Egyptian Informatics Journal 23 (2022) 499–510

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

# Complex attack detection scheme using history trajectory in internet of vehicles

Wonjin Chung [a], Taeho Cho [b,*]

[a] College of Information and Communication Engineering, Sungkyunkwan University, Suwon 16419, Korea
[b] College of Computing and Informatics, Sungkyunkwan University, Suwon 16419, Korea

ARTICLE INFO

*Article history:*
Received 28 January 2022
Revised 10 May 2022
Accepted 22 May 2022
Available online 31 May 2022

ABSTRACT

The internet of vehicles technology provides convenience to drivers and prevents traffic accidents via wireless communication between road infrastructure and autonomous vehicles by sharing real-time traffic information. However, attackers can easily penetrate networks by exploiting the vulnerabilities of wireless communications. An attacker can falsify real-time traffic information and transmit it to a vehicle, causing traffic jams or preventing autonomous vehicles from receiving legitimate real-time traffic information. If autonomous vehicles do not receive accurate information, the arrival time at the destination can be affected, and accidents due to incorrect driving can occur. Because traffic accidents can cause casualties, they must be prevented. Various schemes have been proposed to detect attacks that occur on the internet of vehicles, and these security schemes can prevent traffic accidents by detecting attacks at high speeds. However, the existing schemes focus on quickly identifying a single attack but encounter difficulties when attempting to detect complex attacks that occur simultaneously. The proposed scheme uses a history trajectory to detect complex attacks. The proposed scheme stores behavioral information on all vehicles and road infrastructure using a control center. This information becomes a history trajectory that is used to detect attacks. Thereafter, when the vehicle is abnormally driven, the control center analyzes its driving path. When analyzing the vehicle driving process, the control center determines that an attack is being attempted when the road infrastructure or a vehicle makes an erroneous state transition. In addition, the type of attack is analyzed to identify compromised vehicles or road infrastructure and take measures to prevent further problems. Thus, the proposed scheme can detect complex attacks through history trajectory analysis. The experimental results demonstrate that in 80% of attempted attacks, the proposed scheme detects complex attacks with a probability of 97.56%.



## 1. Introduction

The internet of things (IoT) is widely used owing to the development of the internet and computer technology [1,2]. This technology provides a variety of services that modern people have never experienced before. The IoT is a technology in which all objects connected to the internet communicate and provide various services to users. These technologies have availed various services by connecting cars to the Internet, which has helped to improve autonomous driving technologies. In the internet of vehicles (IoV), the IoT is applied to automobiles to facilitate connectivity to all objects on the road through a roadside unit (RSU) [3–5]. Autonomous vehicles use technologies such as light detection and ranging (LiDAR) to recognize road conditions, communicate with RSUs or other autonomous vehicles, and collect traffic information in real-time to drive safely to their destinations [6–8]. To achieve autonomous driving, the integrity of the road infrastructure must be maintained for the desired communication and the information transmitted. However, because an autonomous vehicle network transmits and receives information wirelessly, an attacker can easily penetrate the network and attempt various attacks. Attacks on autonomous vehicles prevent normal driving by removing necessary information, introducing false information, or communicating altered information to the vehicle. The goal of these attacks is to cause accidents, damage property, and injure

* Corresponding author.
*E-mail addresses:* wonjin12@skku.edu (W. Chung), thcho@skku.edu (T. Cho).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2022.05.002

human beings. Therefore, autonomous vehicles must be able to detect and prevent traffic accidents. Many studies have been conducted to efficiently detect attacks occurring in the IoV [9]. However, these detection schemes do not consider complex attacks that occur simultaneously. Because complex attacks prevent the attack detection process from being performed, it is impossible to detect them using existing detection schemes that focus on single attack detection. If complex attacks continue, the number of transmitted packets increases. Consequently, the autonomous vehicle processing load increases, and incorrect information can be delivered when messages are manipulated by attackers. This leads to road congestion and accidents. To solve this problem, the proposed scheme uses history trajectory information to detect complex attacks. The proposed technique was modeled using the discrete event system specification (DEVS) [10,11] theory-based simulation language, and the models express the behavior of real-world objects. Subsequently, the proposed scheme constructed an IoV environment using the designed model, and its performance was evaluated through simulations.

The contributions of the proposed scheme are as follows. First, security is improved by detecting the complex attacks that occur in the IoV. The proposed scheme detects a single attack at a rate similar to that of existing detection schemes that focus on a single attack, it can detect a complex attack in which the existing detection scheme has not been detected. Second, the proposed scheme can reduce network overhead by detecting complex attacks. By detecting complex attacks, unnecessary packet transmission can be blocked early, and only necessary information can be collected.

The remainder of this paper is organized as follows. Section 2 describes the attack and security schemes relevant to the IoV. Section 3 describes the IoV and DEVS. Section 4 presents the details of the proposed scheme. Section 5 provides an evaluation of the proposed techniques compared to the existing scheme. Section 6 discusses the necessity and the development potential of the proposed scheme. Finally, Section 7 presents conclusions and directions for future research.

## 2. Related work

Autonomous vehicles use real-time traffic information received through network communications to drive safely. Autonomous vehicle networks must use secure channels and deliver accurate data without compromising information. However, because the network of autonomous vehicles uses primarily wireless communication, it is an open system and is thus vulnerable to attacks [12–15]. Malicious attackers use the broadcast characteristics of the controller area network (CAN bus) to gain access to communication and demonstrate that they can attack [16]. The attacker drops or delays packets sent to the autonomous vehicle to paralyze communication [17], and cause accidents using various methods, such as injecting false information into autonomous vehicles or tampering with normal information. VeCure detects an attack in which a false message is transmitted using message authentication code and an attack in which an unnecessary message is continuously transmitted using the message count [18]. However, this scheme cannot detect complex attacks in which the content of unnecessary messages is manipulated and sent repeatedly. Autonomous vehicle accidents can injure drivers and pedestrians; therefore, it is essential to avoid accidents by detecting and responding to attacks. A denial of service (DoS) attack on a vehicle is a threatening attack that can cause accidents; it paralyzes services by continuously sending or dropping packets [19]. Cumulative sum (CUSUM) uses continuous monitoring to detect attacks in which packets are not transmitted or are dropped [20]. However, this scheme is difficult to detect when complex attacks are attempted in special environments, such as on roads where vehicles are rare. In other words, existing schemes that focus on a single attack, such as VeCure or CUSUM, can detect a single attack perfectly, but have problems detecting a complex attack. If these attacks are not detected, autonomous vehicle normal destination arrival times can be delayed because the vehicles create and drive incorrect driving plans, resulting in lost time.

## 3. Background

This section provides an overview of the IoV, autonomous vehicles, and the DEVS simulation theory.

### 3.1. Internet of vehicles and autonomous vehicles

The IoV is a convergence of automobile and IoT technologies and enables efficient driving by providing real-time traffic information to autonomous vehicles. Currently, vehicular ad hoc networks (VANETs) are rapidly evolving into IoV networks, and these advances have helped in the development of autonomous vehicles [21]. The IoV promotes safe driving by enabling autonomous vehicle-to-vehicle communication about the environment in real-time, such as road infrastructure and pedestrians. An autonomous vehicle, a key element of the IoV, recognizes the road environment through sensors, plans its own route to its destination, and drives itself safely to the destination. Most traffic accidents are reportedly caused by driver carelessness or aging. When fully autonomous vehicles are commercialized, traffic accidents can be reduced [22,23]. Autonomous driving technology is divided into six levels—Level 0 to Level 5—and the final Level 5 autonomous driving technology can operate without a driver [7,8]. Currently, autonomous vehicles are about to be put into practical use with Level 3 autonomous driving technology, and for this purpose, much research is being conducted on safe autonomous driving [24–27]. Three systems are required for autonomous driving. First, an autonomous driving system involves sensing, perception, and decision-making algorithms. This method uses raw data collected by sensors to identify the surrounding environment and determine its behavior. Second, an autonomous driving system is a client system that consists of an OS and a hardware platform. The system mixes multiple algorithms to meet real-time and reliability requirements. Finally, the system for autonomous driving is a cloud platform that provides high-definition (HD) maps, simulations, and data storage [28]. The system provides the offline computing and storage capabilities required for autonomous vehicles. Autonomous vehicles can use these subsystems to plan and drive routes to their destinations. Autonomous vehicles must collect real-time information to prevent accidents that are associated with not recognizing pedestrians or vehicles in blind spots, or to avoid road congestion caused by construction or accidents. To collect real-time information, autonomous vehicles use vehicle-to-everything (V2X) communication [29–32]. There are several types of V2X communication, such as vehicle-to-vehicle (V2V), vehicle-to-road infrastructure (V2I), and vehicle-to-network (V2N), based the object that is communicating with the vehicle [29]. V2V communication is a technology for wirelessly exchanging information between vehicles and blocks the risk of collision. V2V communication is mainly used to prevent accidents or prevent the continuous occurrence of accidents and to help the driver avoid vehicles in blind spots. V2I communication wirelessly sends and receives information between vehicles and road infrastructure, and can be guided in real-time traffic conditions and unexpected situations. Real-time traffic information obtained through V2X communication helps autonomous vehicles drive safely.

