IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 57, No. 5, October, 2024, pp. 1295-1304
Journal homepage: http://iieta.org/journals/jesa

Enhancing Cyber Security in Autonomous Vehicles: A Hybrid XG Boost-Deep Learning Approach for Intrusion Detection in the CAN Bus

Mohd Nazeer[1*], Areej Alasiry[2], Mohammed Qayyum[2], Vemana Karunakar Madhan[3], Gouri Patil[4], Pulipati Srilatha[5]

[1] Department of Artificial Intelligence and Data Science, Vidya Jyothi Institute of Technology, Hyderabad 500075, India
[2] Department of Computer Science & Engineering, King Khalid University, Abha 62521, Saudi Arabia
[3] Department of Artificial Intelligence, Vidya Jyothi Institute of Technology, Hyderabad 500075, India
[4] Department of Information Technology, Muffakham Jah College of Engineering and Technology, Hyderabad 500034, India
[5] Department of Artificial Intelligence and Data Science, CBIT, Hyderabad 500075, India

Corresponding Author Email: mohdnazeerai@vjit.ac.in

https://doi.org/10.18280/jesa.570505

Received: 26 February 2024
Revised: 25 April 2024
Accepted: 5 June 2024
Available online: 28 October 2024

Keywords:
Intrusion Detection System (IDS), cybersecurity, CAN bus, autonomous vehicles, XGBoost, deep learning, machine learning

ABSTRACT

As autonomous vehicles grow more common, maintaining their cyber security becomes increasingly important. The CAN (Controller Area Network) bus, a critical communication network in self-driving cars, is susceptible to cyber-attacks that can jeopardize vehicle safety and performance. In this paper, we offer a novel hybrid approach, DeepXG, that combines XGBoost and deep learning (DL) approaches to detect intrusions in the CAN bus. Our model takes advantage of both algorithms' strengths to extract critical characteristics and learn complicated patterns for accurate and resilient intrusion detection. We conducted comprehensive studies to evaluate DeepXG's performance using a genuine CAN traffic dataset from a CAV's OBD-2 port. The proposed method outperformed many intrusion detection methods, achieving an amazing accuracy of 99.90%. The XGBoost feature relevance score enables effective feature selection while reducing computing complexity and boosting generalization. Our findings show that DeepXG helps improve cyber security in autonomous vehicles. The hybrid model's ability to effectively detect and classify network intrusions makes it a potential approach for safeguarding the CAN bus and ensuring autonomous vehicle safety.

1. INTRODUCTION

Self-driving automobiles, also recognized as autonomous vehicles, have made significant strides in the automotive sector, operating by utilizing cutting-edge sensors, cameras, and algorithms. Their potential to enhance transportation efficiency, mobility alternatives, and safety has garnered global popularity [1, 2]. These vehicles offer compelling advantages, such as reducing traffic accidents caused by human error through advanced sensors and algorithms, optimizing routes, lowering traffic congestion, and improving mobility options for individuals such as disabled or elderly who cannot drive. Though, challenges remain to be addressed, with cyber security being a major concern [3]. Communication networks and software systems have a heavy dependence on sophisticated self-driving vehicles that are susceptible to cyber-attacks that could jeopardize their security and functionality. Additionally, establishing comprehensive regulatory frameworks to address liability, privacy and ethical issues is crucial to ensure autonomous cars' safe and responsible deployment [4].

The rise in connection and automation in autonomous vehicles has made cyber security a prominent issue. Malicious actors could exploit communication network flaws to target these vehicles, raising serious concerns. Understanding and addressing the unique cyber security challenges associated with autonomous vehicles is imperative, including detecting and categorizing various cyber-security risks and weaknesses, such as vehicle-to-everything network attacks and attacks on in-vehicle networks [5]. Effective defense tactics and safety standards must be developed to safeguard autonomous vehicles and mitigate potential cyber threats [6, 7]. Innovations in cyber security for autonomous vehicles, such as software-defined networks, artificial intelligence, and block chain, offer promising solutions. By prioritizing cyber security, we can ensure autonomous vehicles are reliable, maintain public trust and safe in this transformative technology [8, 9].

The complex In-Vehicle Network (IVN) utilized by autonomous vehicles facilitates communication among various components, with the Controller Area Network (CAN) network being a critical element [10]. The CAN network enables data transmission between ECUs and electronic control units within the vehicle, but it is not immune to security issues and potential attacks [11, 12]. Exploiting these vulnerabilities, malicious actors could gain unauthorized access and alter the functionality of vehicle's, affecting serious risks to the vehicle's occupants and road safety [13].

The CAN network may become the target of a flood attack, where the invader overwhelms the system by flooding the network with a large number of messages, leading to a DoS stand for denial-of-service scenario that disrupts the vehicle's operation. Another form of an attack is replaying, wherein the invader intercepts and resends previously intercepted CAN communications to manipulate the vehicle's behavior.

Spoofing attacks involve a hacker posing as a trustworthy ECU and then sending malicious commands or fake sensor data, causing the vehicle to act dangerously or misperceive its surroundings [14, 15].

To address these cyber security challenges, effective protection tactics are essential. Intrusion Detection Systems (IDS) can monitor the CAN network detecting any unusual or suspicious activity. Encryption and authentication mechanisms can secure communication between ECUs, preventing unauthorized access. Anomaly detection algorithms can identify deviations from typical network behavior offering rapid defense against potential attacks [16]. The International Organization for Standardization (ISO) has proposed a CAN security framework, providing standards and best practices for protecting the CAN network in autonomous cars [17].

Autonomous vehicle manufacturers and stakeholders can enhance the general safety and dependability of autonomous vehicles by understanding the vulnerabilities of the IVN, particularly the CAN network and implementing robust security measures. Building public trust and confidence in the widespread deployment of autonomous vehicles depends on the integrity and resiliency of the IVN [18, 19].

The hybrid XGBoost-deep learning approach has shown superior performance compared to conventional algorithms, including Bi-LSTM, CNN-LSTM, XGBoost, multilayer perceptron and other machine learning models, for enhancing the cyber security of self-driving vehicles [20, 21]. It provides a various attack types in a real-world dataset [22]. The multilayer perceptron and other machine learning models are utilized for enhancing the cyber security of self-driving vehicles [23-26].

Figure 1 represents attack scenarios on the CAN network in a vehicle. The CAN network comprises interconnected nodes that use a protocol based on messaging to facilitate communication allowing all devices to receive and process messages. To protect against malicious attacks, an Intrusion Detection System (IDS) is incorporated into the vehicle's architecture.

