國立臺灣大學電機資訊學院資訊工程學研究所

碩士論文

Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

基於深度學習之聯網自駕車切換車道異常偵測

Deep-Learning-Based Anomaly Detection for Connected and Autonomous Vehicles in Lane-Changing Scenarios

林謙

Chien Lin

指導教授：林忠緯 博士

Advisor: Chung-Wei Lin, Ph.D.

中華民國 112 年 8 月

August 2023

doi:10.6342/NTU202303898

Acknowledgements

Firstly, I am truly grateful to my advisor, Professor Chung-Wei Lin, for his invaluable guidance, insightful feedback, and beneficial instructions, which have significantly contributed to the refinement and quality of this thesis. I am also thankful to him for accepting me to be a member of CPS Lab, providing me with a supportive research environment and invaluable opportunities for learning and growth. He also provides me with the resources for my research and the opportunity to participate the meeting with foreign professors. The experience during my master's degree with him is extremely enjoyable and far exceeds my expectation.

Secondly, I greatly appreciate Professor Sandip Ray. His guidance is instrumental in refining our approach. Working with you brings me great satisfaction.

Lastly, I am thankful to the member who assists me during my research. Thanks to Sheng-Li Wang. I have gained significant inspiration from his work, and I feel honored to be able to contribute to his research.

Chien Lin

National Taiwan University
July 2023

doi:10.6342/NTU202303898

基於深度學習之聯網自駕車切換車道異常偵測

研究生：林謙　　指導教授：林忠緯 博士

國立臺灣大學資訊工程學研究所

摘要

自動駕駛車輛可以利用各種感測器或或透過通訊系統來獲取周圍環境數據，以便在自主操控中做出決策。然而，感知或接收到的數據可能因惡意攻擊而導致錯誤。這對於具有高度安全性要求的自動駕駛車輛構成了嚴重威脅。在本文中，我們提出了基於深度學習的模型，用於偵測在具有切換車道意圖時是否受到攻擊，包括長短期記憶（LSTM）模型和深度神經網絡（DNN）模型。我們提出了兩種隱蔽攻擊模型，它們可以欺騙基於規則的偵測方法。我們在城市交通模擬軟體（SUMO）中直接部署這些攻擊，以生成異常數據。我們的異常檢測流程具有通用性，可應用於不同的變道環境，我們設計了三種環境進行實驗，包括高速公路、環狀交叉路口和對向超車。結果顯示，我們所提出的基於深度學習的方法對異常具有良好的偵測性能。

關鍵詞：聯網自駕車、智慧車、異常偵測、侵入偵測、深度學習、機器學習

doi:10.6342/NTU202303898

DEEP-LEARNING-BASED ANOMALY DETECTION FOR CONNECTED AND AUTONOMOUS VEHICLES IN LANE-CHANGING SCENARIOS

Student: Chien Lin **Advisor: Dr. Chung-Wei Lin**

Department of Computer Science and Information Engineering
National Taiwan University

Abstract

Autonomous vehicles can use various sensors or wireless networks to acquire their environmental data for making decisions in autonomous maneuvers. However, the sensed or received data can be malicious due to the attacks. This poses a serious threat to autonomous vehicles which are safety-critical systems. In this thesis, we propose deep-learning-based models, which are Long Short-Term Memory (LSTM) model and Deep Neural Network (DNN) model, to detect whether a vehicle is attacked while it has the lane-changing intention. We propose two stealthy attacks as attack models, which can deceive the detection by a rule-based detection approach. Then we directly deploy the attacks into Simulation of Urban Mobility (SUMO) [1] during the simulation to generate the anomalous data. We also establish the standards and specifications for modifying simulation inputs in SUMO. Our anomaly detection workflow has the generality that can be used in different lane-changing environments, we design three environments to conduct experiments, including highway, roundabout, and opposite overtaking. As a result, the proposed

doi:10.6342/NTU202303898

deep-learning-based approach achieves a decent detection performance against the anomaly.

Keywords: Connected and Autonomous Vehicle, Intelligence Vehicle, Anomaly Detection, Intrusion Detection, Deep Learning, Machine Learning

doi:10.6342/NTU202303898

Table of Contents

doi:10.6342/NTU202303898

doi:10.6342/NTU202303898

List of Figures

doi:10.6342/NTU202303898

doi:10.6342/NTU202303898

List of Tables

doi:10.6342/NTU202303898

Chapter 1

Introduction

The development of autonomous driving systems has been rapidly developed in the past few decades. The technology which was once considered impossible is now approaching realization. Today's vehicles are equipped with Advanced Driver-Assistance System (ADAS), such as Collision Avoidance System (CAS) or Lane Keeping Assistance System (LKA). These systems assist drivers in safe and efficient driving. Autonomous vehicles make decisions about their operations based on the sensed or received surrounding information. However, the information can sometimes be incorrect due to various factors, such as poor internet connection, detection of unknown objects, or anomalous attacks. Specifically, when it comes to anomalous attacks deliberately caused by humans, the worst outcomes are expected to occur. Considering that autonomous vehicles are safety-critical systems, it is important to ensure that they can acquire information accurately.

According to studies, one of the most common reasons for accidents during driving is improper lane changing. Changing the lane at the right time and velocity has always been a challenge in driving. Attackers can intercept and modify the data transmitted from surrounding vehicles, which compromise the safety and efficiency of the process of lane-changing maneuvers. The collisions and obstructions are both possible outcomes of the attacks. In this thesis, we detect whether a vehicle is attacked while it has the lane-changing intention by deep-learning-based and

doi:10.6342/NTU202303898

machine-learning-based models.

1.1 Related Work

Over the past few decades, lane-changing detection approaches have been extensively researched. These approached can be classified into three main categories: rule-based approaches, probabilistic models, and deep-learning-based approaches.

1.1.1 Rule-Based Approaches

The rule-based approach detects the goal by setting the rules of human knowledge. These rules can be based on physics rules or other factors such as lateral position, velocity, acceleration from lane boundaries, and steering angle. Nilsson *et al.* developed a lane-changing detection approach based on the rules. They determined whether a vehicle is changing lanes by examining if its lateral velocity or acceleration exceeds a predefined threshold [2]. Additionally, factors such as the vehicle's distance from the left or right lane boundaries [3] and its steering angle [4] were considered in identifying lane-changing cases. To improve the accuracy and effectiveness of the lane-changing detection, the researchers deployed an established rule-based model called MOBIL. This model takes into account the speed and position differences of surrounding vehicles on both the current and target lanes as inputs [5]. The rule-based approach is intuitive since it has interpretable rules. However, the fixed rules may not be suitable for various real-world scenarios.

1.1.2 Probabilistic Models

The probabilistic model is a statistical technique used to predict the probability of future events occurrence based on past data. It performs well in detecting lane changing when time series trajectory data are used as inputs. Park *et al.* used

doi:10.6342/NTU202303898

a Hidden Markov Model (HMM) to detect lane changing [6]. As for the detection performance in lane-changing maneuvers, it can depend on selected features and hidden states of the probabilistic model [7]. In work [8], a combination of a Continuous Hidden Markov Model (CHMM) and a Discrete Hidden Markov Model (DHMM) was used to achieve more precise lane-changing detection. The CHMM generated the best hidden state sequence from trajectory inputs and the DHMM classify the symbolic sequence into different driving behavior. Li *et al.* used a HMM and Gaussian Mixture Model (GMM) hybrid model to detect the lane-changing behavior [9]. Deng and Söffker used a Fuzzy Logic (FL) based HMM to discriminate the driving behavior into very safe, safe, and dangerous driving scenarios, resulting in better detection performance when the HMM was trained on different scenarios [10]. In [11], a Naive Bayes model was proposed to predict whether the driver would take over the leading vehicle.

1.1.3 Deep-Learning-Based Approaches

Deep-learning-based models detect lane changings with high accuracy since they can generate detailed features through iterated training. However, a well-detect performance model requires a large amount of data for the training process. In numerous studies, Recurrent Neural Network (RNN) is the preferred detection approach [12–17]. The RNN model, especially the long short-term memory (LSTM) variant, is highly suitable for processing time series data, making it a widely adopted approach for lane-changing detection. Several studies have leveraged LSTM-based RNN models to achieve impressive results in different aspects of lane-changing detection. Wirthmuller *et al.* proposed a single-layer LSTM-based RNN model to accurately predict the time until a vehicle performs a lane change, achieving a remarkable median error of less than 0.25 seconds in their predictions [12]. Zyner *et al.*

doi:10.6342/NTU202303898

proposed a multi-layer LSTM-based RNN model to predict driver behavior before reaching road intersections [13]. Xing *et al.* proposed an ensemble bi-directional RNN model equipped with LSTM units to achieve a remarkable accuracy of 96.1% in predicting lane-changing intentions, successfully anticipating the maneuvers half a second in advance [14]. Additionally, Xin *et al.* proposed a novel Dual LSTM-based RNNs model, which encompasses two distinct LSTM units. One LSTM is dedicated to recognizing the driver's lane-changing intentions, while the other focuses on trajectory prediction [15]. In addition to the mentioned models, there are other RNN variants utilized for lane-changing prediction tasks. For instance, the Gated Recurrent Units network (GRU) was proposed to predict two specific time points: when the driver initiates the lane-changing maneuver and when the lane-changing maneuver is completed [16]. Furthermore, Park *et al.* proposed an innovative LSTM-based autoencoder model specially designed for predicting vehicle trajectories during lane-changing maneuvers [17].