### 3.2. Discrete event system specification

The DEVS formalism, introduced by Zeigler, provides a means of specifying mathematical objects, and is the basis for establishing new simulation languages with more precise meanings [10]. The DEVS formalism uses input/output, state, and time-based functions to determine the next state and output. It was developed using various programming languages, such as C++ [33], Java [34,35], and Python [36], and is used as the basis of the simulation language. The structure of the DEVS model provides a basis for designing control logic in the form of individual events that can receive confirmation of control commands within a specific time. The DEVS consists of an atomic model of the smallest unit and a coupled model that connects them. The atomic model describes the transition of the state by the input event, and the transition of the internal state over time. The coupled model provides a function to connect models and create a large-scale system, which includes a coupled model and an atomic model, to connect and configure the system. The composition of the atomic model is as follows [10]:

$$M = (X, Y, S, t_a, \delta_{ext}, \delta_{int}, \lambda) \quad (1)$$

X is the input event set, Y is the output event set, S is the state set, $t_a$ is a time advance function used to describe how long it can stay in a state, $\delta_{ext}$ is an external state transition function that describes how an external input event changes the state of the element model, $\delta_{int}$ is an internal state transition function that describes the phenomenon of changing to the next state when the required time reaches the lifetime of the current state, regardless of an external input event, and $\lambda$ is an output function that describes the occurrence of an output event to the outside when an internal state transition occurs. The coupled model is described as follows [10]:

$$N = (X, Y, D, M_d \mid d \in D, EIC, EOC, IC, Select) \quad (2)$$

X is the set of input events and Y is the set of output events. D is the name set of the subcomponents, and $\{M_d\}$ is the set of subcomponents. EIC, EOC, and IC refer to the set of external input couplings, external output couplings, and internal couplings, respectively. Select is a tie-breaking function that describes which model to select when a plurality of the constituent models have an internal state transition scheduled at the same time.

## 4. Proposed scheme

This section introduces the proposed scheme, explains the model configuration of the proposed scheme, describes the elements of the configuration model, provides a detailed explanation of how to detect an attack, and explains the models used in the proposed scheme.

### 4.1. Overview

The IoV provides convenience to drivers; however, because it uses wireless communication, a malicious attacker can break into the vehicle network and attempt an attack that causes a traffic accident. To cause a traffic accident, the attacker manipulates the traffic information or attempts an attack that prevents data from being transmitted to the vehicle. To detect these attacks, security schemes such as VeCure [18], CUSUM [20] and secure and efficient ad hoc distance vectors (SEAD) [37] have been proposed. However, because these schemes specialize in detecting a single attack, it is impossible to detect complex attacks that occur simultaneously. Autonomous vehicles must detect attacks because accidents can occur if complex attacks are not detected.

### 4.2. Detailed procedure

The attacks attempted in the proposed scheme occur in the CAN bus and V2X communications; six of the attacks are likely to be caused by the attack and are of interest to security experts [16,19,38]. The attacks selected in the proposed scheme are blackhole, flooding, replay, spoofing, Sybil, and false data injection (FDI) attacks. Table 1 lists the attacks classified by the vehicle communication network and the type of attack selected [39].

The proposed scheme presents a two-step procedural security technique for detecting attacks that occur in an IoV. The first detection phase focuses on a single attack. It aims at security similar to existing detection schemes such as VeCure. The second detection procedure focuses on complex attacks and analyzes all the actions of IoV objects to detect complex attacks. First, the detection procedure that focuses on a single attack in the proposed method is described. The first procedure of the proposed scheme identifies normal and abnormal behavior information. This identification prevents traffic accidents by stopping the autonomous vehicle when abnormal behavior occurs. Next, the proposed scheme identifies the network in which the attack is attempted during the CAN bus and V2X communication for attack analysis. After identifying the network, a single attack is detected using the algorithms in existing detection techniques such as SEAD and CUSUM. This detection method requires extensive analysis; therefore, high-performance computing is required. Consequently, the proposed scheme delivers damage information to the control center, analyzes it quickly, and then delivers countermeasures to the relevant autonomous vehicle. The purpose of the first procedure is to prevent traffic accidents by minimizing the time required for attack detection and quick response. Fig. 1 shows an example of detecting a single attack using the first procedure of the proposed scheme.

When a victim autonomous vehicle receives a large number of dummy messages from a malignant autonomous vehicle, this is identified as abnormal behavior, and a report is sent to notify the control center through the RSU. The report contains information on abnormal transitions. The control center that receives the report analyzes the information and identifies the attack. The control center analyzes the attack pattern using the algorithms of existing detection techniques, such as SEAD and CUSUM, and identifies that a replay attack is being attempted. When the attack has been detected, the control center delivers new driving information to the autonomous vehicle to prevent accidents. This method can detect a single attack occurring in the IoV, but encounters problems when attempting to detect complex attacks occurring simultaneously. Therefore, a complex attack is detected using the second detection procedure in the proposed scheme. The second procedure in the proposed scheme uses a history trajectory to detect complex attacks. Fig. 2 illustrates the process the control center uses to collect situational information on autonomous vehicles and road infrastructure.

All data generated on the road are collected by the RSU. Thereafter, the RSU transmits the collected information to the control

**Table 1**
Types of attacks that occur on autonomous vehicles.

| Vehicle network | Attack name | Attack type |
|---|---|---|
| CAN bus | Replay attack | Data integrity/data trust attacks |
| | Flooding attack | Availability attacks |
| | Spoofing attack | Authenticity/identification attacks |
| V2X Communication | Blackhole attack | Availability attacks |
| | FDI attack | Data integrity/data trust attacks |
| | Sybil attack | Authenticity/identification attacks |

