Front. Earth Sci. 2021, 15(3): 620–630
https://doi.org/10.1007/s11707-021-0938-1

RESEARCH ARTICLE

A novel spatio-temporal trajectory data-driven development approach for autonomous vehicles

Menghan ZHANG[1,2,3], Mingjun MA[1,2,3], Jingying ZHANG[1,2,3], Mingzhuo ZHANG[1,2,3], Bo LI[1,2,3], Dehui DU (✉)[1,2,3]

1 School of Software and Engineering, East China Normal University, Shanghai 200062, China
2 Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China
3 MOE International Joint Lab of Trustworthy Software, East China Normal University, Shanghai 200062, China

Abstract Nowadays, autonomous driving has been attracted widespread attention from academia and industry. As we all know, deep learning is effective and essential for the development of AI components of Autonomous Vehicles (AVs). However, it is challenging to adopt multi-source heterogenous data in deep learning. Therefore, we propose a novel data-driven approach for the delivery of high-quality Spatio-Temporal Trajectory Data (STTD) to AVs, which can be deployed to assist the development of AI components with deep learning. The novelty of our work is that the meta-model of STTD is constructed based on the domain knowledge of autonomous driving. Our approach, including collection, preprocessing, storage and modeling of STTD as well as the training of AI components, helps to process and utilize huge amount of STTD efficiently. To further demonstrate the usability of our approach, a case study of vehicle behavior prediction using Long Short-Term Memory (LSTM) networks is discussed. Experimental results show that our approach facilitates the training process of AI components with the STTD.

Keywords spatio-temporal trajectory data, data meta-modeling, domain knowledge, LSTM, vehicle behavior prediction, AI component

1 Introduction

1.1 Motivation

In recent years, the rapid development of autonomous driving technology has attracted widespread attention from academia and industry. Although the international standard ISO26262 has established a series of safety standards in this field, there are still a bunch of unsolved safety problems. For example, in June 2020, a Tesla Model 3 crashed into an overturned white van in Chiayi, Taiwan Province, China with the former's auto driving and auxiliary driving systems taking no effective deceleration measures. Such accidents urge practitioners to present more accurate modeling and analysis methods of autonomous vehicle's AI components. Multi-source heterogeneous data is the key to solve this issue. Besides, with excellent performance in processing sequential data, recurrent neural networks, especially Long-Short-Term Memory (LSTM), are now widely used in autonomous driving applications such as vehicle behavior prediction.

Although Geographic Information System (GIS) technology is universally adopted, general data-driven development approaches for autonomous driving is still lacking. A standard set of meta-data modeling methods should be established for the autonomous driving domain to bridge the gaps in the existing model-driven and data-driven methods. In our previous works, we first proposed a meta-modeling approach for autonomous driving scenario modeling (Zhang et al., 2021). We also proposed scenario modeling language called Autonomous Driving Specification Modeling Language (ADSML) (Wang, 2020), which helps domain experts analyze autonomous driving states in a model-driven way. Based on our previous work, we present a data-driven development framework for autonomous driving system in this paper, which integrates various deep learning technologies to process and utilize heterogeneous data. For example, using LSTM for autonomous vehicle or pedestrian's behavior prediction, Convolutional Neural Network (CNN) for intersection traffic light recognition, or even decision making using knowledge-based inference and Bayesian neural networks,

Received March 3, 2021; accepted July 15, 2021

E-mail: dhdu@sei.ecnu.edu.cn

etc. All of these require the use of Spatio-Temporal Trajectory Data as data input. Our aim is to provide a unified and automated data management method for the development of autonomous driving systems to solve data challenges in the industry, to accelerate data processing procedure, and to guide the efficient development of autonomous driving systems.

1.2 Related work

In this section, we review the state of the art in related domains and position our work on autonomous driving.

Data-driven software development for AVs. Although there is already a considerable amount of research, fewer of which give a systematic approach for data engineering or emphasize the use of Spatio-Temporal Trajectory Data with time-series characteristics. In previous work, a DNN-based system was proposed (Seshia et al., 2018) using deep neural network to form a closed-loop system for autonomous driving. Its input is image. It focused on the formal specification of deep neural network, but lacked the complete and comprehensive data pipeline support. And the hybrid-AI approach proposed by Paardekooper et al. (2021) combines data-driven and knowledge-based reasoning to ensure the safety of autonomous driving functions, but there is no complete data meta-model for unified data management. Xu et al. (2020) proposed automated predictive models to train ML models, with little involvement in the efficient processing of real data and the generation of simulation data. Park et al. (2018) applied LSTM and beamed search algorithms to predict the vehicle's best trajectory position. Usually, these data-driven approaches are lack of meta-model to support effective data management. Therefore, when preprocessing data, usually encountered many problems such as weak management of data, difficult and time-consuming. Compared to the existing research, we combine Spatio-Temporal Trajectory Data with time-series characteristics to build a data set of autonomous driving scenarios from a data engineering perspective. Based on our Spatio-Temporal Trajectory Data, LSTM is used to achieve vehicle behavior prediction. Unlike previous studies, we used time-series data to train the model rather than simply use CNN for image recognition, which can help us make effective use of our Spatio-Temporal Trajectory Data.

Data meta-model and domain ontology. There are some studies on data meta-model and domain ontology for autonomous driving. For example, WISE laboratory (Czarnecki, 2018) provided rich domain knowledge and used ontologies to define real-world models of ADS. It is very comprehensive, but very complex and not very friendly to non-domain experts. Toyota (Zhao et al., 2015a) has also proposed an ontology model for autonomous vehicles, and used ontologies to represent the knowledge of map, driving path and driving environment, including Map Ontology, Control Ontology, Car Ontology, to improve the safety of AVs and to facilitate knowledge inference. Their another studies (Zhao et al., 2015b) achieved ontology-based decision making in an uncertain environment, each has its strength, but has not addressed the development approach for AI components in ADS from a data engineering perspective. We believe that it may work well if Spatio-Temporal Trajectory Data were considered for decision making. Zhang et al. (2019) built unified spatio-temporal meta-data expressions, but not combined with domain knowledge of ADS. The development and application of spatio-temporal data and meta-modeling in the field of ADS still requires a lot of work. This paper combines domain knowledge of ADS, unifies the management of multi-source data and makes data processing more efficient. Our work bases on the foundation of the systematic development of autonomous driving systems, and can well support the development of data-driven intelligent systems in the future.

