This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3701285

A Dynamic Decision and Security Framework for Internet of Vehicles by Enhanced Localization Using Deep Reinforcement Learning

Arash Heidari*, Haoran Li, Roberto Passerone*, Nima Jafari Navimipour, Kyu In Lee*

Abstract—With the increasing complexity and interconnectivity of IoV, protecting in-vehicle networks from cyberattacks and controlling mobile dynamics have become increasingly difficult. Currently, most techniques rely on large amounts of labeled data, which are difficult to obtain and expensive to generate. Furthermore, they are not focused on real-time Intrusion Detection Systems (IDSs) and adaptive decision making. To address these issues, this paper proposes an Adaptive Decision Framework for IoV-IDS (ADFII). To ease a policy update in the ADFII, we use Deep Reinforcement Learning (DRL) as it can generalize the policy to different kinds of vehicles by updating them according to specific variables, and we use Variational Autoencoders (VAEs), which ease the detection of hidden features and generalizes anomalies and new attack patterns. This combination of cores ensures that ADFII is both stable and flexible, helping in the efficient detection of intrusions in IoV environments that are subject to property changes, making the decision-making process more secure and reliable. To learn to detect incursions, navigate, and locate, ADFII only has to learn the right rules for the environment. In tests, ADFII has 7.1%, 3.7%, 4.1%, and 5% improvement, respectively, for better decision-making, reward, faster time, and finding an attack than baseline algorithms.

Index Terms—Internet of Vehicles, Intrusion Detection System, Deep Reinforcement Learning, Variational Autoencoders, Anomaly Detection.

I. INTRODUCTION

New technologies, including the Internet of Vehicles (IoV) and Intelligent Transportation System (ITS) made transportation systems much better by making it easier for vehicles, roads, and other parts of the network to talk to each other [1]. Distributed systems like the Internet of Things (IoT), edge computing, and smart cities have a lot to do with IoV networks. Such a network allows people to share information in real time, control traffic, and make objects safer [2]. IoV apps have to figure out how to utilize all the sensory data that vehicles and roads give off, not already categorized [3]. To keep these connected systems secure against unauthorized access, malicious behavior, and cyberattacks, we need stronger, IoV-aware Intrusion Detection Systems (IDS) that can handle the scale and speed of the network [4].

Most of the time, basic Machine Learning (ML) and Deep Learning (DL) algorithms require a lot of data that has been labeled [5]. Getting this information is challenging and expensive, especially in IoV situations where people are continually on the move, objects are always changing, and security issues are growing [6-8]. Deep Reinforcement Learning (DRL) is a very useful method for making decisions and controlling things. It has worked well in several areas [9]. DRL combines DL's capacity to approximate hard functions with reinforcement learning's method of making choices to make policies better in circumstances with a lot of dimensions [10]. On the other hand, most of the DRL frameworks that are available now are only meant to be utilized in fully supervised contexts. This makes them less useful in IoV situations since there is not a lot of labeled data, and the system needs both labeled and unlabeled data to learn properly [11].

Even though there has been a great deal of development in IoV intrusion detection and mobility management, there are still numerous important problems that have not been solved. First, most current IDS frameworks depend significantly on properly labeled datasets, which are hard to find and expensive to get in dynamic vehicular situations where data changes quickly and labeling costs a lot of money. Second, DRL has shown promise in optimizing policies and making decisions, but it is frequently limited to completely supervised or fixed settings, which makes it less useful for IoV systems that are constantly changing and include a multitude of different types of devices. Third, many of the solutions we have now cannot be changed to fit different patterns of movement or make decisions based on the situation, particularly in cities where signal interference and poor localization are common. In the IoV field, there is still substantial work to be done on how to use unlabeled data with unsupervised or semisupervised learning. This gap shows how much we need a flexible, semisupervised framework that can learn from both labeled and unlabeled data, make decisions quickly, and

Arash Heidari is with the Communication and Information Research Center (CIRC), Sultan Qaboos University, Muscat, Oman (a.heidari@squ.edu.om). Arash Heidari, Haoran Li, and Kyu In Lee are with the Department of Information Science Technology, Cullen College of Engineering, University of Houston, Houston, Texas, USA (aheidari@central.uh.edu ,hli80@cougarnet.uh.edu, klee48@central.uh.edu). Arash Heidari and Roberto Passerone are with the Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy (arash.heidari@unitn.it, roberto.passerone@unitn.it, Arash_heidari@ieee.org), Nima Jafari Navimipour is with the Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Kadir Has University, Istanbul, Turkey; and the Research Center of High Technologies and Innovative Engineering, Western Caspian University, Baku, Azerbaijan. ***Corresponding Authors****: Arash Heidari, Roberto Passarone, and Kyu In Lee.*

stay strong in real-time vehicle situations. In this sense, our research presents a new adaptive decision framework made for IoV applications. The *Adaptive Decision Framework for the IoV Intrusion Detection System* (ADFII) employs both labeled and unlabeled data to make better decisions in situations when vehicles are moving about. The model employs Variational Autoencoders (VAEs) to guess missing labels and make the learning process better. This lets the agent apply the best rules to a wide range of situations. The ADFII method solves the issue of having too little labeled data by using the large amount of unlabeled data that IoV systems supply. It does this by extending the DRL paradigm to a semisupervised setting. One of the main things this study looks at is how to locate and regulate the movement of vehicles in cities adaptively. Accurate vehicle localization is critical for navigation, collision avoidance, and route optimization in IoV networks. Traditional localization technologies, like GPS or Wi-Fi-based systems, may not be very accurate or reliable, especially in cities with a lot of people and things that block signals. Our article shows that adding Bluetooth Low Energy (BLE) or 5G-based localization for vehicles to the ADFII model makes IoVs more accurate and better at making decisions that change over time. Here is a summary of the contributions to this publication:

- Putting forth a novel DRL framework that uses VAEs to describe data density statistically and integrates the best parts of DL to handle IoV-specific issues;
- By using IoV's large amount of unlabeled data, our system can combine labeled and unlabeled data to reduce its reliance on costly labeled sets;
- Testing ADFII in moving vehicles, improving localization, and making adaptive IoT policies easier to use;
- Validating in real-world IoV situations, monitoring track of changes in mobility, and boosting reward and localization metrics.

The next parts of this paper are set out in the following order: In Section II, we summarize the most recent findings that are connected to the study issue. Section III goes into further detail on the system model. Section IV also goes into great detail on the proposed ADFII framework. The paper shows the outcomes of the ADFII method and compares it to other methods in Section V. Section VI gives a summary and a conclusion.

II. RELATED WORK

In this section, we will look at key achievements in the field of IDS and security solutions for IoV environments. By assessing existing techniques, we highlight their basic ideas, benefits, and limits, providing support for our suggested framework. In this regard, Zhang, et al. [12] introduced an IDS based on Gaussian Random Incremental Principal Component Analysis (GRIPCA) and the optimal weighted extreme learning machine. GRIPCA reduced data redundancy, whereas DPSO optimized WELM parameters. Using the NSL-KDD and CIC-IDS-2017 datasets, their model obtained an accuracy of 91.02% and 94.67%, respectively, exceeding existing strategies. Also, Fan, et al. [13] presented a DL-based auto-updating IDS for automotive networks that can detect both known and new threats in open environments. Their solution used a two-layer filter and a cloud-edge collaboration model to lessen the computational load on automobiles. However, scalability and reliance on high-performance cloud infrastructure remain important constraints. Moreover, Wu, et al. [14] described a compact ML-based IDS that used an improved Fuzzy Rough Set (FRS) model for feature selection. Their method decreased processing costs and produced high-precision detection, but it struggled with real-time operation and large-scale installations. In the Internet of Medical Things (IoMT) sector, Alzubi, et al. [15] suggested a hybrid DL framework that combined Convolutional Neural Network (CNN) and Long Short-term Memory (LSTM) architectures to improve attack detection accuracy and efficiency. Plus, Zhou, et al. [16] proposed an architecture for detecting anomalies in IoV areas. Their approach used physical invariant qualities to find issues and figure out what they meant. But their methods need a lot of computing power, which makes it too slow for a lot of people to use. In another work, Abou El Houda, et al. [17] developed an edge-based method that combines blockchain and the Federated Learning (FL) method.

In short, the reviewed methods represent a lot of progress in fixing some of the challenges with IoV. But they often fail to meet well with the changing and dynamic needs of IoV systems. Some of these are the ability to scale in real time, make decisions that adapt to new information, and manage resources well. Also, a lot of current solutions use centralized architectures, which cause delays, inefficiencies, and single points of failure. Our ADFII combines DRL and VAEs to fill in these gaps. By using both labeled and unlabeled data, ADFII makes sure that decisions are made correctly, that localization is more accurate, and that mobility is better managed. Experimental findings reveal that the Quality of Services (QoS) characteristics are better than they were with older methodologies. The ADFII system is a scalable, secure, and real-time answer to the most important problems that IoV networks face, including cyber resilience and adaptive operations. Table 1 also displays other works that are related to IoV.

Table 1. Summary of examined works.

Researchers	Core Idea	Advantage	Challenges
Zhang, et al. [12]	Proposing an IDS for IoV using GRIPCA and OWELM.	Reduced data redundancy and high accuracy	Time-consuming detection
Fan, et al. [13]	Proposing an auto-updating IDS with cloud-edge collaboration.	Improved attack detection and reduced computation load.	Scalability and dependence on high-performance cloud infrastructure.
Wu, et al. [14]	Proposing a tiny ML-based IDS with FRS-based feature selection.	High-precision detection and reduced computational cost	Challenges in large-scale deployments and real-time operation.
Alzubi, et al. [15]	Proposing a CNN-LSTM blended framework for IoMT attack detection.	High accuracy and efficiency in real-time detection	Dependence on extensive labeled datasets.
Zhou, et al. [16]	Proposing a KVAE-based method with subspace extraction.	Robust detection and interpretability	High computational resource requirements.
Abou El Houda, et al. [17]	Proposing an edge-based FL and blockchain framework for ITS.	Ensures privacy, reliability, and scalability	Dependence on high-quality datasets for training.
Our work	Proposing the ADFII model using a new DRL and VAEs.	Improved accuracy, intrusion detection, and mobility management	-

III. SYSTEM MODEL

The ADFII system model is meant to reflect the dynamic and distributed nature of the IoV. Emphasizing their roles in localization, mobility management, and intrusion detection, it combines the interactions of automobiles, edge nodes, and cloud nodes. Including Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, BLE/5G-based localization, and an enhanced DRL framework, this model captures the complexity of IoV systems in the real world. The ADFII model lays a strong basis for adaptive decision-making and cybersecurity in-vehicle networks. The system model and its components are shown in Fig. 1.