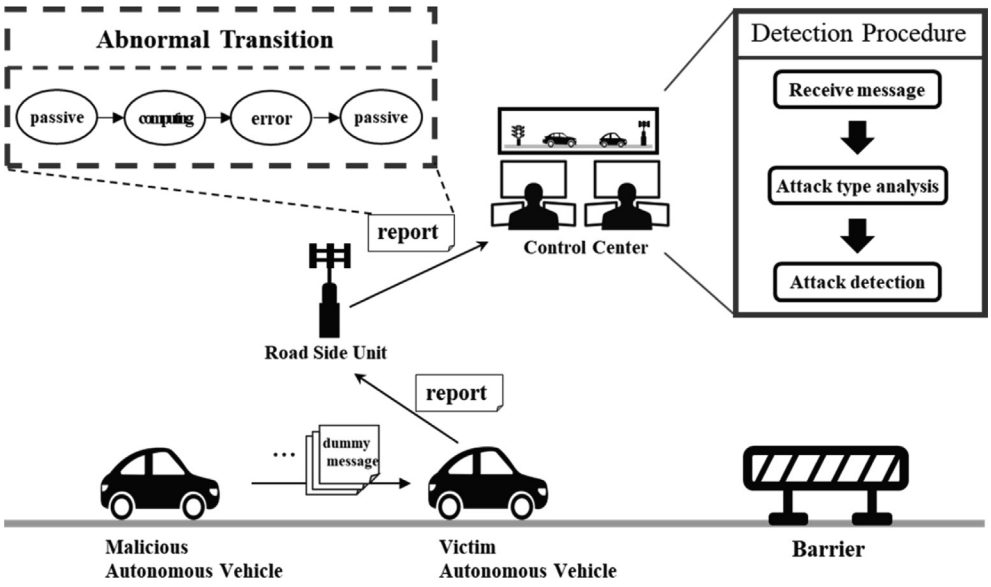


**Fig. 1.** Attack detection process through the proposed scheme (first procedure).

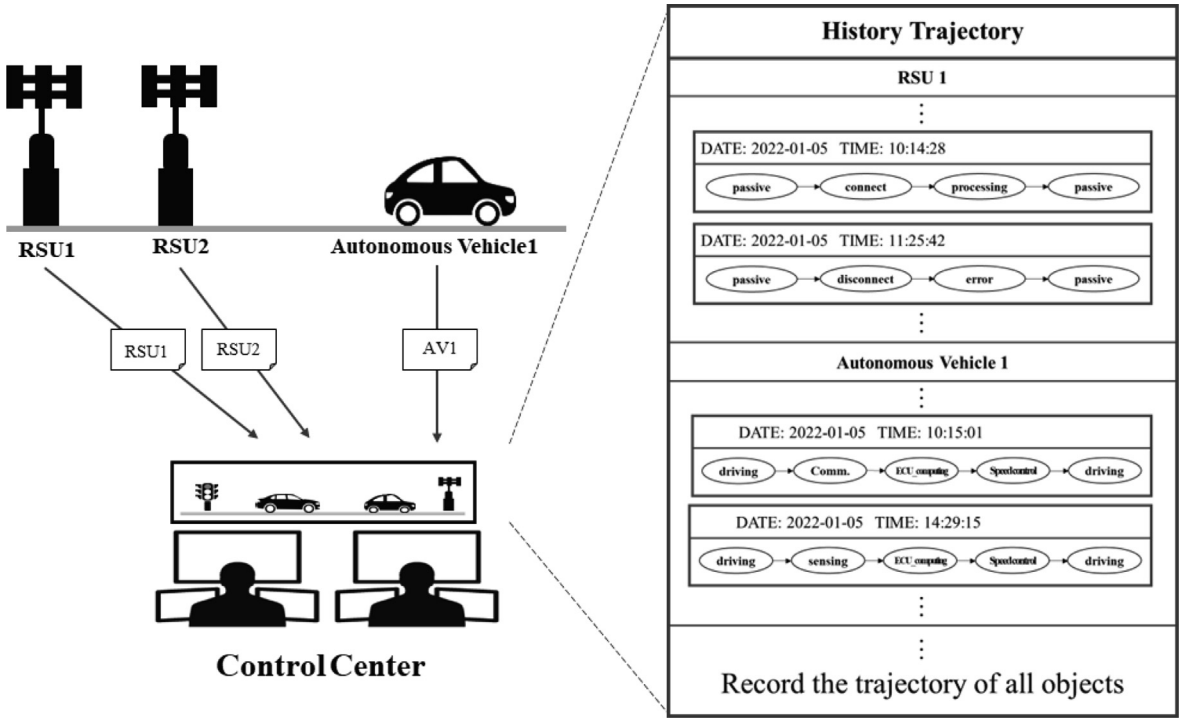


**Fig. 2.** History trajectory collection process.

center without processing. The transmitted information includes the trajectory for vehicle driving or RSU access records. The control center classifies and stores the data received from the RSU by domain. In the proposed scheme, the collected trajectory information is used as an important factor in detecting complex attacks. Complex attacks can occur in various forms and are difficult to detect because they occur simultaneously and interfere with the attack-detection process. For example, attackers can compromise autonomous vehicles and RSUs and attempt Sybil and FDI attacks. Fig. 3 shows Sybil and FDI attacks occurring simultaneously in the IoV and interfering with normal driving.

Attackers compromise autonomous vehicles and RSUs before attempting complex attacks. Because the RSU is installed outside, it can easily be compromised by an attacker [40,41]. Thereafter, the attacker attempts a Sybil attack using a compromised autonomous vehicle to create a traffic jam [42]. These attacks can be

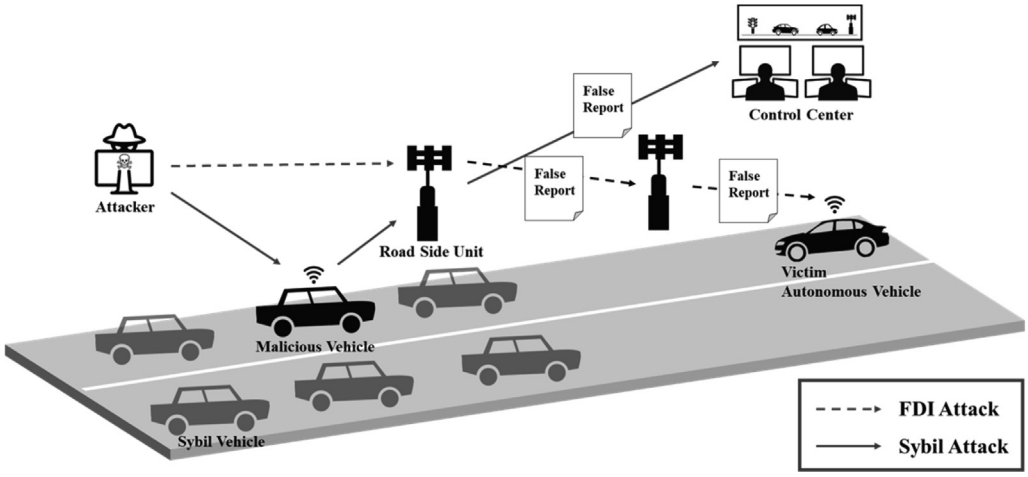


**Fig. 3.** Complex attacks that occur in the IoV (FDI attack, Sybil attack).

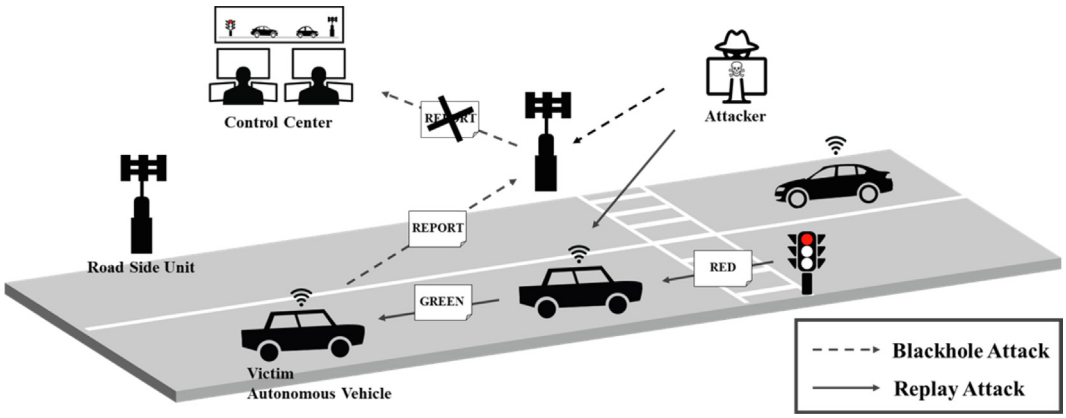


**Fig. 4.** Complex attacks that occur in the IoV (blackhole attack, replay attack).

detected by the RSU, but the compromised RSU information is delivered to the control center without verification. The control center trusts the message delivered by the autonomous vehicle and delivers a road detour message to another vehicle to avoid road congestion. In addition, when a nearby vehicle approaches the attack area, information regarding the attempted attack can be delivered to the control center. To prevent vehicle access in advance, the attacker uses the RSU to generate and deliver a false road-detour message to all vehicles [43–45]. The vehicle that receives the message does not approach the road and instead detours to another road. By attempting such a complex attack, a drivable road can appear to be an undrivable road and confuse autonomous vehicles.

