ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
Received: 13 April 2025; Accepted: 5 January 2026

https://doi.org/10.17559/TV-20240428001508
Original scientific paper

Cloud-Based Secure Data Management for Internet of Vehicles Platforms

Ke XIANG*, Xing YANG, Huihui WANG

Abstract: The rapid growth of Internet of Vehicles (IoV) platforms has raised significant concerns about data security and storage efficiency. This study proposes a novel approach that is integrating cloud computing, machine learning, and advanced database management to enhance IoV data security and storage. MongoDB is utilized for data storage. The logistic regression and Support Vector Machine (SVM) algorithms are combined for security detection. By this way, our method demonstrates improved performance over traditional approaches. Experimental results showed a 4.60% increase in data recognition precision, a 4.66% higher recall, and a 2.55% improvement in the F1 score compared to existing models. The proposed system also exhibits enhanced data storage efficiency and robust security detection capabilities. These findings demonstrate our method has significant potential for improving IoV platform security and data management in real-world applications.

Keywords: cloud computing; data storage; information management; Internet of Vehicles platform; machine learning;

1 INTRODUCTION

The Internet of Vehicles (IoV) platform integrates the mobile Internet with the Internet of Things (IoT), enabling connectivity through vehicle equipment, related devices, communication technologies, and vehicle terminal systems [1]. IoV facilitates the interconnection of people, vehicles, and roads, enabling real-time monitoring and data scheduling for vehicle drivers [2]. The platform collects and exchanges vast amounts of real-time data, including vehicle status, driving conditions, and traffic flow. This data is essential for applications such as intelligent traffic management, predictive maintenance, and energy optimization. Cloud computing, a distributed computing paradigm, processes and organizes large-scale data to simplify its management. It provides a centralized platform to manage IoT devices across diverse geographic locations and dynamically scales resources to accommodate changes in data demand [3]. Advanced cloud computing services, such as machine learning (ML) and big data analytics, further enhance IoV's intelligence. ML leverages computational models to analyze data and apply insights to specific scenarios, with cloud computing serving as the foundational framework for these analyses [4]. ML models can predict potential security threats and implement preventive measures, automating and optimizing security policies by adjusting them based on real-time data and predictions. However, vehicles face security challenges such as hacking, data breaches, and unauthorized access during data collection and transmission. Ensuring data security on IoV platforms has thus become a critical research area. The exponential growth in IoV data has outpaced the capabilities of traditional information management methods, necessitating the development of new technologies and strategies for optimized data storage, retrieval, and sharing. To address these challenges, cloud computing and ML models are employed to analyze and process IoT data. An innovative data storage fusion system based on a cloud computing framework has been designed, along with a network attack detection model leveraging ML and cloud computing to strengthen IoT platform defenses. Additionally, a novel ML-based data processing model has been developed to enhance the management and storage of vehicle data within vehicular networks.

This model acquires data through a traffic data platform and enhances detection performance using multiple data processing methods. The innovative integration of cloud computing, machine learning (ML), and a newly designed system architecture offers a more secure and efficient solution for data processing and security protection in the IoV platform. This study makes significant theoretical and practical contributions to advancing the development and application of IoV technology. It comprises four main parts: a review of domestic and international research, the establishment of a novel data storage (DS) and security detection model, experimental verification of the feasibility of the proposed method and model, and a final summary and analysis of the findings.

2 RELATED WORKS

The rapid development of the Internet of Vehicles (IoV) has brought significant convenience to vehicle usage. However, numerous challenges remain with the IoV platform. Mazhar et al. proposed an innovative solution integrating artificial intelligence, IoT, and smart grid technology. This approach utilized machine learning (ML) to predict energy demand, collected real-time data from smart meters, and employed remote monitoring technology. It not only improved the safety of smart buildings and the comfort of residents but also promoted the integration and expansion of smart grid technology within broader network environments. Despite demonstrating the immense potential of technology integration, the study highlighted ongoing challenges related to management, data privacy, and security in complex systems [5]. Altulaihan et al. tackled network security threats in IoT systems, particularly denial-of-service attacks, by developing an intrusion detection system based on anomaly detection and ML. Using four supervised classifiers and two feature selection algorithms, the study achieved excellent performance on the IoTID20 dataset. A genetic algorithm was integrated with decision tree (DT) and random forest models, with DT showing superior training and testing efficiency. While the research achieved notable technical results, the model's reliance on specific datasets raised concerns about its generalizability. Additionally, the real-time performance and scalability of the model were

not fully validated in practical deployment scenarios [6]. Manoharan et al. introduced a security enhancement scheme combining blockchain and deep learning models to ensure robust data security for IoT applications in smart city development. Blockchain technology safeguarded data transmission, effectively preventing sensitive information leakage, while deep learning models optimized transmission efficiency and accuracy. This study offered strong technical support for the holistic development of smart cities, homes, and industries. However, scalability issues and the high computational resource requirements of the model in handling large-scale data remained challenges [7]. Lopez et al. proposed an innovative IoT intrusion detection method utilizing a clustering-guided flow classification algorithm to address the extreme verification delay problem in IoT environments. This approach effectively adapted to non-stationary data streams and identified emerging threats with high accuracy. Nevertheless, practical deployment faced challenges regarding computing resource demands and storage requirements. Further adjustments were needed to adapt the method to different IoT devices and diverse network conditions [8].