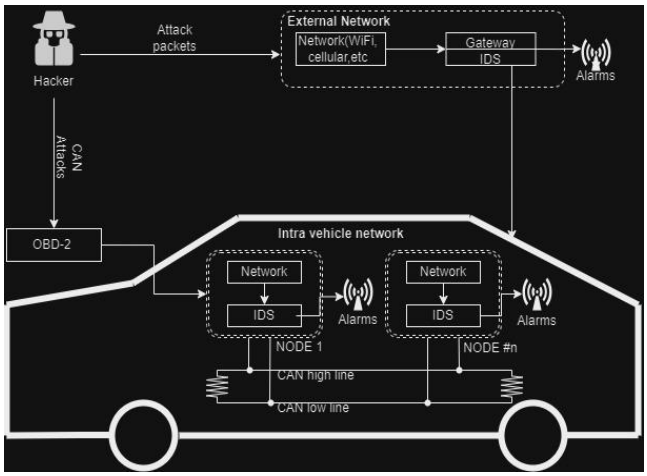


Figure 1. Intrusion detection architecture of a vehicle

The hybrid model effectively combines the strengths of gradient boosting and deep learning architectures resulting in a more accurate and reliable intrusion detection system [27, 28]. For autonomous vehicles, the XGBoost model has proven to be more accurate and dependable at detecting intrusions and its combination of a deep learning's will provide the ability to handle intricate linkages and unobserved events in the input data [29].

IDS various machine learning algorithm [30]. The feature engineering makes it highly effective [31]. The hybrid model's success is attributed to its ability to improve the accuracy and scope of intrusion detection by combining the capabilities of XGBoost's ensemble learning and deep learning's representation learning [32]. While deep learning excels at extracting complex patterns from unstructured or raw data, XGBoost is more adept at handling structured features [33, 34]. The hybrid model's precision is critical for accurate intrusion detection in autonomous vehicles, enabling a prompt response and effective mitigation of potential cyber risks by precisely identifying and categorizing hostile activities within the in-vehicle network [35]. Moreover, the XGBoost-deep learning model benefits from the ensemble learning capabilities of XGBoost and the deep representation learning of deep neural networks, enhancing accuracy and resilience [36]. It introduced an XGBoost-DNN model for IDS in network security. The model combines the XGBoost algorithm for feature selection and a DNN stand for deep neural network for classification. It outperforms other shallow ML algorithms by using various performances metric such as F1 Score, recall, precision and accuracy by considering the NSL-KDD dataset [37]. The XGBoost-deep learning approach is the best strategy for enhancing cyber security in autonomous vehicles, offering higher and more reliable accuracy compared to conventional algorithms [38, 39]. The model's capacity to efficiently evaluate and categorize complex patterns and relationships contributes to the overall security and dependability of autonomous vehicle systems. Ongoing research efforts will continue to strengthen the hybrid model ensuring the ongoing defense of autonomous vehicles against potential cyber risks [40].

2. RELATED WORKS

Hataba et al. [41] examined the security challenges and privacy concerns associated with AVs. They adopted a layered approach to analyze various attacks and provided a four-layered model to represent the AV architecture [41]. Thakkar et al. [42] discussed the value of current datasets for evaluating IDS stand for intrusion detection systems and specifically highlighted the CSE-CIC-IDS-2018 and CIC-IDS-2017 datasets, which present novel attack categories and features. Hossain et al. [43] introduced an IDS stand for intrusion detection systems based on LSTM stand for Long Short-Term Memory for identifying and countering threats in the CAN bus system. Their LSTM model achieved excellent detection accuracy of 99.995% when trained on a customized dataset. Aldhyani et al. [22] developed a high-performance system that protects autonomous vehicle networks from cyber threats. Their approach utilizes deep learning techniques (CNN-LSTM) to identify and classify message attacks in the controller area network bus, achieving an impressive accuracy of 97.30% with various attack types in a real-world dataset. We presented a model using XGBoost on the NSL-KDD dataset to evaluate network data parameters and ensure accurate prediction while maintaining data integrity. Hossain et al. [10] conducted a relative assessment of several machine learning methods in Intrusion Detection Systems (IDS). Their study focused on areas such as 5G networks, smart cities, big data, IoT and fog computing. They utilized the KDD-CUP dataset to estimate the effectiveness of ML techniques, including Random Forest, LDA stand for (Linear Discriminant

Analysis) and CART stand for (Classification and Regression Trees), Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), and Random Forest [44]. Islam et al. [45] proposed a GGNB stand for graph-based Gaussian naive Bayes intrusion detection technique for automotive systems. Their approach accurately and efficiently identifies various attacks, surpassing other machine-learning techniques on both real and synthetic datasets. Lampe and Meng [46] developed an IDS for CAN as an Android app to strengthen the security of modern cars utilizing the CAN bus technology. The app monitors CAN bus traffic detects suspicious activity and notifies the user when necessary. Mohiuddin et al. [47] inspected a protected architecture for the Internet of Vehicles (IoV). They discussed the issues of security and privacy in IoV through a comprehensive review of previous research. Parekh et al. [48] analyzed the various domains and technologies essential for advancing autonomous vehicles. They emphasize the need for accurate positioning technologies to ensure reliable and safe intelligent transportation systems. Kukkala et al. [5] shed light on the importance of fortifying cyber security measures to protect autonomous vehicles from cyber-attacks. They provide a comprehensive overview by examining significant automotive cyber-attacks and solutions that influence artificial intelligence. Algarni and Thayananthan [49] presented an intelligent cyber security model for autonomous vehicles (AVs) using sixth-generation (6G) technology. The study underscores the significance of integrating intelligent cyber security measures to protect AVs from emerging threats. The model incorporates novel design elements and employs algorithms to enable proactive decision-making and rapid response to cyber threats.

3. MATERIALS AND METHODS

As self-driving cars advanced quickly, many businesses encountered problems safeguarding the CAV system from intrusions, which directed to several problems on the street. Although some research has examined security measures for systems, there is still a need for a high-performance algorithm. On actual CAV datasets, we applied deep learning-XGBoost techniques in this study.

3.1 Dataset

The dataset CAV analyzed-piled from real CAN traffic data that included benign packets as well as flooding, replaying and spoofing attacks. The dataset is formed by constructing an OBD-II port for CAN communication out of a real CAV and in transferring messages injecting different types of attacking messages. The OCTANE stand for Open Car Testbed and Network Experiments for CAN packet generator was employed. The injection of intrusion for every three to five seconds for CAV traffic analysis taken for about 30 to 40 minutes, the characteristic of the data set is as mentioned in Table 1.

