Modeling and Risk Analysis of Cooperative Adaptive Cruise Control Systems Based on Petri Nets and Distributed Edge Intelligence

Wangyang Yu, Yumeng Cheng, Xianwen Fang, Xiaojun Zhai, Yinglong Wang, Hongyuan Jing

***Abstract*—Fueled by advancements in intelligent transportation systems, the Internet of Vehicles (IoV) seeks to connect smart vehicles, road infrastructure, and users into a unified network, enhancing traffic efficiency and reducing accident risks. Centralized cloud data collection raises concerns about privacy and communication overhead. To address these, Distributed Edge Intelligence (DEI) reduces transmission costs and improves privacy by implementing machine learning at the network edge. In this context, Cooperative Adaptive Cruise Control (CACC) systems, combined with DEI in the IoV framework, enhance transportation system intelligence through real-time data processing and decentralized decision-making. This paper proposes a modeling and analysis method for CACC systems based on Petri nets. The datasets are automatically generated using tools developed by our team, and machine learning methods are utilized to perform risk prediction analysis on the CACC model. From the perspective of Petri nets synchronization, we propose risk mitigation strategies from a design standpoint. The research results show that the proposed method significantly reduces signal accumulation and enhances synchronization in CACC systems. This improvement provides new theoretical support and technical guidance for the design and implementation of CACC systems, ultimately enhancing their safety and reliability.**

***Index Terms*—Intelligent transportation systems, internet of Vehicles, distributed edge intelligence, cooperative adaptive cruise control, Petri nets**

This work was supported in part by the Open Research Fund of Anhui Province Engineering Laboratory for Big Data Analysis and Early Warning Technology of Coal Mine Safety, China, under Grant CSBD2022-ZD05, in part by the Fundamental Research Funds for the Central Universities under Grant GK202205039, in part by the Open Research Fund of Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education, China, under Grant ESSCKF2023-02, in part by the Beijing Nova Program under Grant 20230484477, and in part by the Beijing Natural Science Foundation Program under Grant L223022 and the Beijing Union University under Grant ZK20202401. This work was also supported by the China Scholarship Council.(*Corresponding author: Yumeng Cheng.*)

Wangyang Yu is with the Key Laboratory of Intelligent Computing and Service Technology for Folk Song, Ministry of Culture and Tourism, Shaanxi Normal University, No. 620, West Chang'an Street, Chang'an District, Xi'an, 710119, China (e-mail: ywy191@snnu.edu.cn).

Wangyang Yu, Yumeng Cheng and Yinglong Wang are with the School of Computer Science, Shaanxi Normal University, No. 620, West Chang'an Street, Chang'an District, Xi'an, 710119, China (e-mail: chengym@snnu.edu.cn; wyl1999@snnu.edu.cn).

Xianwen Fang is with the Anhui Province Engineering Laboratory for Big Data Analysis and Early Warning Technology of Coal Mine Safety, Anhui University of Science and Technology, An'hui, 232001, China (e-mail: xwfang@aust.edu.cn).

Xiaojun Zhai is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK (e-mail: xzhai@essex.ac.uk).

Hongyuan Jing is with the College of Robotics, Beijing Union University, Beijing, 100101, China (e-mail: jqrhongyuan@buu.edu.cn).

I. Introduction

WITH the rapid development of intelligent transportation systems, the Internet of Vehicles (IoV), as a key technology for the next generation of traffic management and optimization, is gradually transforming our travel methods [1], [2]. The IoV seamlessly integrates smart vehicles, road infrastructure, network facilities, and users, offering a comprehensive awareness of the traffic environment, thereby enhancing transportation efficiency and reducing traffic accidents [3]. However, the widespread application of IoV systems also brings significant challenges in data privacy and communication overhead. The traditional centralized cloud data processing model not only increases network load but also faces severe privacy leakage risks [4]. To address these issues, Distributed Edge Intelligence (DEI) has emerged [5]. DEI deploys intelligent computing units at the network edge close to data sources, enabling distributed machine learning and data processing between intelligent vehicles and roadside units (RSUs). This approach eliminates the need for vehicle tasks to travel back and forth to the cloud, achieving real-time decision-making in dynamic environments. It significantly reduces data transmission costs and enhances data privacy protection [6], [7].

Under the impetus of DEI, platoon control systems have become an important research direction for improving road traffic efficiency and safety [8]–[10]. Cooperative Adaptive Cruise Control (CACC), as a pivotal technology, holds a promising and essential role [11]–[13]. Through deep integration with IoV, CACC not only achieves real-time information sharing between vehicles but also enhances the operational efficiency and safety of the entire traffic system through intelligent algorithms and cooperative control [14]. Building on Adaptive Cruise Control (ACC), CACC offers substantial improvements over its predecessor, including earlier collision avoidance and reduced air drag [15]. Therefore, researching and optimizing CACC technology is of great significance for advancing the development of IoV.

The stability of CACC systems is highly dependent on inter-vehicle communication, posing numerous challenges in their design and implementation. The design of the CACC controller will determine the platoon's response to unexpected events [16]. CACC controllers can be classified based on their

communication flow topology into Predecessor-Follower (PF), Predecessor-Leader-Follower (PLF), Bidirectional (BD), and N-Predecessor-Follower [17]. All CACC topologies are proven to be string stable. However, this stability has only been tested under standard driving conditions, without considering sudden events [18]. Traditional analysis and design methods struggle to fully capture the dynamic behavior and potential risks of CACC systems, which may lead to safety issues in practical applications.

The commonly used method for reliability analysis of CACC systems is to construct dynamic models and simulate and predict system behavior through driving scenario simulations [19]–[21]. However, methods based on numerical simulation and modeling have issues such as incomplete process details, unclear workflows, and difficulty in traceability. Additionally, this method requires a lot of test data, which leads to higher testing costs. In contrast, formal modeling methods can analyze the safety of CACC systems from the perspectives of logical reasoning and mathematical analysis. These methods provide significant supplements to traditional software simulation and testing methods, allowing potential design defects and safety vulnerabilities to be identified during the design phase [22].

Petri nets, known for their intuitive graphical representation and solid mathematical foundation, are powerful modeling tools widely used in the analysis of concurrent systems [23], [24]. Consequently, Petri nets can clearly describe the state changes and event dependencies of systems, providing precise analysis methods for the dynamic behavior of complex intelligent transportation systems [25].

Using Petri nets to model CACC systems can help understand and analyze the behavior and interactions of vehicles within a platoon. By defining clear states (such as vehicle speed) and transitions (such as acceleration and deceleration), Petri nets can effectively simulate and analyze how vehicles respond to changes in the speed of the vehicle ahead while maintaining a safe following distance. Additionally, this modeling method supports the analysis of system safety, providing scientific evidence for designing more efficient and safer CACC systems. It also enhances the system's adaptability and robustness in the face of complex traffic scenarios.

This paper uses Petri nets to model the principles of CACC systems. It combines machine learning to analyze state characteristics and explores potential risk vulnerabilities from a design perspective. Petri nets analysis methods will be used to design strategies to mitigate these risks.

Specifically, the main content and innovations of this paper include:

1) Modeling the CACC system using Petri nets;

2) Autonomously generating datasets through team-developed tools and combining machine learning methods to perform risk prediction analysis on the CACC model;

3) From a design perspective, proposing corresponding risk mitigation strategies using the formal analysis methods of Petri nets.

The remainder of the paper is structured as follows: Section II reviews the related work. Section III introduces the fundamental concepts of Petri nets and explores machine learning-based reachability analysis methods for unbounded Petri nets. Section IV details the complete Petri model of the CACC system. Section V discusses the experiments and analysis, including the machine learning-based classifier and the dataset generated using the tool. We also presents the results of risk prediction for the model, proposes a reduction method for the CACC model, and offers risk avoidance strategies from the perspective of synchronization theory, alongside comparative results. Finally, Section VI concludes the article.