The threshold-based ML method proposed by Liyakat et al. effectively identified and located malicious nodes in IoT systems by monitoring network path delays and triggering alerts when threshold conditions were exceeded. This approach significantly enhanced the security of IoT systems, as demonstrated in the NS2 simulation environment. The method also performed exceptionally well in key performance indicators such as throughput, latency, and packet loss rate, showcasing its effectiveness in mitigating potential network attacks. However, the model's reliance on specific threshold settings posed a challenge, requiring adjustments and optimizations to suit different network environments [9]. Alahmadi et al. contributed valuable resources and insights to the academic community through a comprehensive review of research on ML-based DDoS detection in IoT. Their work explored the application of ML algorithms in network attack identification and emphasized their potential in addressing IoT security threats. The literature review provided a robust foundation for further research in the field, fostering the expansion and deepening of IoT DDoS detection studies. Despite its significance, the research highlighted the need for deeper exploration of the breadth and depth of existing literature and the practical application of ML algorithms in real-world scenarios [10]. Krishnamourthy et al. conducted an extensive analysis of IoT intrusion detection systems based on ML from 2011 to 2021, underscoring the significant advantages of ML technology in detecting emerging network attacks. Compared to traditional methods, deep learning techniques demonstrated superior performance in anomaly detection, offering more efficient security measures for IoT systems. However, the study also identified persistent challenges in IoT security research, including high computing resource demands, adaptability to adversarial attacks, and limitations in system scalability [11].

In summary, current research on IoV security models and data resource processing faces challenges such as inadequate security performance, low data processing efficiency, and limited model applicability. To address these issues, this study proposes a novel database network for data storage (DS) that leverages machine learning (ML) and cloud computing to enhance the security of the IoV platform. Complementary defense mechanisms were implemented using the logistic algorithm and Support Vector Machine (SVM), improving the platform's capability to detect and counter network attacks. Additionally, data feature extraction was applied to generate more efficient feature vectors, further enhancing the model's overall efficiency.

3 RESEARCH METHODOLOGY

3.1 Data Information Storage Methods for the Internet of Vehicles Platform

Cloud computing is a flexible data resource model that allows on-demand access and computation based on user needs. These data resources can be accessed, configured, and allocated dynamically to meet specific user requirements. Additionally, cloud computing provides virtualized data resources that can be applied to various platforms. As an extension of cloud computing, cloud storage enables the storage and utilization of these resources, facilitating data integration within IoV platforms. In the IoV platform, data storage (DS) plays a dual role: it stores the resources and information required by the platform and supplies this stored data to external providers. Users can access the IoV platform through a browser to search and retrieve stored data, enabling clients to use and access these resources efficiently. This approach minimizes the waste of storage space caused by redundant or underutilized data [12]. Fig. 1 illustrates the IoV storage model.

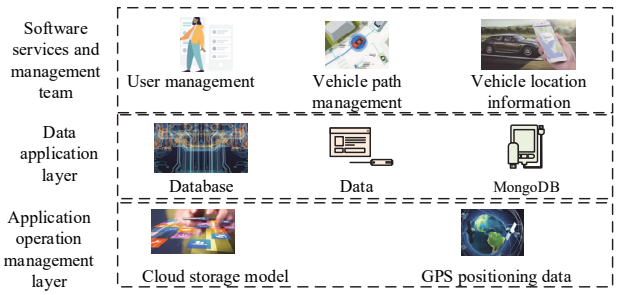

Figure 1 Vehicle connected storage model

From Fig. 1, the IoV data information storage management model consists of three primary operational layers: the software service and management layer, the data application layer, and the application operation management layer. The software service and management layer provides essential services to users, including vehicle location tracking, user system management, and driving path information. The data application layer focuses on offering elastic data services, such as DS file systems and application services tailored to user needs. Lastly, the application operation management layer oversees the operational management of current applications. In this model, vehicle data is stored using cloud storage frameworks. The primary data collected and uploaded include real-time GPS positioning data, vehicle images, and video information gathered through intelligent sensors. Data collection on the DS end of the IoV relies on the

Android data collection framework, enabling efficient acquisition and upload of data to the terminal. Once uploaded, the data is stored in the cloud storage database, allowing for later browsing and retrieval [13]. Fig. 2 illustrates the Android terminal framework structure.

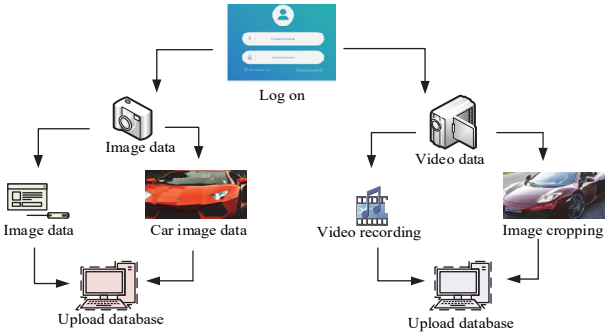

Figure 2 Android terminal framework structure

From Fig. 2, the data storage (DS) process in the terminal begins with the user logging into the client and selecting the camera or video storage option on the homepage. For image data, the images are captured and stored in the image library, then uploaded to the database for storage. For video data, the process involves recording the vehicle video, cropping the collected image data, and subsequently uploading it to the database. The main database employed is MongoDB, a non-relational database written in C language, which provides efficient DS solutions for IoV platform applications. MongoDB excels in quick data reading and writing. As the number of IoT devices increases, MongoDB's scalability allows the addition of more storage space and processing power to meet growing demands. Its powerful search tools enable efficient retrieval of necessary data, making it well-suited for IoT device data storage and utilization. Consequently, this study utilizes the MongoDB library for data storage. The core database infrastructure includes the primary database and cluster service components, with the latter playing a crucial role in ensuring the reliability and normal operation of data storage. The cluster component primarily employs a replication collection mechanism, which not only maintains MongoDB cluster operations during machine failures but also ensures continuous functionality through replication across primary and secondary nodes. This enhances the robustness and reliability of the database system [14]. Fig. 3 depicts the logical structure of the cluster.