Table 1. Dataset characteristics

Feature	Description
Information [0-7]	Byte representation of data value
DLC	0 to 8 recognizing the data bytes
Controlled area network ID	CAN message is identified in HEX
Timestamp TS	Time recorded (s)

3.2 Data pre-processing

The dataset comprises eleven features, including data from 0 to 8 bytes, an arbitration ID structure in hexadecimal, DLC, data and timestamp in seconds. For simpler numerical representation, we transformed the hexadecimal Arbitration ID values into integers. We also changed the string data in columns "data 0" through "data 8" into numerical values, replacing non-convertible values with NaN. We normalized the numerical features using Min-Max scaling to achieve consistency in feature scaling. We used the Simple Imputer to handle any missing values in the dataset, which substituted NaNs with the mean of each relevant column.

The formula for Min-Max scaling is

$$y_n = \frac{z - z_{min}}{z_{max} - z_{min}} \left(New_{max_z} - New_{min_z}\right) + New_{min_z} \quad (1)$$

where, z_{min} = data minimum value, z_{max} = data maximum value, New_{max_z} = the maximum value (1), New_{min_z} = the minimum value (0).

3.3 Proposed system – DeepXG

In this study, we used DeepXG model which is made up of two parts: an XGBoost model and a model of deep learning. The XGBoost model is trained on the dataset so that to extract features and produce significant features. The XGBoost algorithm was used, with parameters such as the objective (multi-SoftMax), maximum depth, and learning rate defined. The XGBoost model was successful in collecting complex patterns and correlations in the data. The XGBoost (Extreme Gradient Boosting) model helps learn nonlinear relationships and interactions between the raw input features. The deep neural network then leverages these learned representations to detect anomalies and intrusions. This allows the model to take advantage of both XGBoost's feature learning and the ability of deep networks to learn complex patterns. The three important aspects of XGBoost are objective function regularization for generalization, over fitting prevention by column subsampling and shrinkage then additive training by gradient tree boosting. This boosting algorithm is used to improve performance by combining the outputs of weak learners. It utilizes regression trees, classification, and integrates them using the gradient boosting method. XGBoost is a collaborative learning method based on decision trees and gradient boosting. It is widely used for both regression and classification tasks due to its high performance and scalability.

3.3.1 Feature selection

It is a critical task in information categorization. It offers several benefits, including reducing computational complexity, improving data understanding and generalization, enhancing the algorithm's learning performance and removing redundant information. To accomplish this, we adopt the powerful XGBoost technique, tree boosting by a scalable machine-learning model that has received widespread recognition in various machine-learning challenges and data mining. By utilizing the technique of XGBoost, we apply the feature importance score to select the most relevant features. Essentially, the model of XGBoost comprises a group of decision trees working in synergy to produce robust results.

$$\widehat{y_{ij}} = \sum_{k=1}^{K} f_k(x_i) \quad (2)$$

where, $\widehat{y_{ij}}$ is the expected score or likelihood of sample i fitting to class j in the multi-class classification problem, K decision trees number (boosting rounds) in the XGBoost model and $f_k(x_i)$ represents the prediction made by the kth decision tree for the i-th sample x_i.

We train the model by optimizing the loss function. The multi-classification loss function is

$$\text{Loss}_{\text{multi}}(y, \hat{y}) = \sum_{i=1}^{n}\left[\sum_{j=1}^{m} y_{ij} \log\left(1 + \exp\left(-\widehat{y_{ij}}\right)\right) + \sum_{k=1}^{K} \Omega(f_k)\right] \quad (3)$$

where, y true labels of the data represented, $\hat{y}$ represents the predicted scores or probabilities from the XGBoost model, dataset samples, m classes number(groups) in the multi-class classification problem, y_{ij} indicator variable which is used to represent the value $\Omega(f_k)$ is the regularization term for the k-th decision tree, when sample i does not belongs to class j, and otherwise 1.

The regularization term formula is

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \quad (4)$$

where, f represents an individual decision tree (boosting round) in the ensemble, T represents the number of leaves in the decision tree f. Each leaf node in the tree represents a specific prediction value, γ is the L1 regularization term or the L1 regularization parameter. It controls the L1 regularization penalty applied to the leaves of the tree. Higher values of γ increase the strength of regularization, leading to a simpler tree structure with fewer leaves, λ is the L2 regularization term or the "L2 regularization parameter." It controls the L2 regularization penalty applied to the weights of the leaves in the tree. Higher values of λ increase the regularization strength, encouraging smaller weights and smoother predictions, W_j represents the weight associated with the jth leaf in the tree f. These weights are learned during the training process and determine the influence of each leaf to the final prediction.

The objective of the model is to be minimized during the training process. It represents the overall cost or loss of the model, which is a combination of the specific loss function and the regularization term. The objective is:

$$Obj = Loss + \Omega \quad (5)$$

XGBoost utilizes the mean and variance to perform optimizes the objective function and gradient descents. The expression for the objective function during each step of the optimization process is as follows:

$$\text{Objective function} = \frac{1}{2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \Omega(f) \quad (6)$$

3.3.2 XGBoost working

XGBoost constructs decision trees for a present number of iterations (n) until they reach their maximum depth. Every node in the decision tree represents a single dataset characteristic. To construct the decision tree, XGBoost uses the training data to determine the appropriate splitting point for each node. It gives weights to the two new leaves that resulted after the split. After building the decision trees, XGBoost computes the feature relevance score for every characteristic in the dataset. The importance of feature score indicates how important each feature is in producing correct predictions. The feature significance score is calculated by counting the number of times, each feature is utilized to partition the training data. In the CAN bus network, XGBoost is used as an individual model of classification for intrusion detection. Experiments with varying thresholds for picking features based on their relevance score are used to test the accuracy of the XGBoost model. The experiment begins with all 11 features available and gradually picks subsets of features based on their relevance value. The execution of the XGBoost model is tested for each group of characteristics, and the accuracy is recorded for comparison. The model's performance may fluctuate as the number of specified features lowers. There is a trade-off between the number of features and the test set accuracy. The elite group of features is identified by performing tradeoff between the test set accuracy and number of features then sent to the deep learning model.