1.3 Contributions

The main contributions of this study are as follows:

1) We propose a Spatio-Temporal Trajectory Data (STTD) driven framework to facilicate the development of AI components for autonomous vehicles.

2) We extend the STTD meta-model for data interoperation and data generation. In addition, a pipeline model for data processing based on STTD is also presented. The aim is to make the best of STTD in AV's decision making process.

3) We implement a LSTM-based vehicle behavior prediction module with the help of simulation data generated by Carla.

This paper is organized as: Section 2 describes the framework for autonomous driving system driven by STTD. Section 3 presents the STTD meta-model and specifies the data preprocessing method. Section 4 presents a case study on vehicle behavior prediction, which exploits LSTM as the core algorithm and STTD as its input. Finally, we summarize this paper and discuss the future work in Section 5.

2 Spatio-temporal trajectory data driven framework for autonomous driving system

As shown in Fig. 1, our closed-loop framework for autonomous driving system driven by STTD includes four modules: data engineering for AVs, STTD-based AI component development, Controller and Plant. Among them, data engineering for AVs includes the collection of driving environment, simulation data and STTD-based data engineering; STTD-based AI component development implements the application of STTD for autonomous driving; Controller receives commands and makes decisions to control the autonomous vehicle; Plant performs actual physical operation of the autonomous vehicle, which affects the collection of environmental data change.

2.1 Data engineering for autonomous vehicles

This part consists of driving environment data, simulation data and STTD-based data engineering.

2.1.1 Driving environment

In the driving process, an autonomous vehicle system generates a huge amount of static data and dynamic data. Detailed descriptions of data categories in driving environment are as follows:

1) Data collected by earth observing devices. Earth observation techniques provide access to global information on autonomous driving. GNSS is used to obtain vehicle's location; 3DGIS is used to obtain environmental information; and high precision maps are used to obtain information of road structure and road networks.

2) Data collected by the vehicle itself. Cameras on autonomous driving vehicles collect picture information which can serve as algorithms inputs to identify objects such as traffic lights, lane lines, taillights of the front car and pedestrians. Millimeter wave radars assist the cameras in collecting distance and speed information of dynamic objects. Ultrasonic radars collect data on obstacles close to the ego vehicle. LIDAR collects three-dimensional geospatial information to help to build a vehicle-centric map, which in turn completes the prediction of the actions of all objects. Vehicle-mounted V2X devices can acquire vehicle network data.

3) Data available to relevant authorities. The development of AI component for autonomous driving also requires historical traffic data and relevant policies, such as traffic light control mechanism, historical traffic flow data and traffic accident information of specific road sections.

These data together form the data source from real world driving environment for the development of AI components for autonomous vehicles. The next part will explain another data source: simulation data.

2.1.2 Simulation data generation by Carla

The amount of data required for training is huge, but the data acquired from real world are limited, incomplete and unbalanced. To solve this issue, simulation engines like

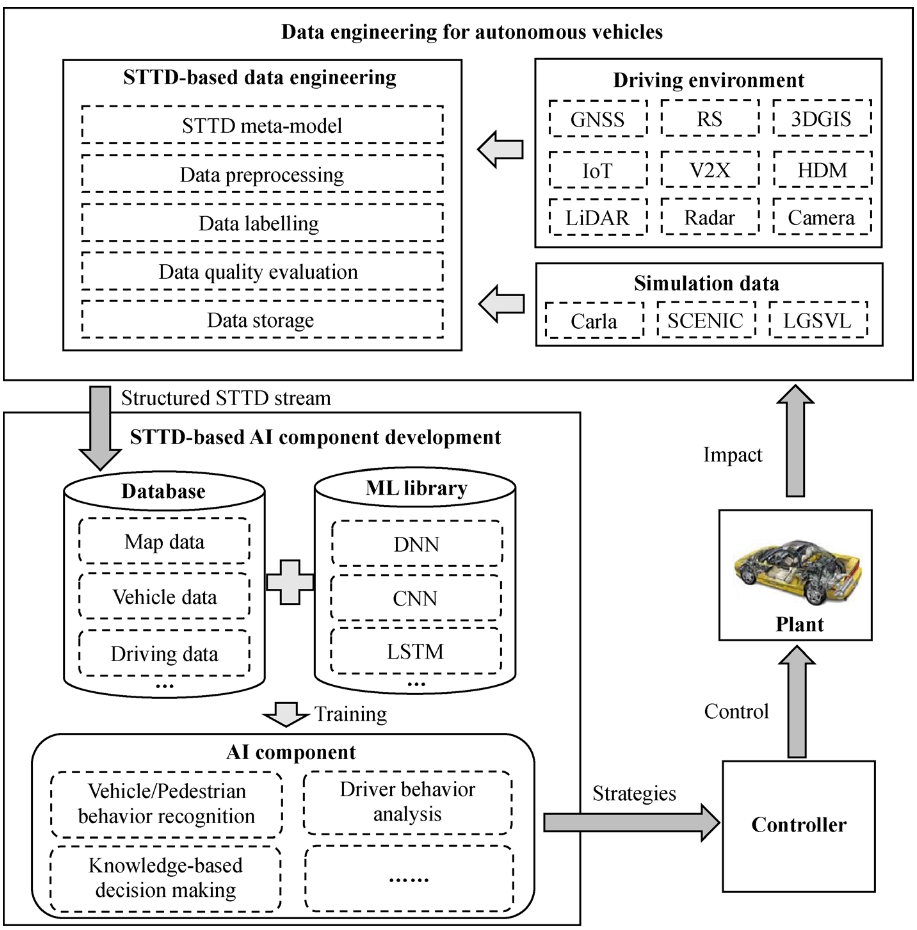

Fig. 1 Framework for autonomous driving system driven by STTD.

