Motion Prediction Using Data-Driven Approach In Autonomous Vehicle

Jaianam Chetankumar Shah
Commercial Vehicle Technology
Technical University Of Kaiserslautern
Kaiserslautern, Germany
jshah@rhrk.uni-kl.de

Varad Vaibhav Avchat
Commercial Vehicle Technology
Technical University Of Kaiserslautern
Kaiserslautern, Germany
avchat@rhrk.uni-kl.de

Pinak Rameshbhai Savaliya
Commercial Vehicle Technology
Technical University Of Kaiserslautern
Kaiserslautern, Germany
savaliya@rhrk.uni-kl.de

***Abstract*—In this paper, our focus is to achieve the optimal error of three parameters: Average Displacement Error (ADE), Average Displacement Error (FDE), and Average Heading Error (AHE) in motion prediction by using different Neural Network approaches like LSTM (Long Short-Term Memory) and FCN (Fully Convolutional Network). In first section we have explained background information of Motion Prediction strategy for Autonomous Vehicle and different types of it. In second section we have explained various Deep learning methods that are used for Motion Prediction. In third section information regarding intersection Drone dataset (inD) is explained. In that part How data collection is done and how data is further processed and prepared in CSV file for easy understanding is elaborated. In the experimental analysis part whole process pipeline of our project is explained. In that basic explanation of python code is given and our own methodology is discussed we have done like developing code for the LSTM model, the impact of changing the different hyperparameters of the model, and selection of different datasets and how is error changing accordingly to it. A comparison of result of LSTM and FCN Keras model is explained to depict which strategy is optimal to train the neural network model and obtain the optimal solution of three errors. In result by doing this project we have achieved the optimal error solution by implementing LSTM model.**

***Keywords*— Neural Network, LSTM model, FCN Keras model, Motion prediction.**

I. INTRODUCTION

Autonomous driving technology has recently emerged as a cutting-edge in global automotive engineering [1]. Both business and academia are actively investing significant resources in the worldwide development of autonomous driving technologies [1]. Here for both the system Advance Driver Assistance System (ADAS) and Autonomous Vehicle safety is the first criterion that must be considered [2]. An autonomous vehicle needs, however, not only to be aware of the present conditions of the neighboring road users for safe and effective operation on roadways but also proactive and foresee what they will do in the future [3]. The definition of safety implies that to estimate the risk associated with a certain scenario, mathematical models that forecast how that condition will evolve in the future are required [2]. In contrast to people's behavior, automobiles' behavior is controlled by their higher inertia, driving laws, and road design, which may assist minimize the problem's complexity [3]. However, there are new issues because of the interdependence of vehicle behavior, the impact of traffic laws, and the environment in which we drive [3]. Practical restrictions on seeing the immediate surroundings and the necessary processing resources to run prediction algorithms all contribute to the problem's difficulties [3]. Different types of approaches are considered by different researchers to evaluate the motion prediction problem statement. According to [4], the review on the method to investigate motion prediction by vehicle surveillance, driver behavior, and intersection safety analysis.

The combination of physics-based models combined with Advance machine learning techniques that are Dynamic Bayesian Networks and deep learning methods are useful to solve the motion prediction problem of an autonomous vehicle [3].

A) Motion Prediction Overview

The motion prediction model can be divided into three further models: physics-based, Maneuver-based, and Interaction-based [2].

a) Physics-based model

This type of model is considered a dynamic model works based on the physics law that is worked based on the law of physics [3]. Here data is used like internal control parameters of vehicle steering, acceleration, and other dynamic and kinematic properties [3]. Apart from that it also considers other external parameters like road conditions also are considered [3]. In Autonomous cars, this model is utilized for trajectory prediction and risk assessments [3].

b) Maneuver-based model

Maneuver-based models presume that the motion of a vehicle on the road network corresponds to a sequence of maneuvers carried out separately from the other cars and depict vehicles as autonomous maneuvering entities [3]. With maneuver-based motion models, trajectory prediction is based on the early identification of the movements that drivers want to execute [3]. Either prototype trajectories or maneuver intention estimates serve as the foundation for maneuver-based motion models [3].

c) Interaction-aware model

In this Model, Vehicles are considered interaction-aware motion models as moving objects that interact with one another [3]. In other words, it is presupposed that a vehicle's motion is affected by the way other vehicles in the area move [3]. In this model, there is much more inter-relation between vehicles that depicts much more accurate motion prediction compared to the Physics-based model and the Maneuver-based model [3].