A. Environments Settings

The IoV environment is represented as a graph $G = (V, E)$, where V is the set of nodes categorized into vehicles V_v, edge nodes V_e, and cloud nodes V_c, and E represents the communication links between these nodes. Such components form a hierarchical structure enabling efficient communication and computation. The vehicles in V_v act as agents, equipped with devices including sensors, communication modules, and processing capabilities, to collect real-time data such as position, speed, traffic conditions, and energy consumption. In our model, the state of a vehicle $v_i \in V_v$ at time t, denoted as $s_{v,t}$ is defined as $s_{v,t} = \{p_{v_i}, v_{v_i}, t_{v_i}, e_{v_i}, c_{v_i}, r_{v_i}\}$, where p_{v_i} is the vehicle current geographical position, v_{v_i} its instantaneous velocity, t_{v_i} indicates the local traffic conditions and situations, encoded as a numerical value extracted from vehicle density and average speed in the surrounding area, e_{v_i} indicates energy consumption metrics, c_{v_i} describes the connectivity status with edge nodes (servers) or other vehicles, and r_{v_i} captures recent sensor information and data, including BLE signals or proximity to nearby objects. Vehicles and the edge nodes share information so they can work together to do things like avoid accidents, change routes based on traffic, and acquire security updates. BLE or 5G signals can assist them in figuring out exactly where they are, which makes it easier for them to go around and find their way.

Edge nodes, including Roadside Units (RSUs) or local base stations, denoted by V_e, are particularly crucial connectors between cars and the cloud nodes. Their main duty is to enable real-time applications to deal with data with the least delay. They also do things like traffic analysis, intrusion detection, and make choices in a certain region. The state of an edge node $e_j \in V_e$ at time t is represented by the tuple $s_{e,t} = \{U_{e_j}, R_{e_j}, C_{e_j}\}$, where U_{e_j} indicates the available computing and memory resources, R_{e_j} is for real-time processing tasks like detecting problems or making localization better, and C_{e_j} denotes the bandwidth that is available for talking to automobiles and cloud nodes.

In our system model, cloud nodes, indicated by V_c, are centralized data centers which are very important to the IoV ecosystem since they manage global regulations and do a lot of analytics. Such nodes collect information from a number of edge nodes to get a better picture of the whole system and train DRL models, which improve the decision-making processes throughout the ADFII system. In this regard, we define the state of a cloud node, $c_k \in V_c$ as $s_{c,k} = \{M_{c_k}, D_{c_k}\}$, here M_{c_k} is several pretrained global models including DRL policies for localization and intrusion detection, and D_{c_k} is a set of historical datasets used to train, fine-tune, and improve these models. Cloud servers make sure that systems can grow by giving them the computing power they need to process the huge amount of IoV data and by keeping policy changes in sync with edge nodes. Cloud nodes make it easy for information to flow freely and ensure that edge nodes and vehicles use the most up-to-date and effective tactics. Cloud nodes also free up resources at the edge and vehicle levels by taking on jobs that need a lot of computing power. This makes the IoV infrastructure more efficient, secure, and strong.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3701285

With V2V, V2I, and edge-to-cloud interactions, the IoV technology allows cars to talk to each other. BLE or 5G signals help cars figure out where they are and convey information. V2V communication helps automobiles avoid crashes by letting them talk to each other. V2I communication connects cars to edge nodes so they can acquire real-time updates and be assigned tasks. Edge-to-cloud communication makes sure that edge nodes obtain information and regulations from all around the world that could help them improve their local operations.

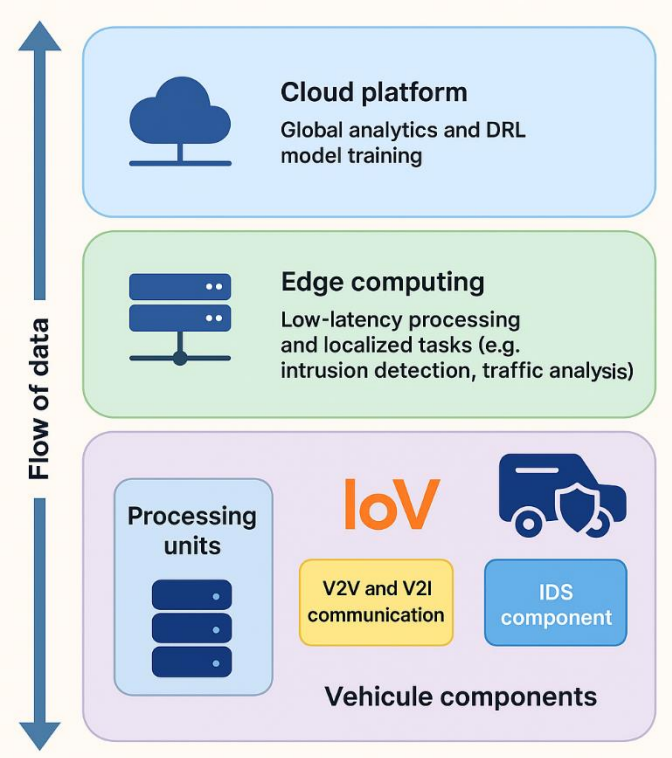

Fig. 1. System model and its components.

B. Markov Decision Process

A Markov Decision Process (MDP) defines how the IoV system makes decisions. This meets the needs for mobility, localization, and cybersecurity while also making IDS and mitigation methods better. Adding intrusion detection to the MDP framework helps keep the IoV ecosystem safer by stopping possible attacks before they happen. This will make cars work better. At any given time t, the system state space includes the status of all vehicles, edges, and cloud nodes, taking into account issues of mobility and security. It is defined as the combination:

$$c_t = s_{v_i,t} \cup s_{e_j,t} \cup s_{e_k} \quad (1)$$

Also, for each vehicle $v_i \in V_i$, the state $s_{v_i,t}$ is augmented with an alert set $alerts_{v_i}$, which saves security warnings received from edge nodes, including anomaly detections, abnormal traffic patterns, or spoofing signals. Such alerts allow us to adapt mobility and safety decisions based on local threats. Similarly, for each edge node $e_j \in V_e$, the state $s_{e_j,t} = \{U_{e_j}, R_{e_j}, C_{e_j}, threat\_level_{e_j}\}$ includes a new parameter $threat\_level_{e_j}$ representing the severity of detected intrusions based on anomaly scores and incoming data streams. So, it allows edge devices to adjust processing load, communication, and mitigation responses dynamically. Finally, cloud nodes $c_k \in V_c$ keep their state s_{c_k} which incorporates a global alert set $global\_alerts_{ck}$ that aggregates IDS updates from multiple edge nodes.

Also, vehicles v_i are responsible for responding proactively by avoiding hazards (a_{evade}), changing their speed or route based on edge alerts, and telling the nearest edge node about any suspicious activity (a_{report}). So, edge nodes, which are indicated by e_j are in charge of local security. They look for strange behavior in streams ($a_{analyze}$), send out alerts ($a_{broadcast}$), and separate suspect nodes ($a_{isolate}$). Also, in our system, cloud nodes c_k watch the system as a whole and they always update and share intrusion-detection patterns (a_{update}), and sync data (a_{sync}) from edges to retain a single, up-to-date perspective of security. At time t, the reward measures how well the ADFII can find, fix, and improve things:

$$r(s_t, a_t, s_{t+1}) = w_1 . r_{security} + w_2 . r_{locacalization} + w_3 . r_{mobility} \quad (2)$$

Here, $r_{security}$ shows how well ADFII acts, focusing on accurate detection, fewer false positives, and stopping system breaches. The $r_{locacalization}$ part rewards keeping the car in the right place, which is important for safety and navigation. Lastly, $r_{mobility}$ looks at how well the system works by measuring how well traffic flows and how much time is saved. Also, we can change the weights w_1, w_2 and w_3 to change how important security, localization, and mobility goals are to each other. Our system can prioritize certain goals by adjusting these weights, based on the situation and operational needs. Such aims may be strengthening cybersecurity, optimizing location, or making vehicles move better. The incentive structure is flexible, so the IoV can change as needed while still accomplishing its basic goals. The state-transition probability shows how probable it is that the system will change from s_t to s_{t+1}, taking into account activities and outside circumstances, including vehicle mobility, traffic, and changing patterns of incursion. In a formal way, as shown in Eq. 3:

$$P(s_{t+1} | s_t, a_t) = f(mobility, traffic, actions, intrusions) \quad (3)$$

This formulation considers both known and unknown breaches, demonstrating how security incidents might alter system behavior. For instance, an attack may change the route of cars, make edge nodes isolate weak parts, or cause rule updates at the cloud level. These factors work together to create a full model of how the system changes over time by affecting the transition probabilities. The goal is to discover the best policy π^$*$ that balances security, localization, and mobility while getting the most long-term returns (Eq. 4):

$$\pi^* = \arg\max \mathbb{E}\ [\sum_{t=0}^{\infty} \gamma^t r\ (s_t\ , a_t\ , s_{t+1}\ )] \qquad (4)$$

Here, γ is the discount factor. It gives more weight to acts that can be taken now than to future rewards. This balance helps the system take care of short-term demands, like responding to intrusions, without losing sight of long-term goals, like keeping traffic flowing and making sure everyone is secure. Adding γ makes it easier to make decisions while changing IoV values.