II. RELATED WORK

With the rapid advancement of IoV technology, edge intelligence—integrating edge computing with distributed machine learning—enables immediate data processing at nodes near vehicles. This capability offers innovative support and optimization for CACC systems. The application of these technologies not only enhances the intelligence level of traffic systems but also offers effective solutions for improving real-time data processing capabilities and accelerating decision-making response times in IoV systems.

Some scholars have proposed a Deep Reinforcement Learning (DRL) strategy for distributed machine learning in the IoV, specifically for selecting working nodes. This study models the node selection problem as a Markov Decision Process (MDP), comprehensively accounts for the timeliness of local model updates, the quality of model parameter transmission, and the sensing range of each vehicle. This method has demonstrated superior performance in high-definition mapping and intelligent driving decision-making, significantly outperforming other existing technologies [26].

Another study explores the integration of Mobile Edge Computing (MEC) with IoV, emphasizing the potential of edge computing in reducing communication delays, improving bandwidth utilization, and enhancing the security of IoV. By combining artificial intelligence technologies, this study proves that edge computing can significantly improve the overall operational efficiency of IoV, enabling faster data processing and decision-making [6].

Additionally, some scholars have proposed a Collaborative Vehicular Edge Computing (CVEC) framework that integrates Software-Defined Networking (SDN) and edge computing. This framework facilitates collaboration among different edge computing nodes, supporting scalable vehicular services and integrating Network Function Virtualization (NFV), intelligent collaborative networks, and blockchain technology. It effectively manages and optimizes vehicular network resources to accommodate diverse vehicular demands, potentially reducing latency and improving communication reliability [27].

Addressing the issue of data redundancy in IoV, another study proposed a differential compression method implemented on the MEC servers of Roadside Units (RSUs). This method effectively reduces the amount of data transmitted to the cloud, saving substantial bandwidth and storage space. By adopting the Maximum Length of Copy (MLOC) algorithm, this technique optimizes the differential compression process, helping to minimize network congestion and reduce storage waste on cloud servers [28].

In contemporary IoV-based intelligent transportation systems, CACC has gained significant attention for its potential to improve traffic efficiency, fuel economy, and vehicle safety. While scholars have made innovations and progress in the field of DEI for IoV, they have also been improving CACC control strategies. Researchers have explored various strategies to enhance inter-vehicle communication efficiency and road safety. These strategies include improving algorithms and enhancing controller capabilities to cope with complex traffic environments and constantly changing driving conditions.

Scholars have proposed two types of CACC systems: acceleration feedforward and control feedforward. They developed a general state-space model for a vehicle platoon based on a predecessor-following topology, ensuring the CACC system's stability and string stability through a series of constraints. The study highlighted that employing feedforward control strategies can significantly enhance the stability and response speed of the platoon [17].

Another study focused on a spacing control algorithm for platoons, which relies on onboard sensors and only considers communication with the preceding vehicle. By utilizing frequency domain methods and Nyquist plot analysis, the research primarily evaluated string stability. Numerical simulations confirmed the effectiveness of the method in maintaining the desired spacing within the platoon and ensuring overall system stability [29].

When the acceleration data of the preceding vehicle is unavailable, some researchers use sliding mode control with disturbance observers to address challenges in CACC systems. The strategy estimates the uncertainty of actuator dynamics and preceding vehicle acceleration, treating these factors as lumped disturbances [30].

To meet the needs of dynamic topologies, a study proposed a robust CACC controller that combines the advantages of All-Predecessor-Following (APF) and Predecessor-Leader-Following (PLF). Through simulation evaluations, this controller demonstrated excellent string stability and robustness under both normal driving conditions and unexpected events [16].

CACC systems are a key part of intelligent transportation systems. They require real-time interaction and coordination among multiple vehicles, which makes system modeling highly challenging. Existing dynamic models may lack sufficient detail, making it difficult to trace dynamic processes accurately. However, traditional formal modeling methods, such as Petri nets, have shown great potential in such applications. Petri nets can not only simulate the independent behavior of each vehicle but also precisely describe the dynamic interaction processes between vehicles.

For example, Hamroun et al. explored the use of Petri nets for modular modeling and evaluating the performance of car-sharing networks, demonstrating the potential of this approach in simulating and optimizing dynamic transportation systems [31]. In another study, Ning et al. proposed a system named the Traffic Warning Message Dissemination System (TWMDS), which utilized three different colored Petri nets to represent and analyze the interactive behaviors within the system. This further verified that the proposed reverse routing protocol had higher application value in different routing protocols [32]. In recent research by Guo et al., Petri nets were used to model and analyze logical defects in Adaptive Cruise Control (ACC) systems. They proposed an optimized modeling solution that significantly improved system reliability and effectively reduced the risk of rear-end collisions [33].

However, solely relying on a single method to achieve effective CACC system modeling and behavior analysis is challenging, as each method has its limitations. Therefore, this paper proposes an organic combination of machine learning-based net learning methods and logic-based Petri nets models. This approach can effectively leverage their respective advantages and compensate for their shortcomings. Through proper design and rigorous validation, this combination aims to enhance the accuracy and reliability of behavior risk analysis in CACC systems.

III. BASIC CONCEPTS

In this section, we present and clarify the fundamental concepts that form the foundation of our study. These concepts are essential for understanding the subsequent discussions and analyses presented in this paper.

A. Petri Nets

Petri nets are a widely used mathematical modeling language for describing and analyzing the dynamics of discrete event systems. The fundamental components of Petri nets consist of places, transitions, and directed arcs. In Petri nets, tokens can be placed in places to represent the state of the system. By triggering transitions, tokens flow between different places, thereby describing the process of state changes within the system. Petri nets have powerful modeling capabilities, capable of representing complex behaviors such as parallelism, synchronization, competition, and conflict.

Definition 1 [34]: An original Petri net is a four-tuple $\Sigma = (S, T; F, M_0)$, where:

1) S is a finite set of places.

2) T is a finite set of transitions, which represents the action of the system, where $S \cup T \neq \emptyset, S \cap T = \emptyset$.

3) $F \subseteq (S \times T) \cup (T \times S)$ is a set of directed arcs, which represents a change in the state of the system, where $\mathrm{dom}(F) = \{x \in S \cup T \mid \exists y \in S \cup T : (x, y) \in F\}, \mathrm{cod}(F) = \{x \in S \cup T \mid \exists y \in S \cup T : (y, x) \in F\}, \mathrm{dom}(F) \cup \mathrm{cod}(F) = S \cup T$.

4) M_0 is an initial marking, which represents the initial state of the Petri net.

Definition 2 [34]: Transition firing rules of Petri nets:

1) For $t \in T$, if $\forall s \in S : s \in {}^\bullet t \rightarrow M(s) \geq 1$, then $M[t >$

2) For $M[t >$, new marking M' will be obtained, where

$$M'(s) = \begin{cases} M(s) - 1, \text{if } s \in {}^\bullet t - t^\bullet \\ M(s) + 1, \text{if } s \in t^\bullet - {}^\bullet t \\ M(s), \text{ others} \end{cases}$$

Definition 3 [34]: Reachable Marking Graph.

Let $\Sigma = (S, T; F, M_0)$ is a bounded Petri net, reachable marking graph of Σ is a three-tuple $RG(\Sigma) = (R(N, M_0), E, P)$, where:

1) $R(N, M_0)$ is the set of all reachable markings from M_0, $N = (S, T; F)$, describing the structural components of a Petri net.

2) $E = \{(M_i, M_j) \mid M_i, M_j \in R(N, M_0), \exists t_k \in T : M_i[t_k > M_j\}$.

3) $P : E \to T, P(M_i, M_j) = t_k$ if and only if $M_i | t_k > M_j$.

Definition 4 [34]: Synchronic Distance.

Let $\Sigma = (S, T; F, M_0)$ is a Petri net, $t_1, t_2 \in T$, the synchronic distance between t_1 and t_2 is given by the following formula:

$$\sigma(t_1, t_2) = \begin{cases} \infty, \text{if } t_1 \text{ and } t_2 \text{ do not exhibit a fair relationship in } \Sigma \\ \max_{t_i, t_j \in T}\{\sharp(t_j/\sigma) | \exists M \in R(N, M_0) : M[\sigma > \wedge \sharp(t_i/\sigma) = 0 \wedge i, j \in \{1, 2\} (i \neq j)\}, \text{ others} \end{cases}$$

Definition 5 [34]: Boundedness and Unboundedness.