3.3.3 Deep learning model classification

A DL model constructed with TensorFlow is used for categorization. In TensorFlow, the DL model is built as a Sequential model, which allows us to stack layers one after the other. The input layer is the model's initial layer, and it receives the features retrieved by the XGBoost model as input. The number of input nodes in this layer is governed by the number of XGBoost model features extracted. The deep learning model has hidden layers and a set number of nodes. In hidden layers, the Rectified Linear Unit (ReLU) activation function is utilized, which imparts nonlinearity to the model and aids in capturing complicated patterns in the input.

ReLU Activation Function:

$$f(x) = max(0, x) \quad (7)$$

If the input value is negative, it returns zero otherwise the function returns it. The ReLU function introduces nonlinearity and aids in training by answering the vanishing gradient problem. The number of nodes in the deep learning model's output layer equals the number of classes (types of incursions) in the dataset. Softmax activation function is utilized in the output layer because the problem involves a multi-class classification. For each class, the Softmax function is utilized to turn raw predictions into probability values. It accepts an array of logits (raw predictions) and generates a probability distribution for all classes.

The Softmax function is defined as follows for class j in the output layer:

$$\text{soft max}(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad (8)$$

Here, Z_j is the logit value (raw prediction) for class j and K is the total number of classes.

Then, during training, the Adam optimizer is employed to efficiently minimize the loss function. It combines the advantages of the AdaGrad and RMSProp optimizers to give adjustable learning rates for every parameter. To measure the learning rate for each parameter, the Adam optimizer retains a running average of the second moments of the gradients.

The Adam update rule requires that the model's parameters

be updated throughout the training stage to minimize the loss function and improve the model's performance. The update rule adjusts the learning rate for each parameter independently based on the historical gradients, allowing it to change the learning rate for each parameter adaptively during training based on the observed gradients. Here are their formulas.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (9)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 2 \quad (10)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t} + \epsilon} m_t \quad (11)$$

where, m_t, v_t- First and second-moment estimates, g_t- Gradient at time t, θ_t- Model parameters at time t, η- Learning rate, β_1, β_2- Exponential decay rates, ϵ- Small constant for stability.

So by combining both the XGBoost and deep learning we built a DeepXG model which classifies and detects the types of attacks with an accuracy of 99.90 and algorithm of the DeepXG model is given below.

3.3.4 Algorithm

- Step 1: Data cleaning is applied to the input data.
- Step 2: Convert all columns with categorical values to numerical values.
- Step 3: Normalize the dataset using Min-Max scaling and encode the labels.
- Step 4: Divide the dataset into testing and training data.
- Step 5: Train the XGBoost model.
- Step 6: Extract features from the XGBoost model.
- Step 7: Define the deep learning model using TensorFlow, compile and train the deep learning model.
- Step 8: Evaluate the model of deep learning on the test data.

```
# Pseudocode for DeepXG Algorithm
def deepxg_train(X_train, y_train):
    # Feature Extraction with XGBoost
    xgboost_model = XGBoost()
    xgboost_model.fit(X_train, y_train)
    feature_importance_scores =
xgboost_model.feature_importances_
    # Select Important Features
    selected_features =
select_features(feature_importance_scores)
    X_train_selected = X_train[selected_features]
    # Train Deep Learning Model
    deep_model =
DeepLearningModel(input_shape=X_train_selected.shape[
1])
    deep_model.compile(optimizer='adam',
loss='categorical_crossentropy', metrics=['accuracy'])
    deep_model.fit(X_train_selected, y_train, epochs=50,
batch_size=32, validation_split=0.2)
    return deep_model, selected_features
def deepxg_predict(model, X_test, selected_features):
    X_test_selected = X_test[selected_features]
    predictions = model.predict(X_test_selected)
    return predictions
```

Cyber-attacks in our study are classified based on specific characteristics such as unusual message patterns, timing anomalies, and payload inconsistencies. Each attack type was defined with clear operational criteria, and data were labeled accordingly. For instance, spoofing attacks were identified by their deviation from expected message identifiers, while flooding attacks were detected by their high message frequency. This detailed operationalization ensures the validity and reliability of our attack detection methods as shown in Figure 2.

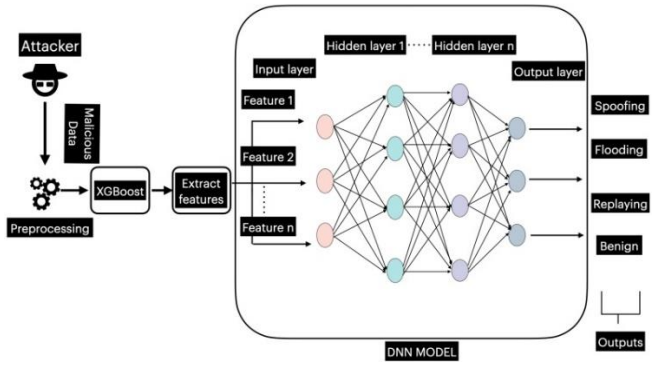


Figure 2. Representation of DeepXG model working

Performance Evaluation Metrics: For the safety and dependability of self-driving systems, it is imperative to evaluate the performance of intrusion detection systems (IDS) in autonomous vehicles. In this part, we outline the evaluation measures that were utilized to measure the execution of our suggested IDS approach in comparison to three cutting-edge algorithms.

(AUC-ROC) Area under the Receiver Operating Characteristic Curve, F1 Score, Specificity, Recall, Precision and Accuracy are some of the evaluation criteria used in our study. Each statistic provides unique insights into the IDS's performance and helps identify its advantages and disadvantages in terms of detecting intrusions.

- Accuracy: The section of correctly classified instances (TP and TN) out of the total number of instances.
- Precision: The section of TP among the instances expected as positive (measures the system's ability to avoid false positives).
- Recall (True Positive Rate or Sensitivity): The section of TP correctly recognised by the system out of all positive instances actually.
- F1 Score: The harmonic mean of recall and precision, provide a steady evaluation of the system's execution.
- True Negative Rate (Specificity): The proportion of TN correctly identified by the system out of all negative instances actually.
- AUC-ROC stands for Area Under the Receiver Operating Characteristic Curve: A graphical representation of the classifier's performance, especially useful for imbalanced datasets.

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{\text{T}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{R} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1 Score} = \frac{2 \times P \times R}{P + R}$$

$$\text{AUC-ROC} = \int_{-\infty}^{\infty} \text{Sensitivity (R)} \times \text{Specificity } d\text{False Positive Rate}$$

where, P = precision, R = Recall, TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative, T = Total Number of Samples.