Carla (Dosovitskiy et al., 2017) have been applied to generate simulation data as a complementary for real world data.

In this paper, we focus on the preprocessing of STTD extracted from specific scenarios. The simulation data is generated by Carla's simulation engine, which is different from the NGSIM (NGSIM) real data set.

2.1.3 STTD-based data engineering

As the development and training of neural network maintains a close relationship with the input data, it is necessary to transform raw data into specified form and execute other preprocessing methods to ensure its quality. First, we further refine the STTD meta-model according to the data requirements, which can be used as the basis for extensive applications, such as domain knowledge-based labeling engineering, data-driven prediction algorithm design, data-driven and domain knowledge combined strategy generation, and map semantic parsing. More details of the meta-model are described in Section 3.1.

Then, we propose a data pipeline for STTD preprocessing. Under the guidance of the STTD meta-model, we extract the features from the multi-source heterogeneous data, fuse these features, label and store the output. Thus, it provides a continuous data stream for autonomous driving AI component, and helps obtain high-quality STTD. We have also designed corresponding databases. More details of the data pipeline are described in Section 3.2 and 3.3.

As above, we complete the work of sensing the environment of the autonomous vehicle by collecting and preprocessing data to get surrounding information like speed, direction and other physical attributes. Such structured data will be stored and fed into neural networks to make behavioral decisions.

2.2 STTD-based AI component development

In autonomous vehicles, AI component such as knowledge-based decision making is the key of autonomous driving. Under the guidance of STTD metamodel, structured data described in Section 2.1 can be obtained and stored in databases in Fig. 1. Then we can select required data from the database and appropriate ML model from the ML library to complete the AI component training and deployment. The most commonly used ML models in autonomous driving includes Deep Neural Network (DNN), Convolutional Neural Network (CNN) and LSTM (Emmer-Streib et al., 2020). Such AI component is further divided into various submodules, for example, vehicle/pedestrian behavior prediction is one of the most important. Learning results of the AI component are exported to Controller to obtain the optimal control strategy. While this paper focuses on LSTM, our approach also can be applied to other neural networks.

2.3 Controller

Controller will use the learning and prediction results of AI component to make the optimal decisions and generate commands to control the execution of the autonomous vehicle. This part has been initially developed by our team and will not be covered in this paper.

2.4 Plant

After getting the decisions generatedby the Controller, Plant will execute corresponding actions, such as going straight, turning left, turning right and so on. The result of its action will inevitably lead to change the vehicle's location and affects current environment. As a result, the vehicle needs to update environmental information according to a certain temporal relationship, thus to form a closed-loop system for autonomous vehicles.

3 Spatio-temporal trajectory data engineering

This section focuses on the STTD-based data engineering, presents STTD meta-model, data preprocessing and storage.

3.1 Spatio-temporal trajectory data meta-model

Figure 2 illustrates the STTD meta-model. According to ISO 26262 standard, we extracted 12 dimensions of STTD by combining the domain knowledge of autonomous driving. Our model includes data labels (Tag), data sources (Source), temporal data (Temporal), spatial data (Spatial), domain knowledge (ADS), environmental data (Environment), human data (Humanity), social data (Society), safety concerns (Critical), scene contracts (Contract), data quality (Quality), and trajectory data (Track). Based on the original meta-model (Zhang et al., 2021), we removed the redundant structure and extended Source, Spatial, ADS, Humanity, and Environment. In the Source, LiDAR, Radar and Camera are added and Sensor is removed. In the Spatial, TrafficInfrastructure is added. In the ADS, TrafficRules, ResponsibilitiesMechanism, HisTrafficData are added. Participant, Passenger and UnknowObstacles are added to the Humanity. Brightness and visibility are added to the Environment.

The data structure are shown as below in Figs. 3–8.

Note that while the STTD meta-model covers a relatively comprehensive range of data in this domain, one specific AI component may only use part of them. For example, in vehicle behavior prediction, the aim is to determine the future behavior of a vehicle (going straight, turning left, turning right, accelerating or decelerating), so in this case we just used a subset of the STTD.