Let $\Sigma = (S, T, F, M)$ be a Petri net. For a place $s \in S$, if there exists a positive integer B such that $\forall M \in R(N, M_0) : M(s) \leq B$, then the place s is said to be bounded; otherwise, s is unbounded. The smallest such B satisfying this condition is called the bound of the place s, denoted as $B(s)$:

$$B(s) = \min\{B \mid \forall M \in R(N, M_0) : M(s) \leq B\}$$

For the net Σ, if every place $s \in S$ is bounded, then Σ is a bounded Petri net, and its bound is denoted as $B(\Sigma)$; otherwise, Σ is unbounded.

$$B(\Sigma) = \max\{B(s) \mid s \in S\}$$

B. Unbounded Petri Nets Reachability Analysis Method Based on Machine Learning

Compared to bounded Petri nets, unbounded Petri nets do not have reachable marking graphs, as an unbounded place can hold an infinite number of tokens, making reachability analysis an NP-hard problem. Some researchers have proposed a machine learning-based approach to predict the reachability of unbounded Petri nets [35]. For discrete event systems with an infinite number of states, despite the extremely large state space, it is possible to estimate the reachability of difficult-to-identify states within a finite time. This method uses known reachable markings and unlabeled markings of the unbounded Petri nets as training data for the model to predict the reachability probability of unlabeled markings. The proposed approach leverages Positive and Unlabeled Learning (PUL) combined with bagging to train a classifier that predicts the reachability probability of markings.

Definition 6 [35]: For an unbounded net system Σ and a training dataset (TD), suppose a marking M is given. A classifier is defined as a scoring function f learned from TD: $M \to [0, 1]$. The decision function $Dr(M)$ is then given by:

$$D_r(M) = \begin{cases} 1, & \text{if } f(M) \geq \varepsilon \\ 0, & \text{else} \end{cases}$$

where ε serves as the threshold for probability reachability. When $Dr(M) = 1$, it indicates that M can probabilistically reach from M_0.

Definition 7 [35]: Let $\Sigma = (N, M_0)$ as an unbounded net system. A sample is defined as a triplet (M, l_M, s_M), where:

1) $M \in \mathbb{N}^{|P|}$ represents a marking of size $|P|$.
2) l_M indicates whether M is reachable from M_0. If M is within the reachability relation $R(N, M_0)$, then $l_M = 1$; otherwise, $l_M = 0$.
3) s_M denotes whether M is labeled. If M is labeled, then $s_M = 1$; otherwise, $s_M = 0$.

Consequently, positive and negative samples are denoted as $(M, l_M = 1, s_M = 1)$ and $(M, l_M = 0, s_M = 1)$, respectively. The unlabeled sample is denoted as $(M, l_M = 0, s_M = 0)$ or $(M, l_M = 1, s_M = 0)$.

IV. Modeling Scheme

Before modeling the CACC system, it is essential to abstract the core elements of the process. While existing CACC controllers each have their strengths and weaknesses, the PLF controller stands out as one of the most stable and advanced options [16]. Fig. 1 illustrates the overall schematic of the CACC system employing the PLF controller [36]. Given that the primary objective of this study is to evaluate and analyze the safety impacts of fundamental CACC systems, the modeling focuses exclusively on the cooperative technology's core functionality. In this system, following vehicles actively adjust their acceleration and deceleration in response to the speed-up and slow-down signals from both the lead vehicle and the directly preceding vehicle.

TABLE I
Meaning of Elements in Figure 2

Element	Type	Meaning
p_0, p_3, p_6, p_9	Place	Low speed state
p_1, p_4, p_7, p_{10}	Place	Medium speed state
p_2, p_5, p_8, p_{11}	Place	High speed state
$p_{12}, p_{14}, p_{16}, p_{18}, p_{20}$	Place	Deceleration signal place
$p_{13}, p_{15}, p_{17}, p_{19}, p_{21}$	Place	Acceleration signal place
$t_0, t_2, t_4, t_6, t_8, t_{10}, t_{12}, t_{14}$	Transition	Deceleration
$t_1, t_3, t_5, t_7, t_9, t_{11}, t_{13}, t_{15}$	Transition	Acceleration

The core of this process lies in the synchronization between the following and leading vehicles [33]. When the leading vehicle's speed changes, the following vehicle receives acceleration or deceleration signals from both the leading vehicle and the directly preceding vehicle. To maintain a consistent distance, the following vehicle must adjust its speed in alignment with the leading vehicle. The degree of synchronization in this process directly influences the risk of rear-end collisions. High synchronization ensures that the speed of both vehicles changes within a controlled range. In contrast, poor synchronization leads to significant variations in the distance between the two vehicles, greatly increasing the likelihood of a rear-end collision. This critical aspect must be accurately represented in the model. The properties and structure of Petri nets are well-suited to capturing the concurrency and synchronization relationships inherent in this system, making them an ideal choice for modeling this process. The related modeling rules are as follows:

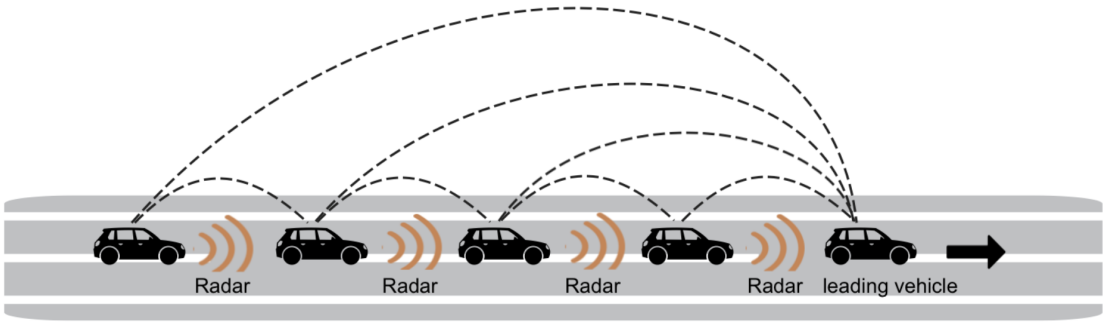


Fig. 1. Schematic diagram of CACC

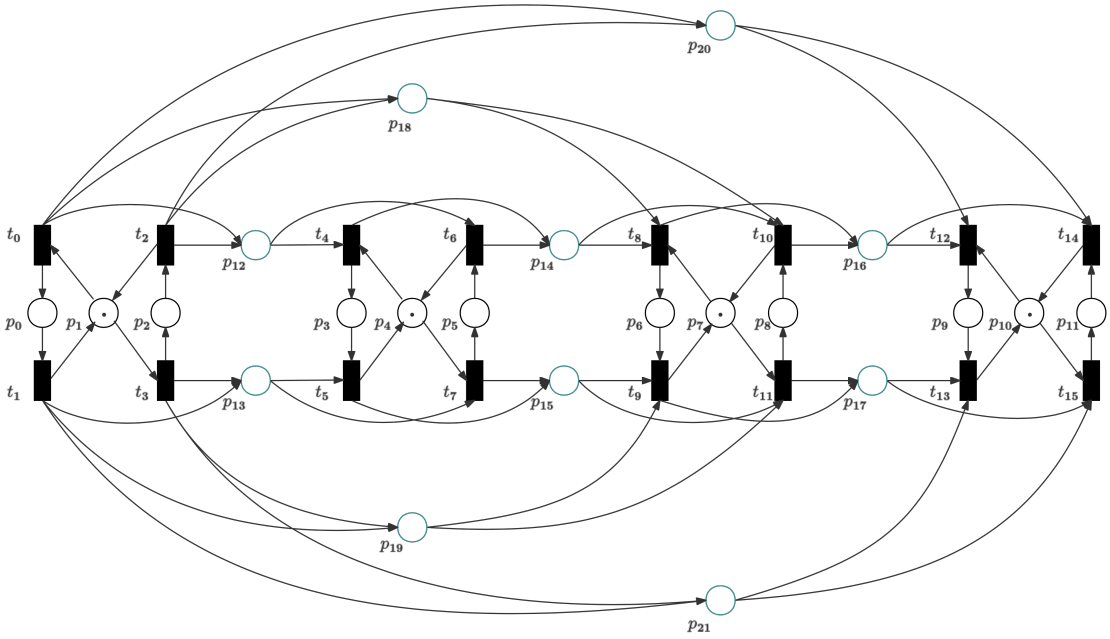


Fig. 2. Petri net model of CACC

1) Places in the Petri net represent the speed levels of the vehicles and the accumulation of signals between the two vehicles.