B) Motion Prediction Parameters

Motion prediction mainly consists of three parameters: stimuli, Modelling approach, and Prediction [3].

a) Stimuli

Most prediction techniques rely on a partial trajectory that has been seen and on sequences of observations of the vehicle and it's surrounding state variables like locations, and velocities [5]. This is frequently offered by a target tracking system and additional inputs like scene geometry is considered [5].

b) Modelling approach

Different methods for predicting human motion have different representations, parametrizations, learning processes, and solutions [5].

c) Prediction

Various techniques result in various parametric, non-parametric, or structured forms of predictions, including Gaussians over agent states, probability distributions over grids, single or many trajectory samples, or motion patterns utilizing graphical models [5].

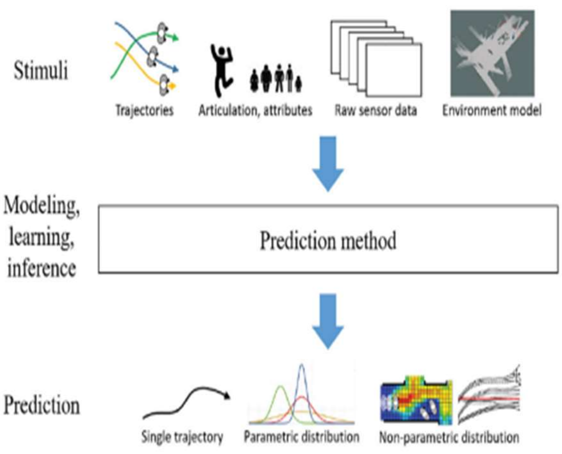

Fig. 1 Parameters of Motion Prediction [5, Fig. 2]

The interconnection of different parameters for motion prediction is shown in [5, Fig. 1].

Interaction-aware models are the more sophisticated models that take into account interaction between vehicles [3]. This model mostly works on the deep-learning principles [3]. According to [3, Fig. 2] classification is divided into three major categories: Input representation, Output Type, and Prediction Method. In input representation methods different internal parameters of vehicles like vehicle's speed, position, The current situation and past data are considered as input and the external environment parameters like traffic signal data, and pedestrian recognition is observed by sensors [3]. In the output type model, the principle of Manoeuvre intention is to predict from the previous data to take a decision on the vehicle's current position Unimodal Trajectory uses the principle of the time window in a continuous manner to predict the output trajectories of vehicles [3]. In the independent of the intended manoeuvre model, the impact of other manoeuvres is neglected to get more accurate results [3]. When the region of interest is long and wide, this strategy can be quite helpful [3]. Multimodal trajectory models worked on the principle that trajectory will depend on separate behaviour of every manoeuvre that is based on static models or dynamic models [3]. In the occupancy map strategy, trajectories are predicted based on the map of the driving environment data stored and using it in the prediction timestamp [3].