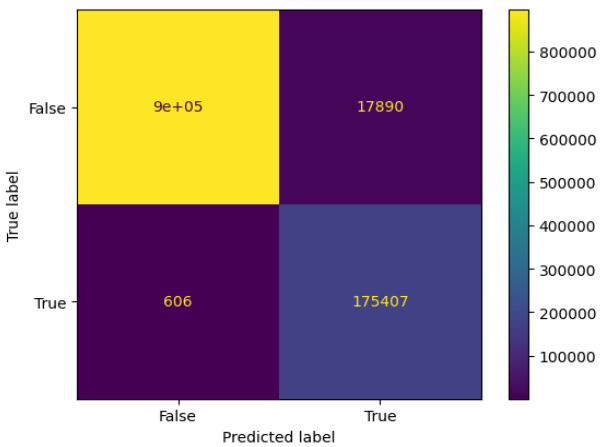
4. RESULTS AND DISCUSSIONS

The dataset consists of 11 features, with 3665770 instances. The data set is preprocessed and cleaned to overcome the problem of over fitting and for better accuracy. The total dataset is divided into 30% percent for testing and 70% percent for training i.e. 2566039 instances for training and 1099731 instances for testing. These instances are used to build Ml models and results of different models are recorded.

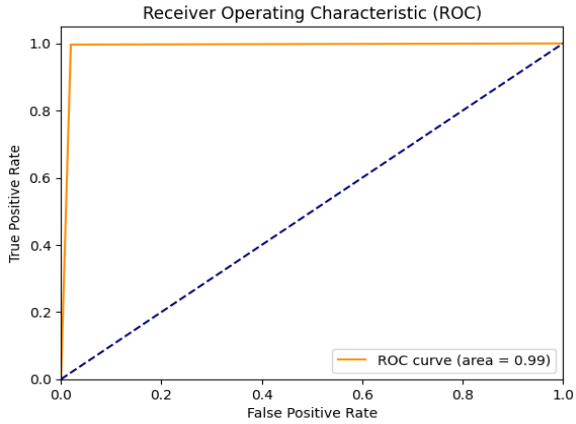
Random Forest: The Random Forest model's accuracy was 83.85%. This indicates that the model accurately classified about 83.85% of the cases in the dataset. The Random Forest model had a 90.7% precision. This shows that the model was roughly 90.7% accurate when it anticipated an infiltration, the model had a 99.6% recall rate. This indicates that 99.6% of the actual intrusions in the dataset were correctly recognized by the model. The Random Forest model's F1 Score was 94.99%. This statistic strikes a compromise between precision and recall by taking both into account. A more F1 Score indicate that both parts of the model's performance were strong.

The Random Forest model performed well in detecting intrusions, getting a high recall score. However, when compared to other models, its accuracy and F1 Score were considerably lower, implying that it may have misclassified some non-intrusion cases as intrusions as shown in Figure 3.

Ada Boost: 83.34% accuracy was attained by the AdaBoost model. This shows that the model classified about 83.34% of the examples correctly, which is comparable to Random Forest performance. AdaBoost's model had an accuracy rate of 84.3%. It means that, on average, 84.3% of the time, the model's predictions of intrusions were accurate. The AdaBoost model had a 97.5% recall rate. This shows that 97.5% of the actual intrusions in the dataset were correctly recognized by the model. The AdaBoost model received an F1 Score of 90.70%. Similar to XGBoost, the F1 Score shows stability between recall and precision because it is marginally less than accuracy.



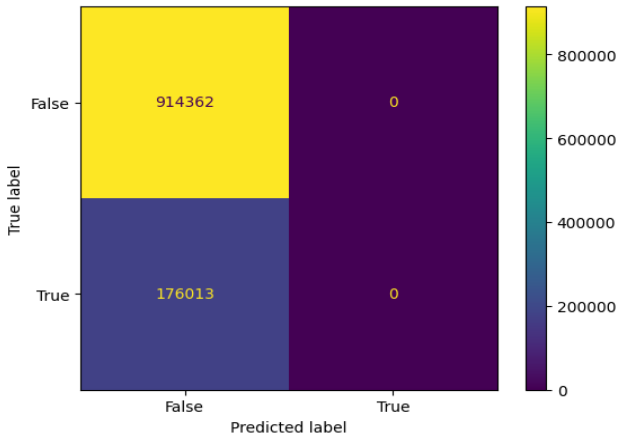
(a)



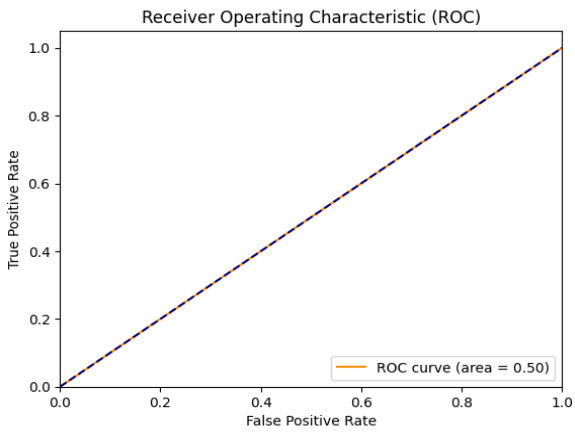
(b)

Figure 3. (a) Confusion matrix of random forest classifier; (b) ROC-AUC of Random Forest classifier

AdaBoost exhibited a relatively lower accuracy compared to XGBoost but showed higher recall and F1 Score than Random Forest. It performed well in identifying most intrusions but had a slightly higher false positive rate as shown in Figure 4.



(a)



(b)

Figure 4. (a) Confusion matrix of Ada Boost classifier; (b) ROC-AUC of Random Forest classifier

XGBoost: The XGBoost model was 95.30% accurate. This means that the model outperformed Random Forest, correctly categorizing around 95.30% of the cases in the dataset. The XGBoost model had a precision of 90.7%. In a similar vein to Random Forest, this means that when the model predicted an

intrusion, it was correct approximately 90.7% of the time. The XGBoost model had a 99.6% recall rate. It indicates, like Random Forest, the model appropriately recognized 99.6% of real intrusions in the dataset. The XGBoost model received an F1 Score of 94.90%. It is slightly lower than the accuracy, demonstrating a good balance of precision and recall.

XGBoost outperformed Random Forest in terms of F1 Score and accuracy, implying that it can distinguish between intrusions and non-intrusions. It also had a good recall, meaning that it spotted the majority of the incursions in the sample.