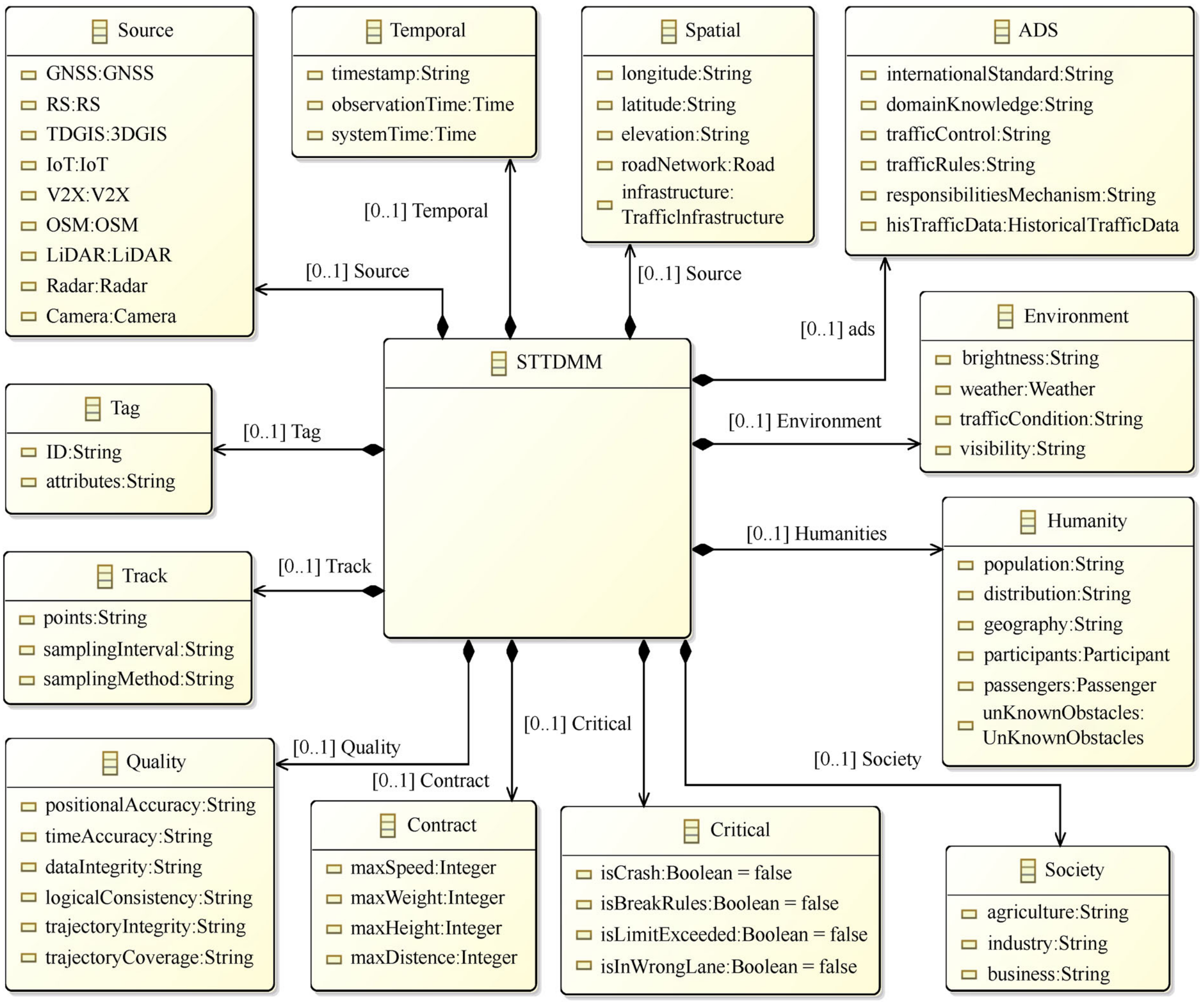

Fig. 2 Spatio-Temporal Trajectory Data meta-model for autonomous driving.

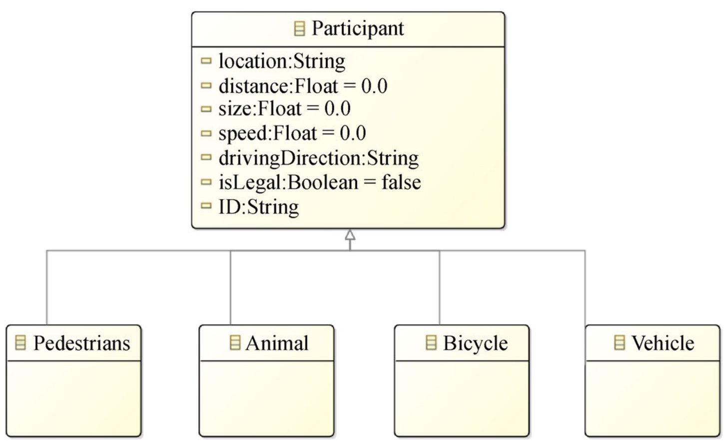

Fig. 3 Traffic participant include pedestrians, animals, bicycles, vehicles. Attributes include ID, location, distance, size, speed, driving direction, and legality of behavior.

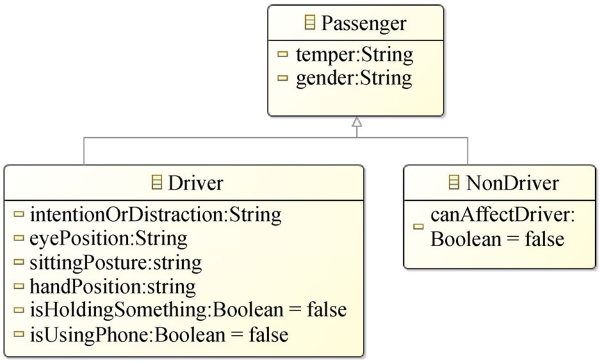

Fig. 4 Passengers are divided into drivers and non-drivers. For non-drivers, it is necessary to consider whether their behavior will affect the driver.

3.2 Data preprocessing

Data preprocessing requires high data quality and processing speed. To solve data quality issues, we need to ensure data accuracy, remove data redundancy and fill in missing data. To solve processing speed issues, we need to ensure the safety of autonomous vehicles and shorten the period of data collection and preprocessing, so that subsequent operations of autonomous vehicle can be completed in time.

In this paper, we propose a data pipeline model for multi-task parallel data preprocessing, which processes data as a continuous stream. As a result, data can be continuously read, processed, fused and transferred to the AI components with low latency. As shown in Fig. 9, the data pipeline model starts from the data sources, and input to the corresponding data processing sub-module. Then based on our STTD meta-model, features are extracted from the clean data, and transferred to the multi-source data fusion module. Finally, high-quality STTD can be obtained. Therefore, the following discussion is of four aspects: data cleaning, data feature extraction, data fusion and data storage.

3.2.1 Data cleaning

The data cleaning of spatio-temporal trajectory mainly deals with outlier data, eliminates redundancy in the data, and fills in missing data, which has an essential effect on subsequent data processing.