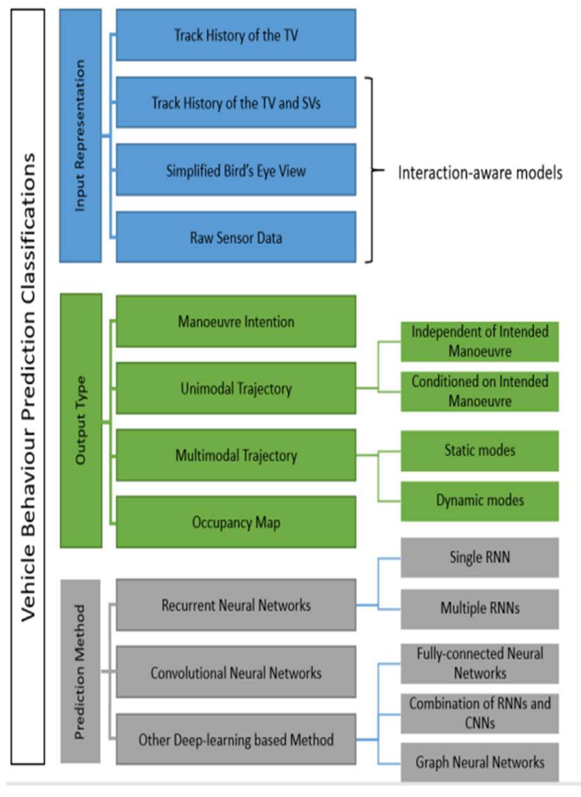

Fig. 2 Classification of vehicle trajectories prediction methods [3, Fig. 2]

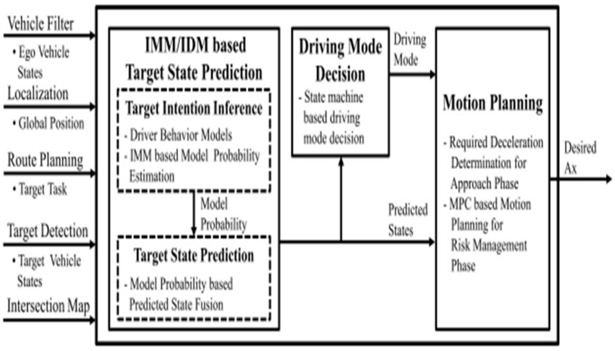

Fig. 3 Architecture of Motion Prediction [6, Fig. 1]

The basic architecture of motion prediction is shown in [6, Fig. 3]. Here input modules are vehicle Fillers, Localization, Route Planning, and Intersection map, and motion planning is performed based on that [6].

II. NEURAL NETWORK-BASED METHODS

Here the Neural network-based methods are further classified into three models: Recurrent neural networks (RNN), Convolutional Neural Networks (CNNs), and different other models [3].

a) Recurrent Neural Networks

RNNs operate on sequential data, including time series presented as text, image, or audio [7]. To create the current output, it makes use of a few prior pieces of information from the sequence [7]. The most basic recurrent neural network may be thought of as an expansion of a two-layer fully connected neural network where the hidden layer has feedback [3]. The RNN analyses the data from the current step's input combined with the memory of prior steps, which is stored in the previously hidden neurons, at each step of the sequence [3]. An RNN model with enough hidden units may theoretically learn to approximate any sequence-to-sequence mapping and give efficient results but for long data sequences, it is not possible to train the model because of gradient exploding [8]. This RNN model is further divided into single RNN and Multiple RNN models. A basic diagram of RNN network is shown below in [9, Fig. 4]

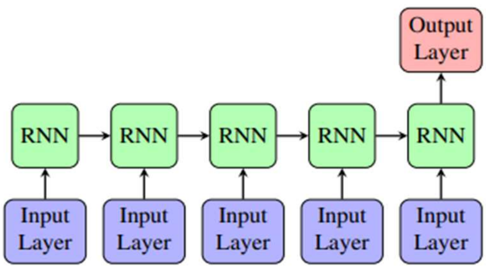

Fig. 4 Model of network for a sequence length of 5 [9, Fig. 5]

1) Single RNN model

This Model uses only a single recurrent neural network or uses the combined second model with the basic primary model [3]. By using a combined model, it can predict the interaction and multimodal forecasting [3]. According to [9] LSTM model used in the RNN network is considered as a sequence classifier and data is fed to each layer of LSTM model. An LSTM can avoid the vanishing gradient issue due to its ability to remember a value for any amount of time [9]. LSTM model eliminate the problem occurred into RNN model in that typically RNN model have exponentially increasing or decreasing input, which makes training difficult [10]. LSTM model has the parameter called gate: input gate, output gate, forgot to get and through this component the LSTM network regulates the status of cells and modifies or adds information to it [11]. The input gate receives input from the latest iteration's output of the LSTM neural network cell, which regulates the volume of input to the memory cell [11]. By deciding when to forget the output, the forget gate may choose the ideal time delay for the input sequence [11]. The LSTM neural network unit, which regulates the output's strength, receives all the computed results and outputs them to the output gate [11]. Below [11, Fig. 5] shows the diagram of basic LSTM network.