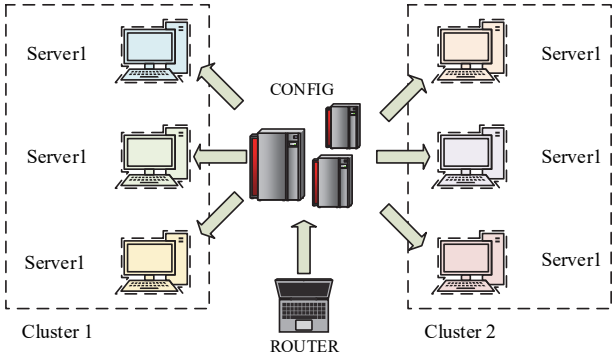

Figure 3 Cluster logic structure

From Fig. 3, the analysis of the terminal data cluster involves the use of three computer hosts, and the main logical components include SERVER, SHARD, CONFIG, and ROUTER. SERVER represents the three computers utilized within the cluster component. SHARD is an automatic data partitioning mechanism that ensures data within the cluster can be recovered and automatically contained as needed. CONFIG serves as the configuration service, ensuring the smooth and reliable operation of the cluster. ROUTER functions as a routing mechanism to enhance the stability of the data storage (DS) process. The MongoDB cluster components evaluate data information in the background and process data requests from various application layers, enabling efficient management of vehicle network data. By utilizing MongoDB for data storage in the IoV platform, the system ensures proper data management, timely inspection, and effective data retrieval, contributing to a robust and reliable data storage solution.

3.2 Internet of Vehicles Data Security Guarantee Method Based on Cloud Computing

The Internet of Vehicles (IoV) often faces network attacks during data usage, resulting in data breaches. Traditional manual methods for data analysis and defense are expensive and may not effectively address real-world security defense challenges. Employing machine learning (ML) algorithms for security performance evaluation and attack defense can significantly reduce costs while providing more accurate judgments and analyses for security defenses. This approach enhances the efficiency and precision of security measures compared to traditional methods [15]. Fig. 4 illustrates the security defense model of traditional IoT platforms.

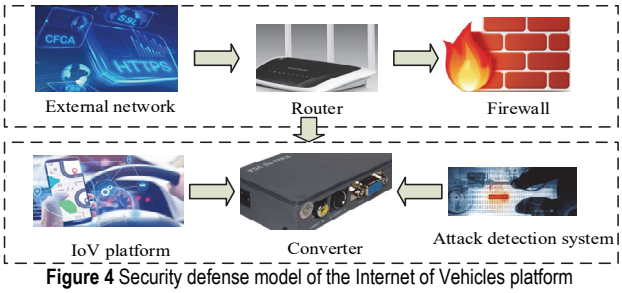

Figure 4 Security defense model of the Internet of Vehicles platform

From Fig. 4, the working components of this model include multiple intrusion prevention structures, such as firewall networks and router blocking mechanisms. The model also involves detecting and analyzing network attacks, including their types, frequency, and intensity. The database functions as a relay for data utilization in network attack defense, ensuring secure storage and use of IoV data while enhancing overall data security. The security model relies on algorithmic models to detect and classify data on the IoV platform. Deploying the IoV platform enables data detection and early warning functions without requiring platform operation, while also improving learning capabilities for offline network attack behaviors. The detection process begins with processing traffic data from the IoV platform runtime layer, converting it into a vector feature set recognizable by machines. These feature models are then trained and tested on datasets to achieve efficient data utilization and scheduling. Using machine learning as

a data source effectively prevents data leakage caused by network attacks. Additionally, the capability for analyzing network attack scenarios can be enhanced, further strengthening network data protection. Fig. 5 illustrates the model training process for network attack detection

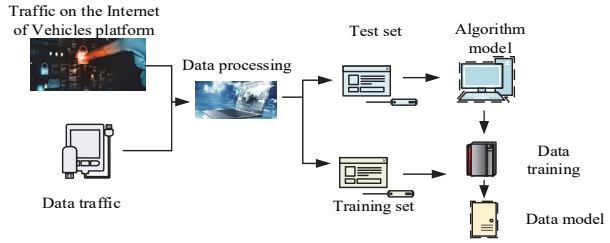

Figure 5 Algorithmic training for network attack detection

From Fig. 5, using the algorithm model for data traffic learning involves processing IoV platform traffic data along with other vehicle-related data traffic. These data are then divided into training and test sets. The training set is used to train the learning algorithms, after which the optimized data are fed into the learning algorithm model. The test set is directly utilized for testing and validating the algorithm's performance. The detection of data network attacks on the IoV platform is a binary classification problem. Therefore, the logistic regression model from machine learning (ML) is applied. This binary classification process involves first identifying the type of network attack to determine whether it constitutes a malicious attack. In IoV security, logistic regression collects behavioral data from devices within the IoV network and selects features that help differentiate between normal and abnormal behavior. Additionally, data cleaning, standardization, or normalization is performed to eliminate biases and noise, ensuring more accurate and reliable classification results. The evaluation of the model is conducted by training a logistic regression model, ultimately determining the type of network attack. To achieve this, the model first labels the data results, which are represented mathematically using Eq. (1) [16].

$$D = (x_1, y_1), (x_2, y_2), ..., (x_n, y_n), x_i \subseteq R^n, \quad y_i \in 0, 1, i = 1, 2, ..., N \tag{1}$$

In Eq. (1), D represents the dataset. (x_i, y_i) refers to the coordinates of the i-th sample data vector in the dataset. x_i is a feature vector. y_i refers to the target value or label corresponding to x_i. R^n means that each feature vector x_i is an n-dimensional real vector. $y_i \in 0, 1, i = 1, 2, \ldots, N$ represents that each target value is a real number y_i. N means a constant. The regression prediction data generated by this model is a continuous real number, which cannot directly represent discrete data variables. Therefore, for discrete data, a transformation of continuous real numbers is required, as represented by Eq. (2).

$$y = \begin{cases} 0, & \eta < 0 \\ 0.5, & \eta = 0 \\ 1, & \eta > 0 \end{cases} \tag{2}$$

In Eq. (2), η represents the magnitude of the data values predicted by the model. If the predicted value is greater than 0, it indicates that the currently attacked network type is classified as a positive sample data type. Conversely, if the predicted value is less than 0, it indicates that the network type under attack is classified as a negative sample data type. At this stage, the predicted calculation method is represented by Eq. (3) [17].