Other deep-learning-based approaches are also used. Lee *et al.* proposed a CNN model to detect the lane-changing intention of surrounding vehicles with an accuracy of 95% [18]. De Candido used three 1-D Convolution-based Deep Autoencoders (DAEs) to analyze various driving maneuvers and detect the driving behavior [19]. Hu *et al.* proposed a Multi-Layer Fully-Connected Deep Neural Network (DNN) model to predict vehicle motion under various driving scenarios [20]. A Deep belief network (DBN) model was also proposed to detect the lane-changing process of a vehicle [21].

1.2 Contributions

In this thesis, our main contributions are as follows:

doi:10.6342/NTU202303898

- We propose deep-learning-based models, which are Long Short-Term Memory (LSTM) model and Deep Neural Network (DNN) model, to detect whether the vehicle is attacked while a vehicle has the lane-changing intention.

- Our anomaly detection workflow has the generality that can be used in different lane-changing environments, providing a broader range of scenarios analysis. We design three traffic environments to experiment, including a highway, roundabout, and opposite overtaking.

- We directly deploy the attacks into Simulation of Urban Mobility (SUMO) during the simulation, which means that vehicles receive the wrong value of surrounding vehicles during the driving in the simulation and generate the data, instead of adding the attack offset values to the data generated by the normal simulation. This approach allows the generated data to be a better reflection of the real-world scenarios. In addition, we also establish the specifications for the operations in SUMO.

- We use two attack models in our experiments. The first approach is Acceleration Bias Attack [22], it causes the anomalous vehicle to transmit incorrect acceleration values to other vehicles, resulting in modified velocity and position values according to the principles of physics rules. The second approach is Mistiming Trajectory Attack, which is proposed by ourselves. The anomalous vehicle transmits outdated data about itself to other vehicles. Both of the attacks are stealthy attacks that enable the attacker deceive the detection by a rule-based model.

doi:10.6342/NTU202303898

1.3 Organization

The chapters in the thesis are as follows. Chapter 2 presents the problem formulation, while Chapter 3 describes the proposed approach. Chapter 4 presents the experimental results, and Chapter 5 summarizes the conclusion and future works.

doi:10.6342/NTU202303898

Chapter 2

Problem Formulation

2.1 System Model

In autonomous driving systems, vehicles use various sensors or wireless networks to acquire their environmental data for making decisions for their next operation. The environmental data can include pedestrians, traffic signals, or information about surrounding vehicles. Particularly, the information about surrounding vehicles is a significant factor for vehicles in the intention and execution of lane-changing maneuvers. Therefore, we choose it as the key feature for our analysis. An overview of the situation is shown in Figure 2.1.

Our goal is to detect whether the vehicle (referred to as the "ego vehicle" hereafter) is attacked while it has the lane-changing intention. The attacked ego vehicle receives false data from anomalous vehicles, resulting in misjudgment during lane changing. For instance, the ego vehicle mistakenly believes that it is blocked by a vehicle on the source lane, however, the ego vehicle is suitable for lane changing, the case is shown in Figure 2.2. Next, we define the format of the training and testing data.

First, we define the received data of the ego vehicle at time step t as a *feature vector* $\mathbf{r}$ with dimension n:

$$\mathbf{r}^{(t)} = \left[\mathbf{f}_1^{(t)}, \mathbf{f}_2^{(t)}, \ldots, \mathbf{f}_n^{(t)}\right]. \tag{2.1}$$

doi:10.6342/NTU202303898

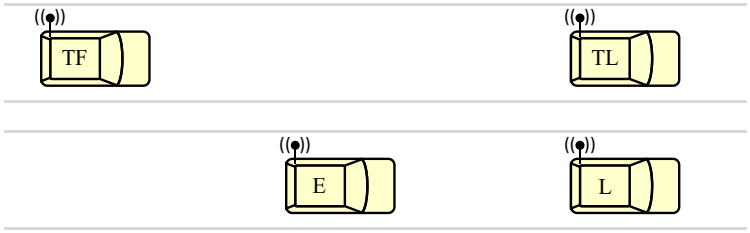


Figure 2.1: The ego vehicle (E) receives data from surrounding vehicles and determines whether to change the lane based on its state and the conditions of the surrounding vehicles, including the positions gap, velocity gap, and acceleration gap between the ego vehicle and the surrounding vehicles. The surrounding vehicles include the leading vehicle on the same lane (L), the leading vehicle on the target lane (TL), and the following vehicle on the target lane (TF).

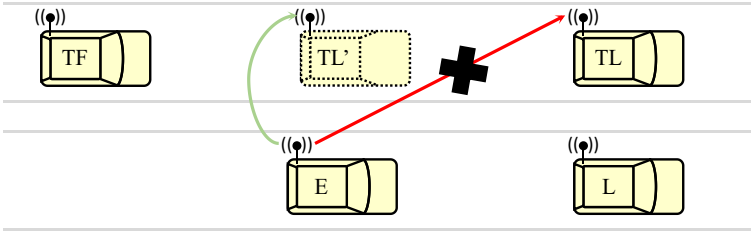


Figure 2.2: While the ego vehicle (E) is driving, it receives data from surrounding vehicles. These vehicles include the leading vehicle on the same lane (L), the leading vehicle on the target lane (TL), and the following vehicle on the target lane (TF). However, if there are attackers among these vehicles, they may transmit false data to the ego vehicle. In this figure, the ego vehicle is unable to change the lane due to receiving false data from the leading vehicle on the target lane.

The feature $\mathbf{f}$ includes the position, velocity, and acceleration of the ego vehicle itself. Additionally, it also includes the position, velocity, and acceleration gaps between the ego vehicle and three other vehicles: the leading vehicle on the source lane, the leading vehicle on the target lane, and the following vehicle on the target lane. Therefore, the dimension n of the feature vector is 12.

As our goal is to detect whether the ego vehicle is attacked while it has the lane-changing intention, we extract the feature vector right before the ego vehicle has the lane-changing intention at time step t, with the length w from time step

doi:10.6342/NTU202303898

$t-w$ to $t-1$ (where 1 time step is 0.1 second). The time series of the trajectory vector can be represented as the trajectory vector $\mathbf{R}$.

$$\mathbf{R} = \left[\mathbf{r}^{(t-w)}, \mathbf{r}^{(t-w+1)}, \ldots, \mathbf{r}^{(t-1)}\right]. \tag{2.2}$$

We use the trajectory vector as the input data for the training and testing process in our detection model. Then, we explain how to generate the attack data by using our attack model.

2.2 Attack Model

2.2.1 Attack 1: Acceleration Bias Attack

We use the approach proposed in the previous work [22] as our first attack model, with some minor modifications. The attack is a stealthy attack that can deceive the detection by a rule-based detection approach. We modify the value of the acceleration which the anomalous vehicle transmits to the ego vehicle and also modify the values of velocity and position based on physics rules.

The acceleration vector in the ego vehicle trajectory vector $\mathbf{R}$ can be represented as $\mathbf{A}$. In acceleration vector $\mathbf{A}$, $\mathbf{a}^{(t)}$ contains the acceleration of the ego vehicle and the three surrounding vehicles at time step t, where w is the trajectory length.

$$\mathbf{A} = \left[\mathbf{a}^{(0)}, \mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(w-1)}\right]. \tag{2.3}$$

We add the offset $\mathbf{o}$ to generate the attack acceleration vector $\mathbf{A}'$, noting that the value of the acceleration of the ego vehicle itself is not modified.

$$\begin{aligned}\mathbf{A}' &= \left[\mathbf{a}^{(0)} + \mathbf{o}^{(0)}, \mathbf{a}^{(1)} + \mathbf{o}^{(1)}, \ldots, \mathbf{a}^{(w-1)} + \mathbf{o}^{(w-1)}\right] \\ &= \left[\mathbf{a}'^{(0)}, \mathbf{a}'^{(1)}, \ldots, \mathbf{a}'^{(w-1)}\right],\end{aligned} \tag{2.4}$$

doi:10.6342/NTU202303898

The offset $\mathbf{o}$ can be derived from the sine function with the magnitude of $\mathbf{m}$. The offset value varies from each vehicle and each time step, since the offset value is generated by the sine function with the variables of the time step t and the random number generated by the seed of the vehicle $\mathbf{s}$.

$$\mathbf{o}^{(t)} = [0, \mathbf{O}(\mathbf{t}, \mathbf{s_L}), \mathbf{O}(\mathbf{t}, \mathbf{s_{TL}}), \mathbf{O}(\mathbf{t}, \mathbf{s_{TF}})]\,. \tag{2.5}$$

$$\mathbf{O}(\mathbf{t}, \mathbf{s}) = \begin{cases} \mathbf{m} \cdot \sin(0.02 \cdot ((\mathbf{s} + \mathbf{t})\%400)), & \text{if the vehicle is anomalous;} \\ 0, & \text{if the vehicle is not anomalous.} \end{cases} \tag{2.6}$$