IV. ADFII METHOD

This section goes into much detail about the ADFII approach. Fig. 2 shows how the ADFII framework works for detecting intrusions in IoV environments. It depicts how a VAE processes sensory data from cars, which lets the DRL agent learn the best actions based on both labeled and unlabeled data. The system talks to edge and cloud nodes to make real-time decisions, including sending out alarms for intrusions, improving localization, and rerouting. Our Deep Q-Network (DQN) method uses the power of Deep Neural Networks (DNNs) to get close to the best action-value function $Q^*(s,a)$, which shows the highest total reward that can be obtained from a certain condition s and action a. This is represented in Eq. (5).

$$Q^* = \max \mathbb{E}\ [r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t = s, a_t = a, \pi] \qquad (5)$$

In our method, γ is the discount factor that prioritizes immediate rewards while still considering future gains, and π is the policy determining the action selection strategy. The Bellman equation changes this function by breaking it down into immediate rewards and future values. This makes the optimization process easier to understand. The Bellman equation reformulates this function, breaking it into smaller steps by defining $Q^*(s,a)$ recursively as shown in Eq. (6).

$$Q^*(s,a) = \mathbb{E}\ [r + \gamma \max Q^*(s',a') | s,a] \qquad (6)$$

Here, s' and a' denote the subsequent state and action, respectively. The Bellman equation enables the iterative estimation of $Q^*(s,a)$ by focusing on immediate rewards r and the discounted future rewards $\gamma \max Q^*(s',a')$. In practice, $Q^*(s,a)$ is approximated using a deep neural network parameterized by θ, where θ denotes the set of neural network weights and biases, such that $Q(s,a;\ \theta) \approx Q^*(s,a)$. The network is trained by minimizing a loss function, which captures the difference between the predicted Q-value and the target value computed from the Bellman equation, as shown in Eq. (7). Experience replay is an important part of the DQN method. It saves prior experiences, $e_t = (s_t, a_t, r_t, s_{t+1})$, in a replay buffer $D_t = \{e_1, \ldots, e_t\}$.

$$L_i(L_i) = \mathbb{E}_{(s,a,r,s')\sim \mathrm{U(D)}}\ [(r + \gamma \max Q^*(s',a';\theta_{i-1}) - Q(s,a;\theta_i))^2] \qquad (7)$$

In this regard, the replay buffer, which is indicated by D plays an important role in the training process by storing the agent's past experiences as tuples, (s,a,r,s'), where s represents the current state, a the action taken, r the reward received, and s' the resulting state. To ensure that the training data remains uncorrelated and diverse, uniform sampling from the buffer $U(D)$ is employed, which helps mitigate issues such as overfitting to specific sequences of experiences. The network parameters at successive iterations are denoted as θ_{i-1} and θ_i, representing the weights before and after an update, respectively. So, the gradient of the loss function is calculated as below:

$$\nabla_\theta L_i(\theta_i) = E_{(s,a,r,s')\sim \mathrm{U(D)}}\ [(r + \gamma \max Q(s',a';\theta_{i-1}) - Q(s,a;\theta_i)) \nabla_\theta Q(s,a,\theta)] \qquad (8)$$

A. VAE in IoV-IDS

Our solution connects supervised tasks to IoV's massive, unlabeled data. The ADFII strategy improves model generalization for dynamic and high-dimensional IoV systems by using labeled and unlabeled observations. In this context, labeled data, $X_l = (x_1, \ldots, x_l)$ with corresponding labels $Y_l = (y_1, \ldots, y_l)$, is augmented with unlabeled data $X_u = (x_{l+1}, \ldots, x_{l+u})$. Semisupervised learning relies on three fundamental assumptions, including smoothness, the cluster assumption, and the manifold assumption.

In the proposed ADFII, a VAE-based deep generative model is utilized for semisupervised learning. VAEs enhance the representation of high-dimensional data by learning latent variable distributions that generalize both labeled and unlabeled observations. For each data point

x_i, the latent variable z_i is modeled as a Gaussian, $p(z) = N(z|0, I)$, where 0 denotes the zero mean vector and I is the identity covariance matrix, ensuring a flexible and compact latent representation. Also, the generative model extends the traditional VAE by incorporating a latent class variable y, alongside the latent variable z. The generative process is defined as follows:

$$P(y) = Cat(y|\pi), \quad p(z) = N(z|0, I), \quad P_0(x|y, z) = f(x, u, z, \theta) \tag{9}$$

Here, $Cat(y|\pi)$ represents a categorical distribution with probabilities π (not to be confused with the policy π introduced earlier in the DQN formulation), and $f(x, u, z, \theta)$, is a nonlinear function parameterized by θ where θ denotes the set of neural network weights. The generative process is defined as:

$$q_\phi(z|x) = N\left(Z|\mu_\phi(x), diag(\sigma_\phi^2(x))\right), q_\phi(y|x) = cat\left(y|\pi_\phi(x)\right) \tag{10}$$

Here, $\mu_\phi(x)$ and $\sigma_\phi^2(x)$ represent the mean and standard deviation vectors of the Gaussian latent variable, and $\pi_\phi(x)$ is the output probability vector for the latent class variable. The objective function balances the generative and discriminative aspects of the model. For labeled data, the Evidence Lower Bound (ELBO) is shown as Eq. (11).

$$logp_\theta(x, y) \geq E_{q_\phi(z|c,y)}[logp_\theta(x, y, z)] - KL[q_\phi(z|x, y)||p(z)] \tag{11}$$

For unlabeled data, y is treated as a latent variable, and the objective becomes as shown in Eq. (12).

$$logp_\theta(x) \geq E_{q_{\phi(y,z|x)}}[logp_\theta(x|y, z)] - KL[q_\phi(y, z|x)||p(y)p(z)] \tag{12}$$

The combined objective over both labeled and unlabeled datasets is defined as their sum, shown in Eq. (13).

$$J = \Sigma L(x, y) + \Sigma U(x) \tag{13}$$

Additionally, a classification loss term is added for labeled data to improve discriminative performance, as shown in Eq. (14).

$$J_\alpha = J + \alpha . E[-logq_\phi(y|x)] \tag{14}$$

So, α controls the trade-off between generative and discriminative learning in our system. In the IoV environment, the ADFII's VAE component evaluates real-time data streams from vehicles, edge nodes, and the cloud to discover anomalies and improve system-wide rules. The semisupervised framework achieves substantial generalization in intrusion detection and response by combining labeled intrusion events with unlabeled data (for example, unexpected traffic or irregular connection patterns). This integration assures that the IoV system can constantly react to changing threats while maintaining efficiency in localization and mobility, making ADFII a reliable and scalable solution for next-generation IoV networks.

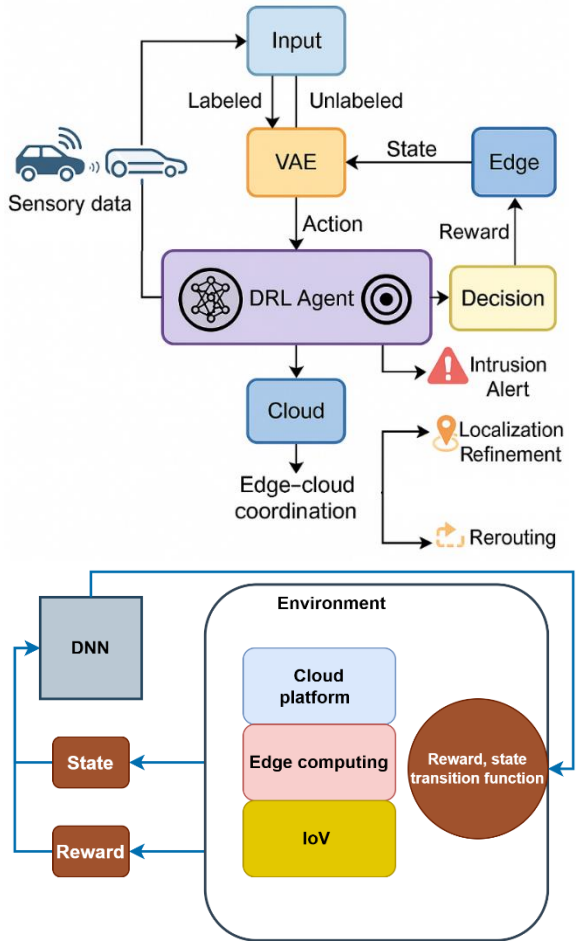

Fig. 2. The complete flow of the proposed ADFII framework, illustrating the integration of the DQN decision module and the VAE-based feature extractor within the adaptive intrusion identification pipeline.

B. Integration of VAE and DQN

The ADFII framework solves the problem of IoV environments being dynamic and only partly labeled by closely integrating VAE with DQN. This allows for adaptive, real-time intrusion detection and decision-making. Instead of using these parts separately, ADFII uses the VAE to turn raw vehicle and network-level input features, like BLE/5G RSSI signals, vehicle speed, connectivity metrics, and communication anomalies, into latent embeddings $z \sim q(z \mid x)$ and inferred class labels

$y \sim q(y \mid x)$. The improved state vector $st = [z_t \| y_t \| c_t]$ is made by adding these outputs to contextual information like traffic density and local resource status. This is what the DQN agent uses as input. The Q-network then uses this semantically rich state to learn the best rules. This helps the system grow closer to the action-value function $Q(s_t, a_t)$, which enables it to choose actions like rerouting, reporting problems, or isolating cars that are thought to be involved in a crime. The DQN's reward function has several weights for things like how accurate the detection is, how many false positives there are, how precise the localization is, and how long it takes to react. This proves that IoV systems need to achieve a lot of things at once. There is a combination of training times at ADFII. At first, the VAE is trained offline using a mix of ELBO and classification loss. Then, its encoder is fixed (or gently fine-tuned) while the DQN is trained online with experience replay. This design makes sure that the DRL agent can use the VAE's unsupervised feature generalization, particularly when there are not enough labels, while still being able to respond and adapt in real time.