$$\eta = w^T x + b \tag{3}$$

In Eq. (3), w^T represents the weight vector. x refers to a feature vector. b means the bias term. However, since the predicted function data are discontinuous, logarithmic processing is required to achieve better data representation, as expressed in Eq. (4).

$$y = \frac{1}{1 + e^{-\left(w^T x + b\right)}} \tag{4}$$

Logarithmic processing enhances the simulation and training performance of the current data, allowing for more accurate modeling. Further data processing is performed to refine the results, as represented by Eq. (5).

$$\ln \frac{y}{1 - y} = w^T x + b \tag{5}$$

In Eq. (5), y refers to the probability of a network attack occurring. $1 - y$ means the probability of negative terms occurring in the predicted data. Comparing these two probabilities allows for the calculation of the logarithmic data occurrence probability as described in Eq. (3). By further comparing and refining the probabilities in Eq. (4), the final probability is represented by Eq. (6) [18].

$$\ln \frac{P(Y = 1 \mid x)}{1 - P(Y = 1 \mid x)} = w^T x + b \tag{6}$$

In Eq. (6), $P(Y = 1|x)$ represents the logarithmic probability of the output value $Y = 1$. Eq. (7) refers to the expression of probability $P(Y = 1|x)$.

$$P\left(Y = 1 \mid x\right) = \frac{1}{1 + e^{-\left(w^T x + b\right)}} \tag{7}$$

When the current probability value approaches 1, the network attack in the sample is classified as a malicious network attack. At this stage, preliminary data training can be performed on the types of network attacks. Irrelevant feature data are eliminated by evaluating the weight values of features, enhancing the model's accuracy in detecting network attacks. The logistic regression model enables the construction of models based on attack types during the initial detection phase, reducing inaccuracies caused by model misclassification. It also makes probability-based judgments on specific attack types, adjusting the feature threshold to achieve a reasonable threshold range. The Support Vector Machine (SVM) algorithm collects data

from the IoV, extracts features relevant for SVM classification, and performs necessary data transformations. Subsequently, an appropriate kernel function is selected to train the model, aiming to find the optimal solution for the data. Once the initial types of network attacks are identified, SVM calculates the optimal solution for the processed data, as represented by Eq. (8).

$$0 = w^T x + b \tag{8}$$

Eq. (8) represents the optimal partitioning formula for the linear equation plane of the model. In determining the optimal plane, multiple linear data points are considered. However, it is crucial to identify the plane that maximizes the distance between these data points. The distance between data points and the hyperplane is calculated and represented by Eq. (9).

$$r = \frac{\left| w^T x + b \right|}{\|w\|} \tag{9}$$

The distance between the plane samples with the maximum margin can be calculated using Eq. (9). Assuming that the current plane classification satisfies the required plane spacing for data construction, the sample classification under these conditions is represented by Eq. (10).

$$\begin{cases} w^T x + b \geq +1, y_i = +1 \\ w^T x + b \leq -1, y_i = -1 \end{cases} \tag{10}$$

In Eq. (10), the upward interval value of the normal vector is classified as a malicious attack network type for forward requests, while the downward interval sample is classified as a normal sample type. The distance between the data intervals is then calculated, leading to Eq. (11) [19].

$$d = \frac{2}{\|w\|} \tag{11}$$

In Eq. (11), d represents the defined distance. Malicious attacks are classified as vector support types. If the current maximum function constraint is identified, the maximum defined distance is achieved by converting $\|w\|$ into the maximum and minimum values, represented by Eq. (12).

$$\min \frac{1}{2} \|w\|^2 \tag{12}$$

In Eq. (12), when the current w and b reach their minimum values, based on the data from the IoV platform, the optimal partition interval is achieved when the interval distance reaches its minimum. The optimal plane derived under these conditions is represented by Eq. (13) [20-22].

$$\begin{cases} \max \sum_{i=1}^{N} l_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} l_i l_j y_i \mathrm{Ker}\left(x_i, x_j\right) \\ \sum_{i=1}^{N} l_i y_i = 0,\ 0 \leq l_i \leq C,\ i = 1, 2, ..., N \end{cases} \tag{13}$$

In Eq. (13), l_i and l_j are Lagrangian factors. Ker(x_i, x_j) means a kernel function. C refers to the penalty factor. The larger the penalty factor, the greater the loss value of the current model, which improves the fitting of the sample. However, overfitting may occur under these conditions. Conversely, when the penalty factor value is smaller, more samples fall near the boundaries, leading to underfitting of the samples and a more pronounced misclassification effect on network types [23, 24]. The IoV security measures process after incorporating the ML model is illustrated in Fig. 6.

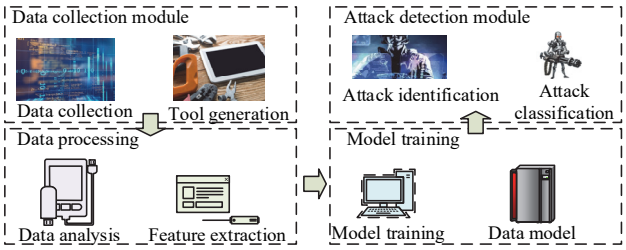

Figure 6 Security detection process for Internet of Vehicles

From Fig. 6, the establishment of a data security platform begins with collecting data from the IoV platform and entering the platform's data storage (DS) into the database. These data are then processed using machine learning (ML). The data undergo analysis and feature extraction to generate feature vectors. Once the feature vectors are created, the attack types associated with the model's data vectors are classified to identify potential network attacks.

