to smaller levels (from city level to road level). Besides, the R^2 and $accuracy$ trained with the EMS label are higher than with the IFM label, mainly because the EMS label's error is $\pm 5\%$ while the IFM label's error is $\pm 0.2\%$. So model trained on EMS label is expected to show 5% error. The model trained with IFM label still has better performance than with the EMS label. In terms of real-time performance, the training of fuel rate prediction models is usually offline while the inference is executing online to predict fuel rate based on current status. The training and inference time of all the models on Intel Fog Reference is shown in Table XVIII. We can find that the inference time are all less than 3ms, which is much less than the data interval of IFM and EMS devices.

TABLE XVI

THE RESULTS OF LEARNING-BASED FUEL RATE MODELS WITH LABEL-1.

Groups	Metrics	LR	PR	MLP	CNN	LSTM	CNN-LSTM	AutoML
G4	R^2	0.8632	0.6092	0.9218	0.9213	0.9483	0.9278	**0.9968**
	accuracy	0.8853	0.9315	0.9578	0.953	0.8295	0.9648	**0.9915**
G5	R^2	0.9128	0.9157	0.9341	0.9309	0.9604	0.9381	**0.9983**
	accuracy	0.9372	0.9513	0.7927	0.9721	0.8944	0.9576	**0.9933**
G6	R^2	0.8646	0.8121	0.9212	0.9165	0.9481	0.9213	**0.9976**
	accuracy	0.8904	0.9333	0.9646	0.9541	0.8247	0.9375	**0.9924**

TABLE XVII

THE RESULTS OF LEARNING-BASED FUEL RATE MODELS WITH LABEL-2.

Groups	Metrics	LR	PR	MLP	CNN	LSTM	CNN-LSTM	AutoML
G4	R^2	0.9379	0.9305	0.9597	0.9626	0.9561	0.9656	**0.9990**
	accuracy	0.9332	0.9711	0.9698	0.9738	0.8547	0.9762	**0.9930**
G5	R^2	0.9567	0.9612	0.9567	0.9687	0.9565	0.9745	**0.9983**
	accuracy	0.9745	0.9712	0.9806	0.9784	0.8478	0.9704	**0.9911**
G6	R^2	0.9409	0.9325	0.9657	0.9644	0.9546	0.9693	**0.9988**
	accuracy	0.9343	0.9695	0.9752	0.9717	0.8608	0.9704	**0.9925**

Observation 3: High-quality engine data helps the fuel rate prediction. AutoML shows the best performance on both EMS and IFM datasets.

TABLE XIX

TRANSFER LEARNING BETWEEN VEHICLES.

	Models	Rows	Training time	R^2
CNN	baseline	20644	**4.5 minutes**	0.9668
	base model	183403	58.3 minutes	0.9332
	fine-tuned model	20644	**4.5 minutes**	**0.9673**
LSTM	baseline	20644	**75 minutes**	0.9045
	base model	183403	10.38 hours	**0.9204**
	fine-tuned model	20644	**75 minutes**	0.9099

TABLE XX

TRANSFER LEARNING BETWEEN DAYS.

	Models	Rows	Training time	R^2
CNN	baseline	15492	**3.5 minutes**	0.9486
	base model	188556	41.7 minutes	0.9349
	fine-tuned model	15492	**3.5 minutes**	**0.9512**
LSTM	baseline	15492	**58 minutes**	0.9090
	base model	188556	11.8 hours	**0.9264**
	fine-tuned model	15492	**58 minutes**	0.9135

E. Transfer Learning of Fuel Rate Modeling

As a data-driven approach, the composition of the dataset determines the learning-based model's performance. However, the longer trucks are driving, the more data is collected, but it is a waste of resources and time to train the model from scratch all the time. Transfer learning is proposed to train the model based on the former models' learned knowledge [42]. We extract a small dataset that collects four trucks' EMS data for several days when driving on the same route. The transfer learning is conducted in two aspects: transfer between trucks and transfer between days. From the model structure of MLP, CNN, LSTM, and CNN-LSTM in Figure 5, the CNN-LSTM model is a stack of CNN layers with LSTM layers together while MLP is a combination of Dense layers. CNN and LSTM are two representative model architectures which contains the dense, convolutional, and LSTM layers. Therefore, we choose CNN and LSTM models for the evaluation of transfer learning.