C. Use Case: IoV IDS and Localization

ADFII leverages DRL to merge a small quantity of labeled threat data with a lot of real-time unlabelled data from autos and infrastructure. The reward function employs security criteria to find the best ways to isolate problematic nodes and provide warnings fast and correctly. To keep cars steady while they are driving, IoV systems require localization. The suggested architecture uses 5G-based localization algorithms to accurately locate things in busy areas where GPS does not operate.

$$RSSI = -1(10n)log_{10}(d) + A \qquad (15)$$

where d is the transmitter–receiver distance, n denotes the path-loss exponent, and A is the RSSI at a reference distance. Also, adding data without labels enables the model to change when the weather is poor or there is a lot of traffic. Also, Geolocation and other vehicle data tell us what every road section is. Vehicles change the environment by moving, interacting, and reacting to outside factors like traffic density or detected intrusions. The agent is the algorithm that informs the car's onboard system what to perform. It can interact with its environment by using sensor data to figure out where to go, how to move, and what threats are likely to be there. The IoV framework enables agents walk about freely, find things exactly where they are, and make everything more secure. The agent's state is a tuple containing real-time sensor data like BLE/5G signal strengths, traffic flow information, and connectivity metrics; the vehicle's geospatial coordinates for precise positioning; threat context in the form of edge node alerts or anomaly indicators; and, for labeled data scenarios, the proximity to the target destination to guide navigation. Stage tasks include relaying agent reports about possible threats to edge nodes, changing speed and heading to find ideal routes, using real-time data from relevant nodes to select a path, and threat avoidance maneuvers that enable the agent to evade a high-threat area. A reward function checks how successfully these tasks are done by delivering rewards for excellent results, such accurate localization, smooth traffic navigation, and effective security threat mitigation, and punishing bad results, like veering off course or ignoring known intrusions. The reward function uses weights (w_1, w_2, w_3) to establish a good balance between security, mobility, and location. Algorithm 1 tells this system how to work.

#	***Algorithm 1: ADFII Framework***
1	***Input:*** *IoV states (vehicles, edge nodes, cloud nodes). Communication data (V2V, V2I, edge-to-cloud interactions). Pretrained models and hyperparameters.*
2	***Output:*** *Real-time decisions for security, optimization, updated policies, and models to enhance IoV system performance*
3	*Initialize system components:* *- Load pretrained DQN and VAE models.* *- Initialize replay buffer* *- Define state space and reward weights.*
4	***While the IoV System True is running***
5	***For each vehicle, do***
6	*Collect real-time data (e.g., position, speed, BLE/5G).*
7	*Perform actions: threat evasion, anomaly reporting, or localization.*
8	*Output: Vehicle updates position, reports anomalies, or refines localization.*
9	*Receive rewards based on security, localization, and mobility.*
10	***For each edge node:***
11	*Input data streams from vehicles.*
12	*Perform actions: analyze threats, broadcast alerts, or isolate compromised nodes.*
13	*Output: Processed data sent to the cloud, alerts broadcast, or compromised nodes isolated.*
14	*Observe system status*
15	***For* every IDS decision**
16	*Store experience in the replay buffer*
17	*Sample a mini-batch and train DQN using collected experiences.*
18	*Use VAE for anomaly detection and policy refinement.*
19	*Transition to the next stage*
20	***End for***
21	*Optimized decision-making policies for vehicles, edge nodes, and cloud nodes.*
22	*Improved intrusion detection, localization, and mobility across the IoV system.*
23	*Optimized decision-making policies for vehicles, edge nodes, and cloud nodes.*
24	***End while***

As an example, we describe a real-world scenario of how ADFII works in a vehicle context by simulating a 5 × 5 km² urban grid with 20 roadside edge nodes and 100 linked automobiles. Each vehicle sends telemetry data (such as its location, speed, BLE/5G signal strength, and connection metrics) to adjacent edge nodes and other cars periodically via V2V/V2I communication (with updates every 5 seconds in our simulation). At simulation time t = 135s, a bad node starts a location spoofing attack by sending fake GPS coordinates and RSSI values that do not match the predicted signal propagation model. When the VAE module gets the vehicle's sensory stream, it encodes it into a latent representation, $z_t \in R^{32}$ and a class probability vector $y_t \in \Delta^2$. This shows that the data is quite different from what it has learnt before. The semisupervised VAE tags an anomaly when the classification entropy is greater than 0.85 and the reconstruction loss is more than 1.7 times the mean during the past 10 seconds. We empirically adjusted the thresholds via initial experiments. We chose the classification entropy threshold (0.85) and the reconstruction loss threshold (1.7× mean) because they gave us the optimal balance between sensitivity and specificity, which is in line with what other VAE-based anomaly detection investigations have found. This latent anomaly encoding is then concatenated with context features such as local traffic density (ρ = 45 vehicles/km), edge buffer load (71%), and recent incident reports to form the state vector, s_t, used by the DQN agent. These statistics (ρ = 45 vehicles/km, load = 71%, Q-value = 0.936) are samples taken from the simulation setting. They are merely meant to help explain how the proposed framework makes decisions. The agent asks its learnt Q-network for the action a_t = {broadcast alert + isolate node} which has the greatest projected Q-value $Q(s_t, a_t)$ = 0.936. There is a 48 ms wait before this action is sent to the closest edge server. As a consequence, the suspect node is cut off from the local mesh, and the edge node's dissemination module sends dynamic rerouting instructions to nearby vehicles. The reward function has three parts: a detection reward of r_{detect} = +5.0, a localization improvement bonus of r_{loc} = +1.4, and a delay penalty of r_{delay} = −0.7. This means that at this point, the net reward went up by +5.7. During this time, the agent's total reward goes up from 373.2 to 378.9. The edge-assisted BLE revalidation phase lowers the localization error from 8.3m to 4.6m after the event. Because both VAE uncertainty and DRL Q-values influence the choice to go from anomaly to alert, the false positive rate stays below 4%. This example shows how ADFII combines unsupervised anomaly encoding with deep reinforcement learning to find and stop intrusions in real time, balancing threat accuracy, reaction speed, and routing stability in a very dynamic and only partly visible IoV network.

We merged both generic IDS datasets (CICIDS2017, UNSW-NB15, TON_IoT, and CSE-CIC-IDS2018) and IoV-specific datasets (VeReMi, CARLA-VANET, AWID-VANET, and V2X-Sim) to make sure that all of the datasets work together for this use case [18-20]. The first group gives us a lot of different types of network traffic that we need to train and test the anomaly detection and DRL modules. The second group gives us telemetry data that is related to vehicle location and behavior, such as vehicle position, speed, heading, and RSSI signals. These datasets provide a complementary perspective, encompassing both communication abnormalities and spatiotemporal motion dynamics, so guaranteeing that the ADFII framework is assessed using adequate and varied data sources. To keep traffic and mobility data in sync across time, all flow-based events and vehicle state vectors were synced using a fixed 1-second sliding window (Δt = 1 s). V2X safety beacons are typically broadcast at 10 Hz (every 100 ms). We use a synchronization window of 1 second since ADFII is a joint mobility + network-flow state representation. Network-side ADFII features (e.g., flow counts, byte/packet rates, burstiness, and short-term entropy indicators) have greater temporal stability and discriminative power when computed in sub-second to second intervals rather than per beacon. The DRL decision cycle (state ↦ action ↦ reward) is designed to correspond to an edge-level control-loop that is strong to transient beacon noise, short-lived RSSI fluctuations, and intermittent packet reordering. The practical 1-second bins are created by averaging the 10 beacon samples (such as mean/standard deviation, last-value, and short-horizon deltas), and aggregating the communication events to strong flow descriptors. This allows the flow director to respond to the state signal every second, thereby stabilizing the state signal the DQN policy utilizes and diminishing the variance of the reward signal the director estimates. Table 2 gives a real-life example of how 10 Hz mobility beacons and network-flow events are combined into synchronized 1-second bins (Δt = 1 s) to make the multimodal state features that ADFII uses.

In our experiments, we used the CARLA driving simulator to generate vehicle trajectories and localization signals for datasets that did not have labeled mobility traces of vehicles (CICIDS2017, UNSW-NB15, CSE-CIC-IDS2018), keeping the temporal order of the network-flow events. On the other hand, for datasets that already had synchronized timestamps and mobility/telemetry signals (TON_IoT, VeReMi, V2X-Sim), we used the temporal relationship of the traffic and localization signals as they were. Also, we evaluate ADFII on the CARLA-VANET dataset, which is a stand-

alone IoV benchmark dataset, in Section V, to see how ADFII performs in high-mobility vehicular scenarios and on attacks based on sensors and locations.

Table 2. Example of 1-second aggregation (Δt = 1 s) over a 3-second interval.

Window (s)	Mobility beacons (10 Hz)	Aggregated mobility features (example)	Aggregated flow features (example)
[0,1)	10 samples	mean speed = 13.8 m/s; std = 0.6; Δpos = 13.5 m; mean RSSI = −64 dBm	packets/s = 420; bytes/s = 310 kB; burst ratio = 0.18
[1,2)	10 samples	mean speed = 14.2 m/s; std = 0.5; Δpos = 14.1 m; mean RSSI = −66 dBm	packets/s = 690; bytes/s = 520 kB; burst ratio = 0.41
[2,3)	10 samples	mean speed = 12.9 m/s; std = 0.9; Δpos = 12.6 m; mean RSSI = −63 dBm	packets/s = 460; bytes/s = 340 kB; burst ratio = 0.22

V. Results and Comparisons

In this part, we are going to have a detailed analysis of the ADFII framework using simulations, dataset analyses, performance metrics, and side-by-side studies.

A. Simulation Environments

We use TensorFlow/PyTorch (GPU-accelerated) with Python for DRL to run our simulation. NetworkX makes the IoV graph, while SUMO makes traffic that seems real. We utilize Matplotlib and Seaborn to keep an eye on things like how accurate our localization is, how efficient our mobility is, and how often we catch intruders. Scikit-learn and OpenCV handle preparing data and sending sensor data in real time. Flask and Django make it easy for mobile, edge, and cloud nodes to talk to each other by giving them RESTful APIs. The solution can be used on multi-core CPUs and NVIDIA GPUs, and it can be added to more cars/nodes. This lets us stress-test the framework in conditions that are similar to those we could find in the real world. This simulation setup shows a complete and flexible IoV ecosystem where automobiles, edge nodes, and cloud servers all work together in real time. In a multi-layer topology, each part is virtually deployed. The bottom layer consists of vehicles that send continuous data streams, the middle layer consists of edge nodes that do localized processing and decision-making, and the top layer consists of cloud servers that manage global learning and policy synchronization. RESTful APIs and socket-based data channels allow these layers to talk to each other, just as genuine V2V, V2I, and edge-to-cloud exchanges. The simulation workflow starts with SUMO generating vehicle telemetry. Then, OpenCV and Scikit-learn modules preprocess the data in real time, and TensorFlow/PyTorch models make predictions in GPU-accelerated environments. Matplotlib and Seaborn dashboards show the output, which includes intrusion alarms, localization accuracy, and mobility data. This integrated and layered setup closely resembles the structure of real-world IoV infrastructures, guaranteeing that both the communication and computational components of ADFII are accurately represented.