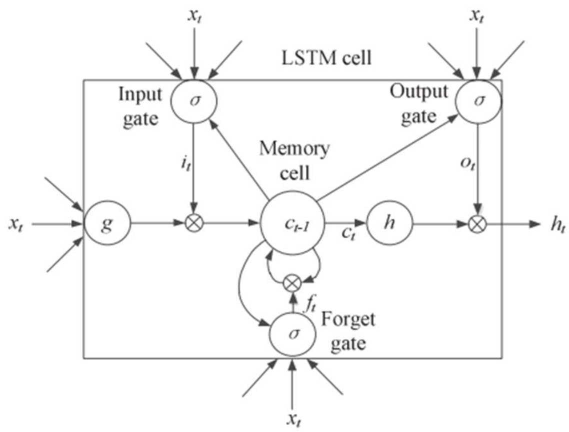

Fig. 5 LSTM model [11, Fig. 2]

Here input variable is t_x and output variable is t_h [11]. The vehicle trajectories from the past are processed using an encoder LSTM [11].

b) Convolutional Neural Networks

A convolution neural network (CNNs) is a deep neural network that uses convolution operations to operate in layers [12]. CNN is normally made up of three layers: a convolutional layer, a pooling layer, and a fully connected layer [13]. The convolution procedure is used in the convolutional layer to learn local characteristics automatically [13]. To decrease the amount of data and training period a pooling layer is introduced [13]. A combination of convolution and pooling layers automatically collected features and utilized it to train a fully connected neural network layer [13]. It is mainly used to extract the data from image to predict the motion [3]. Convolutional neural networks are valued in the prediction of vehicle behavior because they can take image-like data, produce output that looks like it, and maintain the spatial connection of the input data while processing it [3]. Below is the [13, fig. 6] basic diagram of convolution-based network is shown.

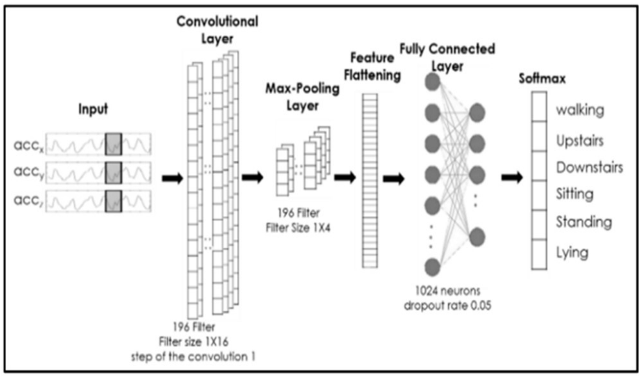

Fig. 6 Block diagram of CNN network [13, Fig. 3]

c) Combination of CNNs and RNNs networks

Due to the capacity of extracting temporal functions of the RNNS model extracting the capacity of spatial layers from convolution in CNNs model combined model can be formulated [3]. CNN is applied to simplified pictures, each of which represents the environment surrounding the Target vehicle in a separate time period [3]. The recovered feature sequence is then fed into an Encoder-Decoder LSTM so that eventually it can learn the behavior dynamics of the raw data [3]. In order to create output visuals that depict how the area surrounding the Target car will change over the coming time steps, the decoder LSTM outputs are sent to a deconvolutional neural network [3].

d) Artificial neural networks (ANNs)

The input layer, hidden layer, and output layer make up the foundation of an ANNs Multiple layers of neurons may make up the hidden layer. [14]. If enough data and the right initialization are provided, ANNs can alter their internal structures to offer the optimal solution and if the right inputs are given to an ANN, it can simulate the learning process by picking up knowledge from its surroundings, and users may subsequently recall this information [15]. The basic diagram of ANNs neural network is shown below in [14, Fig. 7].