When the value of acceleration changes, the value of velocity and position also changes due to the laws of physics:

$$\mathbf{v}'^{(t+1)} = \mathbf{v}^{(t)} + \mathbf{a}'^{(t)} \cdot \Delta t, \tag{2.7}$$

$$\mathbf{l}'^{(t+1)} = \mathbf{l}^{(t)} + \mathbf{v}'^{(t)} \cdot \Delta t + \frac{1}{2} \cdot \mathbf{a}'^{(t)} \cdot \Delta t^2. \tag{2.8}$$

As a result, the attacked feature vector $\mathbf{r}'$ consists of the modified features. The attacked lane-changing trajectory vector $\mathbf{R}'$ is:

$$\mathbf{R}' = \left[\mathbf{r}'^{(0)}, \mathbf{r}'^{(1)}, \ldots, \mathbf{r}'^{(w-1)}\right]. \tag{2.9}$$

2.2.2 Attack 2: Mistiming Trajectory Attack

We propose another attack model which is Mistiming Trajectory Attack to make the anomalous vehicles transmit outdated data about themselves to the ego vehicle. The anomalous vehicles transmit their position, velocity, and acceleration from Δt time steps ago $(t - \Delta t)$. The normal feature vector $\mathbf{r}$ of the ego vehicle at time t is:

$$\mathbf{r}^{(t)} = \left[\mathbf{f}_1^{(t)}, \mathbf{f}_2^{(t)}, \ldots, \mathbf{f}_{12}^{(t)}\right]. \tag{2.10}$$

On the other hand, the anomalous feature vector r' of the ego vehicle at time t is:

$$\mathbf{r}'^{(t)} = \left[\mathbf{f}_1^{(t)}, \mathbf{f}_2^{(t)}, \mathbf{f}_3^{(t)}, \mathbf{f}'^{(t)}_4, \mathbf{f}'^{(t)}_5, \ldots, \mathbf{f}'^{(t)}_{12}\right]. \tag{2.11}$$

doi:10.6342/NTU202303898

The first three features represent the position, velocity, and acceleration of the ego vehicle, and therefore it is not modified. As for the rest of the features, if the feature is transmitted by an anomalous vehicle, the feature is from Δt time steps ago.

$$\mathbf{f}'^{(t)} = \begin{cases} \mathbf{f}^{(t-\Delta t)}, & \text{if the vehicle is anomalous;} \\ \mathbf{f}^{(t)}, & \text{if the vehicle is not anomalous.} \end{cases} \tag{2.12}$$

For example, one of the surrounding vehicles is an anomalous vehicle, it transmits outdated data about itself, causing 3 of the 12 features in the feature vector of the ego vehicle to be incorrect. The attacked feature vector can be represented as $\mathbf{r}'$:

$$\mathbf{r}'^{(t)} = \left[\mathbf{f}_1^{(t)}, \mathbf{f}_2^{(t)}, \mathbf{f}_3^{(t)}, \mathbf{f}_4^{(t)}, \mathbf{f}_5^{(t)}, \mathbf{f}_6^{(t)}, \mathbf{f}_7^{(t)}, \mathbf{f}_8^{(t)}, \mathbf{f}_9^{(t)}, \mathbf{f}_{10}^{(t-\Delta t)}, \mathbf{f}_{11}^{(t-\Delta t)}, \mathbf{f}_{12}^{(t-\Delta t)}\right]. \tag{2.13}$$

Then, the attacked lane-changing trajectory vector $\mathbf{R}'$ is:

$$\mathbf{R}' = \left[\mathbf{r}'^{(0)}, \mathbf{r}'^{(1)}, \ldots, \mathbf{r}'^{(w-1)}\right]. \tag{2.14}$$

Note that this attack is also stealthy, an overview of the situation with $\Delta t = 2$ is shown in Figure 2.3.

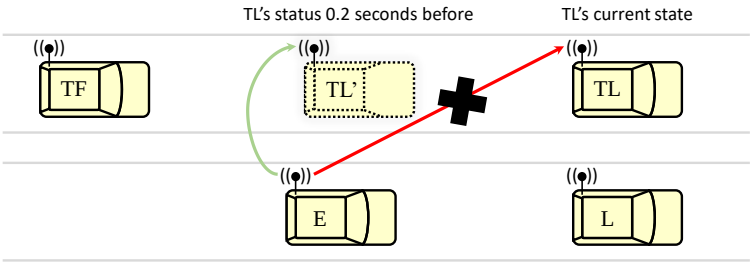


Figure 2.3: The leading vehicle in the target lane (TL) is an attacker and transmits false data to the ego vehicle (E). The false data are the outdated information of TL from 0.2 seconds (two time steps) before. As a result, the ego vehicle misjudges the position of TL and mistakenly believes that the lane-changing route is blocked.

2.3 Detection Goal

If the ego vehicle has the lane-changing intention at time step t without any anomalous vehicle around itself, we have a normal lane-changing trajectory vector

doi:10.6342/NTU202303898

$\mathbf{R}$ with the length w:

$$\mathbf{R} = \left[\mathbf{r}^{(t-w)}, \mathbf{r}^{(t-w+1)}, \ldots, \mathbf{r}^{(t-1)}\right]. \tag{2.15}$$

On the other hand, if the ego vehicle has the lane-changing intention at time step t, and there are anomalous vehicles around itself, we have an anomalous lane-changing trajectory vector $\mathbf{R}'$ with the length w:

$$\mathbf{R}' = \left[\mathbf{r}'^{(t-w)}, \mathbf{r}'^{(t-w+1)}, \ldots, \mathbf{r}'^{(t-1)}\right]. \tag{2.16}$$

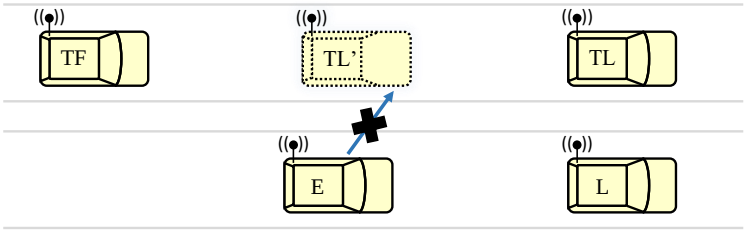


Figure 2.4: The leading vehicle in the target lane (TL) is an attacker and transmits false data to the ego vehicle (E). As a result, E misjudges the position of TL and mistakenly believes that the lane-changing route is blocked by TL.

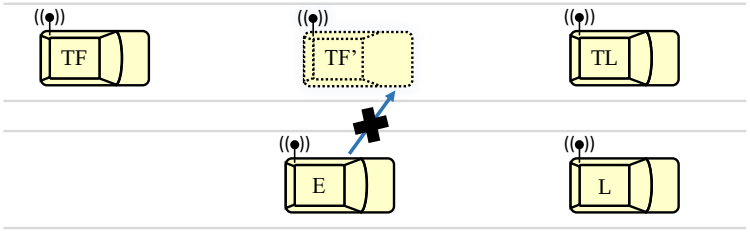


Figure 2.5: The following vehicle in the target lane (TF) is an attacker and transmits false data to the ego vehicle (E). As a result, E misjudges the position of TF and mistakenly believes that the lane-changing route is blocked by TF.

Here, we further explain the details of the lane-changing trajectory vector. We do not use every trajectory vector when the ego vehicle is attacked. Instead, We specifically select the trajectory vector that the ego vehicle is blocked by the

doi:10.6342/NTU202303898

leading or following vehicle in the target lane or the ego vehicle has a collision with the leading or following vehicle in the target lane during the lane-changing maneuver. We consider these lane-changing scenarios to have greater importance in traffic compared to typical scenarios. We select four lane-changing scenarios:

1. The leading vehicle in the target lane blocks the lane-changing route, the figure of the scenario is shown in Figure 2.4.

2. The following vehicle in the target lane blocks the lane-changing route, the figure of the scenario is shown in Figure 2.5.

3. Colliding with the leading vehicle in the target lane during a lane-changing maneuver, the figure of the scenario is shown in Figure 2.6.

4. Colliding with the following vehicle in the target lane during a lane-changing maneuver, the figure of the scenario is shown in Figure 2.7.

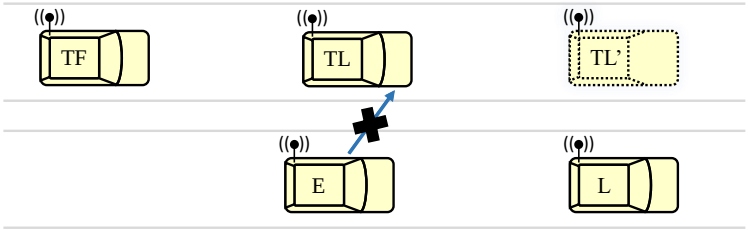


Figure 2.6: The leading vehicle in the target lane (TL) is an attacker and transmits false data to the ego vehicle (E). As a result, E misjudges the position of TL and mistakenly believes that the lane-changing route is clear, leading to a collision between E and TL.

doi:10.6342/NTU202303898

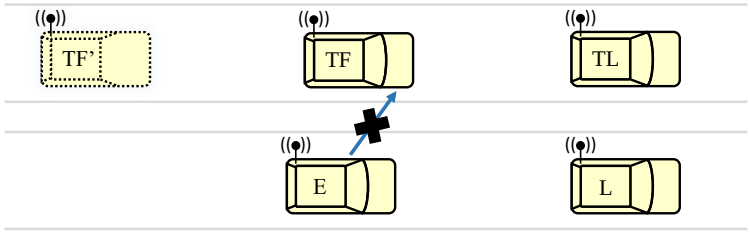


Figure 2.7: The following vehicle in the target lane (TF) is an attacker and transmits false data to the ego vehicle (E). As a result, E misjudges the position of TF and mistakenly believes that the lane-changing route is clear, leading to a collision between E and TF.