Also, the results of the simulations are compared to (1) MOID [21], (2) FLIDS [22], (3) DLIDS [23], and (4) SADNN [24]. This shows that ADFII can improve security, lower latency, and isolate attacks, even when there is pressing demand or when the situation is hostile.

B. Dataset

A broad array of datasets, recognized in IoT and IoV, is employed to evaluate the ADFII architecture. We chose these datasets because they include both current and future security risks in the IoV ecosystem. We may test how well the framework works to find and stop different security risks in IoV environments by combining these datasets. CICIDS2017, UNSW-NB15, TON_IoT, VeReMi, and CSE-CIC-IDS2018 are the most realistic, feature-rich, and relevant datasets for vehicular and IoT systems [25-29].

To maintain rigor and clarity in the domain, we classify the datasets into two primary categories: (i) generic IDS datasets, such as CICIDS2017, CSE-CIC-IDS2018, and UNSW-NB15, which function as baseline corpora for heterogeneous traffic modeling and pretraining; and (ii) IoV-specific datasets, including TON_IoT and VeReMi, which directly represent vehicular communication and misconduct patterns. This classification helps us maintain a balance between diversity and relevance in the many operational contexts of the ADFII system.

The reason for using CICIDS2017 and CSE-CIC-IDS2018 is that they cover a lot of modern attack types, like DDoS, brute force attacks, and attempts to get in. There are more than 80 features, such as packet flows, source and destination IP addresses, and time-series trends. These datasets are from simulated environments that act like real network traffic. The datasets are especially helpful for testing how well the IDS framework can discover faults in IoV situations with a lot of traffic, where the connection between automobiles and infrastructure needs to be safe. These datasets are not specific to IoV, but they are utilized as communication-layer analogs of vehicular networks. This lets the model learn generic patterns of traffic bursts, latency changes, and flooding behaviors that can happen in V2I or RSU-based infrastructures.

The UNSW-NB15 collection also provides exact packet-level data and adds new types of attacks to existing

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3701285

datasets, like fuzzing and SQL injection. This lets us test the IDS against more complicated attacks on IoV networks, such as attacks that inject bad data into edge or cloud nodes. In our research, UNSW-NB15 aids in diversity and negative transfer evaluation, guaranteeing that the acquired latent features are not excessively tailored to certain attack distributions.

We used TON_IoT and VeReMi to fix difficulties that were unique to IoV. By merging data from IoT devices with data from the network, TON_IoT helps the IDS detect more problems. This links data from car sensors to events that happen on the network. This dataset makes it easy to test edge-based IDS sections that find intrusions in real time. To make the link between generic IDS datasets and automotive settings even stronger, TON_IoT acts as a middleman where device telemetry, service logs, and network flows are all looked at together, just like how edge-cloud interactions happen in IoV scenarios.

On the other hand, the VeReMi dataset for the automobile network shows bogus data injection, spoofing, and Sybil attacks. VeReMi checks for cars that supply false information on purpose to mess up traffic or routes. This dataset is focused on IoV hostile node behavior, which makes it easier for the IDS to locate and isolate hacked nodes. These numbers let us see how the IDS operates in a lot of different IoV circumstances, indicating that it can protect big, evolving vehicle networks. VeReMi is the only dataset with clear vehicular ground truth labels, so it is used to test ADFII's ability to detect bad conduct and see if it can apply what it learned from generic datasets to genuine vehicular communication situations.

C. Performance Analysis and Evaluation

This section evaluates the performance of the proposed ADFII framework through a series of comparative experiments across multiple metrics and datasets. The evaluation focuses on cumulative reward, precision, accuracy, False Positive Rate (FPR), Threat Detection Rate (TDR), detection latency, F1 score, and scalability.

- **Reward Convergence Over Epochs**

In this section, we examine the efficacy of the ADFII framework across several contexts. Fig. 3 demonstrates that the total reward keeps going up throughout the course of 100 training epochs. It is about 450 by epoch 50 and about 504 by epoch 100. This upward trend indicates that the system may get better at finding intrusions over time by employing DRL. The agent gets a small reward at first, but it keeps getting better as it trains and plays back experiences offline. This helps it work well with a wide range of vehicle conditions and risk patterns. The consistent and smooth growth of rewards suggests that the learning process is stable and that ADFII can identify the optimum rules. This makes ADFII a strong and adaptable choice for real-time cybersecurity in the future generation of IoT. We found that the overall reward kept steadily rising during training and hit a plateau around Epoch 50, when it was roughly 450. The addition of other epochs would probably increase the achievement, but much less than one per cent. On the other hand, overfitting would also increase, and the calculus would be heavier. We decided that Epoch=50 was a good compromise between speed and convergence. The validation curve indicates that the chosen model is optimal in the sense that it reaches diminishing returns.

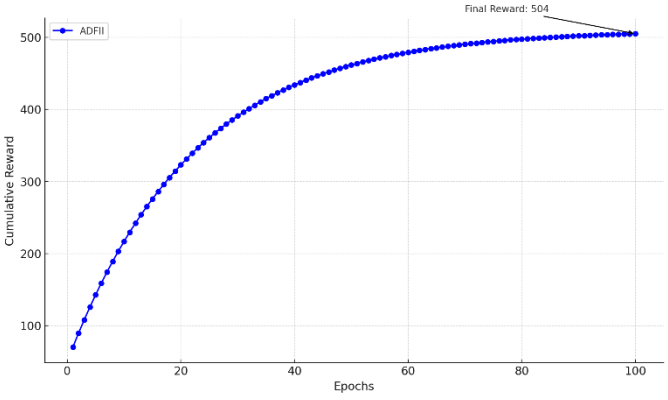

Fig. 3. Reward optimization over epochs for ADFII.

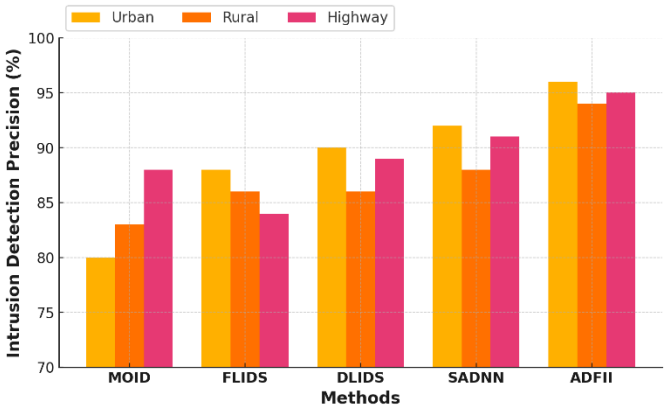

Fig.4. Intrusion detection precision across scenarios for the CICIDS2017, CSE-CIC-IDS2018, TON_IoT, and VeReMi datasets.

- **Precision Across Scenarios**

Fig. 4 compares the precision of intrusion detection achieved by ADFII to MOID, FLIDS, DLIDS, and SADNN in urban, rural, and highway settings. The ADFII framework routinely outperforms rival methods, with precision rates of 96% in urban, 94% in rural, and 95% in highway contexts, well surpassing the next best-performing method, SADNN. This higher performance is credited to ADFII's novel integration of DRL for adaptive

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3701285

decision-making, which dynamically optimizes policies in real-time, as well as its usage of VAEs to refine anomaly detection using unlabeled data efficiently. Our method keeps good performance in urban areas with more interference and data complexity by using BLE and 5G signals. The reason is its distributed architecture, which mixes edge and cloud computing, providing precise and efficient intrusion detection in rural and highway settings despite different mobility and connectivity constraints.

- **Accuracy Across Datasets**

As shown in Fig. 5, we can observe that ADFII outperforms other approaches in terms of accuracy on all datasets. It achieves 91% accuracy on VeReMi and 97% accuracy on CICIDS 2017. ADFII outperforms state-of-the-art methods due to its strong architecture, combining DRL to find the best policies and VAEs to exploit the maximum potential of the available labeled and unlabeled data. While it works well in these datasets, such as CICIDS2017 and CICIDS 2018 (CSE-CIC-IDS2018), with attacks like DDoS, penetration attacks, and more, ADFII also works well on more complex datasets, like VeReMi, which tests the improper behavior of automobiles, even when the amount of infiltration patterns is very low. Two other competing techniques are SADNN and DLIDS. These have a steady, though lower, accuracy level since they are supervised techniques that do not change according to the datasets. The final result below shows how well each of the datasets did, with the ADFII always being the best.

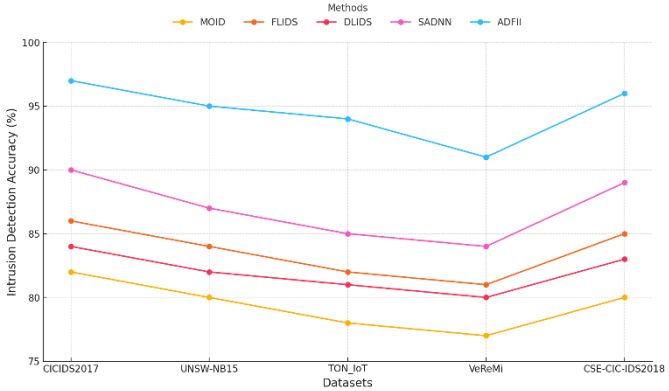

Fig. 5. Intrusion detection accuracy across datasets.

- **False Positive Rate**

The best FPR for each method and dataset is shown in Fig. 6, and ADFII always has the best FPR with a minimum of 3% and a maximum of 6% on the CICIDS2017 and VeReMi datasets, respectively. This shows that the model minimizes false positives while retaining true positives. VAEs are essential for the model to effectively learn the distributions of data and identify benign anomalies from real threats. This prevents false alarms. Since DRL always enforces the best decision policy, it can deal with multiple datasets and the real-time nature of the IoV scenario quite well. While other proposals such as SADNN and DLIDS have also shown improvements under certain circumstances, they suffer from a higher false positive rate because their supervised models were trained on only one dataset each, and do not generalize well to other datasets: the second best performing proposal, SADNN, has a 6% and 9% false positive rate on CICIDS2017 and VeReMi datasets, respectively. Other methods, such as MOID and FLIDS, also have a false positive rate of 12% to 18%, and do not reduce the number of false alarms.