4 RESULTS AND DISCUSSION

This study selected C++ as the programming language, with the initial redundancy value \(n \) set to 3. Additional values of 5 and 10 were used for control experiments. During the experiment, detailed information about the behavior and message transmission of the ECU was monitored via console output. The simulation experiment replicated the complete process of the ESM-VN mechanism, including the looping stages of message sending and updating. The experiment was conducted in three main steps: first, setting up the experimental environment; second, developing the experimental simulator to replicate the model's functionality; and finally, performing the experimental setup to verify the results. Network security tools and IoV platform data were utilized for attack detection to compare the data security effects of current research methods on the IoV platform. The experimental platform was divided into two components: the IoV platform and the network attack platform. The IoV platform simulated the network operation of the front-end, while the attack platform was responsible for real-time monitoring of the host and application layers of the IoV

and conducting batch attacks. Tab. 1 presents the device parameters used for both platforms.

Table 1 Equipment parameters of the platform

/	Telematics platform	Data attack host
Software configuration	Windows10	Kali Linux
	Apache: 2.4.39	Python: 3.6.x
	MySQL: 5.7.26	MySQL: 5.7.26
	PHP: 5.6.9nts	Nginx: 1.15.11
	Telematics platform: 1.0.1	Burpsuite: 2.0.1
	CPU: Intel(R) Core(TM)	CPU: Intel(R) Core(TM)
	i5-8265U	i5-5200U
Hardware configuration	RAM: 8 GB	RAM: 4 GB
	DDR4: 2400 MHz	DDR3: 1600 MHz
	800 GB	500 GB
Network bandwidth	200M WLAN	200M WLAN

IoV attacks and various types of hacker network attacks were simulated using the parameters outlined in Tab. 1. The experiment evaluated the attack detection performance of the current model and the data storage (DS) performance at the application layer. To ensure consistency, three host effects were considered during data storage, and the parameters of the storage hosts used in the tests matched those of the IoV platform hosts. The data used in this study were sourced from publicly available information. The platform contained a total of 20000 samples, evenly divided into two datasets. Each dataset consisted of 10000 entries, including both normal data and anomaly detection data. ECU refers to the number of onboard computers, reflecting the amount of onboard data requiring storage. Fig. 7 illustrates the data proportion effect of the current model compared to the unused model. Data storage capacity is defined as the total amount of data that a computer system, database, storage device, or service can store.

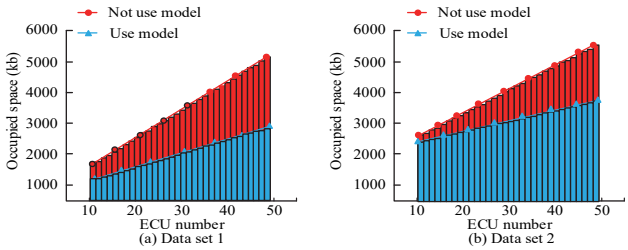

Figure 7 Comparison of different data space occupations

From Fig. 7a and Fig. 7b, when using the model for DS, the storage capacity increased proportionally with the ECU count, while the space occupied by the model was less. The overall space occupancy ratio was significantly lower when comparing two different IoV datasets, indicating that using the model for storage analysis effectively reduced space usage. This demonstrates that the model improves data storage capacity, allowing a greater amount of data to be stored efficiently. To evaluate the security performance of the research, the bus usage rate in the platform was tested. Assuming that the attacker can detect the IoV data bus, the resulting data utilization rate reflects the total amount of data accessible, as shown in Fig. 8. Data usage generally refers to the total volume of data accessed, processed, or consumed by individuals, organizations, or systems within a specific timeframe.

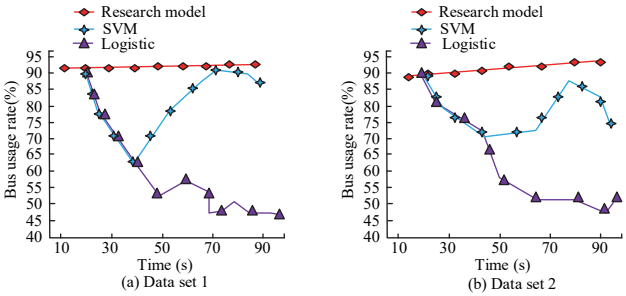

Figure 8 Comparison of data bus usage on connected vehicle platforms

From Fig. 8a and Fig. 8b, as the data simulation time increased, the bus utilization rate of the research model rose but remained stable at approximately 93%. In comparison, the bus usage of the SVM model initially decreased and then exhibited an upward trend, while the logistic regression model showed a gradual overall decline. The comparison of bus utilization among these models indicates that the IoV data security performance was relatively low when using SVM or logistic regression alone. However, the research model demonstrated significantly better IoV data security. This experiment also examined the impact of different vehicle-mounted hosts on data processing time under varying weight values. Fig. 9 illustrates the relationship between data and weight values. Data processing volume typically refers to the total amount or scale of data that a system, organization, or individual processes within a specific time frame.

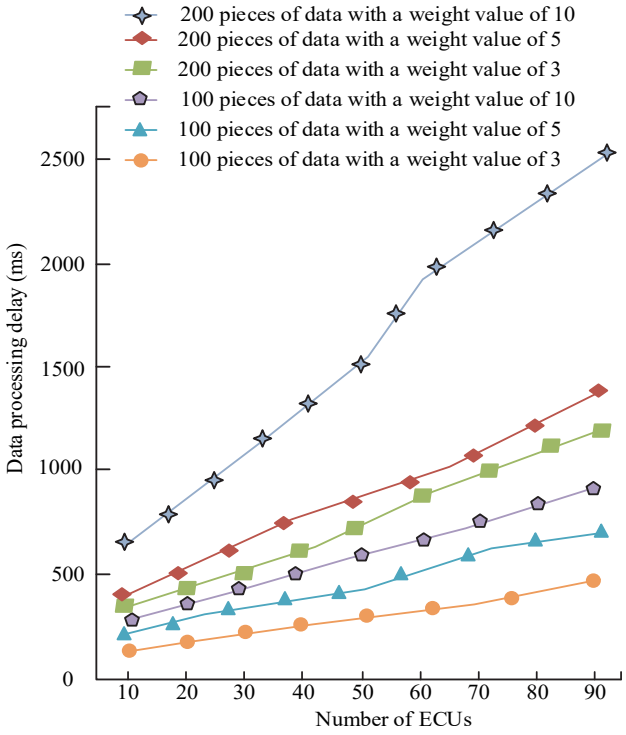

Figure 9 Comparison of processing delays for data with different weights

From Fig. 9, when the data processing volume was 100, the larger the model's weight value, the longer the data processing delay time. At a data processing volume of 200 with a weight value of 10, the delay time reached its