Transfer between trucks. We use three trucks' data as the input to train a base model and transfer the knowledge from this base model into a fine-tuned model with the fourth truck's data as a training dataset. In comparison, a model is trained from scratch with the fourth truck's data.

Transfer between days. We use all the four trucks' EMS data except one day as the input to train a base model and transfer the knowledge from this base model to a fine-tuned model with the EMS data on a remaining day as a training dataset. Similarly, a model with the same structure is trained with one-day EMS data as the baseline.

The transfer learning results between vehicles and days are shown in Table XIX and Table XX, respectively. From both cases, the training time of transferring the knowledge from a base model or training from scratch is the same for both CNN and LSTM. Generally, if there is one base model that could predict the fuel rate consumption of all vehicles under all dates, then the base model should be easily trained for a specific date or vehicle. However, compared with the baseline, which uses the one date or one vehicle dataset and trains from scratch, the base model takes the same time to converge. This result means that there is rare knowledge transfer from the base model to the fine-tuned model. Since the dataset is collected on the same route, the fuel rate prediction model is customized for each vehicle and each date. Besides, the performance of fine-tuned models has a slightly higher R^2 compared with training from the scratch. However, the R^2 of the fine-tuned LSTM model is lower than its base model. One potential explanation is that the base model's training data and training time are higher than the baseline and fine-tuned model. Although more parameters make it possible for deep learning-based models to construct a much complex model than traditional polynomial regression models, it usually requires huge data to train the model from scratch. The more parameters the model has, the more training data it requires [43]. More rows of training data and time are needed to achieve higher regression performance for the LSTM model.

TABLE XVIII

THE TRAINING AND INFERENCE TIME FOR MODELS ON IFM DATASET.

Model	LR	PR	MLP	CNN	LSTM	CNN-LSTM	AutoML
Training (mins)	1	1	15	17	125	100	**300**
Inference (ms)	0.1	0.1	0.1	0.1	0.3	0.2	**2.9**

TABLE XXI

VALIDATION OF CNN-BASED FUEL RATE PREDICTION WITH NEWLY COLLECTED DRIVING DATA.

vid - 9964	first four days	remaining two days	
		before retrain	after retrain
rows of driving data	28,152	23,883	
R^2	**0.9193**	0.9001	0.9131
accuracy	0.9580	0.9162	**0.9611**

Validation with newly collected driving data. When the fuel rate model is deployed, it's necessary to validate the model's performance with newly collected data. Therefore, we choose one truck whose vid ends with 9964 which has six days of driving data is collected on the same route. We use the first four days as a training dataset to train a CNN-based fuel prediction model, while the remaining two days as newly collected data to test the performance of the model. From the results in Table XXI, we can see that before retraining, the performance of the remaining two days is slightly lower than the first four days. The R^2 is decreased to 0.9001 from 0.9193, while the *accuracy* is reduced from 0.9580 to 0.9162. After retraining the model with the newly collected two days driving data, the R^2 is increased to 0.9131 and the *accuracy* is increased to 0.9611. The retraining of the model takes eight minutes. This validation shows that the prediction model shows good performance with newly collected driving data. And the model can be easily fine-tuned to achieve the same level of *accuracy* and R^2 as the original model.

Observation 4: The fuel rate prediction model is customized for each vehicle and each date. For the deployment of fuel rate prediction models on real trucks, the route-level models should be ensembled together into a complete model.

F. Fuel Rate Prediction in Planning and Control

In this experiment, we demonstrate our accurate fuel rate prediction can effectively benefit the planning and control (PnC) algorithm w.r.t the fuel consumption for heavy-truck. One of the classic fuel-efficient PnC algorithms for heavy-duty trucks is predictive cruise control (PCC) [44], [45], which is an optimization algorithm calculating the optimal speed according to the road profile w.r.t fuel consumption, travel time, and speed band. The essential part of PCC's optimization function is the cost of fuel consumption rate. Thus, we integrate our AutoML model, which has the highest fuel rate prediction accuracy, into the PCC optimization function.

In PCC evaluation experiments, we adopt a simplified vehicle model, its longitudinal dynamics is formulated as:

$$F = \eta(sin(\theta(s)) + \mu(\dot{s}))gM_{veh} + \frac{1}{2}\rho_{air}A_f C_d \dot{s} + \ddot{s}M_{veh} \tag{6}$$

in which η is total power efficiency from engine torque to propulsion force, $\theta(s)$ is road gradient with respect to the distance ahead of vehicle, $\mu(\dot{s})$ is tire rolling friction, M_{veh} indicates the vehicle mass, ρ_{air} is the density of air, A_f means the front area of vehicle and C_d is the air drag constant. The parameters in this model are shown in Table. XXII. We adopt the original solving method in [44], [45] and change the fuel model into our model and VT-CPFM model for evaluation.