Notice that we do not select any trajectory vector which the leading vehicle is the anomalous vehicle. In SUMO simulation, the leading vehicle in the source lane has less impact on the feasibility of lane changing. For example, if the leading vehicle in the source lane is driving too slowly, it may enhance the intention of the ego vehicle to change the lane. However, the key factors that determine the feasibility and safety of lane changing are the spatial relationships between the ego vehicle and the leading or following vehicle in the target lane.

Here, we treat the anomaly detection as a binary classification problem. Based on a lane-changing trajectory vector $\mathbf{R}$ with w time steps, we label the normal lane-changing trajectory vector as normal data with label 0 and anomalous lane-changing trajectory as anomalous data with label 1.

$$\mathbf{R} = \begin{cases} 0, & \text{there is truly no anomaly in } \mathbf{R}; \\ 1, & \text{there is truly anomaly in } \mathbf{R}. \end{cases} \tag{2.17}$$

Meanwhile, based on the lane-changing trajectory vector $\mathbf{R}$, we use the anomaly detection approach to determine whether this trajectory is anomalous. If $F(\mathbf{R}) = 0$, the trajectory vector is considered anomalous, indicating an attack on the ego vehicle. On the other hand, if $F(\mathbf{R}) = 1$, the trajectory is considered normal,

doi:10.6342/NTU202303898

indicating no attacks on the ego vehicle.

$$F(\mathbf{R}) = \begin{cases} 0, & \text{the anomaly detection approach considers there is no anomaly in } \mathbf{R}; \\ 1, & \text{the anomaly detection approach considers there is anomaly in } \mathbf{R}. \end{cases} \tag{2.18}$$

We use the F_1 score to evaluate the detection performance. It is computed as follows:

$$\text{TP} = |\{\mathbf{R} = 1 \mid F(\mathbf{R}) = 1\}|, \tag{2.19}$$

$$\text{FN} = |\{\mathbf{R} = 1 \mid F(\mathbf{R}) = 0\}|, \tag{2.20}$$

$$\text{FP} = |\{\mathbf{R} = 0 \mid F(\mathbf{R}) = 1\}|, \tag{2.21}$$

$$\text{TN} = |\{\mathbf{R} = 0 \mid F(\mathbf{R}) = 0\}|, \tag{2.22}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{2.23}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{2.24}$$

$$F_\beta \text{ Score} = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precision}) + \text{Recall}}. \tag{2.25}$$

Then, we further explain our detection goal. As our detection goal is to detect whether the ego vehicle is attacked when it has the lane-changing intention, we can achieve the goal by identifying whether the obstruction or collision is due to the false information transmitted by an anomalous vehicle. However, the anomalous trajectory vector has similar positional patterns as the normal trajectory vector, it is a challenge to classify the normal and anomalous trajectory vector solely based on positional differences.

We need to know that an operation of a vehicle at each time step, namely the driving behavior, is calculated based on various parameters and complex conditions. The position, velocity, and acceleration of the surrounding vehicles play a significant role in the calculation.

doi:10.6342/NTU202303898

In the case of the Acceleration Bias Attack, the learning-based model can detect the acceleration and velocity offsets in the false information, which diverges from normal driving behavior. However, for the Mistiming Trajectory Attack, the situation is different since the false information originates from the trajectory of the anomalous vehicles several time steps ago, and the anomalous vehicle performs the normal driving behavior throughout the simulation. Therefore, the point of detecting Mistiming Trajectory Attack is to classify whether false information from the past aligns with the normal driving behavior at the current point in time. An example is shown in Fig 2.8.

In summary, Our detection objective goes beyond "detecting whether the ego vehicle is attacked when it has the lane-changing intention" and also the additional aspect of "detecting whether a vehicle performs the normal driving behavior." Our experiments address the limitation in previous work [22], which focused on detecting the rationality of lane changing. Our detection model can detect whether an individual vehicle performs normal driving behavior. By integrating these two approaches, we can achieve a comprehensive detection of all vehicle behaviors in traffic flows, resulting in an enhanced anomaly detection approach.

doi:10.6342/NTU202303898

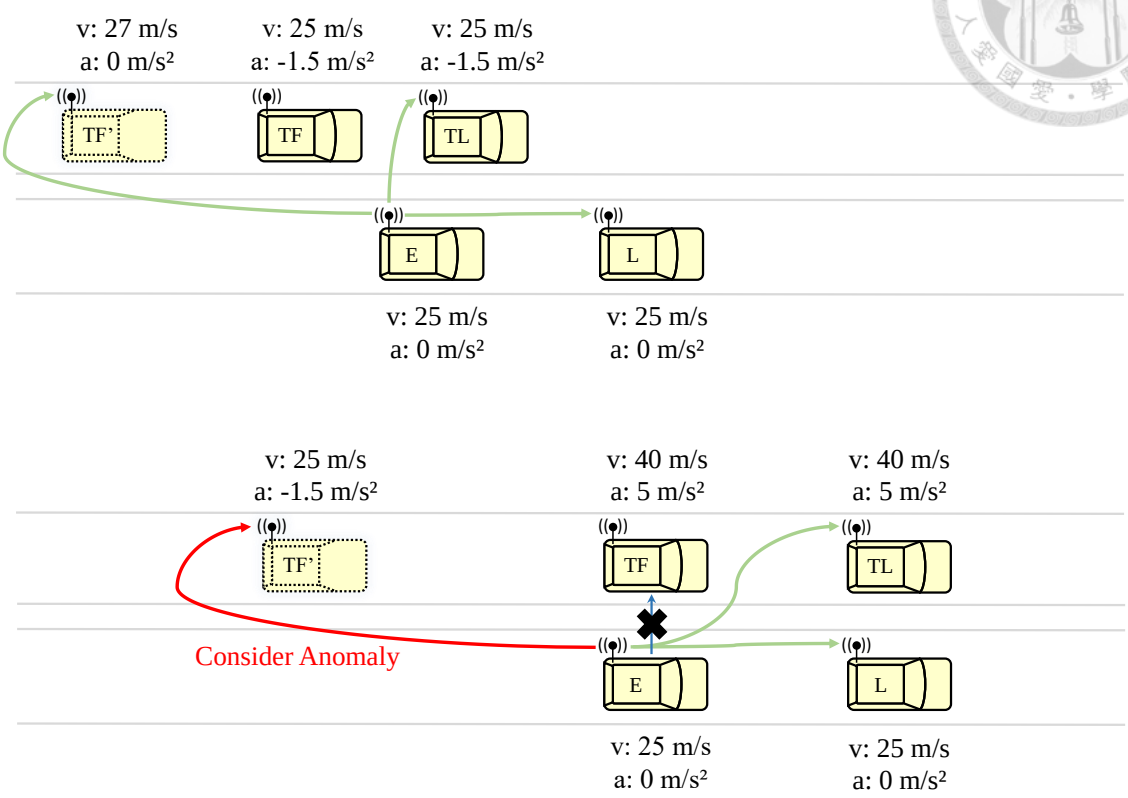


Figure 2.8: The upper part of the image represents the state of vehicles on the road at time step 0, while the lower part of the image represents the state at time step 5. In the case of a Mistiming Trajectory Attack, TF as the anomalous vehicle transmitted its past data from 5 time steps ago. When the ego vehicle has the lane-changing intention at time step 5, the ego vehicle receives the right timing data from vehicles L and TL, but it receives data from anomalous vehicle TF at time step 0. Our learning-based model detects that the data transmitted by TF is problematic since TF should accelerate forward when TL accelerates. This behavior is classified as abnormal driving behavior. Therefore, the model detects the ego vehicle is attacked.

2.4 Traffic Environments

We design three types of traffic environments for simulation in SUMO, to provide a more comprehensive scenarios analysis. In this section, we explain the reasons behind designing these three traffic environments and the characteristics of driving behaviors within them. The detailed setup of the traffic environments is in

doi:10.6342/NTU202303898

Section 4.4.

2.4.1 Highway

It is widely recognized that the establishment of highways has brought convenience to people, but it has also raised significant risks. Vehicles traveling at high velocity on highways can have life-threatening accidents with a single operational mistake. Also, there are frequent lane changing to maintain a smooth traffic flow. Therefore, we need to detect the attacks since allowing attackers to attack vehicles during lane changing on high-velocity and high-traffic highways can be extremely dangerous. The figure of a highway in SUMO simulation is shown in Figure 4.1.

2.4.2 Roundabout

Roundabouts are complex traffic environments that require vehicles to enter from different entrances and perform lane-changing at appropriate timing. In roundabouts, the frequency of lane-changing is often higher compared to regular roads. This is mainly because roundabouts are typically designed with multiple lanes, vehicles need to drive on the inner lane during general circulation and change to the outer lane when they approach their desired exit. Also, there are frequent acceleration and deceleration when the vehicle is near exits, to the position for suitable lane changing. Therefore, it is meaningful to conduct lane-changing anomaly detection in roundabouts, where frequent lane-changing and specific driving behavior occurs. The figure of a roundabout in SUMO simulation is shown in Figure 4.2.