As another type of compound attack, a blackhole and a replay attack may be attempted simultaneously. The purpose of these attacks is to cause traffic jams by stopping vehicles. Fig. 4 shows a blackhole and a replay attack occurring simultaneously. This type of complex attack can only be performed if the attacker has damaged the RSU and the vehicle in advance, and the autonomous vehicle is later waiting for a signal. A compromised autonomous vehicle continuously transmits a stop signal message to another vehicle, even though it is changed to a driving signal [46,47]. Because the waiting vehicle receives continuous messages, it detects that a replay attack has been attempted and delivers an attack intrusion-related message to the control center through the RSU. However, the compromised RSU collects these messages and removes them immediately without delivering them to the control center [48]. Messages are also removed in the same manner because other vehicles monitor road congestion and deliver situational information through compromised RSUs. As a result, the road becomes congested because the waiting vehicle cannot drive and continues to wait. The proposed scheme uses the history trajectory to detect complex attacks that occur in the situations shown in Figs. 3 and 4. Because the proposed scheme stores all history trajectories in the control center, information suitable for the attack situation can be used. The proposed scheme can detect complex attacks by analyzing past behavior through the trajectories of all RSUs and autonomous vehicles.

Fig. 5 shows attack detection using the history trajectory, which is a two-step procedure in the proposed scheme. Because the control center collects all situational information generated on the road, it is possible to grasp the process of collecting and delivering information using the RSU. Based on this information, it is possible for the control center to compare the context information of all

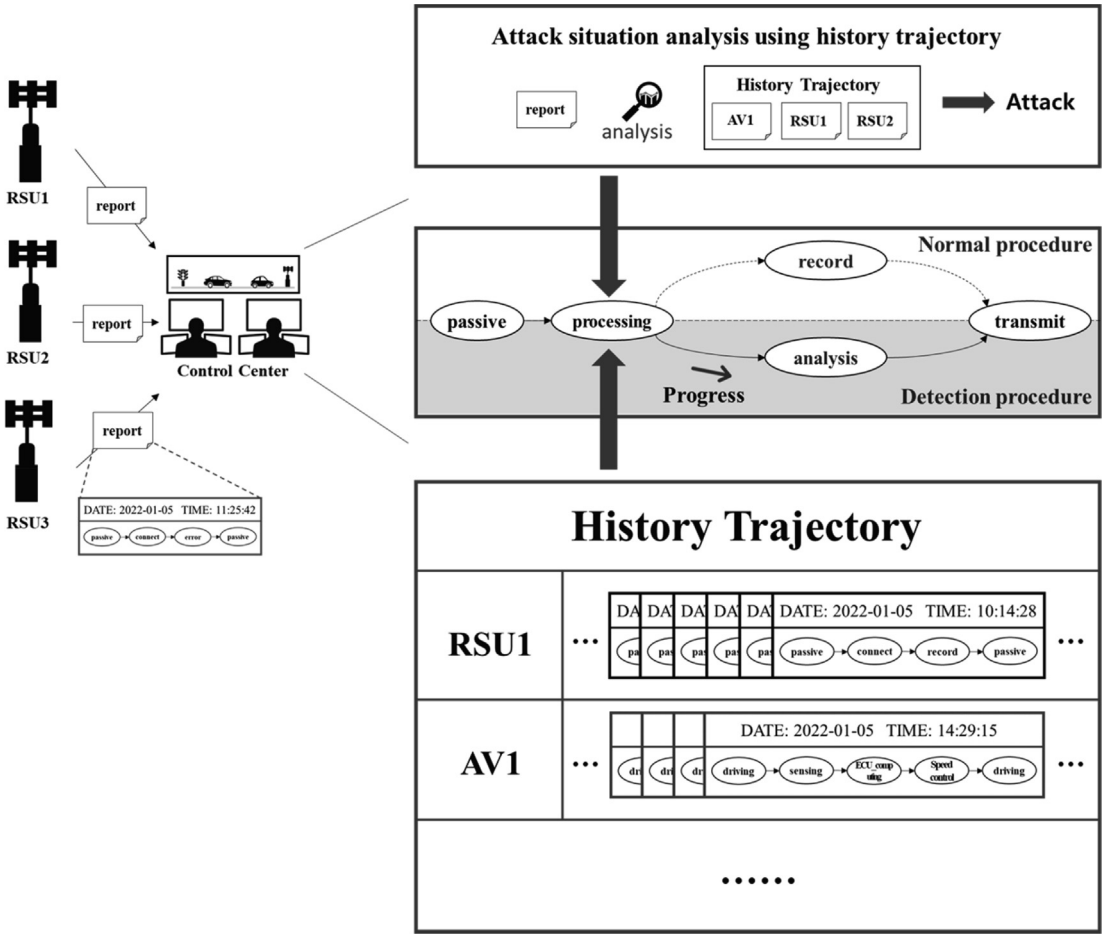


**Fig. 5.** Attack detection process through the proposed scheme (second procedure).

RSUs. When comparing the situation information of the RSU, it may be confirmed that the information related to the vehicle entry of the compromised RSU is different from that of the other RSUs.

Fig. 6 shows the method of coping with the attack described above. The control center determines that this is an attack, writes the IDs of the corresponding vehicle and RSU on the blacklist, and then delivers them to all RSUs and vehicles [49]. Thereafter, the normal RSU and vehicle check the blacklist and drop the RSU on the blacklist and the message transmitted from the vehicle. Using this method, all vehicles can drive on a road that appears to be congested.

### 4.3. Detailed procedure

This section introduces the DEVS-based models that are used in the proposed scheme. The model in the proposed scheme describes the behavior of objects in the real world, and only the functions necessary for the proposed scheme were designed. Fig. 7 shows the overall structure of the proposed scheme using a model diagram and an SES tree [50,51].

The proposed scheme is largely composed of a road model and a cloud computing model. The road model consists of driving-related models, including autonomous vehicles. The cloud computing model collects all road situation information based on the control center and instructs the autonomous vehicle to take appropriate actions when an attack is attempted.

Fig. 8 shows a road model diagram. The road model collects information on various road infrastructures through the RSU, centering on autonomous vehicles. The autonomous vehicle uses V2X communication, which collects information from the outside, and CAN bus communication, which detects a situation from an internal sensor and transmits it to the electronic control unit (ECU). Fig. 9 shows a state diagram of the advanced driver assistance system (ADAS) model that is responsible for driving an autonomous vehicle.

The ADAS model was designed to self-adjust the speed during the speed control phase to express the autonomous driving function. If road condition information is collected through V2X communication or the CAN bus, it transitions to the ECU_computing phase, adjusts the speed according to the situation, and then transitions to the speed control phase again. The autonomous vehicle model drives to the destination by repeating these transitions. Fig. 10 illustrates a cloud computing model that communicates with autonomous vehicles and monitors all traffic conditions.

The broker model classifies the collected information and delivers it to the model, and the monitoring system model stores the autonomous vehicle trajectory information and the road infrastructure classified in the broker model. Finally, the security system model analyzes the attack situation when an autonomous vehicle behaves abnormally and detects an attack using an existing scheme or history trajectory. Algorithm 1 shows the process of storing real-time traffic information in the history trajectory.

**Algorithm. 1:** Real-time driving information storage

```
1:HT ← Ø // Initialization history trajectory (HT)
2:repeat
3: while AV_ID = input_ID do
4: if lastData_time < input_time then
5: HT ← inputData
6: update lastData_time
7: else
8: Alarm ← T
9: end if
10: end while
11: if Alarm changed then
12: return alarm message // To check whether an attack occurs, an alarm message
13: end if and input data are transmitted to the security model.
14:end repeat // until the input of the new vehicle information
```

The proposed scheme compares the ID of the autonomous vehicle and the input ID to store the history trajectory. After the ID verification, the proposed scheme compares the input data transmission time with the last data transmission time to confirm the presence of an attack. If the input data is the latest transmission time value, it is stored in the history trajectory, and the last transmission time is updated. If the input data transmission time value is less than or equal to the latest transition time, it is suspected to be an attack, and a notification message is sent to the control center. Fig. 11 shows the state diagram for the monitoring system model, which stores all the traffic situation information.

The cloud computing model sends a request message to the monitoring system model for complex attack detection. Upon receiving the request message, the monitoring system model delivers the necessary information from the recorded situation to the cloud computing model. Subsequently, the cloud computing model analyzes the attack using the history trajectory. Algorithm 2 shows the attack detection process using V2X communication while driving an autonomous vehicle.