2) Transitions in the Petri net depict the changes in the vehicles' states.

3) Arcs in the Petri net indicate the transmission of synchronization information.

Following the above modeling rules, the Petri net model of the CACC system is illustrated in Fig. 2, with the primary elements and their corresponding meanings detailed in Table I. In the longitudinal domain, the vehicle's primary actions are acceleration and deceleration. Consequently, each vehicle's state is categorized into three conditions: low speed, medium speed, and high speed. When a vehicle transitions from low speed to high speed, it must pass through the buffering state. Likewise, transitioning from high speed to low speed also requires passing through the medium state.

In the lateral domain, each set of four transitions and three places forms the basic model for a vehicle. The current vehicle is connected to both the directly preceding vehicle and the leading vehicle via signal places. For example, transitions t_8 and t_{10} represent acceleration, while transitions t_9 and t_{11} represent deceleration. The places p_6, p_7, and p_8 represent low speed, moderate speed, and high speed, respectively, constituting the basic states and actions of the third vehicle. Vehicle 3 is connected to the directly preceding vehicle (Vehicle 2) through places p_{14} and p_{15} and to the leading vehicle (Vehicle 1) through places p_{18} and p_{19}, receiving acceleration or deceleration signals from them.

At this point, Vehicle 1 is in a moderate speed state, and transitions t_0 and t_3 can both fire, indicating that the leading vehicle can either accelerate to high speed or decelerate to low speed. Assuming that transition t_3 fires and the leading vehicle completes acceleration, the signal places

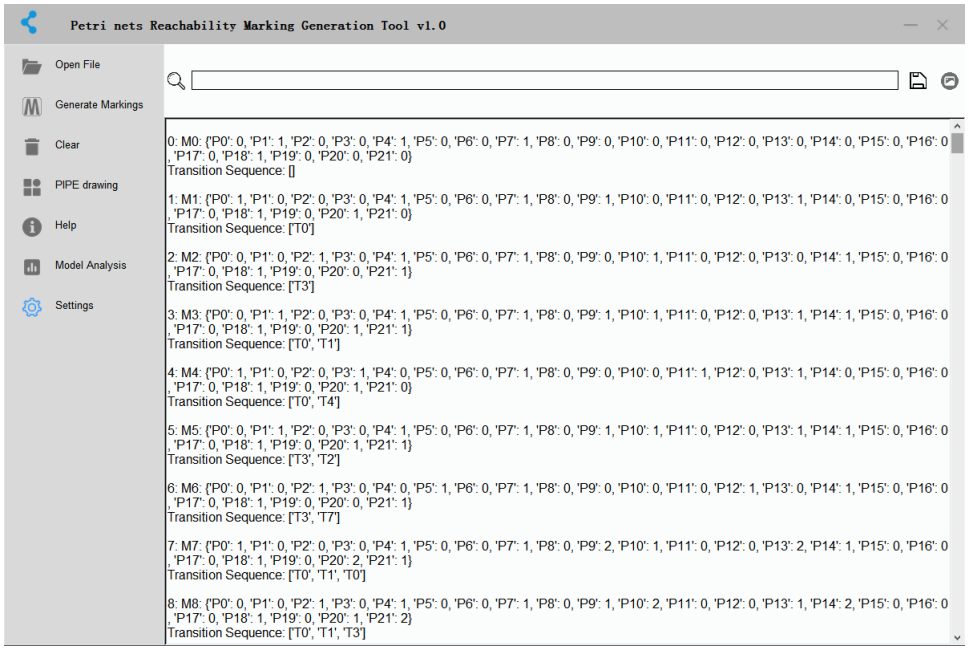


Fig. 3. Petri nets Reachable Marking Generation Tool v1.0

p_{13} and p_{19} receive tokens. The token in the signal place p_{13} enables transition t_7, indicating that the current Vehicle 2 receives acceleration information from the leading vehicle. Vehicle 2 needs to accelerate in the same manner as the leading vehicle to maintain a constant distance between the two vehicles, thereby avoiding rear-end collisions caused by excessive distance changes. Therefore, transitions t_3 and t_7 are in a synchronized relationship. Similarly, other transitions also follow such sequential synchronization relationships.

V. Experiments and Analysis

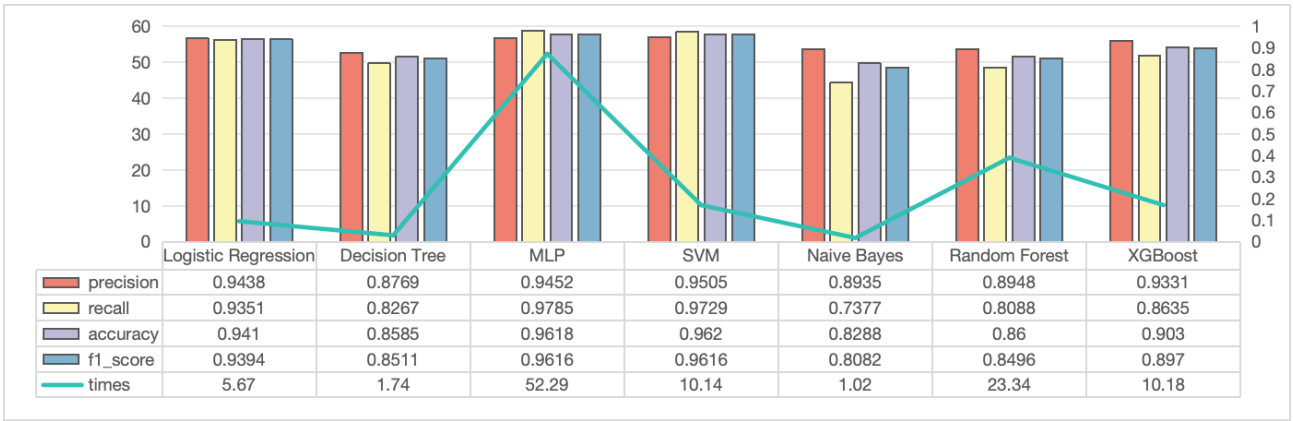
A. Machine Learning Methods and Dataset Generation Tools

In [35], the model's prediction ability is positively correlated with the amount of data, meaning that the larger the dataset, the stronger the prediction ability. However, for the complex models we are studying, generating and recording all reachable markings is very challenging.

Therefore, our team developed a custom software for generating reachable states, named Petri nets Reachable Marking Generation Tool v1.0. As shown in Fig. 3, its interactive interface allows users to create Petri nets models and automatically generates a large amount of reachability data for training. According to the method described in [35], we successfully generated 10,000 reachable states and 10,000 unknown label markings using this tool for training purposes.