Due to the environmental noise and the limitation of the

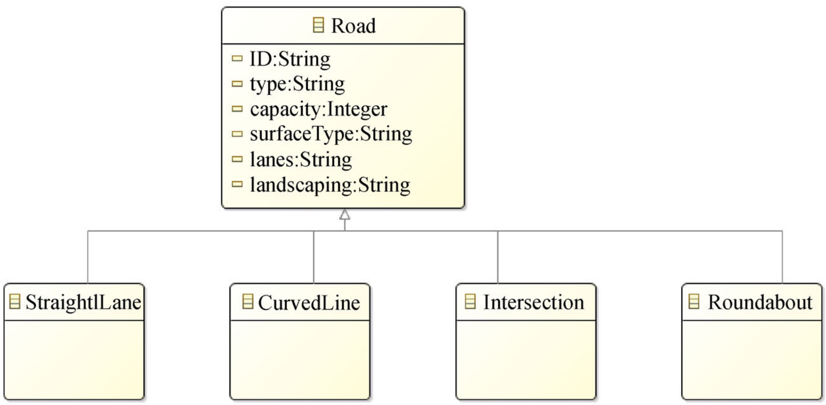

Fig. 5 Road types include straight roads, curved roads, junctions, roundabouts, etc. Attributes include ID, road type, road surface condition, lanes, landscaping.

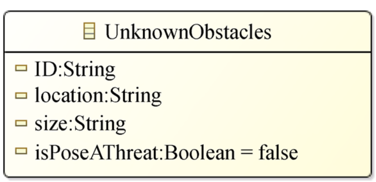

Fig. 6 Unknown obstacles are objects that cannot be anticipated, such as rubbish bins, branches, and so on.

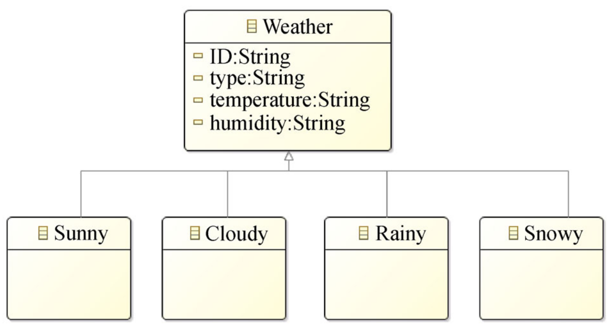

Fig. 7 Weather data include Sunny, Cloudy, Rainy, Snowy. Attributes include ID, weather type, temperature, humidity.

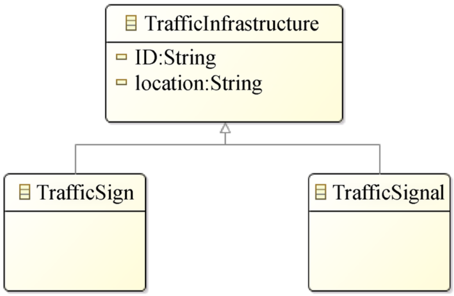

Fig. 8 Traffic infrastructure includes traffic signs, traffic signals, etc.

accuracy of the sensor itself, the data captured by autonomous vehicle have a certain degree of errors, causing outlier data, such as the deviation of the vehicle from the road center. Different noise filtering methods are applied to different types of outlier data. There are three kinds of noise filtering methods (Pannekoucke et al., 2016). Mean filter is a commonly used method in image processing. It can help eliminate sharp noises in images and achieve image smoothing and blurring. Kalman filter is usually used for predicting state over time. It uses a series of measured values observed to generate estimates of the object. For example, the estimation of vehicle position and vehicle velocity can be done via the Kalman filter. The particle filter is widely applied in object tracking, car navigation, and position estimation.

Due to the different structures, sampling frequencies, and data formats of data collection devices, trajectory data can be redundant and missing. Data redundancy includes duplicate records or meaningless data segments. Several methods can be applied to solve such problems, such as trajectory compression (Meratnia and de By, 2004), clustering (de Vries and van Someren, 2010), and binning (Fung and Zheng, 2018). Low signal and insufficient equipment reliability result in missing data. Fitting methods and regression methods are usually used to fill the missing data. Regression methods can also be used to smooth large volumes of data. If data are seriously missing (over 50% of the corresponding feature), we will consider discarding them.

3.2.2 Data feature extraction based on STTD meta-model

When constructing data structure of spatio-temporal trajectory, some data characteristics are important for autonomous driving while irrelevant raw data are negligible. Useful data features for STTD need to be extracted (Li et al., 2018). For example, we choose a subset of the clean data's most critical features, leaving irrelevant or redundant features unused. This operation can help to optimize model accuracy and accelerate the training process. A useful method in data feature extraction is Principal Component Analysis (PCA). In PCA, data features are compressed through a linear transformation. It helps to reduce the number of data dimensions while remaining the most significant aspects in the data set.

Therefore, guided by the STTD meta-model, we extracte features from the clean data obtained by previous section. The feature includes car's information of current location, speed and characteristics, and traffic participants' information, road network information, traffic signs information, weather and other information. We organize these features as $X = \{x_1, x_2, ..., x_n\}$.

3.2.3 Data fusion

After data cleaning and data feature extraction procedure, we have already ontained clean and correct STTD.

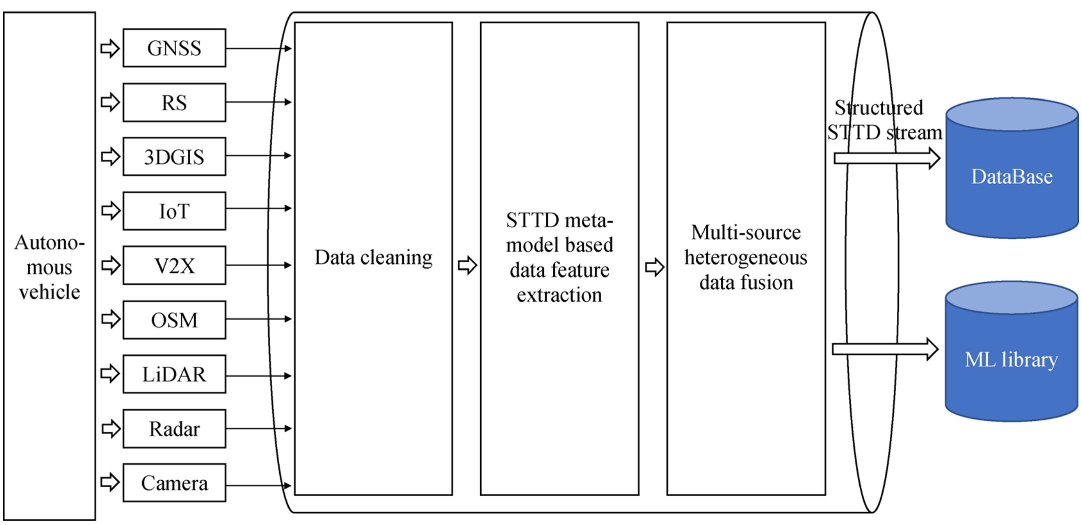

Fig. 9 Spatio-Temporal Trajectory Data preprocess for autonomous driving system.