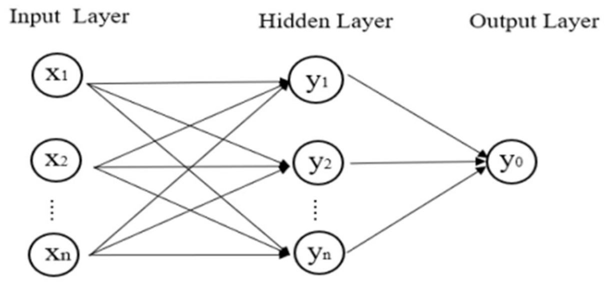

Fig. 7 ANNs (Artificial Neural Networks) [14, Fig. 1]

III. Data set used in the experiment

In this experiment, we have used the data set that is provided by [15]. For the data collection, camera-equipped drone is used [15]. High longitudinal and vertical resolution cameras on drones enable for the recording of traffic from a so-called bird's eye perspective that helps to achieve optimal result of longitudinal and lateral data [15]. The traffic behavior observed was normal given that the drone was hovering at a height of more than 60 meters [15]. In [15] naturalistic road user trajectory dataset is created called intersection Drone dataset at German intersections and it was compared with Stanford Drone dataset to validate German inD dataset [15]. The following requirements like a sufficient amount of data size, maintain Natural behaviour of pedestrians, integrate different types of data with respect to time, considering other influencing parameters other than pedestrians was fulfilled in order to achieve optimum result [15].

To generate inD dataset at intersections process pipeline is followed: Selection Of Recording Sites according to safety criteria developed, Recording the data in normal weather condition and pre-processing the data with High resolution camera, Detection the data with 2D position of pedestrians to create the trajectory dataset and classification of dataset by using semantic segmentation process, Tracking the dataset by tracking algorithms and post-processing the data to obtain x- and y-directional smooth locations, speeds, and accelerations by Bayesian smoothing process, and in the end Dataset is formatted meta data and each dataset is constructed in CSV files. From 10 hours of footage, we used deep learning algorithms to extract the pixel-accurate trajectories of 13 599 road users, including automobiles and pedestrians and recordings made over four different sites.

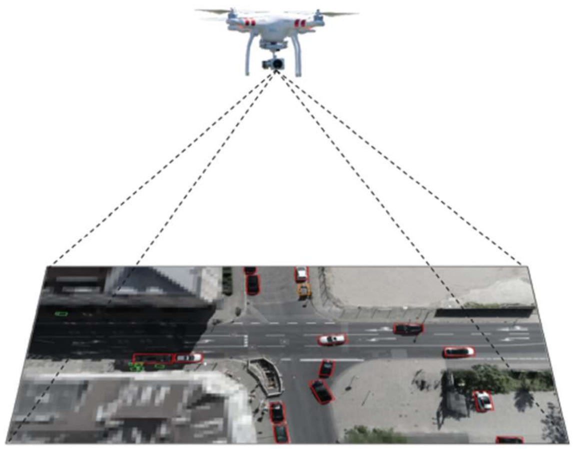

Fig. 8 Image Capturing with Drone [15, Fig. 2]

In the above [15, Fig. 8] the how intersection of location is captured by drone is shown.

IV. Experiemental analysis

The main aim of this experiment is to achieve optimal errors: the Average displacement Error (ADE), Final

Displacement Error (FDE), and Average Heading Angle (AHE) for the motion prediction competition in the class of seminar electromobility 2022. Firstly, the python code is given by the faculty of EIT at TU Kaiserslautern. In that python code, we have changed in main python file line no 9 by making DATA_DRIVEN_PREDICTION = True to use the Data-driven approach. We also have changed the Testing ID from ‘19’ to ‘28’ as per the rule for competition.

In main Python code following process is observed. Firstly, libraries that are used for this project is imported by import command. After that the code for reading the dataset with a particular location ID. After that code for down sampling and normalizing the data is written. After the code for data preparation that will later use for training and testing purpose is written. In next part Neural Network training code is executed and that is compared in the part of the code called ‘collect ground truth’. Then an Excel sheet is automatically generated for the evaluation part and in final part of code result is obtained.