2.4.3 Opposite Overtaking

In certain circumstances, such as traffic congestion, some drivers choose to perform opposite overtaking. This includes emergency vehicles with urgent missions

doi:10.6342/NTU202303898

or drivers with poor driving behavior. Opposite overtaking puts both the oncoming vehicles and the overtaking vehicle itself in significant danger. The overtaking driver tends to change back to the original lane due to the approaching of the oncoming vehicle, however, it is probably a struggle in finding a suitable moment to return to their original lane. If an accident occurs, the consequences can be severe since the vehicles collide in opposite directions. Therefore, it is highly possible for attackers to launch the attacks in the scenarios of opposite overtaking. The figure of the opposite overtaking case in SUMO simulation is shown in Figure 4.3.

doi:10.6342/NTU202303898

Chapter 3

Proposed Approaches

Our goal is to detect whether the ego vehicle is attacked while it has the lane-changing intention. Therefore, we propose two deep-learning-based models for the detection. The models are the Long short-term memory model (LSTM) and Deep Neural Network model (DNN). The workflow of the learning-based classifier approach is shown in Figure 3.1. Additionally, we use other detection models which are a rule-based model (RBF), and two machine-learning-based models, which are the Support Vector Machine model (SVM) and Random Forest model (RF) for the comparison to the deep-learning-based models. The detail of the comparative model is in Section 4.3.

We treat the anomaly detection as a binary classification problem. If the ego vehicle has the lane-changing intention at time step t and without any anomalous vehicle around itself simultaneously, we have a normal lane-changing trajectory vector $\mathbf{R}$, with the length w:

$$\mathbf{R} = \left[\mathbf{r}^{(t-w)}, \mathbf{r}^{(t-w+1)}, \ldots, \mathbf{r}^{(t-1)}\right]. \tag{3.1}$$

On the contrary, if the ego vehicle has the lane-changing intention at time step t, and there are any anomalous vehicles around itself simultaneously, we have an anomalous lane-changing trajectory vector $\mathbf{R}'$, with the length w:

$$\mathbf{R}' = \left[\mathbf{r}'^{(t-w)}, \mathbf{r}'^{(t-w+1)}, \ldots, \mathbf{r}'^{(t-1)}\right]. \tag{3.2}$$

doi:10.6342/NTU202303898

The two classified data have additional meaning. The normal data represent the vehicle that blocks or collides with the ego vehicle is normal, it performs the normal driving behavior. The anomalous data represent the vehicle that blocks or collides with the ego vehicle is the false information transmitted by an anomalous vehicle, it performs the abnormal driving behavior.

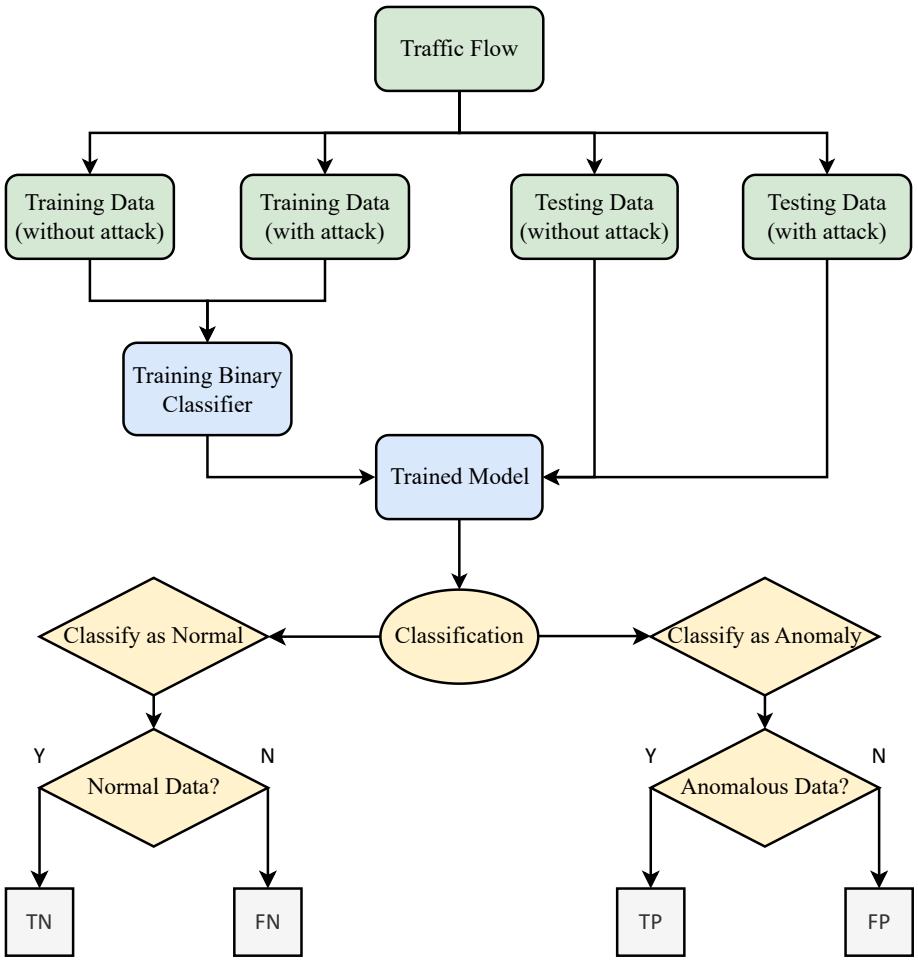


Figure 3.1: The workflow of the learning-based classifier model.

doi:10.6342/NTU202303898

3.1 Long Short-Term Memory

Long short-term memory (LSTM) is a type of Recurrent Neural Network (RNN), LSTM is well-suited for processing time series data because it can preserve information about previous states and use it in the next step. Unlike other types of neural networks, LSTM has a feedback loop design where neurons can receive inputs and states from previous time steps and pass them to the next time step. This design is ideal for processing time series data because it can integrate time-dependent information and use previous states to determine current outputs.

The architecture diagram of the LSTM model used in our experiments can be referred to Figure 3.2. We directly take the trajectory vector data as the inputs of the training and testing section without flattening. We use both normal data $\mathbf{R}$ and anomalous data $\mathbf{R}'$ as inputs. Note that the data are time series data. The time series data enter the LSTM layer, each LSTM layer contains memory units that can capture and retain long-term dependencies within the time series. The long-term dependencies include how the behavior of the ego vehicle is influenced by the surrounding vehicles. Afterward, the LSTM layer processes the time series data and generates the corresponding outputs. In the training process, the model compares the outpust of the LSTM layer with known target values and calculates the loss function. Based on the loss function, the model uses a backpropagation algorithm to adjust the parameters in the LSTM layer. The outputs of the LSTM layer further propagates to a fully-connected DNN layer. The fully-connected DNN layer maps the features to the final prediction outcome. As for the prediction process, the testing data are used as inputs of the trained LSTM, the model generates the predictions based on the outputs and internal memory states of the LSTM layer.

doi:10.6342/NTU202303898

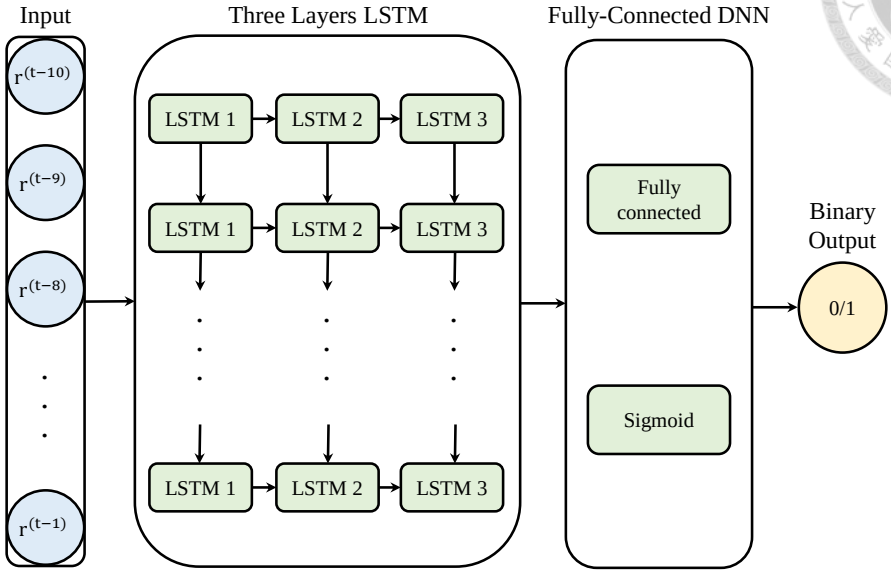


Figure 3.2: The architecture diagram of the LSTM model used in the experiments.