Figure 10 shows our test road's altitude and the optimal speed from PCC using different fuel rate models. The test road is around 14 kilometers and costs about 700 seconds for the whole drive. From Figure 8, we can observe that VT-CPFM accelerates earlier than our model, because PCC with VT-CPFM produced a worse speed allocation result and led the vehicle to accelerate too early during the period of upward slope, which is a very crucial impact to fuel consumption. The distance difference in Figure 10 is not obvious because the value is small compared with the x-axis units. However, from the same distance range in Figure 11, we can find a noticeable velocity difference between these two methods, and the fuel consumption is also different. It is reflected in Figure 9, there is a sharp increase of fuel consumption during 200 300s. We also plot the velocity difference ($V_{ours} - V_{VT-CPFM}$), and fuel consumption in distance domain, as shown in Figure 11, to explain why our precise fuel model can save more fuel. From a macro perspective, our model's speed is lower than VT-CPFM's speed at the very beginning uphill road which lead to fuel saving. From a micro perspective, each significant change of fuel consumption is the result of lower acceleration and lower speed compared to VT-CPFM. Since the truck is heavy, even a very small acceleration will lead to large fuel consumption on an uphill road. This figure also reflects that VT-CPFM accelerates the truck too much around the hilltop. Overall, the AutoML and VT-CPFM's total fuel consumption is 10570 grams and 11240 grams, respectively, which is 5.97% total fuel saving on this road.

TABLE XXII
PARAMETERS OF VEHICLE

Parameter	Metric Units	Value
Vehicle Weight	kg	49000
Tire Rolling Coefficient μ	-	0.0065
Front Area A_f	m^2	9.7737
Air Drag Constant C_d	-	0.62

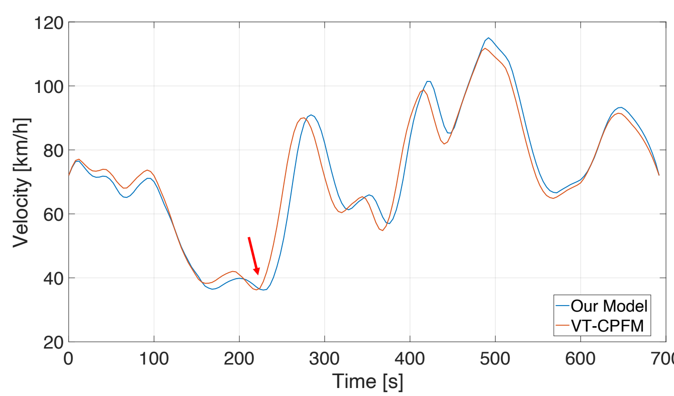


Fig. 8. Optimal speed using our AutoML fuel prediction model (blue line) and VT-CPFM's noisy model (red line).

Observation 5: Accurate fuel rate prediction can effectively benefit the planning and control (PnC) algorithm w.r.t the fuel consumption for heavy-trucks.

V. CONCLUSIONS

Fuel rate prediction is an essential step for the decrease of fuel consumption for heavy-duty trucks. In this paper, we studied the fuel rate prediction of autonomous driving trucks with a high-accuracy dataset collected from EMS and IFM devices. We propose a sampling-based route matching algorithm

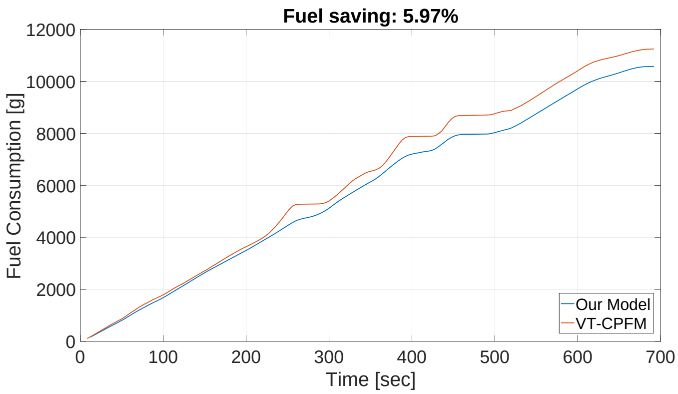


Fig. 9. Fuel consumption of PCC evaluation with our model (blue line) and VT-CPFM's noisy model (red line).