**Algorithm. 2:** V2X communication attack detection

```
1:input: Real-time driving information
2:if network_type = V2X_communication then
3: using existing scheme (VeCure) // First attack detection (single attack)
4: if attack detection then
5: Attack_type ← current attack information
6: Attack_info ← inputData
7: else
8: doubtID ← inputID
9: return R_HT // Request history trajectory for complex attack detection
10: end if
11:end if
12:
13:{
14:Requests history trajectory information to MS model and receives necessary information.
15:} // The MS model transmits autonomous vehicle information or RSU information according to the attack situation.
16:
17:if RSU_information = doubtID then
18: for all AV_ID do
19: if RSU_information = AV_information then
20: return error
21: else
22: return attack
23: end if
24: end for
25:end if
```

The cloud computing model performs the first procedure in detecting a single attack. The first procedure runs all the existing detection scheme algorithms, such as VeCure, to detect a single attack. If there is no matching attack type in the detection process, the cloud computing model receives the RSU history trajectory information through the MS model and proceeds with the second detection procedure. When the history trajectory is transmitted, the cloud computing model analyzes the information by backtracking the actions that occur based on the time when the data were

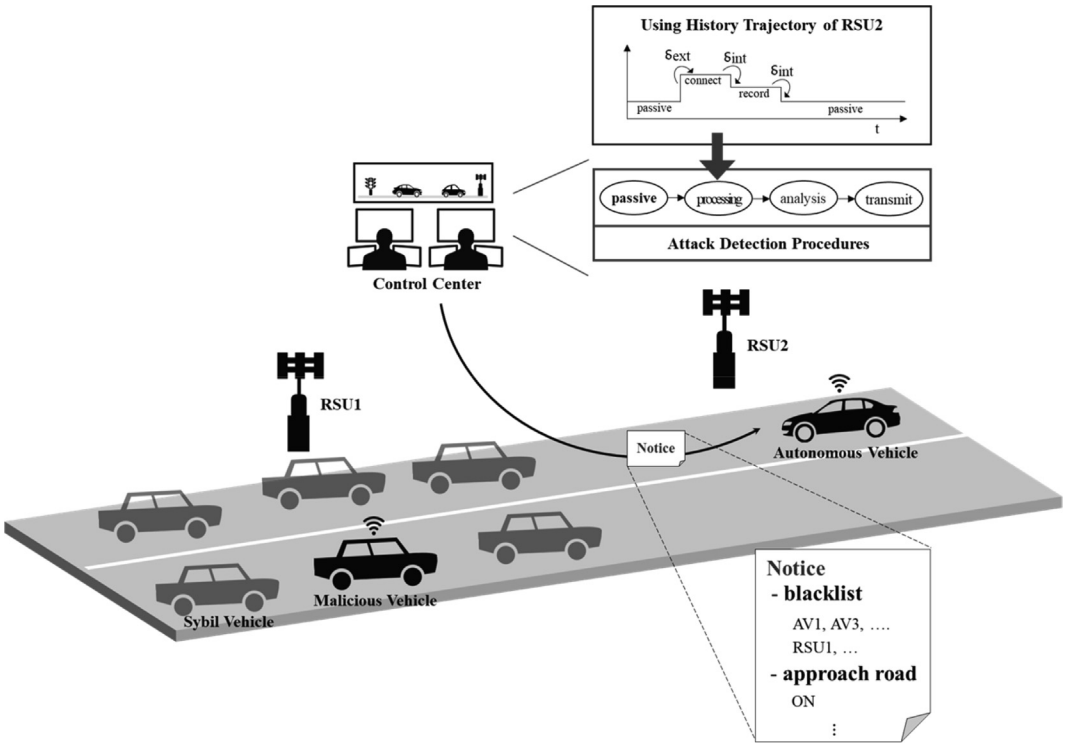


**Fig. 6.** Coping with complex attacks occurring in IoV.

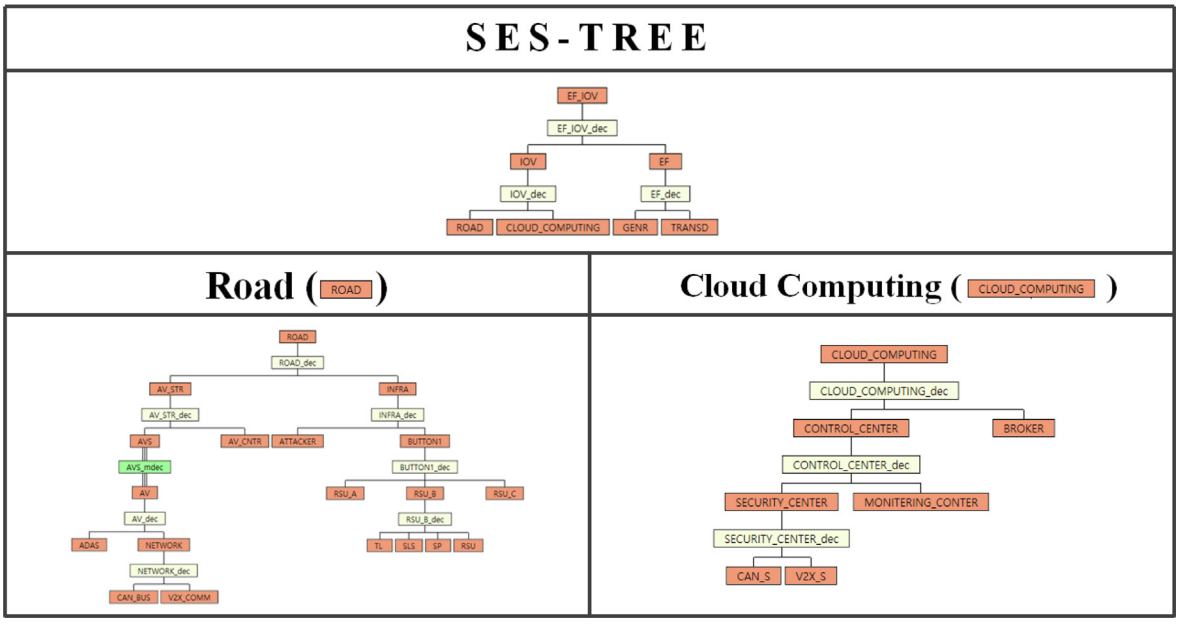


**Fig. 7.** Structure of the proposed scheme.

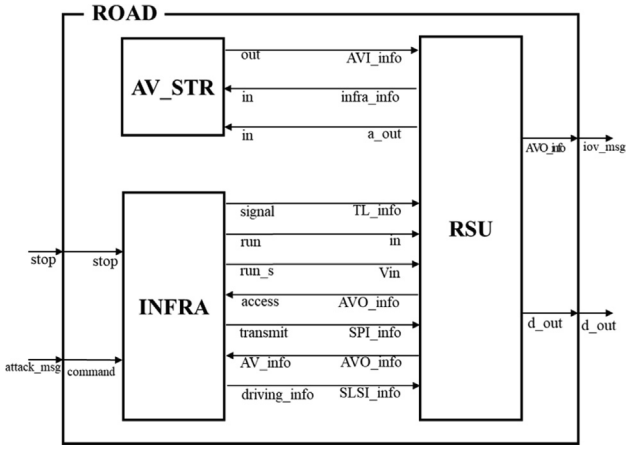


**Fig. 8.** Model diagram of the road model.

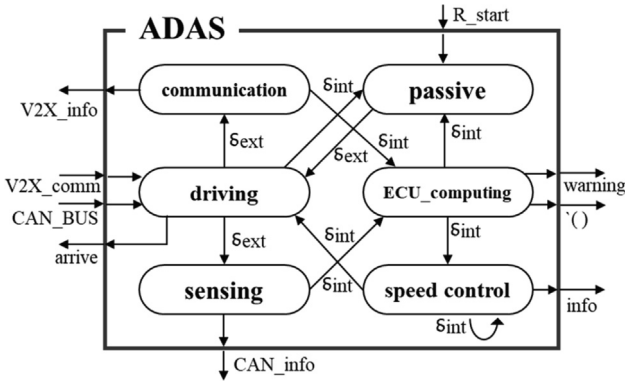


**Fig. 9.** State transition diagram of the ADAS model.

injected. If the RSU information concerning the history trajectory and the input autonomous vehicle driving information do not match, the cloud computing model determines that a complex attack has been attempted. Fig. 14 shows the timing diagram of the security system model that detects attacks. The security system model consists of a CAN bus security model and V2X security model. The state transition diagram is the same, and only the algorithms for the analysis phases were designed differently.