Table 3.1: The architecture summary of the LSTM model used in the experiments.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 5, 20)	2640
dropout (Dropout)	(None, 5, 20)	0
lstm_1 (LSTM)	(None, 5, 50)	14200
dropout_1 (Dropout)	(None, 5, 50)	0
lstm_2 (LSTM)	(None, 100)	60400
dropout_2 (Dropout)	(None, 100)	0
dense (Dense)	(None, 32)	3232
dense_1 (Dense)	(None, 2)	66

3.2 Deep Neural Network

The Deep Neural Network (DNN) is a feed-forward neural network that can also be used for time series data, although its detection performance may not be as well as LSTM in time series data. Unlike LSTM, it is not designed with feedback loops and therefore cannot preserve previous state information. However, the DNN model as a traditional deep-learning-based model is also effective in the field of

doi:10.6342/NTU202303898

anomaly detection. There are various techniques can be used in the DNN model to enhance detection performance, such as incorporating appropriate features or utilizing different loss functions.

The architecture diagram of the DNN model used in our experiments can be referred to Figure 3.3. We flatten the trajectory vector data as the inputs of the training and testing section. In the training process, the DNN layer extracts the features from the time series data. The model uses a backpropagation algorithm to adjust the parameters in the DNN layer, and dropout layers are also used to address the issue of overfitting. The features extracted by the DNN may not capture long-term dependencies, but they can still detect abnormal driving behaviors such as significant differences in position, velocity, or acceleration compared to surrounding vehicles.

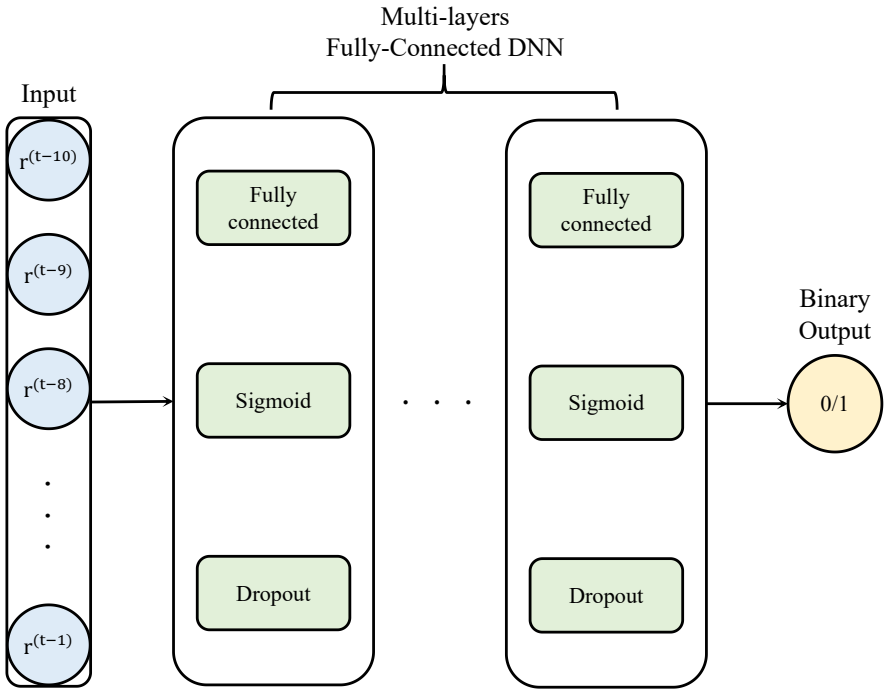


Figure 3.3: The architecture diagram of the DNN model used in the experiments.

doi:10.6342/NTU202303898

Table 3.2: The architecture summary of the DNN model used in the experiments.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 5, 20)	2640
dropout (Dropout)	(None, 5, 20)	0
lstm_1 (LSTM)	(None, 5, 50)	14200
dropout_1 (Dropout)	(None, 5, 50)	0
lstm_2 (LSTM)	(None, 100)	60400
dropout_2 (Dropout)	(None, 100)	0
dense (Dense)	(None, 32)	3232
dense_1 (Dense)	(None, 2)	66

doi:10.6342/NTU202303898

Chapter 4

Experimental Results

4.1 Experimental Setup

We conduct the traffic simulation using the Simulation of Urban Mobility (SUMO) [1]. We build three traffic environments: highway, roundabout, and opposite overtaking, to provide a broader range of analysis.

In the simulation setup, we keep the default setting for the size, speed limit, and acceleration limit of vehicles. However, we slightly enhance the lane-changing willingness of vehicles, as they tend to change the lane to drive more smoothly. The anomalous vehicles used in the attacks account for 10% of the total vehicles in the simulation. In the case of highway and roundabout, there are 5000 training data and 1000 testing data, with a 1:1 ratio between normal data and anomalous data in both the training and testing datasets. However, in the case of opposite overtaking, generating the data that meets our requirements is more challenging. Therefore, we generate 1500 training data and 300 testing data in opposite overtaking.

4.2 Attack Deployment

In our experiments, we directly deploy the attacks into SUMO during the simulation. We modify the position, velocity, and acceleration of the anomalous vehicle that is transmitted to other vehicles during the simulation. To clarify, ve-

doi:10.6342/NTU202303898

hicles in the simulation receive false data from anomalous vehicles. However, the anomalous vehicles still maintain their original states during the simulation. The situation can be referred to Figure 2.3. In summary, we directly modify the inputs during the simulation, instead of adding the attack offset values to the simulation data generated by the simulation without modified inputs.

Nevertheless, most experiments conducted in SUMO use Traci (Traffic Control Interface) to modify various settings, such as inputs, during the simulation. Although Traci allows users for adjusting the settings during simulation, there are limitations in Traci, as Traci is an API interface that does not allow direct modification of internal functions. Modifying the source code of SUMO is necessary to achieve the goal of the experiments.

Unfortunately, due to the complexity of the SUMO source code, achieving our goal by modifying the inputs during the simulation is not a simple task. We provide an approach for modifying the SUMO inputs in our experiments. We hope that the approach can benefit anyone interested in using SUMO.

Modifying SUMO inputs involves several steps:

1. Download SUMO:

   SUMO is an open-source and free software.

2. Set up road network (.rou) and network (.net) files:

   You can also find pre-existing files from the tutorials provided in SUMO for convenience.

3. Find the target file:

   In our example, the data related to vehicle positions are stored in the “microsim” folder. After that, you can access the “MSVehicle.h” file for further

doi:10.6342/NTU202303898

modifications. In most cases, vehicle-related data can be found in the "MSVehicle.h" file. If you want to adjust road-related data, you can refer to the "MSLane.h" file. Similarly, if you need to modify traffic signal-related data, you can look into the "MSTrafficlight.h" file. Similarly, the specific files within the SUMO source code correspond to different components of the simulation.

4. Find the target function:

   Most of the functions in SUMO have well-named identifiers, making it relatively easy to locate the desired functions for modification. Take our work for example, functions that require modification are "getSpeed()", "getAcceleration()", and "getPositionOnLane()". By changing the return values within these functions, you can achieve the desired modifications. Note that, if you modify the "getPosition()" function instead of "getPositionOnLane()" function, other vehicles in the simulation still receive the unmodified data during the simulation process. Therefore, it's crucial to consider the behavior of each function and its impact on the simulation.

5. Find the target function:

   The modified code must be compiled. First, navigate to "sumo/build/cmake-build" and enter the command "make -j$(nproc)" to initiate the compile process. Once the compile is successful, you can proceed with the simulation.

4.3 Comparative Detection Model

We use some detection models for comparison to the deep-learning-based models LSTM and DNN. These include a rule-based model (RBF) and two machine-learning-based models, which are the Support Vector Machine model (SVM) and Random Forest model (RF).

doi:10.6342/NTU202303898

4.3.1 Rule-Based Approach

In the rule-based approach, the physics rules and the traffic phenomena are used as the standard for the detection in our experiments. First, the position, velocity, and acceleration of vehicles need to follow the linear motion of the physics rules, any inconsistency in these parameters is unacceptable. The second rule is that the ego vehicle has no sudden brake or acceleration while it has the lane-changing intention. If such behavior occurs, the ego vehicle may have encountered abnormal driving behavior from the surrounding anomalous vehicles. If the data satisfy these two rules, we classify it as normal data; otherwise, it is considered anomalous data.

4.3.2 Support Vector Machine

Support Vector Machine (SVM) is a machine-learning-based model that can effectively deal with high dimensional data and nonlinear problems. Therefore, when we use time series data as inputs, SVM can handle more complex relationships, which can achieve better accuracy. However, the training time of SVM is relatively long, especially when the dataset is large.

4.3.3 Random Forest

Random forest (RF) is an ensemble machine-learning-based model, consisting of multiple decision trees. It has high robustness and flexibility, which can achieve better accuracy on complex time series data. However, since random forest uses multiple decision trees that are built independently, it may not capture temporal relationships in time series data.

doi:10.6342/NTU202303898

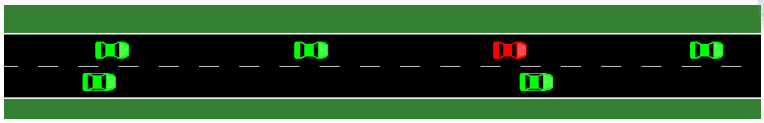


Figure 4.1: The figure of the highway in our SUMO simulation.