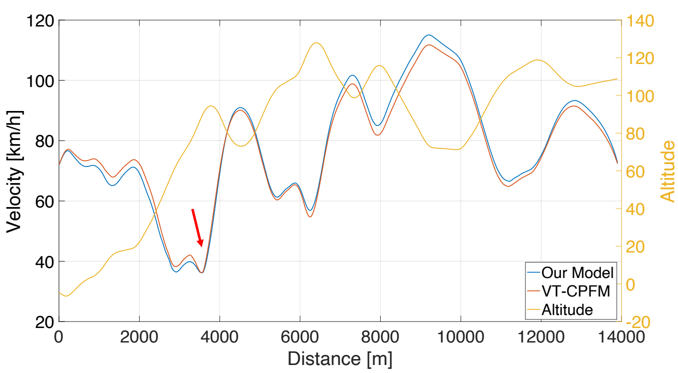


Fig. 10. Road profile (yellow line), and optimal speed using our AutoML fuel prediction model (blue line) and VT-CPFM's noisy model (red line).

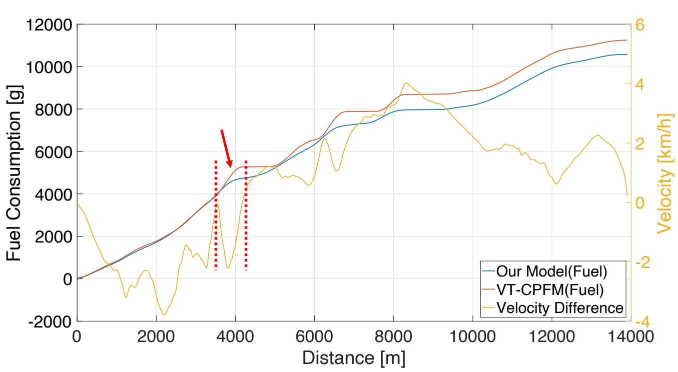


Fig. 11. Velocity Difference ($V_{ours} - V_{VT-CPFM}$) (yellow line), and fuel consumption of our model (blue line) and VT-CPFM's noisy model (red line).

to prepare engine data for model training. We propose seven models, including LR, PR, MLP, CNN, LSTM, CNN-LSTM, and AutoML, trained to predict the fuel rate consumption. Besides, we include the VT-dataset to train these models as the baseline to evaluate the EMS and IFM dataset. At the same time, the VT-CPFM is implemented as the baseline to fuel rate prediction models. Metrics like R^2 and $accuracy$ are used for the evaluation of the fuel rate prediction.

Compared with traditional approaches like linear and polynomial regression, learning-based approaches show better regression performance for fuel rate prediction on all the dataset. This is mainly owing to the huge number of parameters in learning-based models. However, learning-based models need much longer time to train compared with linear and polynomial regression.

We also observed that current throttle and history torque play a significant role in fuel rate prediction. In the LSTM model, the performance of R^2 converges when the window size increases to a certain point. The choice of window size is a trade-off between regression performance and real-time prediction.

The VT dataset does not include torque and throttle data. The EMS and IFM dataset contains millisecond-level torque, throttle, and engine RPM data. We found that high-quality engine data helps the fuel rate prediction. AutoML shows the best performance on both EMS and IFM datasets.

We also studied the transfer learning of the fuel rate prediction model between different trucks and days. We found that there is rare knowledge transferred between different vehicles and dates. The fuel rate prediction model is customized for each vehicle and each date.

The fuel rate prediction model is integrated into the path planning of autonomous trucking. We implemented AutoML and VT-CPFM -based fuel rate prediction into the predictive cruise control (PCC) optimization function to show the actual fuel saving when using fuel rate prediction model,. The evaluation results show that accurate fuel rate prediction can effectively benefit the planning and control (PnC) algorithm w.r.t the fuel consumption for heavy-truck.

ACKNOWLEDGEMENT

The authors would like to thank Jinghui Wang for providing us the VT dataset.