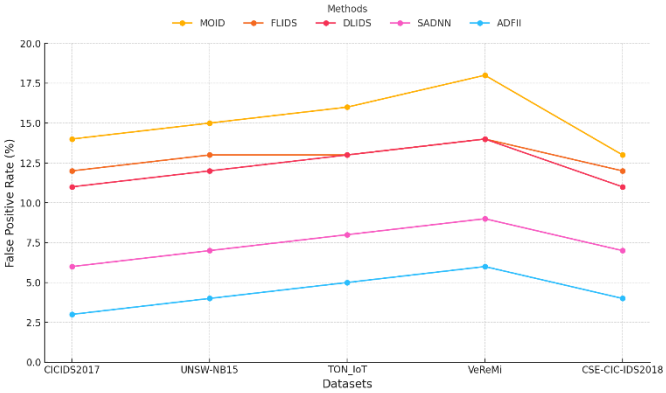

Fig. 6. FPR across datasets.

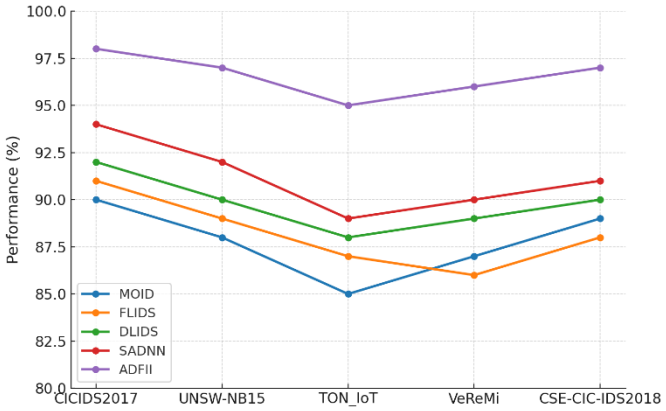

Fig. 7. TDR across datasets.

- **Threat Detection Rate**

Fig. 7 shows a comparison of the Threat Detection Rates (TDR) of different approaches across several datasets. In this regard, ADFII consistently surpasses all other approaches, achieving the greatest TDR across all datasets, ranging from 92% on VeReMi to 98% on CICIDS2017. The result illustrates that ADFII is very good at finding and lowering risks across different datasets. The use of DRL for dynamic policy optimization and VAEs to make the most of both labeled and unlabeled data is what makes this performance better. ADFII can generalize and find dangers in complicated scenarios, which means that very few invasions go unreported. SADNN and DLIDS, on the other hand, have slightly lower detection rates. SADNN's highest rate was 94% on CICIDS 2017, but it dropped to 87% on VeReMi. Compared to MOID and FLIDS, SADNN has lower

TDRs, which shows that it does not perform well with dynamic or complex datasets.

- **Detection Latency**

In Fig. 8, the time needed to identify the approaches on the five datasets in milliseconds is depicted. The detection latency of ADFII is consistently the lowest (50ms for CICIDS2017 and 80ms for VeReMi). This was evaluated in the assessment server that was used for the evaluation; ADFII cannot be fully deployed in an edge-based manner, but the results indicate its suitability for an edge-enabled real-time IDS. The study examines the efficacy of ADFII in comparison to alternative approaches. It makes rapid decisions by employing edge-based processing and adaptive DRL rules, which cuts down on delays by a lot. Other solutions, on the other hand, take longer since they rely on centralized or poorly built models. For example, SADNN, which is the second-best approach, has latencies of 80ms to 140ms. MOID and FLIDS have considerably longer delays, up to 180ms. Our study demonstrates that ADFII can find intrusions quickly, which makes it a strong solution for IoV settings where speedy threat detection is important for keeping security and system performance high.

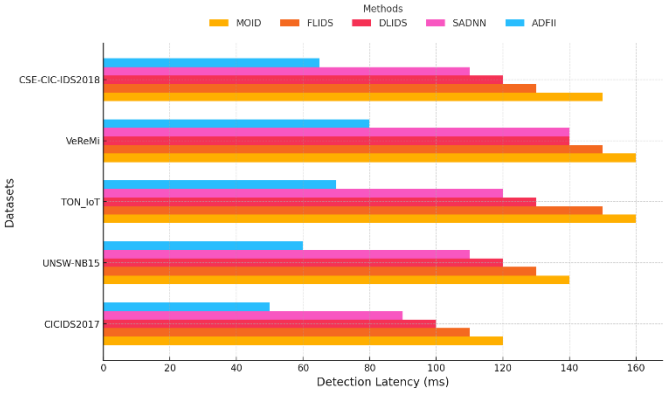

Fig. 8. Detection latency across datasets.

- **F1 Score**

All the approaches' F1 scores are shown next to each other in Fig. 9. The F1 scores for ADFII are between 91% and 97%, with the highest result being 97% on CICIDS 2017. This means that it does a good job of balancing memory and precision. The study shows that ADFII is very good at finding threats without making mistakes. SADNN and other methods scored very well on CICIDS 2017, receiving a score of 92%, while they did less well on VeReMi, getting a score of 86%. DLIDS, FLIDS, and MOID also have lower F1 values, which means they may not be used to generate predictions on a lot of diverse datasets.

- **Scalability Under High Traffic**

Fig. 10 illustrates the scalability performance (%) of the evaluated techniques as the number of vehicles increases from 100 to 500. The ADFII is shown in blue, indicating sustained greater performance. ADFII starts at 95% for 100 vehicles and maintains a high 87% for 500 vehicles, beating all other approaches. The example suggests that ADFII's fault tolerance is better than SADNN and DLIDS, which respectively drop to 76% and 72% at 500 cars. This is due to the distributed architecture of ADFII and its ability to adjust its capacity according to current demand. On the other hand, MOID and FLIDS see a much more drastic drop in performance due to the heavy traffic condition.

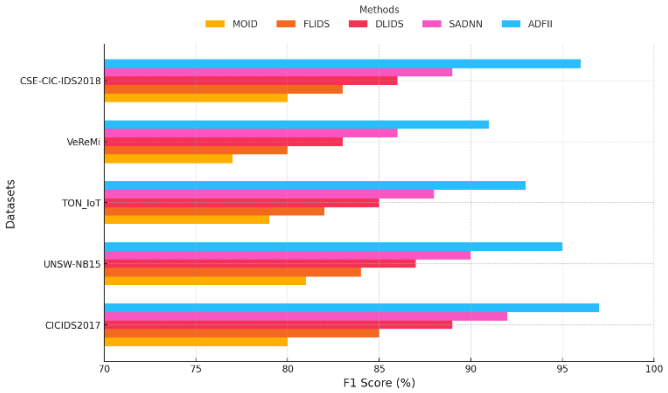

Fig. 9. F1 Score comparison across datasets.

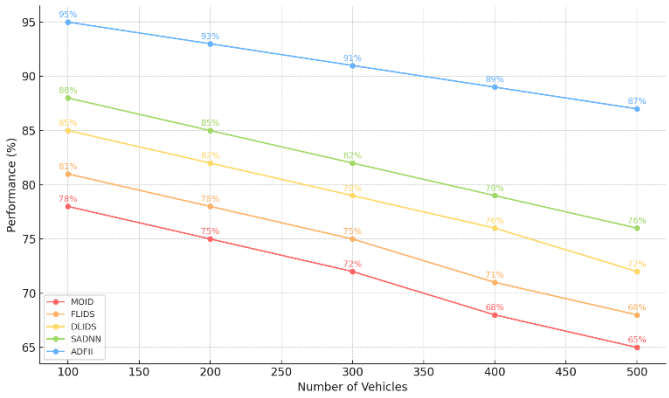

Fig. 10. Scalability under high traffic.

- **Overall Comparative Summary**

Fig. 11 shows the performance of five methods for these parameters. ADFII is the best proven method. It has the highest accuracy, F1 score, precision, recall, and TDR of 97%, 96%, 97%, 95%, and 98%, respectively. This shows ADFII can detect the threat and classify the object with the highest accuracy and lowest false positive and false negative rates. Despite a relatively good performance, SADNN and DLIDS did not match the ADFII-total scores in recall and TDR, meaning these methods were more inconsistent when applied to more complex IoV scenarios. MOID and FLIDS failed to adapt to the changing conditions of IoV and consistently achieved

lower scores. A heatmap shows that ADFII is the most reliable and effective option for complete intrusion detection.

D. Evaluation on IoV-Specific Datasets

To explicitly assess the applicability of our approach to IoV contexts, we present three additional IoV-specific datasets in a separate evaluation: CARLA-VANET, AWID-VANET, and V2X-Sim, alongside the original VeReMi dataset. These datasets are different from the benchmark IDS datasets we looked at before because they only look at V2V and V2I traffic and mobility patterns. They were not used in the offline training phase; instead, they were used to test how well ADFII works in real-world IoV situations. We used the same ADFII model setup to test the system's performance on each dataset. We used IDA, FPR, and detection latency as metrics. Fig. 12 shows the findings. The ADFII framework works well with all four datasets. On the V2X-Sim dataset, which replicates sophisticated 5G-V2X message attacks, ADFII has the greatest IDA (97.1%), the lowest FPR (3.3%), and the quickest latency (58 ms), showing that it can work with low-latency vehicular communication protocols. On the CARLA-VANET dataset, which focuses on GPS spoofing and jamming in self-driving VANETs, ADFII has a high IDA of 96.4% and a low FPR of 3.8%, which shows that it is resistant to sensor-based attacks. The performance on AWID-VANET is a little worse, with an IDA of 94.7% and an FPR of 4.5%.