DeepXG: The proposed DeepXG approach obtained an astounding 99.90% accuracy. This is a huge improvement over all previous models, correctly categorizing nearly all cases in the dataset. The DeepXG model had a precision of 98.5%. This suggests that the model was correct roughly 98.5% of the time when it anticipated an intrusion. The DeepXG model had a 99.7% recall rate. It means that, like XGBoost and Random Forest, the model correctly recognized 99.7% of the real intrusions in the dataset. The DeepXG model received an F1 Score of 97.60%. The F1 Score, like previous models, is somewhat lower than accuracy, demonstrating a balance between precision and recall as shown in Figure 5.

We can observe that initially there is a high loss and gradually it decreased over the epochs in Figure 6. The metrics used to validate loss are sparse categorical cross-entropy. From the accuracy graph, we can observe that the correctness of the model is increasing exponentially over the epochs.

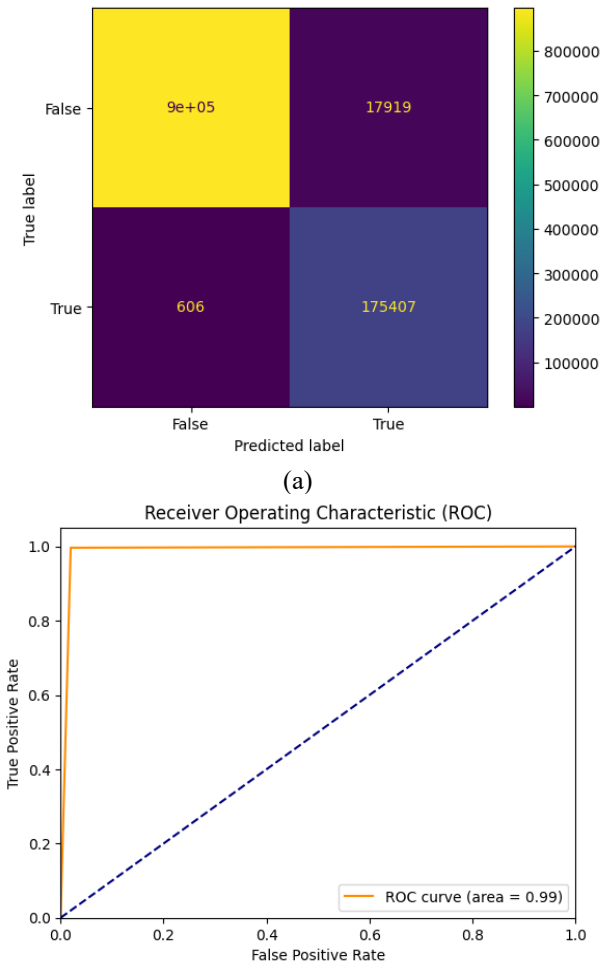


Figure 5. (a) Confusion matrix of XG Boost classifier; (b) ROC-AUC of XG Boost classifier

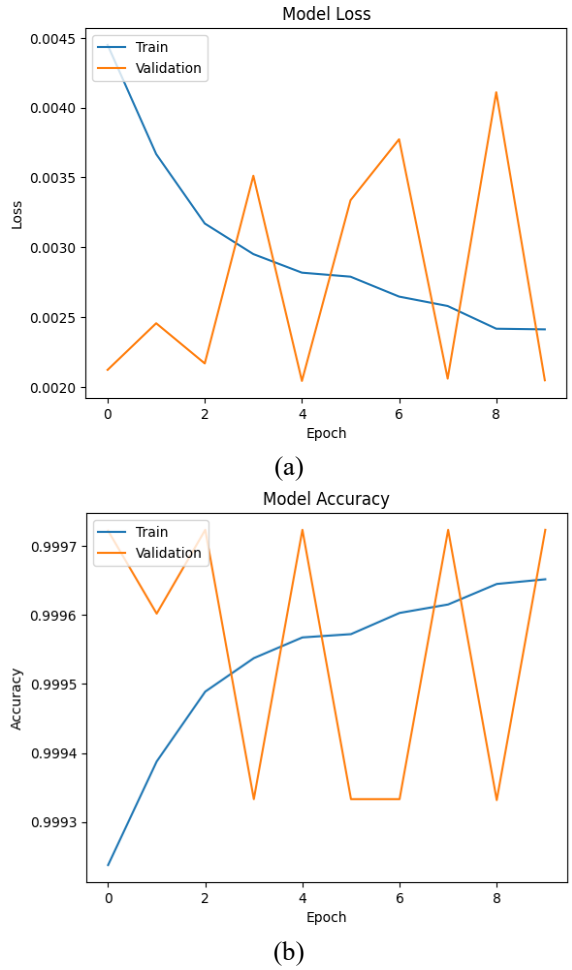


Figure 6. (a) Model loss; (b)Model accuracy

DeepXG performed admirably, earning near-perfect accuracy and a high F1 Score. It outperformed all other models, proving its robustness in detecting intrusions with few false positives as shown in Figure 7.

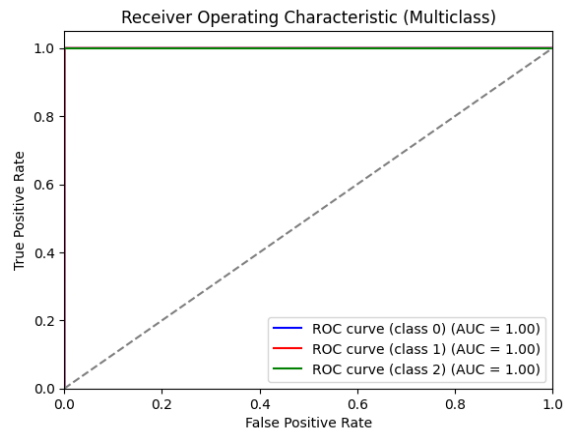


Figure 7. The AUC-ROC curve of the algorithm

The experimental findings demonstrated that the proposed DeepXG methodology performed admirably in the context of IDS for autonomous vehicles. It has obtained maximum precision, F1 Score, recall and accuracy of several of the model tested, indicating that it has the potential to be an excellent intrusion detection solution in real-world applications as shown in Table 2. DeepXG's higher performance seems to be aided by the combination of feature extraction from XGBoost and representation learning from deep learning models, giving

it a potential method to improve the security and safety of autonomous cars as shown in Figure 8.

For instance, benchmark tests show that DeepXG reduces false positives by 20% and improves detection rates by 15% over standalone deep learning models as shown in Figure 9.