Appendix

A. Abbreviation and Definition

Abbreviation	Definition
TCO	total cost of ownership
RMB	Ren Min Bi
MEMS	comprehensive modal emission model
EMS	engine management system
IFM	instant fuel meter
MSE	mean squared error
R^2	coefficient of determination
MLP	multi-layer perceptron
CNN	convolutional neural network
DNN	deep neural network
ANN	artifical neural network
LR	linear regression
PR	polynomial regression
LSTM	long-short term memory
AutoML	automated machine learning
PCC	predictive cruise control
VT-CPFM	Virginia Tech comprehensive power-based fuel consumption model
PnC	planning and control

B. AutoML Model Details

References

[1] (2020) The Real Cost of Trucking – Per Mile Operating Cost of a Commercial Truck. [Online]. Available: https://www.thetruckersreport.com/infographics/cost-of-trucking/

[2] A. Engholm, A. Pernestål, and I. Kristoffersson, "Cost analysis of driverless truck operations," *Transportation research record*, vol. 2674, no. 9, pp. 511–524, 2020.

[3] H. A. Rakha, K. Ahn, K. Moran, B. Saerens, and E. Van den Bulck, "Virginia tech comprehensive power-based fuel consumption model: model development and testing," *Transportation Research Part D: Transport and Environment*, vol. 16, no. 7, pp. 492–503, 2011.

[4] J. Wang and H. A. Rakha, "Fuel consumption model for heavy duty diesel trucks: Model development and testing," *Transportation Research Part D: Transport and Environment*, vol. 55, pp. 127–141, 2017.

[5] A. Tansini, G. Fontaras, B. Ciuffo, F. Millo, I. P. Rujas, and N. Zacharof, "Calculating heavy-duty truck energy and fuel consumption using correlation formulas derived from vecto simulations," *SAE Technical Paper Series*, 2019.

[6] "VECTO - Overview," https://ec.europa.eu/clima/system/files/2018-12/201811_overview_en.pdf.

[7] D. I. Arkhipov, D. Wu, T. Wu, and A. C. Regan, "A parallel genetic algorithm framework for transportation planning and logistics management," *IEEE Access*, vol. 8, pp. 106 506–106 515, 2020.

[8] M. Barth, G. Scora, and T. Younglove, "Modal emissions model for heavy-duty diesel vehicles," *Transportation Research Record*, vol. 1880, no. 1, pp. 10–20, 2004.

[9] M. Barth, F. An, T. Younglove, C. Levine, G. Scora, M. Ross, and T. Wenzel, "Development of a comprehensive modal emissions model. national cooperative highway research program," *Transportation research board of the national academies*, 2000.

[10] H. Rakha, K. Ahn, and A. Trani, "Development of vt-micro model for estimating hot stabilized light duty vehicle and truck emissions," *Transportation Research Part D: Transport and Environment*, vol. 9, no. 1, pp. 49–74, 2004.

[11] (2019) TuSimple Study Finds 10% Fuel Savings for Autonomous Trucks. [Online]. Available: https://www.truckinginfo.com/346711/tusimple-announces-10-fuel-economy-boost-for-autonomous-trucks

[12] S. Xu, S. E. Li, X. Zhang, B. Cheng, and H. Peng, "Fuel-optimal cruising strategy for road vehicles with step-gear mechanical transmission," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3496–3507, 2015.

[13] B. M. Aizenbud, Y. B. Band, and O. Kafri, "Optimization of a model internal combustion engine," *Journal of Applied Physics*, vol. 53, no. 3, pp. 1277–1282, 1982.

[14] I. B. Dhaou, "Fuel estimation model for eco-driving and eco-routing," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 37–42.

[15] A. K. Madhusudhanan, X. Na, A. Boies, and D. Cebon, "Modelling and evaluation of a biomethane truck for transport performance and cost," *Transportation Research Part D: Transport and Environment*, vol. 87, p. 102530, 2020.

[16] C. Langness and C. Depcik, "Statistical analyses of cng constituents on dual-fuel compression ignition combustion," SAE Technical Paper, Tech. Rep., 2016.

[17] E. Siami-Irdemoosa and S. R. Dindarloo, "Prediction of fuel consumption of mining dump trucks: A neural networks approach," *Applied Energy*, vol. 151, pp. 77–84, 2015.

[18] F. Rahimi-Ajdadi and Y. Abbaspour-Gilandeh, "Artificial neural network and stepwise multiple range regression methods for prediction of tractor fuel consumption," *Measurement*, vol. 44, no. 10, pp. 2104–2111, 2011.