4.4 Traffic Environments

4.4.1 Highway

We design a two-lane highway with a length 1500 meters, where each lane has different speed limits. The speed limits are set to 35 m/s (126 km/h) for the inner lane and 30 m/s (108 km/h) for the outer lane. By setting various speed limits, we ensure that the simulation meets the real-world scenarios, and also enhance the drivers' willingness to change lanes. The figure of the highway in our SUMO simulation is shown in Figure 4.1.

4.4.2 Roundabout

We design a roundabout with a radius of 55 meters. It includes four entrances and exits, evenly spaced at every 90 degrees around the perimeter of the roundabout. The figure of the roundabout in our SUMO simulation is shown in Figure 4.2.

4.4.3 Opposite Overtaking

We design a four-lane highway with two lanes in each direction as the traffic environment of the opposite overtaking simulation. In our simulation, emergency vehicles perform opposite overtaking maneuvers to reach calls for service on time. We extract lane-changing data of emergency vehicles during their opposite overtaking maneuvers. The figure of the opposite overtaking scenario in our SUMO simulation is shown in Figure 4.3.

doi:10.6342/NTU202303898

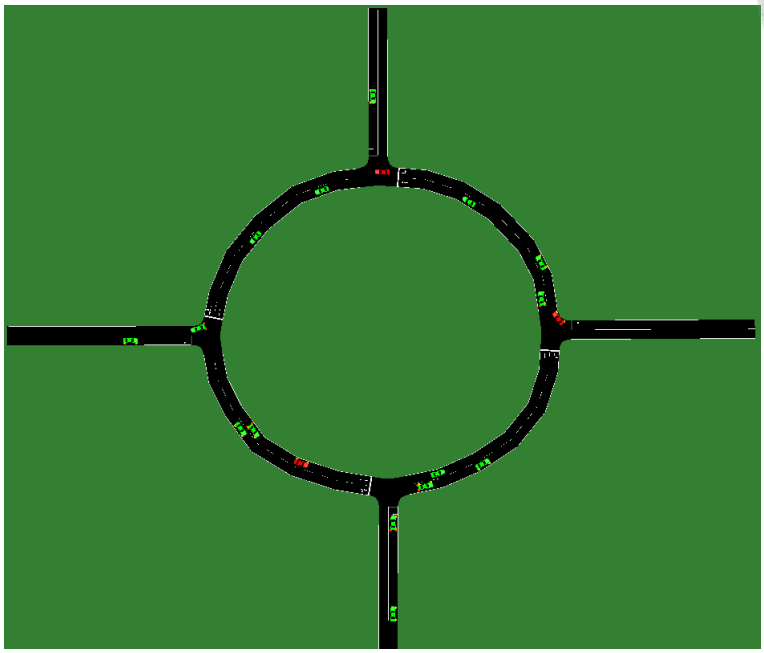


Figure 4.2: The figure of the roundabout in our SUMO simulation.

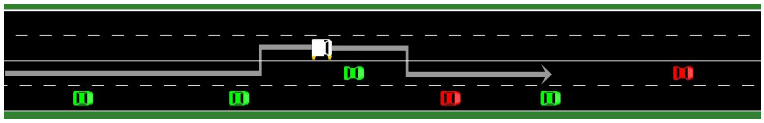


Figure 4.3: The opposite overtaking maneuver performed by an emergency vehicle is represented by a white-colored vehicle in the opposite lane. The route of the emergency vehicle for the overtaking maneuver is depicted by the gray line segment.

4.5 Experimental Results with Acceleration Bias Attack

The simulation step is 0.1 second, and the input size (w) of the data is 5 and 10, respectively. It means that the model uses 0.5-second and 1-second information to identify whether the vehicle is attacked while it has the lane-changing intention. We consider different lengths of data to analyze whether the detection model generates better predictions in the longer time series data. The magnitude $\mathbf{m}$ of the acceleration offset in Equation (2.6) is 0.8 in the highway and the opposite-overtaking cases, 0.3 in the roundabout case.

doi:10.6342/NTU202303898

Table 4.1: Experimental results (F_1 scores) with Acceleration Bias Attack on the highway.

w	LSTM	DNN	SVM	RF	RBS
5	0.92	0.86	0.74	0.79	0.03
10	0.95	0.92	0.72	0.80	0.04

Table 4.2: Experimental results (F_1 scores) with Acceleration Bias Attack in the roundabout.

w	LSTM	DNN	SVM	RF	RBS
5	0.90	0.87	0.78	0.76	0.03
10	0.94	0.92	0.77	0.77	0.05

Table 4.3: Experimental results (F_1 scores) with Acceleration Bias Attack in the opposite overtaking.

w	LSTM	DNN	SVM	RF	RBS
5	0.84	0.78	0.63	0.64	<0.01
10	0.85	0.81	0.63	0.61	<0.01

The experimental results of the highway, roundabout, and opposite overtaking are listed in Table 4.1, 4.2, and 4.3, respectively. In all of the three traffic environments, the rule-based model can not detect the anomaly. The deep-learning-based models, LSTM and DNN, outperform traditional machine-learning-based models, SVM and RF. Deep-learning-based models have more powerful learning capabilities, allowing them to automatically learn higher-level features from data. Among the two deep-learning-based models, the LSTM model performs better than the DNN model since LSTM is proficient in processing the time series data. Additionally, the detection performance improves with longer input data length due to the increased amount of information available.

The detection models perform better on the highway than in the other two cases. We believe that it is because of the low tolerance for operational errors in

doi:10.6342/NTU202303898

high-velocity environments, where a small mistake can lead to serious accidents. Therefore, the standard of normal driving behavior on the highway is much stricter, which makes the detection model easier to detect the added attack offset on acceleration and velocity.

The detection performance is slightly lower than highway due to the frequent lane-changing occurrence in the roundabout. which results in more frequent acceleration and deceleration of vehicles in both normal and anomalous data. Consequently, it becomes more challenging for the detection model to detect the added attack offset on acceleration and velocity.

As for the opposite overtaking, the detection performance is even lower. This is because the leading vehicle and the following vehicle in the target lane accelerate and decelerate frequently, providing sufficient space for the opposite overtaking vehicle to safely change back to the original lane. The scenario of the frequent acceleration and deceleration is similar to the roundabout case, which leads to weak detection performance.

4.6 Experimental Results with Mistiming Trajectory Attack

The simulation step is 0.1 second, and the input size (w) of the data is 10. This means that the model uses 1-second information to identify whether the vehicle is being attacked or not. We deploy the attacks by transmitting false data with different time step intervals (Δt). One attack transmits the false data from 10 time steps ($\Delta t = 10$) ago, while the other attack transmits the false data from 20 time steps ($\Delta t = 20$) ago. We observe whether transmitting the false data for a longer time results in larger differences in driving behavior, making it easier for the detection model to identify the attack.

doi:10.6342/NTU202303898

Table 4.4: Experimental results (F_1 scores) with Mistiming Trajectory Attack on the highway.

Δt	LSTM	DNN	SVM	RF	RBS
10	0.72	0.66	0.60	0.51	<0.01
20	0.81	0.72	0.63	0.56	<0.01

Table 4.5: Experimental results (F_1 scores) with Mistiming Trajectory Attack in the roundabout.

Δt	LSTM	DNN	SVM	RF	RBS
10	0.86	0.83	0.65	0.64	<0.01
20	0.89	0.84	0.71	0.67	<0.01

Table 4.6: Experimental results (F_1 scores) with Mistiming Trajectory Attack in opposite overtaking.

Δt	LSTM	DNN	SVM	RF	RBS
10	0.70	0.61	0.61	0.59	<0.01
20	0.75	0.65	0.64	0.60	<0.01

The experimental results of the highway roundabout, and opposite overtaking are listed in Table 4.4, 4.5 and 4.6, respectively. Similar to the experimental results of the Acceleration Bias Attack, the rule-based model can not detect the anomaly. The deep-learning-based models LSTM and DNN outperform machine-learning-based models SVM and RF in all three traffic environments. Also, the LSTM model performs better than the DNN model. Meanwhile, transmitting false data from a longer time ago results in better detection results.

However, the results show that detection performance in the Mistiming Trajectory Attack is worse than that in the Acceleration Bias Attack. In the Acceleration Bias Attack, the detection model can improve its detection performance by detecting the added offset values, which diverge from normal driving behavior.

doi:10.6342/NTU202303898

Nevertheless, in the Mistiming Trajectory Attack, the detection model can only detect the anomaly by classifying whether the false information is from the past, conforming to normal driving behavior at the current time step.

As we mentioned before, the standard of normal driving behavior on the highway is much stricter. Therefore, some cases of abnormal driving behavior can be easily detected. However, not all of the anomalous data corresponds to abnormal driving behavior. In some cases, the surrounding vehicles of the ego vehicle are far apart from each other due to the scope of the highway. The surrounding vehicles have little impact on the driving behavior of anomalous vehicles, the pattern of the anomaly driving behavior is similar to normal driving behavior. Therefore, the detection model cannot easily detect anomalous data.

In the case of the roundabout, due to the smaller scope of the traffic environment, the distances between vehicles are relatively closer compared to those on the highway. As a result, the driving behavior of anomalous vehicles is more likely to be affected by surrounding vehicles, making it more possible to be detected as abnormal driving behavior, namely anomalous data.