This tool is primarily used for automatically generating reachable markings in Petri net models. The core function of the tool is to simulate the state transitions of a Petri net to generate possible reachable markings. The process is as follows:

1) Input: The user constructs a Petri net model using Pipe



	Logistic Regression	Decision Tree	MLP	SVM	Naive Bayes	Random Forest	XGBoost
precision	0.9438	0.8769	0.9452	0.9505	0.8935	0.8948	0.9331
recall	0.9351	0.8267	0.9785	0.9729	0.7377	0.8088	0.8635
accuracy	0.941	0.8585	0.9618	0.962	0.8288	0.86	0.903
f1_score	0.9394	0.8511	0.9616	0.9616	0.8082	0.8496	0.897
times	5.67	1.74	52.29	10.14	1.02	23.34	10.18

Fig. 4. Comparison of different machine learning model metrics

TABLE II
PART OF THE DATASET

p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}	p_{18}	p_{19}	p_{20}	p_{21}	label
1	0	0	1	0	0	0	0	1	0	1	0	11	11	2	0	1	2	13	11	14	13	1
1	0	0	0	0	1	0	0	1	0	1	0	12	10	1	1	1	2	13	11	14	13	1
1	0	0	0	1	0	0	1	0	0	1	0	11	10	2	2	1	1	13	12	14	13	1
1	0	0	1	0	0	1	0	0	0	1	0	11	11	1	1	2	1	12	12	14	13	1
1	0	0	0	0	1	1	0	0	0	1	0	12	10	0	2	2	1	12	12	14	13	1
1	1	0	0	1	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0
0	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	1	0
1	1	0	0	1	0	0	1	0	0	1	0	1	1	0	0	0	0	1	1	1	1	0
1	0	0	1	1	0	0	1	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0
1	0	0	1	0	0	0	1	0	0	1	0	1	0	1	0	0	0	1	0	1	0	0

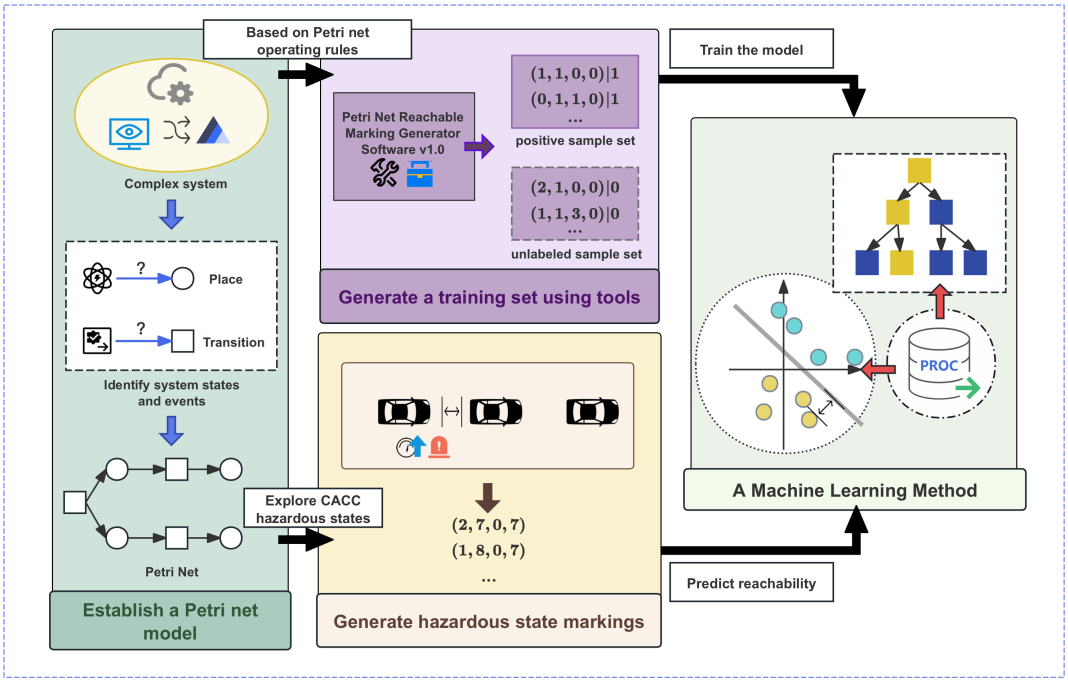


Fig. 5. Flowchart of the machine learning-based risk analysis method

or other Petri net simulation tools, and then imports the corresponding XML file into the Petri Nets Reachable Marking Generation Tool v1.0. The tool automatically reads the Petri net structure and its initial markings, including transitions, places, directed arcs, and the initial distribution of tokens.

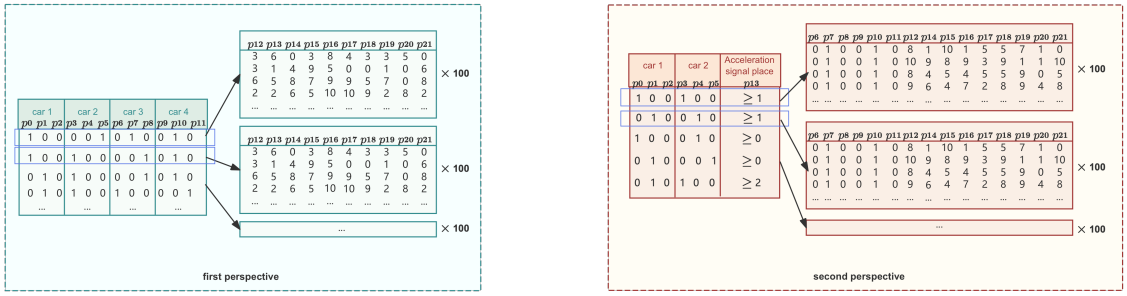
2) Reachable Markings Generation: Based on the input Petri net, the tool executes a state traversal algorithm. It simulates the triggering of different transitions to explore the possible marking combinations.

3) Unknown Label Markings Generation: For generating unknown label markings, the tool first runs to generate reachable markings. It then automatically adds or removes tokens from any place, creating a new unknown reachable marking. Most of these markings are unreachable, and together they form a set of unknown label markings.

4) Output: Finally, the tool outputs a specified number of data sets and provides the markings for subsequent analysis.

This significantly alleviates the challenge of manually simulating the generation of training data. These unknown labels include both reachable and unreachable states, providing a rich data foundation for model training and validation. As shown in Table II, we present the data from rows 9996 to 10005, which are at the junction between reachable and unknown labels. A label of 1 indicates reachable, while 0 indicates unknown reachability.

We employed a range of machine learning models to train, test, and validate the generated data, including Logistic Regression, Decision Tree, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), and Random Forest. Throughout the experiment, we rigorously evaluated the prediction accuracy of each model. After conducting a comprehensive comparative analysis of the extensive experimental results, it became evident, as shown in Fig. 4, that the SVM outperformed the other models in terms of accuracy and exhibited a shorter training duration when handling our data. Therefore, we selected SVM as the base classifier for the bagging ensemble method.



(a) (b)

Fig. 6. CACC dangerous situation dataset. (a) First perspective (b) Second perspective

B. Risk Analysis

Fig. 5 illustrates the overall process of risk analysis for the CACC Petri net model using the classifier from subsection V-A. The entire process is divided into the following parts:

1) Establishing the Petri Net: First, identify the main states and events within a complex system and map them to Places and Transitions in the Petri net. Based on the identified system states and events, establish the corresponding Petri net model.

2) Generating the Training Set: Use the Petri nets Reachable Marking Generation Tool v1.0 to generate the training set. This software generates reachable markings according to the operational rules of the Petri net and categorizes the generated markings into a positive sample set and an unlabeled sample set.

3) Generating Dangerous State Markings: In the Petri net, explore the dangerous states of the CACC system and generate the corresponding marking dataset.

4) Machine Learning Training and Prediction: Leverage the generated training set to train the model using machine learning techniques. Subsequently, the trained model is applied to predict the reachability of dangerous states within the CACC system.

The Petri net has already been established in Section IV, and the training set has been generated using the Petri nets Reachable Marking Generation Tool v1.0. Next, we need to explore the characteristics of dangerous state markings specific to the CACC model to generate these dangerous state markings, which will ultimately be evaluated by the trained classifier.

The dangerous states of CACC can be explored from two perspectives:

1) Speed Difference Perspective: A dangerous state is identified when the leading vehicle is in a low-speed state while the following vehicle is in a high-speed state. This scenario presents a significant risk of a rear-end collision, making it a typical example of a dangerous situation.

2) Signal Place Perspective: Focus on the accumulation within the signal places. A dangerous state is identified if the accumulation of tokens in the acceleration signal place between the two vehicles is sufficient to cause the following vehicle to accelerate to a speed exceeding that of the leading vehicle. This indicates that the following vehicle may over-accelerate and rear-end the leading vehicle.