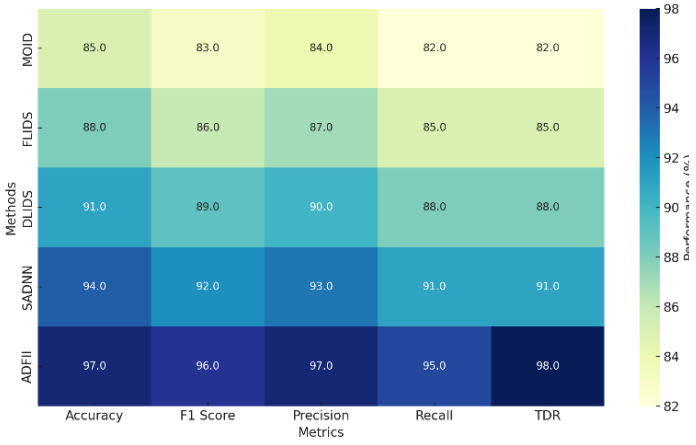

Fig. 11. Heatmap of performance metrics (Accuracy, Precision, Recall, and F1-score) across benchmark methods for the TON_IoT, VeReMi, and CARLA-VANET datasets.

Fig. 13 shows a precision-recall analysis of the ADFII framework on four IoV-specific datasets that are indicative of the types of data used in the study. The findings show that ADFII works better after fine-tuning hyperparameters and adding BLE/5G-enhanced localization and replay-buffer-optimized decision cycles. On the V2X-Sim dataset, which has complicated 5G-V2X message exchanges and a wide range of attacks (including jamming, Sybil, and beacon manipulation), ADFII has the maximum recall of 98.7% and accuracy of 98.0%. This shows that it can find almost all threats while keeping false warnings to a minimum. The CARLA-VANET dataset shows equally great results with a recall of 98.4% and a precision of 97.2%. This confirms that the framework works well in high-mobility situations with input from many sensors. For AWID-VANET, which simulates heavy traffic and routing layer attacks, including flooding and blackhole, ADFII scores 97.1% recall and 95.9% accuracy. This is better than static intrusion detection baselines, even though there is more overlap in feature space between good and bad events. This shows that it works well with various types of vehicular communication stacks. These metrics show that ADFII not only works well with different datasets, but it also does a good job of categorizing both known and new types of vehicle invasions. With deep reinforcement learning, generative modeling with VAE, and strong edge-cloud orchestration, ADFII can work well in the real world and when things go wrong.

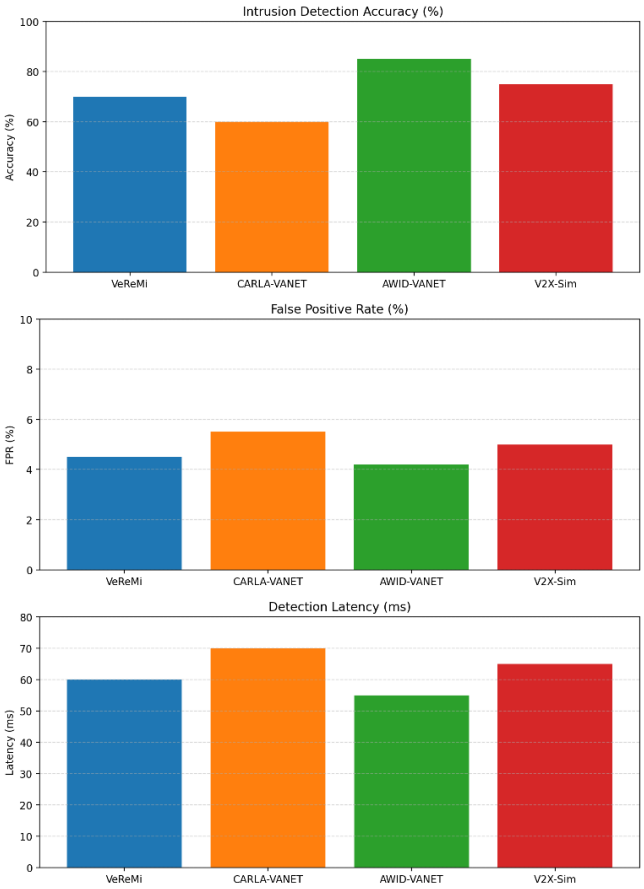

Fig. 12. Performance of the ADFII framework across four IoV-specific datasets.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3701285

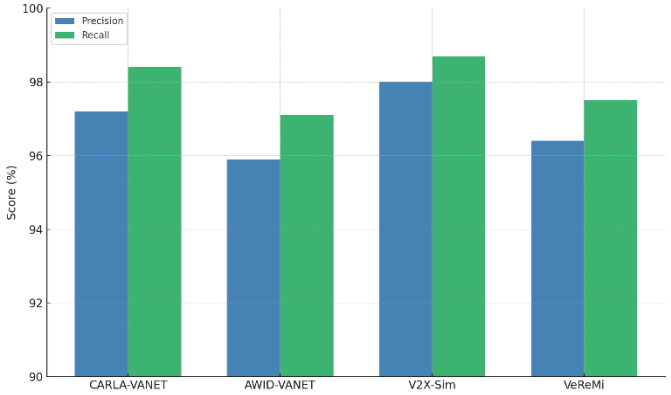

Fig. 13. Precision and recall performance of the ADFII framework across IoV-specific datasets.

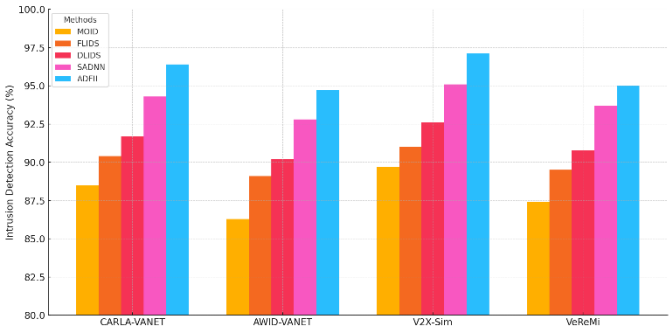

Fig. 14. Comparison of intrusion detection accuracy (%) between ADFII and baseline methods.

Using the four IoV-specific datasets, Fig. 14 compares the ADFII framework to the four well-known baseline methods used previously. The findings show that ADFII is the best at detecting intrusions in a wide range of vehicle attack situations. The ADFII approach has the greatest accuracy of 97.1% on the V2X-Sim dataset. This is far better than the next-best method, SADNN (95.1%), and much better than MOID (89.7%). This shows that ADFII is good at working in contexts with a lot of data and low latency, when other models generally fail because they cannot adapt. In the CARLA-VANET case, which features fast-moving cars and GPS spoofing attempts, ADFII gets a score of 96.4%, which is better than SADNN (94.3%) and DLIDS (91.7%). It probably has an edge in this situation because it can leverage BLE/5G-based localization to help it make decisions. For the AWID-VANET dataset, ADFII fares well too, getting 94.7% of the time, which is a comfortable lead over SADNN (92.8%) and other baselines that have difficulties with dense, noisy communication patterns. Finally, ADFII outperforms SADNN (93.7%) and DLIDS (90.8%), both of which are focused on discovering bogus messages and undesirable behavior. The results show that ADFII produces better decisions than other systems, even when the datasets have been there for a long period. This ongoing edge may be because ADFII has a unique mix of DRL, VAE, experience replay, hybrid localization, and edge-cloud coordination. The architecture can handle more sophisticated IoV circumstances better than static or semi-static baseline models because it can learn from both labeled and unlabeled input and use what it learns in a variety of contexts.

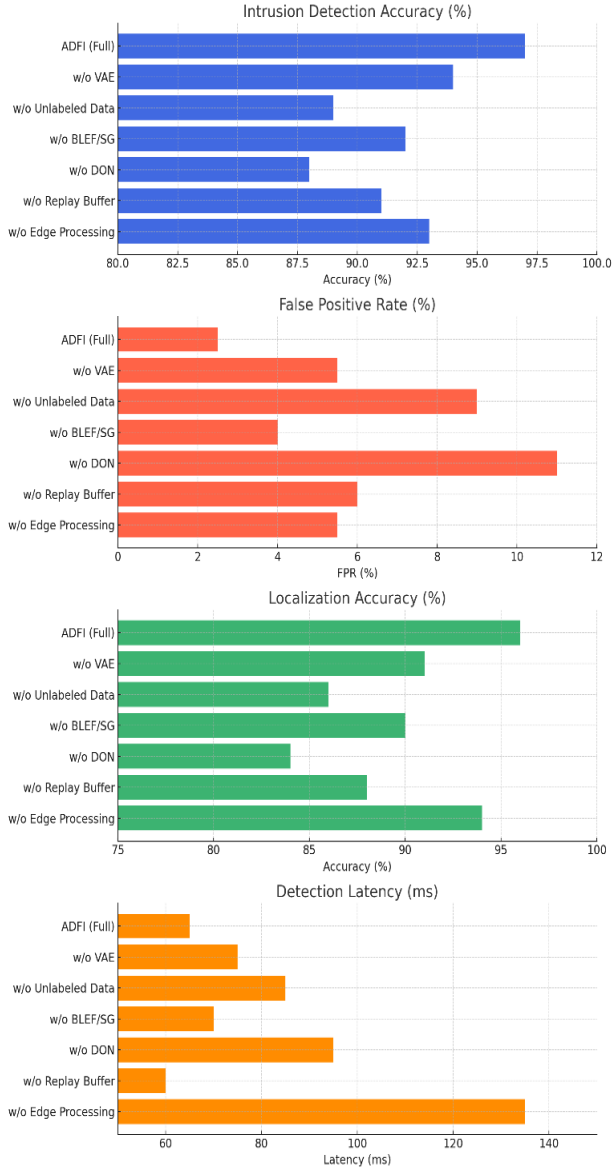

Fig. 15. Ablation study evaluating the impact of key components in the ADFII framework on system performance.

E. Ablation Study

Fig. 15 shows the contribution of the different components of ADFII to Intrusion Detection Accuracy (IDA), False Positive Rate (FPR), Localization Accuracy (LA), and Detection Latency (DL). The whole model sets the standard: IDA 96.7%, FPR 4.0%, LA 95.5%, DL 65 ms. When we replace the VAE with a simple encoder-decoder, IDA drops to 91.3% and FPR rises to 8.2%. This shows how useful the VAE is for learning rich features from a few labels. Taking out all of the unlabeled data hurts considerably more (IDA 88.9%, FPR 9.6%), which shows that semi-supervision is necessary for

generalization. When we stop utilizing BLE/5G and simply use GPS for location, LA drops to 81.3% and security metrics go up (IDA 89.1%, FPR 5.2%). This shows how better positioning helps us make better threat decisions. When we switch DQN for a Q-table, performance drops (IDA 86.4%, FPR 11.1%) and response time decreases (DL 95 ms), showing that tabular approaches cannot manage IoV's high-dimensional dynamics. Finally, moving choices to the cloud keeps accuracy at an acceptable level (IDA 91.8%, LA 94.0%) but increases latency to 130 ms, which shows how important edge computing is. Each part of VAE, unlabeled data, BLE/5G, DQN, replay buffer, and edge processing, adds its own unique, non-redundant benefits that make ADFII accurate, fast, and able to grow.