[19] O. Wysocki, L. Deka, D. Elizondo, J. Kropiwnicki, and J. Czyżewicz, "Heavy duty vehicle fuel consumption modelling based on exploitation data by using artificial neural networks," in *International Work-Conference on Artificial Neural Networks*. Springer, 2019, pp. 794–805.

[20] F. Perrotta, T. Parry, and L. C. Neves, "Application of machine learning for fuel consumption modelling of trucks," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 3810–3815.

[21] C. Fang, S. Song, Z. Chen, and A. Gui, "Fine-grained fuel consumption prediction," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2783–2791.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[23] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[24] A. J. Hoffman, "Neural models for benchmarking of truck driver fuel economy performance," 2019.

[25] "Vehicle Application Layer J1939/71_202002," https://www.sae.org/standards/content/j1939/71_202002/.

[26] (2022) Coefficient of determination. [Online]. Available: https://en.wikipedia.org/wiki/Coefficient_of_determination

[27] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[28] T. Mikolov, S. Kombrink, L. Burget, J. Černockỳ, and S. Khudanpur, "Extensions of recurrent neural network language model," in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2011, pp. 5528–5531.

[29] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[30] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[31] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[32] J. Dean, "The deep learning revolution and its implications for computer architecture and chip design." 2020.

[33] J. Xiao, Z. Xiao, D. Wang, V. Havyarimana, C. Liu, C. Zou, and D. Wu, "Vehicle trajectory interpolation based on ensemble transfer regression," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[34] D. Wu, H. Xu, Z. Jiang, W. Yu, X. Wei, and J. Lu, "Edgelstm: Towards deep and sequential edge computing for iot applications," *IEEE/ACM Transactions on Networking*, vol. 29, no. 4, pp. 1895–1908, 2021.

[35] S. Lu, B. Luo, T. Patel, Y. Yao, D. Tiwari, and W. Shi, "Making disk failure predictions smarter!" in *18th {USENIX} Conference on File and Storage Technologies ({FAST} 20)*, 2020, pp. 151–167.

[36] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," *Advances in neural information processing systems*, vol. 28, 2015.

[37] M. Feurer, K. Eggensperger, S. Falkner, M. Lindauer, and F. Hutter, "Auto-sklearn 2.0: The next generation," *arXiv preprint arXiv:2007.04074*, 2020.

[38] R. R. Picard and R. D. Cook, "Cross-validation of regression models," *Journal of the American Statistical Association*, vol. 79, no. 387, pp. 575–583, 1984.

TABLE XXIII
AUTOML MODEL FOR G4 ON IFM DATASET.

Metrics	weights	data_preprocessing			feature_preprocessor	regressor
		encoding	impulation	rescaling		
regressor 1	0.96	no_encoding	most_frequent	robust_scaler	feature_agglomeration	extra_trees
regressor 2	0.02	no_encoding	mean	standardize	select_rates_regression	k_nearest_neighbors
regressor 3	0.02	one_hot_encoding	mean	standardize	extra_trees	mlp

TABLE XXIV
AUTOML MODEL FOR G5 ON IFM DATASET.

Selection	weights	data_preprocessing			feature_preprocessor	regressor
		encoding	impulation	rescaling		
regressor 1	0.86	no_encoding	most_frequent	standardize	feature_agglomeration	extra_trees
regressor 2	0.04	no_encoding	mean	standardize	select_rates_regression	k_nearest_neighbors
regressor 3	0.04	no_encoding	median	standardize	extra_trees	mlp
regressor 4	0.02	one_hot_encoding	most_frequent	standardize	extra_trees	random_forest
regressor 5	0.02	no_encoding	median	standardize	extra_trees	k_nearest_neighbors
regressor 6	0.02	no_encoding	median	standardize	extra_trees	k_nearest_neighbors

TABLE XXV
AUTOML MODEL FOR G6 ON IFM DATASET.

Selection	weights	data_preprocessing			feature_preprocessor	regressor
		encoding	impulation	rescaling		
regressor 1	0.48	one_hot_encoding	median	robust_scaler	feature_agglomeration	extra_trees
regressor 2	0.48	one_hot_encoding	most_frequent	robust_scaler	select_rates_regression	extra_trees
regressor 3	0.02	no_encoding	most_frequent	standardize	select_percentile_regression	k_nearest_neighbors
regressor 4	0.02	no_encoding	most_frequent	minmax	feature_agglomeration	libsvm_svr

[39] H. Jin, Q. Song, and X. Hu, "Auto-keras: An efficient neural architecture search system," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 1946–1956.