Fig. 12 shows how various attacks are detected by changing the state in the time order in the security system model. When it is a single attack, the security system model can detect the attack during the processing phase. However, when a compound attack is attempted, the security system model cannot detect the attack in the processing phase; therefore, it transitions to the request phase to obtain the history trajectory. After receiving information from the monitoring system model, the security system model transi-

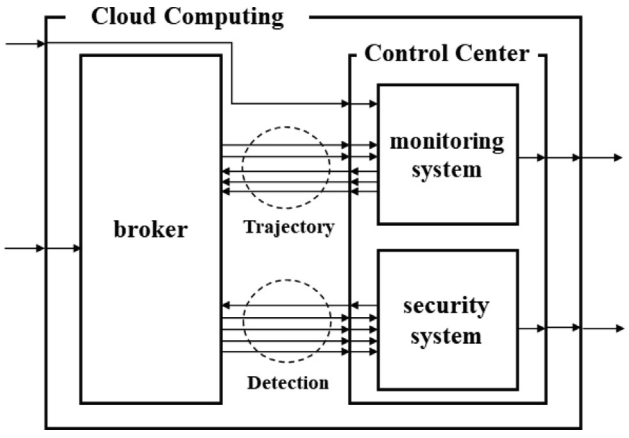


**Fig. 10.** Model diagram of the cloud computing model.

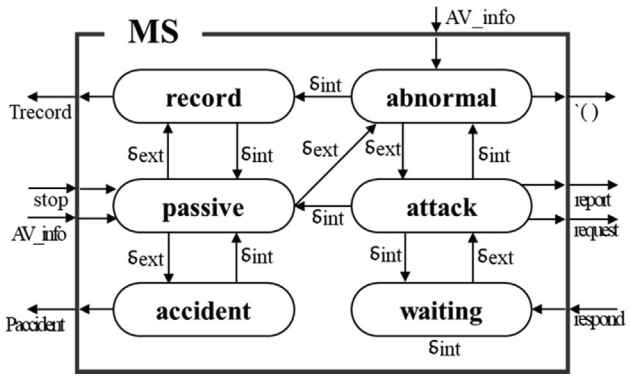


**Fig. 11.** State transition diagram of the MS model.

tions to an analysis phase to compare the situation information and detect a complex attack.

## 5. Simulation evaluation

The proposed scheme was modeled using the DEVS-ObjC program, based on DEVS formalism, and simulated using a model that reflects the real world. In an experimental environment to evaluate the performance of the proposed scheme, the road distance for vehicles to travel was 6 km, and RSUs were deployed every 2 km and communicated with autonomous vehicles. In addition, four vehicles were designed, and the vehicles departed consecutively at 0 km intervals. Attacks in the experimental environment were attempted through the RSU_B model, and the type of attack, whether blackhole, flooding, replay, spoofing, FDI, or Sybil, was determined through the GNER model. A complex attack was simulated by simultaneously generating two attacks. In the experiment,

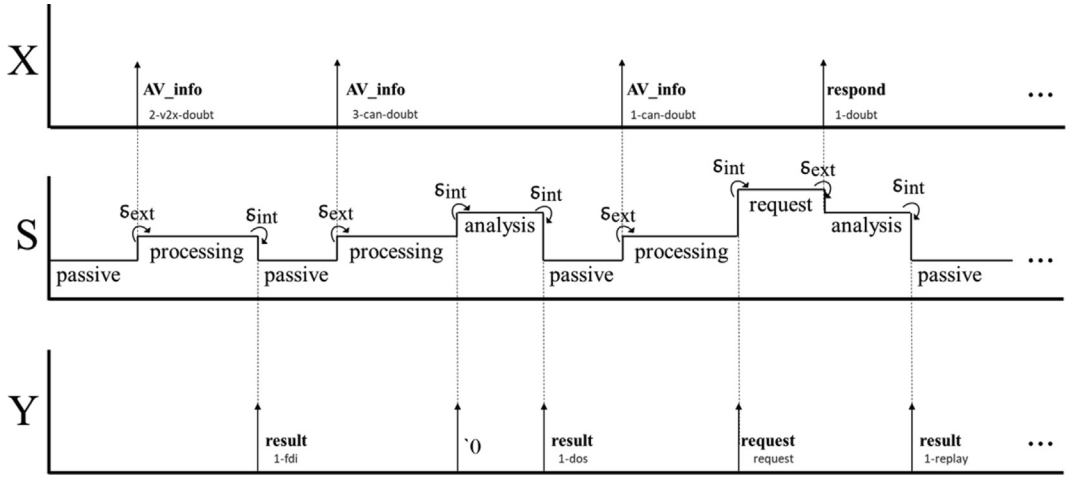


**Fig. 12.** Timing diagram of the SS models.

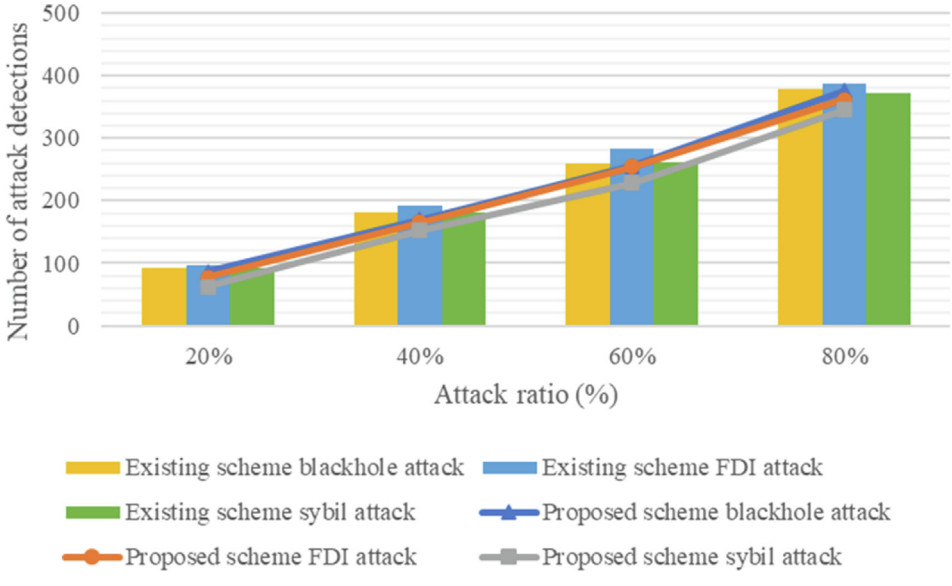


**Fig. 13.** Number of attack detections by attack rate (V2X communication).

only complex attacks that combined Sybil and FDI or replay and blackhole attacks occurred. The pedestrian model randomly changed its location, and the TL model was changed in the order of green, yellow, and red at a 6:1:3 time ratio. A speed limit sign was placed between 1500 m and 2500 m, and the speed was constrained below 60 km in this section. The existing detection scheme focusing on a single attack for performance evaluation comparison with the proposed scheme modeled the CUSUM detection algorithm [20] to detect attacks attempted in V2X communication, and modeled the detection algorithm of VeCure [18] to detect attacks attempted in the CAN bus.

Fig. 13 shows the detection rate for attacks occurring in V2X communication when the proposed scheme is compared with an existing scheme. In the experiment, the types of attacks attempted in V2X communication were blackhole, FDI, and Sybil attacks; Fig. 13 shows the number of attack detections and the attack rate when the event was executed 100 times. Early detection of blackhole attacks is difficult because a certain number of packets must be accumulated. Therefore, its detection rate is lower than the other attack detection rates. The proposed scheme shows an average detection rate of 80.3458% when an attack is attempted in V2X communication.

Fig. 14 shows the detection rate for attacks occurring in the CAN bus by comparing the proposed scheme with the existing scheme. The types of attacks in the CAN bus are flooding, spoofing, and replay attacks, and the number of attack detections is shown under the same conditions as V2X communication. The proposed scheme showed an average detection rate of 93.6217% when an attack was attempted in the CAN bus. Therefore, the proposed scheme shows an attack detection rate similar to that of the existing detection scheme focusing on a single attack.