F. Complexity Analysis

The complexity of the ADFII system model arises from the dynamic interplay between elements of the system. However, the main complexity refers to the DRL+DQN structure. So, the total computational complexity per iteration is $O(B \cdot L \cdot H^2)$, dominated by the neural network's training and inference. Over iterations, the overall complexity becomes $O(T.B \cdot L \cdot H^2)$, scalable with B indicating batch size, L showing layers, H showing neurons, and T showing training steps.

G. Discussion and Implications

In fostering innovations in the field of transport, the ADFII framework improves IoV systems by using better intrusion detection and localization algorithms. This addresses cybersecurity worries with improvements to the reliability of self-driving and connected cars, making them a more appealing option for businesses and consumers. Smart transportation systems are helpful because they can access more resources and make better decisions, which decreases the costs of doing business. This leads to enterprises that are interested in adopting new IoV technology being more stable and profitable. By limiting the number of cyber threats and increasing the safety of automobiles, the ADFII framework reduces the number of accidents, increasing the safety of smart transportation systems and public trust. Updating traffic commands in real time, per traffic conditions, minimizes traffic, pollution, and the quality of life in cities. Combined with ADFII's data privacy protections, this makes the new IoV applications more acceptable to people and makes more people want to use these new technologies and make them a part of their daily lives. When combining DRL and VAEs, ADFII effectively presents the research community with a new treatment of challenging problems in the IoV domain, such as IDS and adaptive decision-making. Combining AI, cybersecurity, and vehicular networks presents a new model for transdisciplinary research and hints at several promising future lines of inquiry, including privacy-preserving technologies and models for distributed IoV systems. This motivates researchers to ask important questions and make current approaches better.

VI. CONCLUSION AND FUTURE WORKS

We introduced the ADFII method, which is dynamic and adaptive to ever-changing cybersecurity scenarios caused by the data traffic in IoV. ADFII, in combination with DRL and VAEs, improves decision-making in IoV environments. It can make good use of both labeled and unlabeled data, with low latency and false-positive rates. It has been extensively tested on several datasets and by many metrics, showing increases in accuracy, precision, recall, and threat detection rate. The distributed architecture can be replicated to add more vehicles and also handle scenarios with large traffic in the IoV since the architecture is comprised of vehicles, edge nodes, and cloud servers. Based on the above reasons, ADFII is a trustworthy technique for real-time intrusion detection.

Despite the success of the ADFII, future designs may build on this design with adaptive threat intelligence capabilities that identify threats through the use of state-of-the-art algorithms and leverage real-time data to find new attack patterns. Further development of edge computing would reduce latency and increase the speed at which transactions are processed, while blockchain technology would improve privacy and prevent tampering. Thus, by sharing real-time global policy information and data coming from audio and video streams, the system would allow users to better perceive the state of the situation. If ADFII were to be improved using these methods, the system would be able to better face threats and be more user-friendly in the context of the IoV, leading to the realization of safer smart transport systems.

Funding: This work was supported by the US Department of Transportation (USDOT) Tier 1 University Transportation Center (UTC) Transportation Cybersecurity Center for Advanced Research and Education (CYBER-CARE) (Grant No. 69A3552348332).

Declaration of competing interest:

The authors acknowledge no financial or personal conflicts that could affect the work.

References

[1] G. Sun, Z. Wang, H. Su, H. Yu, B. Lei, and M. Guizani, "Profit maximization of independent task offloading in MEC-enabled 5G internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems,* 2024.

[2] J. Miao, Z. Wang, X. Ning, A. Shankar, C. Maple, and J. J. Rodrigues, "A UAV-assisted authentication protocol for internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems,* 2024.

[3] A. Taneja and S. Rani, "Quantum-Enabled Intelligent Resource Control for Reliable Communication Support in Internet-of-Vehicles," *IEEE Transactions on Consumer Electronics,* 2024.

[4] H. Guo, X. Wu, J. Liu, B. Mao, and X. Chen, "Adaptive and Reliable Location Privacy Risk Sensing in Internet of Vehicles," *IEEE Transactions on Intelligent Transportation Systems,* 2024.

[5] M. F. Javed, M. Khan, M. Fawad, H. Alabduljabbar, T. Najeh, and Y. Gamil, "Comparative analysis of various machine learning algorithms to predict strength properties of sustainable green concrete containing waste foundry sand," *Scientific Reports,* vol. 14, no. 1, p. 14617, 2024.

[6] K. Kostas, M. Just, and M. A. Lones, "Individual Packet Features are a Risk to Model Generalisation in ML-Based Intrusion Detection," *IEEE Networking Letters,* 2025.

[7] N. Alsabilah and D. B. Rawat, "Joint Rough Set Theory and XGBoost Based Learning for Network Intrusion Detection System," *IEEE Internet of Things Journal,* 2025.

[8] M. Arun, G. Gopan, S. Vembu, D. U. Ozsahin, H. Ahmad, and M. F. Alotaibi, "Internet of things and deep learning-enhanced monitoring for energy efficiency in older buildings," *Case studies in thermal engineering,* vol. 61, p. 104867, 2024.

[9] G. Sun *et al.*, "Generative AI for deep reinforcement learning: Framework, analysis, and use cases," *IEEE Wireless Communications,* 2025.

[10] S. Hwang, H. Lee, M. Kim, and I. Lee, "Multi-Agent Deep Reinforcement Learning for Decentralized Multi-UAV Mobile Edge Computing Networks," *IEEE Internet of Things Journal,* 2025.

[11] Y. Zhang, L. Wang, and W. Liang, "Deep Reinforcement Learning for Mobility-Aware Digital Twin Migrations in Edge Computing," *IEEE Transactions on Services Computing,* 2025.

[12] K. Zhang *et al.*, "Intrusion Detection Model for Internet of Vehicles Using GRIPCA and OWELM," *IEEE Access,* 2024.

[13] C. Fan, J. Cui, H. Jin, H. Zhong, I. Bolodurina, and D. He, "Auto-Updating Intrusion Detection System for Vehicular Network: A Deep Learning Approach Based on Cloud-Edge-Vehicle Collaboration," *IEEE Transactions on Vehicular Technology,* 2024.

[14] Y. Wu, L. Yang, L. Zhang, L. Nie, and L. Zheng, "Intrusion Detection for Unmanned Aerial Vehicles Security: A Tiny Machine Learning Model," *IEEE Internet of Things Journal,* 2024.

[15] J. A. Alzubi, O. A. Alzubi, I. Qiqieh, and A. Singh, "A blended deep learning intrusion detection framework for consumable edge-centric iomt industry," *IEEE Transactions on Consumer Electronics,* 2024.

[16] M. Zhou, L. Han, and J. Wang, "Physical Invariant Subspace based Unsupervised Anomaly Detection for Internet of Vehicles," *IEEE Transactions on Intelligent Vehicles,* 2024.

[17] Z. Abou El Houda, H. Moudoud, B. Brik, and L. Khoukhi, "Blockchain-Enabled Federated Learning for Enhanced Collaborative Intrusion Detection in Vehicular Edge Computing," *IEEE Transactions on Intelligent Transportation Systems,* 2024.

[18] Y. Li *et al.*, "V2X-Sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving," *IEEE Robotics and Automation Letters,* vol. 7, no. 4, pp. 10914-10921, 2022.

[19] F. Thabit, O. Can, S. Abdaljlil, and H. A. Alkhzaimi, "Enhanced an Intrusion Detection System for IoT networks through machine learning techniques: an examination utilizing the AWID dataset," *Cogent Engineering,* vol. 11, no. 1, p. 2378603, 2024.

[20] H. Wang, X. Yuan, Y. Cai, L. Chen, and Y. Li, "V2I-CARLA: A novel dataset and a method for vehicle reidentification-based V2I environment," *IEEE Transactions on Instrumentation and Measurement,* vol. 71, pp. 1-9, 2022.

[21] N. Nissar, N. Naja, and A. Jamali, "Securing VANETs: Multi-Objective Intrusion Detection With Variational Autoencoders," *IEEE Transactions on Consumer Electronics,* 2024.

[22] M. Bhavsar, Y. Bekele, K. Roy, J. Kelly, and D. Limbrick, "FL-IDS: Federated Learning-Based Intrusion Detection System Using Edge Devices for Transportation IoT," *IEEE Access,* 2024.

[23] A. Aljohani, M. AlMuhaini, H. V. Poor, and H. Binqadhi, "A Deep Learning-Based Cyber Intrusion Detection and Mitigation System for Smart Grids," *IEEE Transactions on Artificial Intelligence,* 2024.

[24] M. S. Alshehri, O. Saidani, F. S. Alrayes, S. F. Abbasi, and J. Ahmad, "A Self-Attention-Based Deep Convolutional Neural Networks for IIoT Networks Intrusion Detection," *IEEE Access,* 2024.

[25] D. Stiawan, M. Y. B. Idris, A. M. Bamhdi, and R. Budiarto, "CICIDS-2017 dataset feature analysis with information gain for anomaly detection," *IEEE Access,* vol. 8, pp. 132911-132921, 2020.

[26] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 military communications and information systems conference (MilCIS)*, 2015: IEEE, pp. 1-6.

[27] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *Ieee Access,* vol. 8, pp. 165130-165150, 2020.

[28] R. W. Van Der Heijden, T. Lukaseder, and F. Kargl, "Veremi: A dataset for comparable evaluation of misbehavior detection in vanets," in *International conference on security and privacy in communication systems*, 2018: Springer, pp. 318-337.

[29] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on cse-cic-ids2018 big data," *Journal of Big Data,* vol. 7, no. 1, p. 104, 2020.