A) Methodology Used

Firstly, we have used the FCN Keras model with different hyperparameters. In that, we have used 6 hidden layers with different numbers of neurons: 900, 750, 550, 375,100,50,10. We have used the activation function ‘tanh’ and the epoch was used at 100. The result was that AHE was 8.136 m, FDE was 8.967 m, and AHE is 9.96 degrees. That shows not as much as error reduction as we want. After that, we changed the parameters: the activation function used is ‘relu’, the number of epochs we have increased to 125 and neurons changed in the power of 2, the number of hidden layers reduced from 6 to 3 and we got a positive result that was for 27 number Location ID and it was AHE was 5.176 m, FDE was 5.857 m and AHE was 7.45 degree. From this, it can be depicted that by reducing the number of Hidden layers and increasing Epochs and selecting the number of neurons in the power of 2 is a better combination of parameters.

We have trained the FCN Keras model excluding location ID ‘28’. In all Location ID. we have used the same parameters that are mentioned above.

After this result to achieve a more optimal result we have to train the model on the LSTM network. Due to the advantage of handling more complex data accurately, we have implemented that code to train our model. Below Fig. 16 shows the hyperparameters that we have used to train different location IDs. In the LSTM model firstly, we have selected the two hidden layers, Epoch was 50 and the activation function selected was ‘tanh’. Number of neurons selected 64, 64,32,32 respectively. The average Result was recorded is not optimal compared to FCN Keras model. After that we have changed the parameters. we have used two hidden layers with 64 and 32 neurons respectively and a dense layer with 10 neurons. In input layer number of neurons was 128. All the neurons are selected in the multiple of two. Result achieved was optimum. It was trained on Location ID 27. AHE was 1.568 m, FDE was 1.713 m and AHE was 6.48 degrees. In below [Fig. 09] the code of the hyperparameter is shown with that we have achieved optimum result.

```
model = keras.Sequential()
model.add(LSTM(128, return_sequences=True,
          input_shape=(n_input,1)))
model.add(LSTM(64, return_sequences=True))
model.add(LSTM(32))
model.add(Dense(10, activation='relu'))
model.add(Dense(n_output))

model.compile(loss='mae', optimizer='adam')
# callback =  tf.keras.callbacks.EarlyStopping(monitor='loss', patience=3)
history = model.fit(xTrain, yTrain, epochs=50, batch_size=64,
          validation_data=(xTest, yTest), verbose=1, shuffle=False)
```

Fig. 09 Parameters of the LSTM model

We have collected all the data that we have trained on different IDs and it is shown below in two different tables.

TABLE 1 LSTM Model result

Location ID	ADE (Average Displacement Error)	FDE (Final Displacement Error)	AHE (Average Heading Error)
21	4.716 m	4.657 m	12.03 degree
22	4.243 m	4.258 m	3.29 degree
23	2.216 m	1.196 m	2.09 degree
24	4.817 m	5.152 m	7.08 degree
25	7.453 m	7.39 m	8.45 degree
26	3.072 m	3.2 m	4.08 degree
27	1.568 m	1.713 m	6.48 degree

TABLE 2 FCN Keras Model Result

Location ID	ADE (Average Displacement Error)	FDE (Final Displacement Error)	AHE (Average Heading Error)
21	8.681 m	8.975 m	24.35 degree
22	5.821 m	5.607 m	8.42 degree
23	3.957 m	4.131 m	8.06 degree
24	4.958 m	5.334 m	10.03 degree
25	9.447 m	9.357 m	9.82 degree
26	3.897 m	3.896 m	11.51 degree
27	3.163 m	3.104 m	8.62 degree

From above two table comparison can be made for FCN Keras model and LSTM model. It can be shown from the result LSTM model has provided optimal output compared to FCN Keras model.

V. Conclusion

In conclusion, by using different hyperparameters for different location IDs for LSTM and FCN Keras Model it can be proved that we have achieved optimal errors for seminar electromobility 2022. That is for location ID 27 and it is for AHE was 1.568 m, FDE was 1.713 m and AHE was 6.48 degrees in LSTM model using hyperparameters: two hidden layers with 64 and 32 neurons respectively and a dense layer with 10 neurons. In input layer number of neurons was 128.