After identifying the dangerous states of the CACC system from these two perspectives, the generated dangerous state markings will be evaluated by the trained classifier to predict the dangerous states of the CACC system during actual operation.

The dangerous state markings are generated by specifying the number of tokens in particular places according to the specific conditions of the dangerous states in the CACC Petri net. The number of tokens in other places is randomly generated within a small range. This approach ensures that the token composition of all dangerous state markings is not fixed, thereby maintaining the generality of the description of dangerous states.

Specifically, each dangerous state generates 100 marking data, the final total of 1,500 test data entries were generated. If any one of these markings is determined to be "reachable," the model is considered to have a dangerous state. Using this method, we can more accurately describe and evaluate potential dangerous states in the CACC system.

The generated data are presented in Fig 6. Fig 6a shows the data situation from the first perspective. The speed states of each vehicle are represented by three places, listing several different markings of the leading vehicle at low speed and the following vehicle at high speed. For each "low-speed leading, high-speed following" state, 100 additional markings for other places are automatically generated. Fig. 6b shows the data situation from the second perspective. Since this situation is noteworthy for any two adjacent vehicles, we use the first and second vehicles as examples. The low-speed state is designated as -1, the medium-speed state as 0, and the high-speed state as 1. Besides setting the speed state markings for the two vehicles, the acceleration signal place between them is also specified. The specific settings are as follows:

a) When the speed of the following vehicle minus the speed of the leading vehicle is 0, the tokens in the acceleration signal place must be at least 1;

b) When the speed of the following vehicle minus the speed of the leading vehicle is 1, the tokens in the acceleration signal place must be at least 0;

c) When the speed of the following vehicle minus the speed of the leading vehicle is -1, the tokens in the acceleration signal place must be at least 2.

These configurations ensure that the token count in the acceleration signal place is adequate, enabling the following

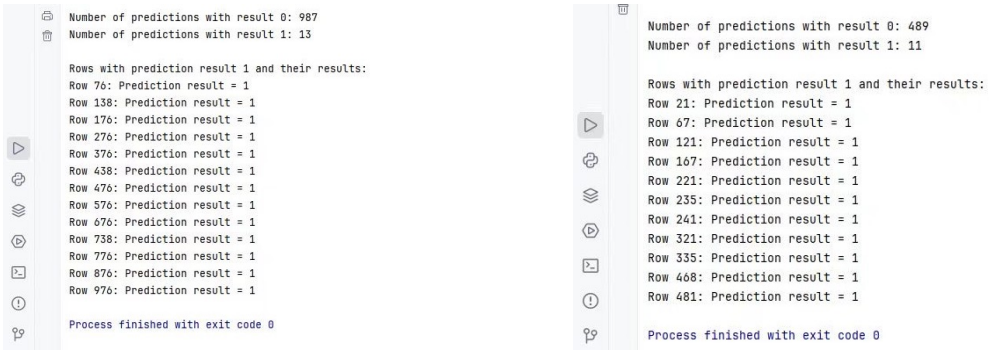


Fig. 7. Reachability test results for dangerous situations. (a) First perspective result (b) Second perspective result

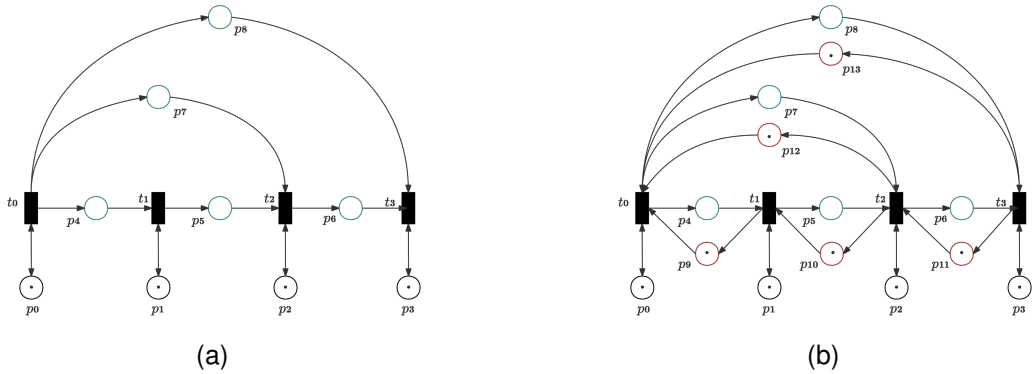


Fig. 8. (a) Reduced CACC model (b) Improved CACC model

Algorithm 1: Reduction Algorithm for the CACC Model

Input: $CP = (S, T; F, M_0)$
Output: $CP' = (S', T'; F', M_0')$
1. Copy $CP' = CP$;
2. Remove all connecting arcs;
3. Merge speed state places belonging to the same vehicle;
4. Merge each pair of signal places between two vehicles;
5. Merge all transitions belonging to the same vehicle;
6. **for** *any transition in* CP' **do**
   Direct single-headed arcs to the signal places between the vehicle it belongs to and the following vehicle;
7. **for** *any place in* CP' **do**
   if *it is a signal place* **then**
      Direct single-headed arcs to the transitions of the following vehicle between the two vehicles it belongs to;
   else
      Use double-headed arcs to connect to the transitions of the vehicle it belongs to;
8. Insert tokens into each vehicle as the initial marking;

vehicle to accelerate beyond the speed of the leading vehicle.

Subsequently, similar to the first perspective, each dangerous state marking is supplemented with 100 additional automatically generated markings for other places, thus forming a complete dataset.

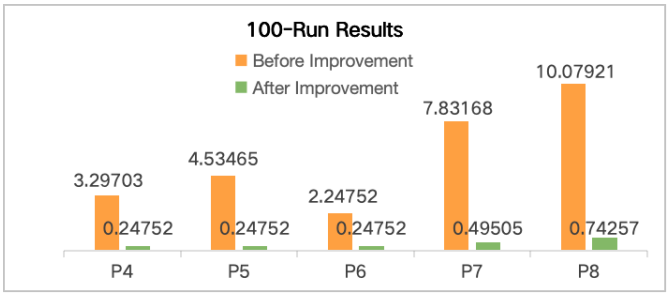
This method not only ensures data diversity but also better simulates the complex situations encountered in real scenarios. By generating a large volume of data in this manner, we provide ample validation resources for our model, greatly enhancing its generalization capability.

Figs. 7a and 7b show our experimental results. It can be observed that dangerous states from both perspectives have markings classified as reachable by the classifier. This indicates the presence of dangerous situations, highlighting the need for careful design to avoid these risks.

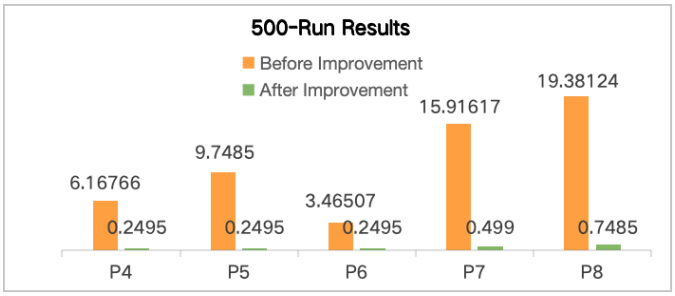
C. Model Reduction

As shown in Fig 8a, the model reduced using Algorithm 1 is simpler and more abstract compared to the original model, enabling us to analyze its synchronous cooperation characteristics more clearly.