Lastly, in the case of the opposite overtaking, the vehicle driving behavior is more inconsistent. The inconsistency makes it difficult for the detection model to classify between normal and abnormal driving behavior.

4.7 Runtimes

The training times and testing times of machine-learning based models are crucial factors that impact various aspects of model development. The training time has a direct impact on cost-effectiveness. Training complex machine-learning models can be computationally expensive. Reducing training time can lead to significant

doi:10.6342/NTU202303898

cost savings in computation resources. Also, the scalability of a model relies on its training time. A model with a shorter training time makes the model can be trained efficiently even with massive datasets.

The training times with Acceleration Bias Attack are shown in Table 4.7 and the training times with Mistiming Trajectory Attack are shown in Table 4.8. The training times of detection models in our two attack models are reasonable, none of the training times exceed 30 minutes. It indicates that our models do not require significant computational resources. This characteristic makes our models scalable.

Table 4.7: Training times (minutes) with Acceleration Bias Attack.

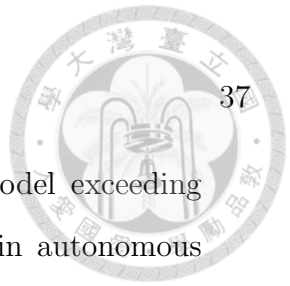
Environment / Data Length	LSTM	DNN	SVM	RF
Highway / 5	11	15	2	5
Highway / 10	17	22	3	6
Roundabout / 5	12	17	2	4
Roundabout / 10	20	25	3	6
Opposite Overtaking / 5	18	21	3	5
Opposite Overtaking / 10	27	30	3	8

Table 4.8: Training times (minutes) with Mistiming Trajectory Attack.

Environment / Time step intervals	LSTM	DNN	SVM	RF
Highway / 10	17	21	3	4
Highway / 20	19	21	3	4
Roundabout / 10	17	18	3	3
Roundabout / 20	17	20	4	3
Opposite Overtaking / 10	26	23	4	3
Opposite Overtaking / 20	28	23	4	3

The testing times of detection models are important since the autonomous vehicle system is safety-critical. The detection model have to minimize the time it takes to identify anomalies. The testing times with Acceleration Bias Attack are shown in Table 4.9 and the testing times with Mistiming Trajectory Attack

doi:10.6342/NTU202303898



are shown in Table 4.10. With no testing time of a detection model exceeding 1 millisecond, it indicates that our detection models are suitable in autonomous vehicle systems.

Table 4.9: Testing times (milliseconds per data) with Acceleration Bias Attack.

Environment / Data Length	LSTM	DNN	SVM	RF
Highway / 5	0.12	0.03	0.01	0.01
Highway / 10	0.24	0.07	0.04	0.03
Roundabout / 5	0.09	0.03	0.01	0.04
Roundabout / 10	0.15	0.07	0.03	0.06
Opposite Overtaking / 5	0.17	0.04	0.02	0.05
Opposite Overtaking / 10	0.26	0.11	0.06	0.07

Table 4.10: Testing times (milliseconds per data) with Mistiming Trajectory Attack.

Environment / Time step intervals	LSTM	DNN	SVM	RF
Highway / 10	0.18	0.08	0.05	0.04
Highway / 20	0.20	0.10	0.04	0.03
Roundabout / 10	0.24	0.07	0.03	0.06
Roundabout / 20	0.16	0.07	0.03	0.06
Opposite Overtaking / 10	0.31	0.15	0.07	0.06
Opposite Overtaking / 20	0.28	0.12	0.05	0.08

doi:10.6342/NTU202303898

Chapter 5

Conclusions

In this thesis, we detect whether a vehicle is attacked while it has the lane-changing intention. The attacked vehicle receives false information from anomalous vehicles, resulting in the lane-changing route is blocked or colliding with another vehicle. To address this issue, we propose two deep-learning-based models for anomaly detection, which are the LSTM model and the DNN model. Also, we propose two stealthy attack models. The first one is the Acceleration Bias Attack, where vehicles receive modified acceleration values of anomalous vehicles. The other is the Mistiming Trajectory Attack, where vehicles receive outdated data of anomalous vehicles. Meanwhile, we directly deploy the attacks into SUMO during the simulation, instead of adding attack offset values to the simulation-generated data. Additionally, our anomaly detection workflow has the generality that can be used in different lane-changing environments, we design three environments to experiment, including highway, roundabout, and opposite overtaking. Our proposed detection models achieve some decent *F1* scores against the anomaly. We also establish the standards and specifications for modifying simulation inputs in SUMO.

In future work, we can improve the detection performance by exploring more efficient detection approaches, such as applying other deep-learning-based models like the Convolutional Neural Networks model (CNN) or Generative Adversarial Networks model (GAN). Additionally, we can propose more powerful attack models,

doi:10.6342/NTU202303898

such as Collaborative Attacks. The more powerful attacks provide comprehensive testing and evaluation of existing detection systems. This helps to strengthen the system robustness in more complex attack scenarios.

doi:10.6342/NTU202303898

Bibliography

[1] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using SUMO. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[2] Julia Nilsson, Jonas Fredriksson, and Erik Coelingh. Rule-based highway maneuver intention recognition. In *IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2015.

[3] Jennie Lioris, Annie Bracquemond, Gildas Thiolon, and Laurent Bonic. Lane change detection algorithm on real world driving for arbitrary road infrastructures. In *Annual Computer Software and Applications Conference*, 2018.

[4] Puttipong Leakkaw and Sooksan Panichpapiboon. Real-time lane change detection through steering wheel rotation. In *IEEE Vehicular Networking Conference (VNC)*, 2018.

[5] Basma Khelfa and Antoine Tordeux. Lane-changing prediction in highway: Comparing empirically rule-based model mobil and a naıve bayes algorithm. In *IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021.

doi:10.6342/NTU202303898

[6] Seungjin Park, Wonteak Lim, and Myoungho Sunwoo. Robust lane-change recognition based on an adaptive hidden markov model using measurement uncertainty. In *International Journal of Automotive Technology*, 2019.

[7] Omveer Sharma, N.C. Sahoo, and N.B. Puhan. Recent advances in motion and behavior planning techniques for software architecture of autonomous vehicles: A state-of-the-art survey. In *Engineering Applications of Artificial Intelligence*, 2021.

[8] Omveer Sharma, N. C. Sahoo, and N. B. Puhan. Highway discretionary lane changing behavior recognition using continuous and discrete hidden markov model. In *IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021.

[9] Junde Li, Navyata Gattu, and Swaroop Ghosh. An efficient gmm-hmm fpga implementation for behavior estimation in autonomous systems. In *International Joint Conference on Neural Networks (IJCNN)*, 2020.

[10] Qi Deng and Dirk Söffker. Improved driving behaviors prediction based on fuzzy logic-hidden markov model (FL-HMM). In *IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[11] Liu Tong, Shuo Shi, and Xuemai Gu. Naive bayes classifier based driving habit prediction scheme for vanet stable clustering. In *Mobile Networks and Applications*, 2020.

[12] Wirthmüller Florian, Klimke Marvin, Schlechtriemen Julian, Hipp Jochen, and Reichert Manfred. Predicting the time until a vehicle changes the lane using lstm-based recurrent neural networks. In *IEEE Robotics and Automation Letters*, 2021.

doi:10.6342/NTU202303898

[13] Alex Zyner, Stewart Worrall, James Ward, and Eduardo Nebot. Long short term memory for driver intent prediction. In *IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[14] Y. Xing, C. Lv, H. Wang, D. Cao, and E. Velenis. An ensemble deep learning approach for driver lane change intention inference. In *Transportation Research Part C: Emerging Technologies*, 2020.

[15] L. Xin, P. Wang, C.-Y. Chan, J. Chen, S. E. Li, and B. Cheng. Intention aware long horizon trajectory prediction of surrounding vehicles using dual lstm networks. In *IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2018.

[16] Z. Yan, K. Yang, Z. Wang, B. Yang, T. Kaizuka, and K. Nakano. Time to lane change and completion prediction based on gated recurrent unit network. In *IEEE Intelligent Vehicles Symposium (IV)*, 2019.

[17] Seong Hyeon Park, ByeongDo Kim, Chang Mook Kang, Chung Choo Chung, and Jun Won Choi. Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture. In *IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[18] D. Lee, Y. P. Kwon, S. McMains, and J. K. Hedrick. Convolution neural network-based lane change intention prediction of surrounding vehicles for acc. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2017.

[19] Oliver De Candido, Maximilian Binder, and Wolfgang Utschick. An interpretable lane change detector algorithm based on deep autoencoder anomaly detection. In *IEEE Intelligent Vehicles Symposium (IV)*, 2021.

doi:10.6342/NTU202303898

[20] Y. Hu, W. Zhan, and M. Tomizuka. Probabilistic prediction of vehicle semantic intention and motion. In *IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[21] D.-F. Xie, Z.-Z. Fang, B. Jia, and Z. He. A data-driven lane-changing model based on deep learning. In *Transportation Research Part C: Emerging Technologies*, 2019.

[22] Sheng-Li Wang, Chien Lin, Srivalli Boddupalli, Chung-Wei Lin, and Sandip Ray. Deep-learning-based anomaly detection for lane-changing decisions. In *IEEE Intelligent Vehicles Symposium (IV)*, 2022.

doi:10.6342/NTU202303898