maximum. Conversely, when the data processing volume was 100, smaller weight values resulted in shorter processing delays. Additionally, as the number of ECUs increased, the data processing delay time became longer. A smaller data processing volume combined with a lower model weight led to shorter delay times, ensuring faster and more cost-effective IoV data processing. The effectiveness of data processing with different datasets was compared, and the results are shown in Tab. 2. Positive samples represent data that have not been attacked, while reverse samples represent data subjected to network attacks. In classification problems, positive samples are classified as belonging to the positive class, while reverse samples belong to the negative class. Reverse samples often indicate network attacks, whereas positive samples may refer to normal network traffic. Precision measures the proportion of correctly predicted positive samples out of all samples predicted as positive. Recall measures the proportion of actual positive samples correctly identified by the model. F1 score, the harmonic mean of precision and recall, provides a balanced evaluation of the model's accuracy and completeness.

Table 2 Security performance comparison of different data

Data samples	Positive samples	Reverse samples	Precision	Recall	F1
1	300	100	87.50%	70.30%	79.40%
2	800	600	90.20%	89.50%	88.60%
3	1500	1200	92.50%	90.30%	94.60%
4	4000	3500	94.60%	91.60%	95.60%
5	4000	4000	97.80%	95.60%	96.20%

From Tab. 2, when the positive and reverse samples were equal, the precision achieved by the research model was approximately 10.30% higher than that of group 1. Its recall was about 25.30% higher, and its F1 score was around 16.80% higher than group 1. When the positive and reverse samples were nearly equal, the overall testing performance was better, with smaller differences yielding improved results. To compare the testing outcomes of the research model with traditional models, various data models, including decision trees, Naive Bayes algorithms, and the research model, were evaluated for their precision in identifying attack types, as shown in Fig. 10.

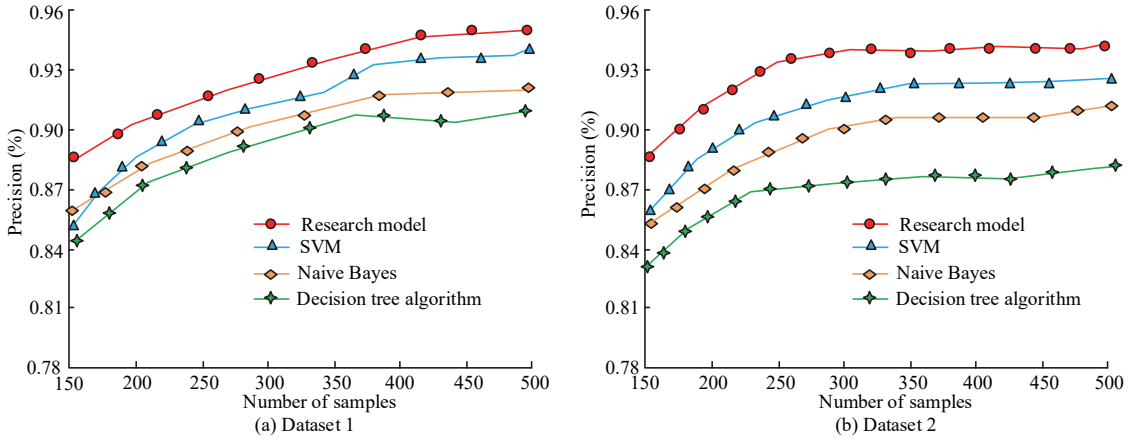

Figure 10 Comparison of network attack recognition precision for different model data

From Fig. 10a, in dataset 1, the research model achieved an average data recognition precision of 93.25%, which was 4.60% higher than the decision tree model, the algorithm with the lowest recognition precision at 88.65%. From Fig. 10b, the research model's average recognition precision was 94.32%, which was 7.97% higher than the decision tree's average of 86.35%. These results indicate that the research model significantly outperformed other algorithms in accurately identifying network attacks, demonstrating superior recognition of network attack types. To further evaluate the recall and F1 scores of different models, the parameters of these models were compared across varying data volumes, as shown in Fig. 11.

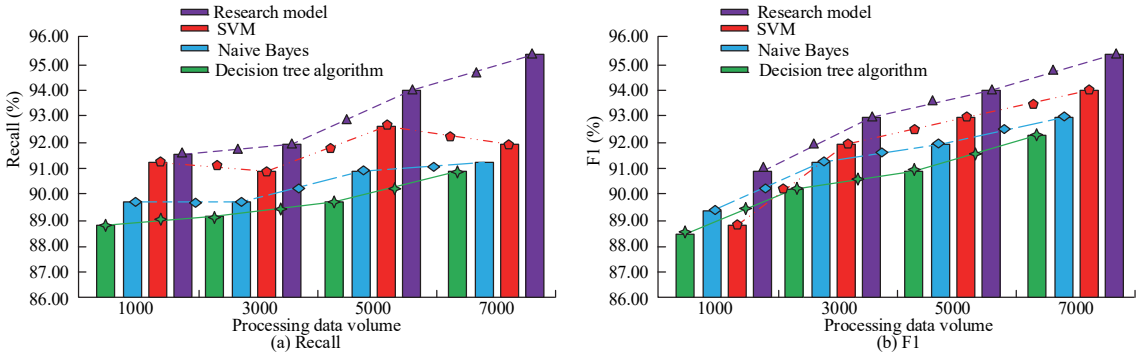

Figure 11 Comparison of network attack type recognition effects among different models

From Fig. 11a, the recall of the research model significantly increased with the growth in data volume. When the data volume reached 7000, the recall of the research model was 94.98%, which was approximately

4.66% higher than the 90.32% achieved by the decision tree. From Fig. 11b, the F1 score of the research model followed a trend consistent with the recall. At a data volume of 7000, the F1 score of the research model was 94.68%, which was 2.55% higher than the decision tree's 92.13%. These results indicate that the research model could more effectively identify the types of attacks on IoV data, demonstrating superior security performance for IoV data compared to other models.