VI. FUTURE WORK

In this seminar project, we have worked on inD dataset and used LSTM model and FCN Keras networks. It is possible to integrate two model at the same time to reduce the error for motion prediction. It is also possible to work on physics-based method by extending the constant velocity model to constant acceleration model.

VII. REFERENCES

[1] Y. Ma, X. Zhu, et al., "Trafficpredict: Trajectory prediction for heterogeneous traffic-agents," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 1, 2019, pp. 6120–6127.(Article)

[2] Lefèvre, S., Vasquez, D. & Laugier, C. A survey on motion prediction and risk assessment for intelligent vehicles. *Robomech J* **1,** 1 (2014). https://doi.org/10.1186/s40648-014-0001-z. (Article)

[3] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings and A. Mouzakitis, "Deep Learning-Based Vehicle Behavior Prediction for Autonomous Driving Applications: A Review," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 33-47, Jan. 2022, doi: 10.1109/TITS.2020.3012034. (Article)

[4] M. S. Shirazi and B. T. Morris, "Looking at intersections: A survey of intersection monitoring, behavior and safety analysis of recent studies," IEEE Trans. Intell. Transp. Syst., vol. 18, no. 1, pp. 4–24, Jan. 2017. (Article)

[5] Rudenko A, Palmieri L, Herman M, Kitani KM, Gavrila DM, Arras KO. Human motion trajectory prediction: a survey. The International Journal of Robotics Research. 2020;39(8):895-935. doi:10.1177/0278364920917446. (Article)

[6] Y. Jeong and K. Yi, "Target Vehicle Motion Prediction-Based Motion Planning Framework for Autonomous Driving in Uncontrolled Intersections," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 1, pp. 168-177, Jan. 2021, doi: 10.1109/TITS.2019.2955721. (Article)

[7] C. Du, Z. Wang, A. A. Malcolm and C. L. Ho, "Imitation Learning for Autonomous Driving Based on Convolutional and Recurrent Neural Networks," 2021 International Conference on High-Performance Big Data and Intelligent Systems (HPBD&IS), 2021, pp. 256-260, doi: 10.1109/HPBDIS53214.2021.9658480. (Article)

[8] A. Graves, Long Short-Term Memory. Berlin, Germany: Springer, 2012, pp. 37–45, doi: 10.1007/978-3-642-24797-2_4. (Article)

[9] A. Zyner, S. Worrall, J. Ward and E. Nebot, "Long short term memory for driver intent prediction," 2017 IEEE Intelligent Vehicles Symposium (IV), 2017, pp. 1484-1489, doi: 10.1109/IVS.2017.7995919. (Article)

[10] A. Graves, Supervised Sequence Labelling with Recurrent Neural Networks (Studies in Computational Intelligence). Springer, 2012.

[11] Y. Du et al., "The Vehicle's Velocity Prediction Methods Based on RNN and LSTM Neural Network," 2020 Chinese Control And Decision Conference (CCDC), 2020, pp. 99-102, doi: 10.1109/CCDC49329.2020.9164532. (Conference)

[12] . Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensorbased activity recognition," IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 6, pp. 790–808, Nov. 2012. (Article)

[13] H. Nematallah and S. Rajan, "Comparative Study of Time Series-based Human Activity Recognition using Convolutional Neural Networks," 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2020, pp. 1-6, doi: 10.1109/I2MTC43012.2020.9128582. (Conference)

[14] Y. Chen and J. Zhang, "Prediction of Liquefaction of Iron Concentrate with Artificial Neural Network Model," 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA), 2022, pp. 165-169, doi: 10.1109/CVIDLICCEA56201.2022.9825161. (Conference)

[15] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater and L. Eckstein, "The inD Dataset: A Drone Dataset of Naturalistic Road User Trajectories at German Intersections," 2020 IEEE Intelligent Vehicles Symposium (IV), 2020, pp. 1929-1934, doi: 10.1109/IV47402.2020.9304839. (Article)