[40] M. Feurer and F. Hutter, "Hyperparameter optimization," *Automated machine learning: Methods, systems, challenges*, pp. 3–33, 2019.

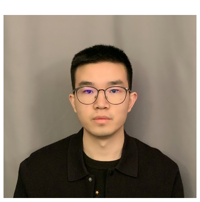
[41] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artificial intelligence review*, vol. 18, pp. 77–95, 2002.

[42] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.

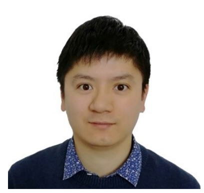
[43] W. Wang, M. Zhang, G. Chen, H. Jagadish, B. C. Ooi, and K.-L. Tan, "Database meets deep learning: Challenges and opportunities," *ACM SIGMOD Record*, vol. 45, no. 2, pp. 17–22, 2016.

[44] F. Lattemann, K. Neiss, S. Terwen, and T. Connolly, "The predictive cruise control–a system to reduce fuel consumption of heavy duty trucks," *SAE transactions*, pp. 139–146, 2004.

[45] B. Asadi and A. Vahidi, "Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time," *IEEE transactions on control systems technology*, vol. 19, no. 3, pp. 707–714, 2010.



Liangkai Liu received the B.S. degree in telecommunication engineering from Xidian University, Xi'an, China, in 2017. He is currently working toward the Ph.D. degree at Wayne State University, Detroit, MI, USA. His current research interests include edge computing, distributed systems, and autonomous driving.



Wei Li received his Ph.D. degree with the College of Automation Engineering, Nanjing University of Aeronautics and Astronautics. He is the staff engineer at Inceptio and the postdoctoral researcher at Peking University. His research interests include robotics, computer vision and computer graphics. He has published 10+ papers in top journals and conferences including Science Robotics / SIGGRAPH / TVCG / AAAI / CVPR / ICRA.

Dawei Wang received the B.S. degree in computer science from Dalian University of Technology in 2018. He is currently pursuing the Ph.D.degree with the Department of Computer Science, The University of Hong Kong. His research interests include robotics and machine learning.

Yi Wu received her Ph.D. degree with the College of Automation Engineering, Nanjing University of Aeronautics and Astronautics. From 2015 to 2018, she was a Visiting Research Scholar with the Center for Advanced Life Cycle Engineering, University of Maryland at College Park. She is now a lecturer with the College of Automation and College of Artificial Intelligence, Nanjing University of Posts and Telecommunications. Her research interests include electronic vehicles, battery management, health monitoring and prediction for electronic devices.

Ruigang Yang received his Ph.D. degree from University of North Carolina at Chapel Hill and M.S degree from Columbia University. He is the CTO of Inceptio and a full professor of computer science at the University of Kentucky. He was the director of Robotics and Autonomous Driving Lab at Baidu Research. He has published over 130 papers, which, according to Google Scholar, has receive over 14000 citations with an H-index of 61. He has received a number of awards, including US NSF Career award in 2004 and the Deans Research Award at the University of Kentucky in 2013. He is a fellow of the IEEE.

Weisong Shi is a Professor and Chair of the Department of Computer and Information Sciences at the University of Delaware (UD), where he leads the Connected and Autonomous Research (CAR) Laboratory. Dr. Shi is an internationally renowned expert in edge computing, autonomous driving and connected health. His pioneer paper entitled "Edge Computing: Vision and Challenges" has been cited more than 5700 times. Before he joins UD, he was a professor at Wayne State University (2002-2022) and served in multiple administrative roles, including Associate Dean for Research and Graduate Studies at College of Engineering and Interim Chair of Computer Science Department. Dr. Shi also served as a National Science Foundation (NSF) program director (2013-2015) and chair of two technical committees of the Institute of Electrical and Electronics Engineers (IEEE) Computer Society. Dr. Shi has published more than 270 articles in peer-reviewed journals and conferences and served in editorial roles for more than 10 academic journals and publications, including EIC of Smart Health, AEIC of IEEE Internet Computing Magazine. He is the founding steering committee chair of several conferences, including ACM/IEEE Symposium on Edge Computing (SEC), IEEE/ACM International Conference on Connected Health (CHASE), and IEEE International Conference on Mobility (MOST). He is a fellow of IEEE, a distinguished member of ACM. More information can be found at http://weisongshi.org.