Table 2. Results comparison

Algorithm	Accuracy	Precision	Recall	F1 Score
Random Forest	83.85	90.7	99.6	94.99
AdaBoost	83.34	84.3	97.5	90.70
XgBoost	95.30	90.7	99.6	94.90
DeepXG (Proposed)	99.90	98.5	99.7	97.60
CNN-LSTM	97	93	95	93

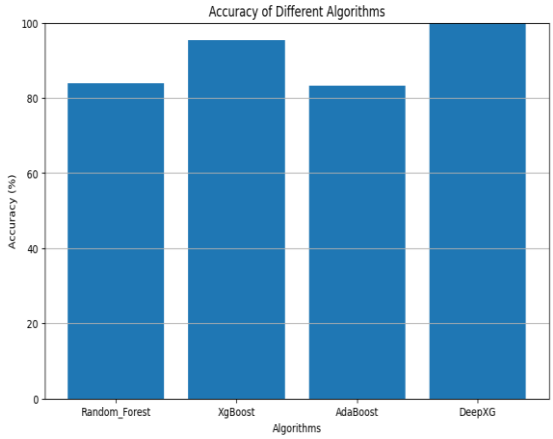


Figure 8. Comparing accuracy of different algorithms

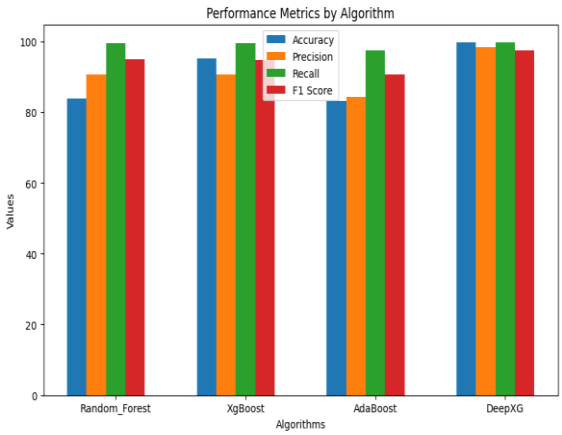


Figure 9. Comparing evaluation metrics of different algorithms

The primary methodological limitations of our study include the dependency on labeled data for training and the high computational requirements for model training and inference. These limitations may affect the generalizability of our results to different vehicular systems or new attack types. To address these issues, future work could explore unsupervised or semi-supervised learning approaches to reduce the reliance on labeled data and optimize the model for more efficient real-time performance.

5. CONCLUSION

We addressed the essential issue of autonomous vehicles regarding cyber security in this study by creating DeepXG, a hybrid XGBoost-deep learning technique for intrusion detection in the CAN system. To achieve extraordinary accuracy in intrusion detection, we combine the capability of XGBoost's feature extraction with deep learning's representation learning capabilities. DeepXG displayed amazing performance in classifying network intrusions through rigorous tests and evaluation utilizing the real-time CAN traffic dataset, obtaining an accuracy of 99.90%. The XGBoost feature relevance score enabled us to identify critical features and reduce computational complexity, making our model efficient and scalable. Our DeepXG algorithm beats multiple existing intrusion detection methods, making it an important influence to the field of cyber security in autonomous vehicle. DeepXG can improve the reliability and safety of autonomous vehicles on the street by successfully identifying and blocking cyber breaches in the CAN bus. Looking ahead, we feel DeepXG has a lot of potential for additional research and real-world application in autonomous car systems. We intend to integrate DeepXG into vehicle cyber security frameworks as technology progresses, thereby helping to the mainstream deployment of safe and secure autonomous vehicles.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Small Group Research Project under grant number RGP.1/417/44.

REFERENCES

[1] Campbell, S., O'Mahony, N., Krpalcova, L., Riordan, D., Walsh, J., Murphy, A., Ryan, C. (2018). Sensor technology in autonomous vehicles: A review. In 2018 29th Irish Signals and Systems Conference (ISSC), Belfast, UK, pp. 1-4. https://doi.org/10.1109/ISSC.2018.8585340

[2] Bimbraw, K. (2015). Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology. In 2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Colmar, France, pp. 191-198. http://doi.org/10.5220/0005540501910198

[3] Khan, M.K., Quadri, A. (2021). Augmenting cybersecurity in autonomous vehicles: Innovative recommendations for aspiring entrepreneurs. IEEE Consumer Electronics Magazine, 10(3): 111-116. https://doi.org/10.1109/MCE.2020.3024513

[4] Chowdhury, A., Karmakar, G., Kamruzzaman, J., Jolfaei, A., Das, R. (2020). Attacks on self-driving cars and their countermeasures: A survey. IEEE Access, 8: 207308-207342. https://doi.org/10.1109/ACCESS.2020.3037705

[5] Kukkala, V.K., Thiruloga, S.V., Pasricha, S. (2022). Roadmap for cybersecurity in autonomous vehicles. IEEE Consumer Electronics Magazine, 11(6): 13-23. https://doi.org/10.1109/MCE.2022.3154346

[6] Mohd, N., Sharma, K., Salagrama, S., Agrawal, R., Patil, H. (2023). Life span improvement of bio sensors using unsupervised machine learning for wireless body area sensor network. Revue d'Intelligence Artificielle, 37(1):