4.1 Statistical Validation

The experiment is repeated 10 times to reduce random errors. The research model @0.5: 83.5, 82.9, 83.4, 83.2, 83.0, 83.5, 83.6, 83.0, 82.9, 83.1

The initial SVM @0.5: 77.2, 77.5, 78.4, 77.6, 78.3, 78.2, 78.1, 78.3, 79.4, 78.2

Assumption 1: The use of models and raw materials SVM@0.5 is studied. There is no significant difference in the above average values.

Assumption 2: The average value of the research model @0.5 in the study is significantly higher than that of the original SVM model.

The mean and standard deviations of the two samples were calculated, followed by an independent sample *t*-test. The *P*-values were computed, and the results were analyzed. The statistical analysis yielded \($P < 0.05$ \), indicating that the model used in the study achieved AP@0.5. This improvement is statistically significant, confirming that Assumption 2 holds true.

4.2 Discussion

When comparing the data processing performance of the models, the results showed that data processing was more efficient when the models were used. The decision tree and Naïve Bayes algorithms exhibited higher data processing performance, likely due to the incorporation of SVM and logistic regression models in the study, which enhanced the models' data processing capabilities. Adding different models improved the overall performance, making the proposed model superior to traditional models. In testing the types of cyberattacks, the accuracy of identifying attack types using the model's data parameters was higher than that of other algorithms. This improvement is likely attributed to the enhanced ability of SVM and logistic regression models to identify cyberattacks. When comparing the performance of different data parameters, the performance was consistently higher with the proposed model. This may be due to the model's superior data processing capabilities when handling large data volumes. These findings indicate that the model used in the study provides better data preprocessing capabilities and maintains high data processing performance, even when dealing with larger datasets, compared to other models.

In summary, incorporating different models into the study significantly enhances overall model performance. Compared with other models, the research model demonstrates superior accuracy and recognition in terms of data processing capability, data security performance, and network attack identification. Additionally, its ability to handle larger datasets is significantly better than that of other models. The research model consistently outperforms others, delivering better and higher overall performance.

5 CONCLUSION

This study introduces a novel approach to enhancing data security and storage efficiency in IoV platforms by integrating cloud computing, machine learning (ML), and advanced database management. The proposed system demonstrated significant improvements in data recognition precision, recall, and F1 scores compared to existing models. The use of MongoDB for data storage, combined with logistic regression and SVM algorithms for security detection, established a robust framework for managing IoV data. However, certain limitations were identified, including the need for further validation in real-world environments and challenges related to scalability. Future research should address these limitations by testing the model with more real-world data and exploring the integration of advanced ML techniques. Additionally, the complexity introduced by the machine learning models may increase the computational burden of the system. Future studies will focus on reducing computational complexity and enhancing the model's efficiency. Another limitation is the lack of adequate testing against sophisticated cyberattacks, such as Advanced Persistent Threats (APT). Subsequent research will aim to strengthen the model's security to ensure better protection against such advanced threats. The findings of this study provide valuable insights for improving IoV platform security and data management, contributing to the development of more secure and efficient smart transportation systems.

Acknowledgements

This work was supported by the 2024 scientific research project of Sichuan Post and Telecommunication College, "Research on Security Management Methods of Internet of Vehicles Platforms Based on Cloud Computing" (Project No. YDXJKY202432).

6 REFERENCES

[1] Zakariyya, I., Kalutarage, H., & Al-Kadri, M. O. (2023). Towards a robust, effective and resource efficient machine learning technique for IoT security monitoring. *Computers & Security*, *133*(10), 103388-103389. https://doi.org/10.1016/j.cose.2023.103388

[2] Harahsheh, K. M. & Chen, C. H. (2023). A survey of using machine learning in IoT security and the challenges faced by researchers. *Informatic*a, *47*(6), 1-54. https://doi.org/10.31449/inf.v47i6.4635

[3] Karthikeyan, M., Manimegalai, D., & RajaGopal, K. (2024). Firefly algorithm based WSN-IoT security enhancement with machine learning for intrusion detection. *Scientific Reports*, *14*(1), 231-232. https://doi.org/10.1038/s41598-023-50554-x

[4] Adekunle, T. S., Alabi, O. O., Lawrence, M. O., Adeleke, T. A., Afolabi, O. S., Ebong, G. N., Egbedokun, G. O., & Bamisaye, T. A. (2024). An intrusion system for internet of things security breaches using machine learning techniques. *Artificial Intelligence and Applications*, *2*(3), 188-194. https://doi.org/10.47852/bonviewAIA42021780

[5] Mazhar, T., Irfan, H. M., Haq, I., Ullah, I., Ashraf, M., Shloul, T. A., Ghadi, Y. Y., Imran, & Elkamchouchi, D. H. (2023).