Fig. 15 shows the number of attack detections when a complex attack is attempted. The proposed technique can detect complex attacks using a history trajectory. In the experiment, it was possible to detect a compound attack with a maximum probability of 83.1032%. Existing schemes cannot detect complex attacks for the reasons described in Figs. 6 and 7. Existing detection schemes that focus on a single attack detect an attack by analyzing one attack pattern. When these schemes are used in detecting a complex attack, while one attack is detected, another attack interferes with the detection algorithm. This situation makes it impossible to detect complex attacks with existing detection schemes. However, the proposed scheme has an improved detection rate compared to the existing scheme, and accidents can be prevented.

Fig. 16 shows the number of packets transmitted between the vehicle and the RSU over time. In the experiment, a single attack and complex attack occurred randomly. Attacks were attempted on autonomous vehicles at a 40% rate, and the total simulation time was 300; Fig. 16 shows the number of packets transmitted every 30 simulation units. Because the proposed scheme detects complex attacks, it transmits fewer unnecessary packets. Therefore, the proposed scheme requires fewer packet transmissions than the existing scheme.

Fig. 17 shows the number of packet drops per simulation time. The conditions are the same as in Fig. 16, and it can be seen that the

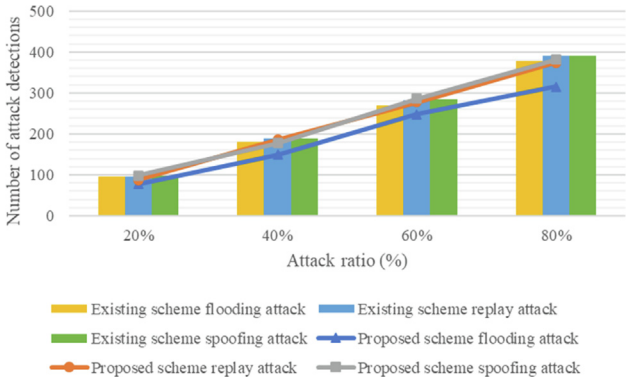


**Fig. 14.** Number of attack detections by attack rate (CAN bus).

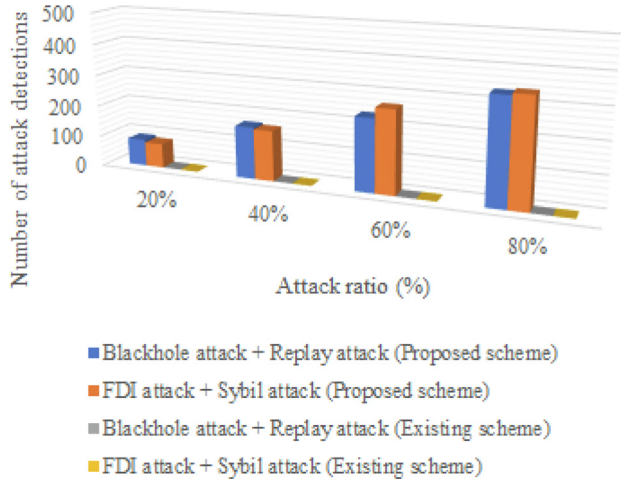


**Fig. 15.** Number of attack detections by attack rate (complex attack).

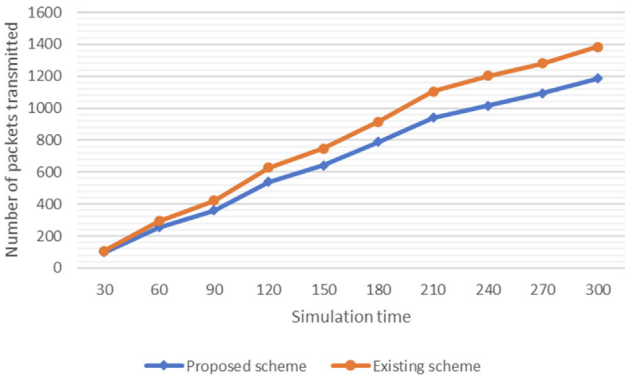


**Fig. 16.** Number of packet transmissions during the simulation time.

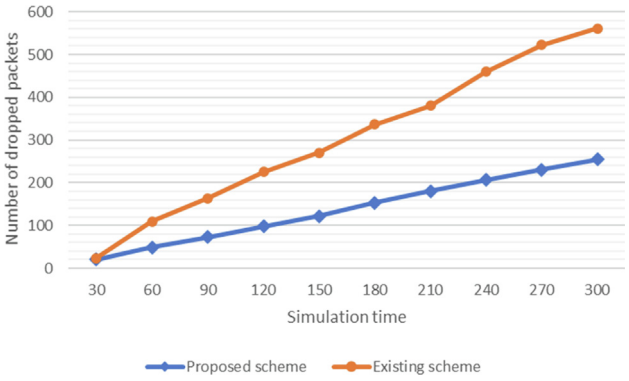


**Fig. 17.** Number of packet drops for simulation time.

number of packet drops increases when a complex attack, which cannot be detected using the existing scheme, is attempted. As shown in Figs. 16 and 17, the proposed scheme reduces the network overhead by detecting complex attacks. This helps drive efficiency by reducing the network communication overhead of autonomous vehicles and is effective in preventing traffic accidents. In the experiment, the proposed scheme reduced the number of packet transmissions and packet drops compared to the existing scheme; thus, it can be expected to improve communication efficiency over time.

## 6. Discussion

The main purpose of the proposed scheme is to prevent accidents by detecting complex attacks that occur in the IoV. It also reduces the network overhead by detecting and responding to attacks. The proposed scheme utilizes a history trajectory to detect complex attacks that existing schemes have not detected, thereby improving security. In addition, by reducing network overhead, packet drops due to network overload can be minimized, and traffic information can be collected in real-time. In addition, the complex attack detection scheme using the history trajectory in the IoV is expected to detect complex attacks that are more complex than the attack situation presented in this paper, if a significant amount of security knowledge is accumulated.

## 7. Conclusion and future research

The IoV collects information in real-time to plan an optimal route and communicate with road infrastructure or other autonomous vehicles to prevent accidents due to situations of which drivers are unaware, such as blind spots. Most attacks that occur in the IoV are detected through existing security schemes, but these schemes do not consider complex attacks. If damage from an attack accumulates, autonomous vehicles cannot drive normally, which can lead to traffic accidents. To reduce such damage, the proposed scheme uses history trajectories to detect complex attacks that cannot be detected by existing schemes. The proposed scheme can use trajectory analysis of objects to identify a damage situation in which normal packets cannot be delivered to the control center and thus detect a complex attack. The proposed scheme prevents traffic accidents by detecting complex attacks and allows real-time traffic information to be collected by reducing network overhead. However, the proposed scheme has difficulty managing a large amount of security knowledge, and as security knowledge increases, the detection time is delayed because various situations need to be identified.

To solve these problems, future research will study a temporal logic, rule-based security system that is applicable to the IoV. Because BM-DEVS [52] expresses security knowledge using temporal logic rules, it can easily express knowledge, and can cope with various attack situations by easily adjusting security knowledge. In addition, because the proposed scheme focuses on detecting complex attacks, it is necessary to take appropriate measures after the attack has been detected. Future research plans include additional investigations to prevent further incidents by using RG-DEVS [53] to detect attacks and respond effectively.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. NRF-2021R1A2C2005480).