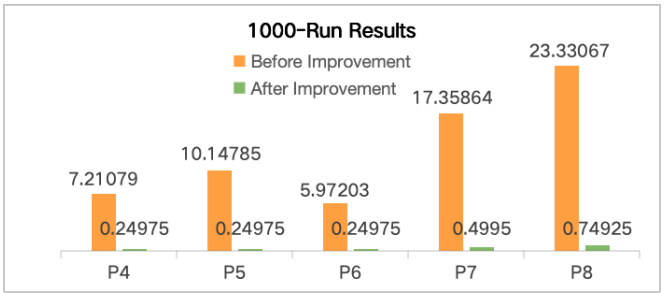
Following this, we proceed to analyze the time complexity of Algorithm 1. Initially, in Step 1, copying $CP' = CP$ involves an operation with a time complexity of $O(1)$. In Step 2, the removal of all connecting arcs has a time complexity of $O(m)$, where m denotes the number of arcs within the



(a)



(b)



(c)

Fig. 9. Comparison of token accumulation before and after improvement of the CACC Petri net model

network. Subsequently, in Step 3, each vehicle's three speed state places are merged. Given n vehicles, this step has a time complexity of $O(3n)$.

In Step 4, the two signal places between each pair of vehicles are merged into one. Since there are n vehicles in total, the time complexity for this step is $O(2n)$. Step 5 entails merging four transitions for each vehicle into a single transition. This operation is carried out across all n vehicles, resulting in a time complexity of $O(4n)$.

Step 6 focuses on operations concerning each transition in CP'. After merging, there is only one transition per vehicle, leading to n transitions overall and a time complexity of $O(n)$ for this step. In Step 7, operations are conducted on each place in CP'. Considering the reduced number of places, this step also carries a time complexity of $O(n)$. Lastly, Step 8 involves the insertion of tokens, a constant-time operation with a complexity of $O(1)$.

In conclusion, the overall time complexity of the Algorithm 1 is governed by the aforementioned steps, yielding a total complexity of $O(n+m)$, where n represents the number of vehicles, and m corresponds to the number of arcs in the network.

The original model is an unbounded Petri net, and after reduction, it retains the unbounded property. We know that if two transitions t_1 and t_2 are not in a fair relationship, i.e., for any positive integer k, if there exists a reachable marking M and a transition sequence σ such that $M[\sigma>$, and $\sharp(t_i/\sigma) = 0 \wedge \sharp(t_j/\sigma) \geq k$ $(i,j \in \{1,2\}$ and $i \neq j)$, then we consider the synchronization distance between them to be infinite [34]. Then we consider their synchronization distance to be infinite.

TABLE III
MEANING OF ELEMENTS IN FIGURE 8

Element	Type	Meaning
p_0, p_1, p_2, p_3	Place	State place
p_4, p_5, p_6, p_7, p_8	Place	Signal place
t_0, t_1, t_2, t_3	Transition	Information Processing
$p_9, p_{10}, p_{11}, p_{12}, p_{13}$	Place	Control place

For the reduced model, the specific meanings of the elements are referenced in Table III. The state places are retained solely for symbolic purposes. All other acceleration and deceleration transitions are simplified into a single information processing transition, representing the vehicle's response to signals. It is not difficult to observe that any two transitions are not in a fair relationship. This is the reason why the CACC model has dangerous states. The synchronization distance between the transitions of the two vehicles is infinite, indicating poor synchronization. Such poor synchronization may cause the following vehicle to fail in processing the leading vehicle's signals promptly, which could lead to a rear-end collision, especially when the leading vehicle is moving slowly and the following vehicle is at a higher speed.

From a design perspective, we consider adding control places to the CACC model to enhance the synchronization between transitions, as shown in Fig. 8b. According to the calculations, it can be determined that the synchronization distance between any two transitions is 1, thereby establishing a strong synchronization relationship [37].

We let the model run automatically for 100, 500, and 1000 times before and after the improvement, and the results show the average token accumulation in the signal places, as illustrated in Fig. 9. Before the improvement, there was significant signal accumulation, indicating that the vehicles in the CACC system were not processing signals in a timely manner, resulting in poor synchronization. This poor synchronization is a primary reason for potential accidents.

After improving the model, the token accumulation in each place became more consistent and significantly reduced, indicating that the synchronization of the model has been enhanced. This greatly reduces the likelihood of accidents, demonstrating the significant improvements in synchronization and safety after adding control places to the model.

In order to provide a clearer comparison of the different performance indicators before and after the synchronization improvement, we have added Table IV, which presents the key indicators before and after the improvement. These indicators include: fairness between any two transitions, where the model was "Unfair" before the improvement and "Fair" after; synchronization distance between any two transitions,

TABLE IV
COMPARISON OF INDICATORS BEFORE AND AFTER IMPROVEMENT

Indicator	Before Improvement	After Improvement
Fairness between any two transitions	Unfair	Fair
Synchronization distance between any two transitions	∞	1
Average token accumulation in signal places (after 100 runs)	5.598	0.396
Average token accumulation in signal places (after 500 runs)	10.936	0.399
Average token accumulation in signal places (after 1000 runs)	12.804	0.400

where the synchronization distance was "∞" before the improvement and "1" after; average token accumulation in signal places, with a comparison of the token accumulation after different numbers of runs. After the improvement, the token accumulation decreased significantly, indicating that the improved synchronization mechanism significantly enhanced the system's stability and response speed.

The core goal of the synchronization improvement is to reduce the potential risks caused by signal processing delays. In the original model, the synchronization between vehicles was poor, and there were significant delays in signal transmission, which caused the following vehicles to respond late to acceleration or deceleration from the leading vehicle, leading to unstable inter-vehicle distances. When the leading vehicle decelerated, the following vehicles might have over-accelerated, increasing the risk of rear-end collisions. The improved synchronization mechanism significantly reduced the signal transmission delay, enabling the following vehicles to respond more promptly and smoothly to the speed changes of the leading vehicle, thus maintaining a more stable inter-vehicle distance. Specifically, the improved synchronization mechanism ensures:

1) Real-time vehicle response: It reduces the response delay of the following vehicle when the leading vehicle accelerates or decelerates, preventing excessive changes in the inter-vehicle distance.

2) Stability of inter-vehicle distance: After the synchronization improvement, the distance between vehicles remains stable, reducing the risk of collisions caused by fluctuations in the inter-vehicle distance.

3) System smoothness: The synchronization improvement reduces instances of over-acceleration and over-deceleration, making the overall system run more smoothly and effectively preventing dangerous situations.

VI. CONCLUSION

To address signal processing delays and synchronization issues in intelligent transportation systems, we propose a synchronization improvement mechanism for the Cooperative Adaptive Cruise Control (CACC) system based on Petri nets. Our study employs a machine learning method suitable for the reachability analysis of unbounded Petri nets to perform a risk analysis of the model. By reducing the response delay between vehicles, the system's stability and safety are enhanced, ensuring more consistent inter-vehicle distances and smoother operation. Furthermore, we developed tools to generate datasets for classifier training. These improvements are of significant importance for the practical deployment of CACC systems, potentially reducing accident risks in real-world scenarios and enhancing traffic efficiency.

Although promising results have been achieved in this study, there are still areas that can be further improved. For instance, future research could explore the scalability and robustness of the system in larger-scale vehicular networks. Addressing these issues would provide deeper insights into the system's actual performance and potential enhancements.

Building on the current work, future efforts will integrate more realistic traffic conditions and test the system under varying levels of communication reliability. Additionally, further optimization of the synchronization mechanism, particularly for high-density traffic scenarios, will be explored to ensure real-time responsiveness. This research aims to contribute new theoretical foundations and technical insights for improving the synchronization and safety of CACC systems, driving progress in intelligent transportation systems.

REFERENCES

[1] Sudha Anbalagan, Ali Kashif Bashir, Gunasekaran Raja, Priyanka Dhanasekaran, Geetha Vijayaraghavan, Usman Tariq, and Mohsen Guizani. Machine-learning-based efficient and secure rsu placement mechanism for software-defined-iov. *IEEE Internet of Things Journal*, 8(18):13950–13957, 2021.

[2] Shubing Liao, Yaxin Wu, Kanghua Ma, and Yunyun Niu. Ant colony optimization with look-ahead mechanism for dynamic traffic signal control of iov systems. *IEEE Internet of Things Journal*, 11(1):366–377, 2024.

[3] Wei Duan, Jinyuan Gu, Miaowen Wen, Guoan Zhang, Yancheng Ji, and Shahid Mumtaz. Emerging technologies for 5g-iov networks: Applications, trends and opportunities. *IEEE Network*, 34(5):283–289, 2020.