However, the fusion of heterogeneous data from multiple sources is the biggest bottleneck in big data processing and analysis. So, multi-source data need to be transformed into an appropriate format that machine learning algorithms can understand. Aggregation and Normalization are the two types of data transformation techniques. Data aggregation is the crucial method for creating structured spatio-temporal trajectories, like property society and environment, which is done on a large amount of cleaned data through tools known as data aggregators. Data normalization turns the data into unified scale. The different scales without scaling will make ML algorithms take more time to train and get low accuracy.

3.3 Data storage

The STTD collected by AVs is massive, to store it, advanced data storage technology is required. In the data storage module, we have designed four databases: vehicle feature database, map database, vehicle driving information database, and autonomous driving domain knowledge database; after completing the collection and feature extraction, STTD can be stored in these databases for later model training, knowledge inference, tagging engineering, or driver behavior analysis (Bai et al., 2019). With the help of modern advanced technology, it is possible to use MySQL, Redis, and HBase for data storage. And combining technologies such as cloud computing, onboard computing platforms and edge computing to improve the efficiency of data processing (Zhang et al., 2018).

4 Case study: vehicle behavior prediction based on spatio-temporal trajectory data

Data-driven modules play an increasingly important role in autonomous driving systems, such as the AI component of vehicle behavior prediction module. In this section, vehicle behavior prediction is used as an example to illustrate how the data-driven approach can be applied to autonomous driving systems.

As shown in Fig. 10, the training process goes through the following steps: first, the STTD will be preprocessed with the method we proposed in Section 3.2 to construct the vehicle trajectory samples. We use simulation data generated by Carla as experimental data input. Second, vehicle trajectory samples will be labeled automatically or manually. If the data is collected by the real-world sensors, people need to do label extraction by calculating the vehicle's heading angle. But simulation data can generate label autonomously. Third, labeled vehicle trajectory samples will be fed into the vehicle behavior prediction model to do the classification task. By comparing the prediction label (y_1,y_2,y_3) and the true label (t_1,t_2,t_3), we calculate the cost of this model and iterate it to a convergent state. Since LSTM has an inherent advantage in processing time-series data, we use LSTM to predict autonomous vehicle's driving behavior combined with high-quality STTD.

4.1 Data set

This subsection describes data collection and data

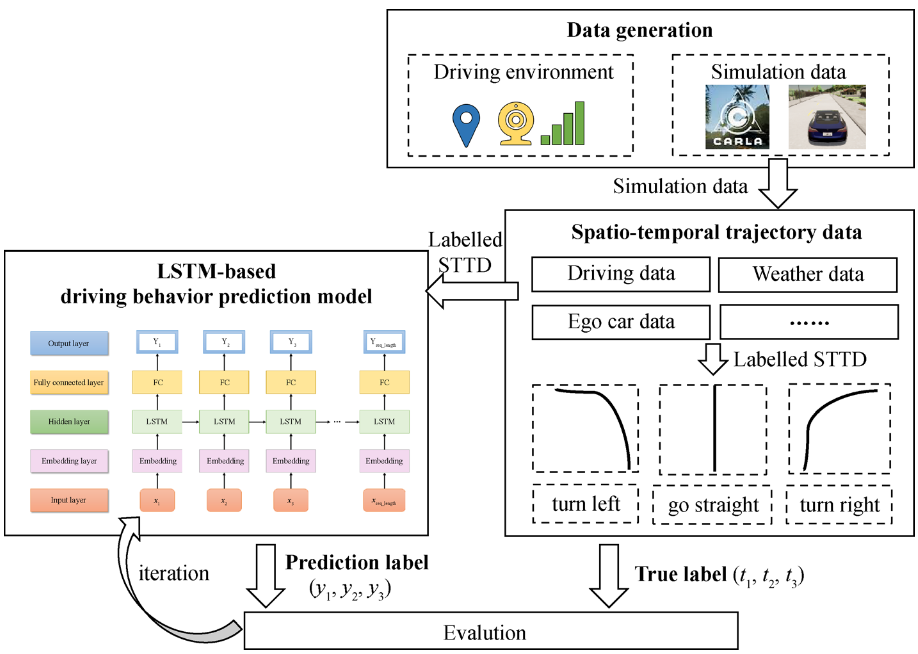

Fig. 10 Vehicle behavior prediction based on STTD.

preprocessing, and data set used for the experiment are present.

Due to the high cost of the data collection in the real-world and the poor quality of real-world data sets, generating the simulation data is another effective way to provide data source of the experiment. We use SCENIC (Fremont et al., 2019) and Carla to generate the simulation data. It allows for the generation of enough data with reasonable control of the ratio of data types, which is very important for the model training and if the ratio is not balanced, it will make the model perform poorly. What's more, we can generate STTD for specific scenarios, such as intersection.

We define three kinds of vehicle behavior ("Turning left", "Turning right", "Going straight") in SCENIC. For the vehicle trajectory data for intersection, every sample contains 4 s of data, and the frequency of data collection is 0.1 s, so each sample is formed of 40 rows of records. To explore some experimental comparisons, we apply different data generation strategies to the "Going straight" example with data generated in 3 s. The data set is split into the training set and the testing set with a 7:3 ratio. Final data set consists of 3000 pieces of data, divided into three groups: 1000×30 pieces of straight-ahead data, 1000×40 pieces of left-turn data, and 1000×40 pieces of right-turn data. The data set is shown in the Table 1. As shown in Table 1, four features (longitude, latitude, brightness, weather) are considered in our experiment, corresponding to the elements in the metamodel, respectively. Longitude and latitude are all real value types, while light and weather are both discrete value types. And we deal the discrete data using one-hot algorithm. The light condition has three types: strong light, sufficient light, and weak light. The weather feature has 4 different types: Sunny, Cloudy, Rainy, and Snow.