## References

[1] Atzori L, Iera A, Morabito G. The internet of things: A survey. Comput Netw 2010;54(15):2787–805.
[2] Xia F, Yang LT, Wang L, Vinel A. Internet of things. Int J Commun Syst 2012;25 (9):1101.
[3] Cheng J, Cheng J, Zhou M, Liu F, Gao S, Liu C. Routing in internet of vehicles: A review. IEEE Trans Intell Transp Syst 2015;16(5):2339–52.
[4] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in 2014 IEEE world forum on internet of things (WF-IoT), 2014: IEEE, pp. 241- 246.
[5] Yang F, Wang S, Li J, Liu Z, Sun Q. An overview of internet of vehicles. China Commun 2014;11(10):1–15.
[6] Liu X. Airborne LiDAR for DEM generation: some critical issues. Prog Phys Geogr 2008;32(1):31–49.
[7] J. Levinson *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE intelligent vehicles symposium (IV)*, 2011: IEEE, pp. 163-168.
[8] Caesar H et al. nuscenes: A multimodal dataset for autonomous driving. In: in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. p. 11621–31.
[9] V. L. Thing and J. Wu, "Autonomous vehicle security: A taxonomy of attacks and defences," in *2016 ieee international conference on internet of things (ithings) and ieee green computing and communications (greencom) and ieee cyber, physical and social computing (cpscom) and ieee smart data (smartdata)*, 2016: IEEE, pp. 164-170.
[10] Zeigler BP. DEVS representation of dynamical systems: Event-based intelligent control. Proc IEEE 1989;77(1):72–80.
[11] Zeigler BP, Muzy A, Kofman E. Theory of modeling and simulation: discrete event & iterative system computational foundations. Academic press; 2018.
[12] S. Checkoway et al., "Comprehensive experimental analyses of automotive attack surfaces," in *USENIX Security Symposium*, 2011, vol. 4, no. 447-462: San Francisco, p. 2021.
[13] Yan C, Xu W, Liu J. Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. Def Con 2016;24(8):109.
[14] Koopman P, Wagner M. Autonomous vehicle safety: An interdisciplinary challenge. IEEE Intell Transp Syst Mag 2017;9(1):90–6.
[15] A. Ferdowsi, U. Challita, W. Saad, and N. B. Mandayam, "Robust deep reinforcement learning for security and safety in autonomous vehicle systems," in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018: IEEE, pp. 307-312.
[16] K. Koscher et al., "Experimental security analysis of a modern automobile," in *2010 IEEE Symposium on Security and Privacy*, 2010: IEEE, pp. 447-462.
[17] Amoozadeh M, Raghuramu A, Chuah C-N, Ghosal D, Zhang HM, Rowe J, et al. Security vulnerabilities of connected vehicle streams and their impact on cooperative driving. IEEE Commun Mag 2015;53(6):126–32.
[18] Q. Wang and S. Sawhney, "VeCure: A practical security framework to protect the CAN bus of vehicles," in 2014 International Conference on the Internet of Things (IOT), 2014: IEEE, pp. 13-18.
[19] Sakiz F, Sen S. A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV. Ad Hoc Netw 2017;61:33–50.
[20] Z. Jin and A. L. Bertozzi, "Environmental boundary tracking and estimation using multiple autonomous vehicles," in 2007 46th IEEE Conference on Decision and Control, 2007: IEEE, pp. 4918-4923.
[21] Indu SK. Internet of vehicles (IoV): evolution, architecture, security issues and trust aspects. Int J Recent Technol Eng 2019;7(6):2019.
[22] Pérez-Marín AM, Guillen M. Semi-autonomous vehicles: Usage-based data evidences of what could be expected from eliminating speed limit violations. Accid Anal Prev 2019;123:99–106.
[23] J. Z. Varghese and R. G. Boone, "Overview of autonomous vehicle sensors and systems," in *International Conference on Operations Excellence and Service Engineering*, 2015: sn, pp. 178-191.
[24] Lv C, Cao D, Zhao Y, Auger DJ, Sullman M, Wang H, et al. Analysis of autopilot disengagements occurring during autonomous vehicle testing. IEEE/CAA J Autom Sin 2018;5(1):58–68.
[25] Hafeez F, Sheikh UU, Alkhaldi N, Al Garni HZ, Arfeen ZA, Khalid SA. "Insights and strategies for an autonomous vehicle with a sensor fusion innovation: a fictional outlook," *IEEE*. Access 2020;8:135162–75.
[26] K. Min, S. Han, D. Lee, D. Choi, K. Sung, and J. Choi, "SAE Level 3 Autonomous Driving Technology of the ETRI," in 2019 International Conference on Information and Communication Technology Convergence (ICTC), 2019: IEEE, pp. 464-466.
[27] Badue C et al. Self-driving cars: A survey. Expert Syst Appl 2021;165:113816.
[28] F. Poggenhans et al., "Lanelet2: A high-definition map framework for the future of automated driving," in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018: IEEE, pp. 1672-1679.
[29] Wang J, Shao Y, Ge Y, Yu R. A survey of vehicle to everything (V2X) testing. Sensors 2019;19(2):334.

[30] Chen S, Hu J, Shi Y, Peng Y, Fang J, Zhao R, et al. Vehicle-to-everything (V2X) services supported by LTE-based systems and 5G. IEEE Communications Standards Magazine 2017;1(2):70–6.
[31] Campolo C, Molinaro A, Iera A, Menichella F. 5G network slicing for vehicle-to-everything services. IEEE Wirel Commun 2017;24(6):38–45.
[32] Xu X et al. Edge server quantification and placement for offloading social media services in industrial cognitive IoV. IEEE Trans Ind Inf 2020;17(4):2910–8.
[33] B. P. Zeigler, Y. Moon, D. Kim, and J. G. Kim, "DEVS-C++: A high performance modelling and simulation environment," in *Proceedings of HICSS-29: 29th Hawaii International Conference on System Sciences*, 1996, vol. 1: IEEE, pp. 350-359.
[34] C. Seo, B. P. Zeigler, R. Coop, and D. Kim, "DEVS modeling and simulation methodology with MS4 Me software tool," in *SpringSim (TMS-DEVS)*, 2013, p. 33.
[35] Zeigler BP, Sarjoughian HS. Introduction to devs modeling and simulation with java: Developing component-based simulation models. Technical Document: University of Arizona; 2003.
[36] L. Capocchi, J. F. Santucci, B. Poggi, and C. Nicolai, "DEVSimPy: A collaborative python software for modeling and simulation of DEVS systems," in 2011 IEEE 20th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2011: IEEE, pp. 170-175.
[37] Shringar Raw R, Kumar M, Singh N. Security challenges, issues and their solutions for VANET. International journal of network security & its applications 2013;5(5):95–105.
[38] Sharma S, Kaushik B. A survey on internet of vehicles: Applications, security issues & solutions. Veh Commun 2019;20:100182.
[39] Cui J, Liew LS, Sabaliauskaite G, Zhou F. A review on safety failures, security attacks, and available countermeasures for autonomous vehicles. Ad Hoc Netw 2019;90:101823.
[40] Hao Y, Cheng Y, Zhou C, Song W. A distributed key management framework with cooperative message authentication in VANETs. IEEE J Sel Areas Commun 2011;29(3):616–29.
[41] Azees M, Vijayakumar P, Deboarh LJ. EAAP: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks. IEEE Trans Intell Transp Syst 2017;18(9):2467–76.
[42] S. Park, B. Aslam, D. Turgut, and C. C. Zou, "Defense against sybil attack in vehicular ad hoc network based on roadside unit support," in MILCOM 2009-2009 IEEE Military Communications Conference, 2009: IEEE, pp. 1-7.
[43] Y. Fraiji, L. B. Azzouz, W. Trojet, and L. A. Saidane, "Cyber security issues of Internet of electric vehicles," in 2018 IEEE Wireless Communications and Networking Conference (WCNC), 2018: IEEE, pp. 1-6.
[44] Nanda A, Puthal D, Rodrigues JJ, Kozlov SA. Internet of autonomous vehicles communications security: overview, issues, and directions. IEEE Wirel Commun 2019;26(4):60–5.
[45] D. Anadu, C. Mushagalusa, N. Alsbou, and A. S. Abuabed, "Internet of Things: Vehicle collision detection and avoidance in a VANET environment," in 2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2018: IEEE, pp. 1-6.
[46] Ghosal A, Conti M. Security issues and challenges in V2X: A survey. Comput Netw 2020;169:107093.
[47] Miller C, Valasek C. Adventures in automotive networks and control units. Def Con 2013;21(260–264):15–31.
[48] Alnasser A, Sun H, Jiang J. Cyber security challenges and solutions for V2X communications: A survey. Comput Netw 2019;151:52–67.
[49] A. Kchaou, R. Abassi, and S. G. El Fatmi, "Towards a secured clustering mechanism for messages exchange in vanet," in 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2018: IEEE, pp. 88-93.
[50] Park HC, Lee WB, Kim TG. RASES: A database supported framework for structured model base management. Simul Pract Theory 1997;5(4):289–313.
[51] Zeigler BP. Hierarchical, modular discrete-event modelling in an object-oriented environment. Simulation 1987;49(5):219–30.
[52] Cho TH. Simulation Methodology-Based Context-Aware Architecture Design for Behavior Monitoring of Systems. Symmetry 2020;12(9):1568.
[53] Cho TH. Embedding intelligent planning capability to DEVS models by goal regression method. Simulation 2002;78(12):716–30.