[4] Arooj Masood, Demeke Shumeye Lakew, and Sungrae Cho. Security and privacy challenges in connected vehicular cloud computing. *IEEE Communications Surveys Tutorials*, 22(4):2725–2764, 2020.

[5] Taiyuan Gong, Li Zhu, F. Richard Yu, and Tao Tang. Edge intelligence in intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 24(9):8919–8944, 2023.

[6] Jun Zhang and Khaled B. Letaief. Mobile edge intelligence and computing for the internet of vehicles. *Proceedings of the IEEE*, 108(2):246–261, 2020.

[7] Guozhi Yan, Kai Liu, Chunhui Liu, and Jie Zhang. Edge intelligence for internet of vehicles: A survey. *IEEE Transactions on Consumer Electronics*, pages 1–1, 2024.

[8] Bo Yang, Xuelin Cao, Kai Xiong, Chau Yuen, Yong Liang Guan, Supeng Leng, Lijun Qian, and Zhu Han. Edge intelligence for autonomous driving in 6g wireless system: Design challenges and solutions. *IEEE Wireless Communications*, 28(2):40–47, 2021.

[9] Veronika Lesch, Martin Breitbach, Michele Segata, Christian Becker, Samuel Kounev, and Christian Krupitzer. An overview on approaches for coordination of platoons. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):10049–10065, 2021.

[10] Dongyao Jia, Kejie Lu, Jianping Wang, Xiang Zhang, and Xuemin Shen. A survey on platoon-based vehicular cyber-physical systems. *IEEE communications surveys & tutorials*, 18(1):263–284, 2015.

[11] Yi Liu and Wei Wang. A safety reinforced cooperative adaptive cruise control strategy accounting for dynamic vehicle-to-vehicle communication failure. *Sensors*, 21(18):6158, 2021.

[12] Tianci Yang, Carlos Murguia, Dragan Nešić, and Chen Lv. A robust cacc scheme against cyberattacks via multiple vehicle-to-vehicle networks. *IEEE Transactions on Vehicular Technology*, 72(9):11184–11195, 2023.

[13] Sinan Öncü, Jeroen Ploeg, Nathan Van de Wouw, and Henk Nijmeijer. Cooperative adaptive cruise control: Network-aware analysis of string stability. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1527–1537, 2014.

[14] Bart van Arem, Cornelie J. G. van Driel, and Ruben Visser. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):429–436, 2006.

[15] Vicente Milanés, Steven E. Shladover, John Spring, Christopher Nowakowski, Hiroshi Kawazoe, and Masahide Nakamura. Cooperative adaptive cruise control in real traffic situations. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):296–305, 2014.

[16] Lian Cui, Zheng Chen, Aobo Wang, Jia Hu, and Byungkyu Brian Park. Development of a robust cooperative adaptive cruise control with dynamic topology. *IEEE Transactions on Intelligent Transportation Systems*, 23(5):4279–4290, 2021.

[17] Yuanheng Zhu, Dongbin Zhao, and Haibo He. Synthesis of cooperative adaptive cruise control with feedforward strategies. *IEEE Transactions on Vehicular Technology*, 69(4):3615–3627, 2020.

[18] Kakan C Dey, Li Yan, Xujie Wang, Yue Wang, Haiying Shen, Mashrur Chowdhury, Lei Yu, Chenxi Qiu, and Vivekgautham Soundararaj. A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (cacc). *IEEE Transactions on Intelligent Transportation Systems*, 17(2):491–509, 2015.

[19] Yeojun Kim, Jacopo Guanetti, and Francesco Borrelli. Compact cooperative adaptive cruise control for energy saving: Air drag modelling and simulation. *IEEE Transactions on Vehicular Technology*, 70(10):9838–9848, 2021.

[20] Faris Alotibi and Mai Abdelhakim. Anomaly detection for cooperative adaptive cruise control in autonomous vehicles using statistical learning and kinematic model. *IEEE Transactions on Intelligent Transportation Systems*, 22(6):3468–3478, 2020.

[21] Taehooie Kim and Kshitij Jerath. Congestion-aware cooperative adaptive cruise control for mitigation of self-organized traffic jams. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):6621–6632, 2021.

[22] Wangyang Yu, Yadi Wang, Lu Liu, Yisheng An, Bo Yuan, and John Panneerselvam. A multiperspective fraud detection method for multiparticipant e-commerce transactions. *IEEE Transactions on Computational Social Systems*, 2023.

[23] Iwona Grobelna and Andrei Karatkevich. Challenges in application of petri nets in manufacturing systems. *Electronics*, 10(18):2305, 2021.

[24] Wangyang Yu, Chungang Yan, Zhijun Ding, Changjun Jiang, and Mengchu Zhou. Analyzing e-commerce business process nets via incidence matrix and reduction. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(1):130–141, 2016.

[25] Ali Saleh, Manuel Chiachío, Juan Fernández Salas, and Athanasios Kolios. Self-adaptive optimized maintenance of offshore wind turbines by intelligent petri nets. *Reliability Engineering & System Safety*, 231:109013, 2023.

[26] Junyu Dong, Wenjun Wu, Yang Gao, Xiaoxi Wang, and Pengbo Si. Deep reinforcement learning based worker selection for distributed machine learning enhanced edge intelligence in internet of vehicles. *Intelligent and Converged Networks*, 1(3):234–242, 2020.

[27] Kai Wang, Hao Yin, Wei Quan, and Geyong Min. Enabling collaborative edge computing for software defined vehicular networks. *IEEE network*, 32(5):112–117, 2018.

[28] Zhijuan Hu, Danyang Wang, Zan Li, Meng Sun, and Weizhi Wang. Differential compression for mobile edge computing in internet of vehicles. In *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 336–341. IEEE, 2019.

[29] Duc Lich Luu, Ciprian Lupu, Laith S Ismail, and Hamid Alshareefi. Spacing control of cooperative adaptive cruise control vehicle platoon. In *2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, pages 1–6. IEEE, 2020.

[30] Jaswandi Sawant, Uttam Chaskar, and Divyesh Ginoya. Robust control of cooperative adaptive cruise control in the absence of information about preceding vehicle acceleration. *IEEE Transactions on Intelligent Transportation Systems*, 22(9):5589–5598, 2020.

[31] Ali Hamroun, Karim Labadi, Mourad Lazri, and Jean-Pierre Barbot. A petri nets-based simulation methodology for modular modeling and performance evaluation of car-sharing networks. *IEEE Transactions on Automation Science and Engineering*, 2022.

[32] Haijing Ning, Yisheng An, Yaxin Wei, Naiqi Wu, Chen Mu, Hanhan Cheng, and Chenxing Zhu. Modeling and analysis of traffic warning message dissemination system in vanets. *Vehicular Communications*, 39:100566, 2023.

[33] Qi Guo, Wangyang Yu, Fei Hao, Yuke Zhou, and Yuan Liu. Modelling and analysis of adaptive cruise control system based on synchronization theory of petri nets. *Electronics*, 11(21):3632, 2022.

[34] Wu Zhehui. Introduction to petri nets. *Beijing, Press of Machinery and Industry*, pages 15–21, 2006.

[35] Hongda Qi, Mingjian Guang, Junli Wang, Chungang Yan, and Changjun Jiang. Probabilistic reachability prediction of unbounded petri nets: a machine learning method. *IEEE Transactions on Automation Science and Engineering*, 2023.

[36] Ye Li, Hao Wang, Wei Wang, Lu Xing, Shanwen Liu, and Xueyan Wei. Evaluation of the impacts of cooperative adaptive cruise control on reducing rear-end collision risks on freeways. *Accident Analysis & Prevention*, 98:87–95, 2017.

[37] Yumeng Cheng, Wangyang Yu, Xiaojun Zhai, Fei Hao, and Yuan Liu. Grading and calculation of synchronic distance in petri nets for trustworthy modeling and analyzing. In *2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 605–611. IEEE, 2023.