4.2 Vehicle behavior prediction

The vehicle behavior prediction model is constructed based on LSTM model, a many-to-one structure. As shown in Fig. 11, the LSTM has five layers, including the input layer, embedding layer, LSTM hidden layer, fully connected layer, and output layer. The loss function is chosen as "CrossEntropyLoss," which is shown in Eq. (1). Moreover, "Adam" is our choice for the optimizer. The

Table 1 Simulation data set based on STTD meta-model

Sample	Timestamp	Longitude	Latitude	Brightness	Weather	True label
X_1	x_1	–51.0548	15.16971	Strong light	Sunny	Turn left
	x_2	–51.0548	15.16938	Strong light	Sunny	
	⋮	⋮	⋮	⋮	⋮	
	x_{40}	–36.61	–2.90651	Strong light	Sunny	
	⋮	⋮	⋮	⋮	⋮	⋮
X_n	x_1	–67.8006	–6.21437	Sufficient light	Cloudy	Turn right
	x_2	–67.8002	–6.21437	Sufficient light	Cloudy	
	⋮	⋮	⋮	⋮	⋮	
	x_{40}	–53.5013	–19.7462	Sufficient light	Cloudy	

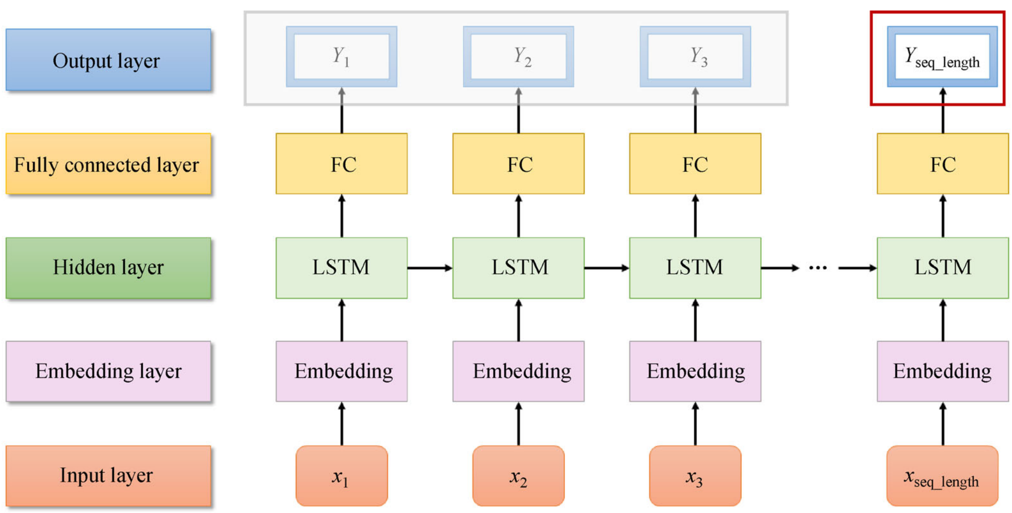

Fig. 11 LSTM for training vehicle behavior prediction module.

input $x_1, x_2, \ldots, x_n$ of the model in Fig. 11 is corresponding to Table 1:

$$L = -\sum_{i}^{N} y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right), \quad (1)$$

and hyperparameters are shown in Table 2.

4.3 Experiment result

The training process and the testing process are shown in Fig. 12. Model accuracy of the training set achieves the highest 98.97% in the No. 37 epoch, the model accuracy of testing set achieves highest 98.58% in the No. 23 epoch, the minimum model loss of training set is 0.0477 in the No.48 epoch, and the minimum model loss of testing set is 0.0744 in the No.48 epoch.

As is shown in Tab. 3, in addition to model accuracy, we also output other model evaluation metrics in the last epoch, including F1, Recall and Precision. Average of evaluation results for metrics is high, but we can find that the evaluation value of “Going Straight” is lower than other labels, we guess the reason is that we only generate 30 records for a “Going Straight” sample.

5 Conclusions and outlook

In this paper, a STTD-driven approach is presented from a data engineering perspective. For the development of autonomous driving, it has four module: a domain data meta-model, data collection, data preprocessing and data storage. We also give a case study, using STTD and LSTM model to predict the vehicles’ behaviors. The experiment results show that our approach facilitates the data-driven development for AVs.

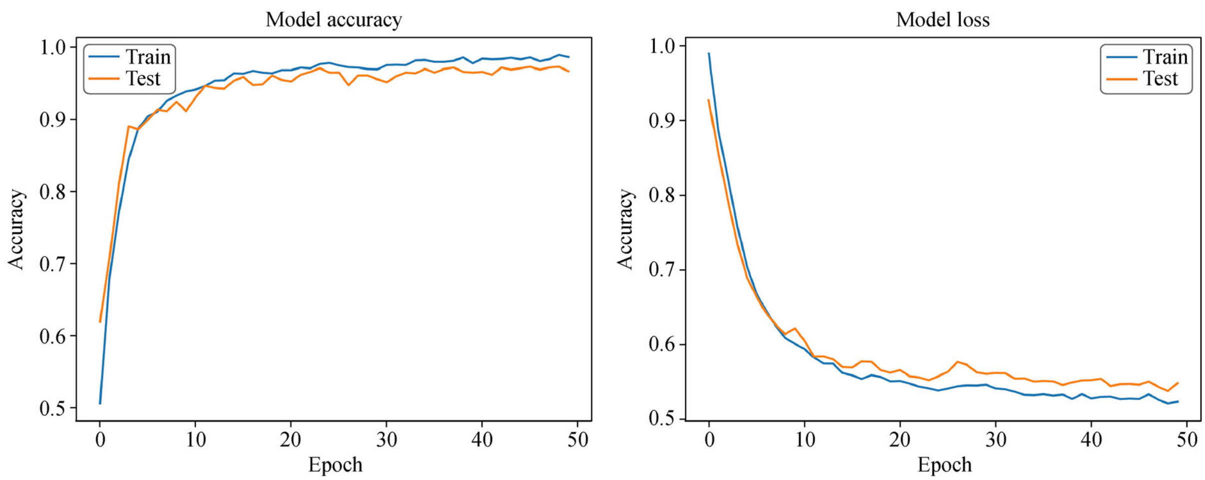

Fig. 12 Model accuracy and model loss of the training set and the testing set.