Analysis of challenges and solutions of IoT in smart grids using AI and machine learning techniques: *A review. Electronics*, *12*(1), 242-243. https://doi.org/10.3390/electronics12010242

[6] Altulaihan, E., Almaiah, M. A., & Aljughaiman, A. (2024). Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms. *Sensors*, *24*(2), 713-714. https://doi.org/10.3390/s24020713

[7] Manoharan, H., Manoharan, A., Selvarajan, S., & Venkatachalam, K. (2023). Implementation of internet of things with blockchain using machine learning algorithm: Enhancement of security with blockchain. *Handbook of research on blockchain technology and the digitalization of the supply chain*, *32*(6), 399-430. https://doi.org/10.4018/978-1-6684-7455-6.ch019

[8] Lopez, M. M., Shao, S., Hariri, S., & Salehi, S. (2023). Machine learning for intrusion detection: Stream classification guided by clustering for sustainable security in iot. *Proceedings of the Great Lakes Symposium on VLSI 2023*, *5*(1), 691-696. https://doi.org/10.1145/3583781.3590271

[9] Liyakat, K. K. (2023). Machine learning approach using artificial neural networks to detect malicious nodes in IoT networks. *International Conference on Machine Learning, IoT and Big Data*, *10*(4), 123-134. https://doi.org/10.1007/978-981-99-4577-1_3

[10] Alahmadi, A. A., Aljabri, M., Alhaidari, F., Alharthi, D. J., Rayani, G. E., Marghalani, L. A., Alotaibi, O. B., & Bajandouh, S. A. (2023). DDoS attack detection in IoT-based networks using machine learning models: a survey and research directions. *Electronics*, *12*(14), 3103-3104. https://doi.org/10.3390/electronics12143103

[11] Krishnamoorthy, G. & Sistla, S. M. (2023). Exploring Machine Learning Intrusion Detection: Addressing Security and Privacy Challenges in IoT-A Comprehensive Review. *Journal of Knowledge Learning and Science Technology*, *2*(2), 114-125. https://doi.org/10.60087/jklst.vol2.n2.p125

[12] Hulayyil, S. B., Li, S., & Xu, L. (2023). Machine-learning-based vulnerability detection and classification in internet of things device security. *Electronics*, *12*(18), 3927-3928. https://doi.org/10.3390/electronics12183927

[13] Alsharif, N. A., Mishra, S., & Alshehri, M. (2023). IDS in IoT using Machine Learning and Blockchain. *Engineering, Technology & Applied Science Research*, *13*(4), 11197-11203. https://doi.org/10.3390/electronics12183927

[14] Cherbal, S., Zier, A., Hebal, S., Louail, L., & Annane, B. (2024). Security in internet of things: a review on approaches based on blockchain, machine learning, cryptography, and quantum computing. *The Journal of Supercomputing*, *80*(3), 3738-816. https://doi.org/10.1007/s11227-023-05616-2

[15] Kaur, B., Dadkhah, S., Shoeleh, F., Neto, E. C., Iqbal, S., Lamontagne, P., Ray, S., & Ghorbani, A. A. (2023). Internet of things (IoT) security dataset evolution: Challenges and future directions. *Internet of Things*, *22*(6), 100780-100781. https://doi.org/10.1016/j.iot.2023.100780

[16] Fedotov, S., Lipatiev, A., Lipateva, T., & Lotarev, S. (2021). Femtosecond laser-induced birefringent microdomains in sodium-borate glass for highly secure data storage. *Journal of the American Ceramic Society*, *104*(9), 4297-4303. https://doi.org/10.1111/jace.17868

[17] Alhalabi, W., Al-Rasheed, A., Manoharan, H., Alabdulkareem, E., Alduailij, M., Alduailij, M., & Selvarajan, S. (2023). Distinctive measurement scheme for security and privacy in internet of things applications using machine learning algorithms. *Electronics*, *12*(3), 747-748. https://doi.org/10.3390/electronics12030747

[18] Usoh, M., Asuquo, P., Ozuomba, S., Stephen, B., & Inyang, U. (2023). A hybrid machine learning model for detecting cybersecurity threats in IoT applications. *International Journal of Information Technology*, *15*(6), 3359-3370. https://doi.org/10.1007/s41870-023-01367-8

[19] Hebbi, C. & Mamatha, H. (2023). Comprehensive Dataset Building and Recognition of Isolated Handwritten Kannada Characters Using Machine Learning Models. *Artificial Intelligence and Applications*, *1*(3), 179-190. https://doi.org/10.47852/bonviewAIA3202624

[20] Sarker, I. H., Khan, A. I., Abushark, Y. B., & Alsolami, F. (2023). Internet of things (iot) security intelligence: a comprehensive overview, machine learning solutions and research directions. *Mobile Networks and Applications*, *28*(1), 296-312. https://doi.org/10.1007/s11036-022-01937-3

[21] Sudharsanam S. R., Sivathapandi P., & Surampudi Y. (2023). Cloud-Based Telematics and Real-Time Data Integration for Fleet Management: A Comprehensive Analysis of IoT-Driven Predictive Analytics Models. *Journal of Artificial Intelligence Research and Applications*, *3*(1), 622-657.

[22] Ajao L. A., Olaniyi M. O., Agajo J, Olutoye M. A., & Ajao A. O. (2023). An Enhanced Telematics-Based Automobile Tracking and Volume Monitoring System for the Supply Chain Sustainability in the Petroleum Industry. *International Conference on Smart Technologies in Urban Engineering 2023*, *8*(6), 462-473. https://doi.org/10.1007/978-3-031-46874-2_40

[23] Truby J, Brown R. D., & Antoine Ibrahim I. (2024). Regulatory options for vehicle telematics devices: balancing driver safety, data privacy and data security. *International Review of Law, Computers & Technology*, *38*(1), 86-110. https://doi.org/10.1080/13600869.2023.2242671

[24] Kenfack P. D., Abana A. B., Tonye E., Ekam P. S., & Mbang G. H. (2023). Optimizing Telematics Network Performance through Resource Virtualization in a Disruptive Environment: The Case of the IP/MPLS Core Network. *Network and Communication Technologies*, *8*(2), 1-34. https://doi.org/10.5539/nct.v8n2p1

Contact information:

Ke XIANG
(Corresponding author)
Sichuan Post & Telecommunication College,
ChengDu, 610067, China
E-mail: xiangke@sptc.edu.cn

Xing YANG
Geely University of China,
641423, ChengDu, China
E-mail: 18980091368@163.com

Huihui WANG
Sichuan Post and Telecommunication College,
610067, ChengDu, China
E-mail: 17864808780@139.com