Table 2 Hyperparameters of LSTM for vehicle behavior prediction module

No.	Hyperparameter	Value
1	Embedding size	1
2	Epoch	50
3	Batch size	10
4	Hidden size	128
5	Sequence length	40
6	Hidden layer	1

Table 3 Result of LSTM model based on STTD data set

Item	Turning left	Turning right	Going straight	All
Accuracy	94.63%	96.1%	94.57%	96.06%
F1	92.52%	94.43%	85.98%	96.05%
Recall	99.60%	99.20%	100%	96.00%
Precision	86.38%	90.10%	92.46%	96.10%

The tool development is very important for applying our approach. In our future work, we will build a data-driven integrated development platform to integrate and automate data preprocessing, model training and intelligent system building.

Acknowledgements Financial supports for this work, provided by the National Natural Science Foundation of China (Grant No. 61972153), the National Key Research and Development Program (No. 2018YFE0101000), the Key projects of the Ministry of Science and Technology (No. 2020AAA0107800), are gratefully acknowledged.

References

Bai X, Xu C, Ao Y, Chen B, Du D (2019). Learning-based Probabilistic Modeling and Verifying Driver Behavior using MDP. In: 2019 International Symposium on Theoretical Aspects of Software Engineering (TASE). IEEE, 152–159

Czarnecki K (2018). Automated driving system (ads) task analysis—part 2: structured road maneuvers. Waterloo Intelligent Systems Engineering Lab (WISE) Report, University of Waterloo

Emmert-Streib F, Yang Z, Feng H, Tripathi S, Dehmer M , (2020)). An introductory review of deep learning for prediction models with big data. Front Artificial Intell, 3: 4–88

de Vries G, van Someren M (2010). Clustering vessel trajectories with alignment kernels under trajectory compression. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Berlin: Springer. 2010: 296–311

Dosovitskiy A, Ros G, Codevilla F, Lopez A, Koltun V (2017). CARLA: an open urban driving simulator. In: Conference on robot learning. PMLR, 2017: 1–16

Fremont D J, Kim E, Dreossi T, Ghosh S, Yue X, Sangiovanni-Vincentelli A L, Seshia S A (2019). Scenic: a language for scenario specification and scene generation. In: Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation, 63–78

Fung S F, Zheng A (2018). Effects of data binning techniques on results of analyzing solar wind and geomagnetic indices data. In: AGU Fall Meeting Abstracts, 2018: SM31D–3525

Li J D, Cheng K W, Wang S H, Morstatter F, Trevino R P, Tang J L, Liu H (2017). Feature selection: a data perspective. ACM Comput Surv 50(6): 94

Meratnia N, de By R A (2004). Spatiotemporal compression techniques for moving point objects. In: Bertino E, Christodoulakis S, Plexousakis, Chriltophides V, Koubarakis M, Böhm K, Ferrari E, eds. Advances in Database Technology-EDBT 2004. Conference Proceedings EDBT 2004. Berlin: Springer

Pannekoucke O, Cébron P, Oger N, Arbogast A (2016). From the Kalman filter to the particle filter: a geometrical perspective of the curse of dimensionality. Adv Meteor, 2016: 9372786

Paardekooper J P, Comi M, Grappiolo C, Snijders R, van Vught W, Beekelaar R (2021). A hybrid-AI approach for competence assessment of automated driving functions. In: CEUR Workshop Proceedings. CEUR-WS, 2808(2808)

Park S H, Kim B D, Kang C M, Chung C C, Choi J W (2018). Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture. In: 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018: 1672–1678

Seshia S A, Desai A, Dreossi T, Fremont D J, Ghosh S, Kim E, Shivakumar S, Vazquez-Chanlatte M, Yue X Y (2018). Formal specification for deep neural networks. In: International Symposium on Automated Technology for Verification and Analysis. Cham: Springer, 2018: 20–34

Wang Y (2020). Modeling, simulation and verification of autonomous driving scenario based on model-driven. Dissertation for the Master's Degree. Shanghai: East China Normal University (in Chinese)

Xu K, Xiao X, Miao J, Luo Q (2020). Data driven prediction architecture for autonomous driving and its application on apollo platform. In: 2020 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2020: 175–181

Zhang L, Zhang C X, Du D H, Liu B, Tian B, Yuan Q (2019). Spatial information spatial metadata construction method. Shanghai: CN110532340A, 2019–12–03

Zhang M H, Du D H, Zhang M Z, Zhang L, Wang Y, Zhou W T (2021). Spatio-temporal trajectory data-driven autonomous driving scenario meta-modeling approach. J Softw, 32(4): 973–987 (in Chinese)

Zhang Q, Wang Y, Zhang X, Liu L, Wu X, Shi W (2018). OpenVDAP: an open vehicular data analytics platform for CAVs. In: 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2018: 1310–1320

Zhao L, Ichise R, Mita S, Sasaki Y (2015a). Core ontologies for safe autonomous driving. In: International Semantic Web Conference

Zhao L, Ichise R, Yoshikawa T, Naito T, Kakinami T, Sasaki Y (2015b). Ontology-based decision making on uncontrolled intersections and narrow roads. In: 2015 IEEE intelligent vehicles symposium (IV). IEEE, 2015: 83–88