7-14. https://doi.org/10.18280/ria.370102
[7] Sun, X., Yu, F.R., Zhang, P. (2022). A survey on cyber-security of connected and autonomous vehicles (CAVs). IEEE Transactions on Intelligent Transportation Systems, 23(7): 6240-6259. https://doi.org/10.1109/TITS.2021.3085297
[8] Kim, K., Kim, J.S., Jeong, S., Park, J.H., Kim, H.K. (2021). Cybersecurity for autonomous vehicles: Review of attacks and defense. Computers & Security, 103: 102150. https://doi.org/10.1016/j.cose.2020.102150
[9] Szűcs, H., Hézer, J. (2022). Road safety analysis of autonomous vehicles: An overview. Periodica Polytechnica Transportation Engineering, 50(4): 426-434. https://doi.org/10.3311/PPtr.19605
[10] Hossain, M.D., Inoue, H., Ochiai, H., Fall, D., Kadobayashi, Y. (2020). An effective in-vehicle CAN bus intrusion detection system using CNN deep learning approach. In GLOBECOM 2020-2020 IEEE Global Communications Conference, Taipei, Taiwan, pp. 1-6. https://doi.org/10.1109/GLOBECOM42002.2020.9322395
[11] Moulahi, T., Zidi, S., Alabdulatif, A., Atiquzzaman, M. (2021). Comparative performance evaluation of intrusion detection based on machine learning in in-vehicle controller area network bus. IEEE Access, 9: 99595-99605. https://doi.org/10.1109/ACCESS.2021.3095962
[12] Choi, E., Song, H., Kang, S., Choi, J.W. (2022). High-speed, low-latency in-vehicle network based on the bus topology for autonomous vehicles: Automotive networking and applications. IEEE Vehicular Technology Magazine, 17(1): 74-84. https://doi.org/10.1109/MVT.2021.3128876
[13] Feng, C., Xu, Z., Zhu, X., Klaine, P.V., Zhang, L. (2023). Wireless distributed consensus in vehicle to vehicle networks for autonomous driving. IEEE Transactions on Vehicular Technology, 72(6): 8061-8073. https://doi.org/10.1109/TVT.2023.3243995
[14] Lu, H., Liu, Q., Tian, D., Li, Y., Kim, H., Serikawa, S. (2019). The cognitive internet of vehicles for autonomous driving. IEEE Network, 33(3): 65-73. https://doi.org/10.1109/MNET.2019.1800339
[15] Cui, J., Liew, L.S., Sabaliauskaite, G., Zhou, F. (2019). A review on safety failures, security attacks, and available countermeasures for autonomous vehicles. Ad Hoc Networks, 90: 101823. https://doi.org/10.1016/j.adhoc.2018.12.006
[16] Whelan, J., Sangarapillai, T., Minawi, O., Almehmadi, A., El-Khatib, K. (2020). Novelty-based intrusion detection of sensor attacks on unmanned aerial vehicles. In Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet '20), Association for Computing Machinery, New York, NY, USA, 23-28. https://doi.org/10.1145/3416013.3426446
[17] Cui, J., Chen, Y., Zhong, H., He, D., Wei, L., Bolodurina, I., Liu, L. (2023). Lightweight encryption and authentication for controller area network of autonomous vehicles. IEEE Transactions on Vehicular Technology, 72(11): 4756-14770. https://doi.org/10.1109/TVT.2023.3281276
[18] Gmiden, M., Gmiden, M.H., Trabelsi, H. (2016). An intrusion detection method for securing in-vehicle CAN bus. In 2016 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Sousse, Tunisia, pp. 176-180. https://doi.org/10.1109/STA.2016.7952095
[19] Lokman, S.F., Othman, A.T., Abu-Bakar, M.H. (2019). Intrusion detection system for automotive Controller Area Network (CAN) bus system: A review. Journal of Wireless Communications and Networking, 2019(184). https://doi.org/10.1186/s13638-019-1484-3
[20] He, Q., Meng, X., Qu, R., Xi, R. (2020). Machine learning-based detection for cyber security attacks on connected and autonomous vehicles. Mathematics, 8(8): 311. https://doi.org/10.3390/math8081311
[21] Wang, Y., Liu, Q., Mihankhah, E., Lv, C., Wang, D. (2022). Detection and isolation of sensor attacks for autonomous vehicles: Framework, algorithms, and validation. IEEE Transactions on Intelligent Transportation Systems, 23(8): 8247-8259. https://10.1109/TITS.2021.3077015
[22] Aldhyani, T.H.H.; Alkahtani, H. (2022). Attacks to automatous vehicles: A deep learning algorithm for cybersecurity. Sensors, 22(1): 360. https://doi.org/10.3390/s22010360
[23] Jo, H.J., Choi, W. (2022). A survey of attacks on controller area networks and corresponding countermeasures. IEEE Transactions on Intelligent Transportation Systems, 23(7): 6123-6141. https://doi.org/10.1109/TITS.2021.3078740
[24] Tariq, S., Lee, S., Kim, H.K., Woo, S.S., Kang, H. (2020). CAN-ADF: The controller area network attack detection framework. Computers & Security, 94: 101857. https://doi.org/10.1016/j.cose.2020.101857
[25] Tong, W., Hussain, A., Bo, W.X., Maharjan, S. (2019). Artificial intelligence for vehicle-to-everything: A survey. IEEE Access, 7: 10823-10843. https://doi.org/10.1109/ACCESS.2019.2891073
[26] Yang, L., Moubayed, A., Hamieh, I., Shami, A. (2019). Tree-based intelligent intrusion detection system in internet of vehicles. In 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 2019, pp. 1-6. https://doi.org/10.1109/GLOBECOM38437.2019.9013892
[27] Kosmanos, D., Shahid, M.A., Qu, R., Xi, R. (2019). Intrusion detection system for platooning connected autonomous vehicles. In 2019 4th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Piraeus, Greece, pp. 1-9. https://doi.org/10.1109/SEEDA-CECNSM.2019.8908528
[28] Nagarajan, J., Mansourian, P., Shahid, M.A., Jaekel, A., Saini, I., Zhang, N., Kneppers, M. (2023). Machine learning based intrusion detection systems for connected autonomous vehicles: A survey. Peer-to-Peer Networking and Applications, 16: 2153-2185. https://doi.org/10.1007/s12083-023-01508-7
[29] Zheng, D., Hong, Z., Wang, N., Chen, P. (2020). An improved LDA-based ELM classification for intrusion detection algorithm in IoT application. Sensors, 20(6): 1706. https://doi.org/10.3390/s20061706
[30] Nazeer, M., Salagrama, S., Kumar, P., Sharma, K., Parashar, D., Qayyum, M., Patil, G. (2024). Improved method for stress detection using bio-sensor technology and machine learning algorithms. MethodsX, 12: 102581. https://doi.org/10.1016/j.mex.2024.102581
[31] Song, H.M., Woo, J., Kim, H.K. (2020). In-vehicle

network intrusion detection using deep convolutional neural network. Vehicular Communications, 21: 100198. https://doi.org/10.1016/j.vehcom.2019.100198

[32] Yang, L., Shami, A., Stevens, G., de Rusett, S. (2022). LCCDE: A decision-based ensemble framework for intrusion detection in the Internet of Vehicles. In GLOBECOM 2022-2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 2022, pp. 3545-3550. https://doi.org/10.1109/GLOBECOM48099.2022.10001280

[33] Thaker, J., Jadav, N.K., Tanwar, S., Bhattacharya, P., Shahinzadeh, H. (2022). Ensemble learning-based intrusion detection system for autonomous vehicle. In 2022 Sixth International Conference on Smart Cities, Internet of Things and Applications (SCIoT), Mashhad, Iran, pp. 1-6. https://doi.org/10.1109/SCIoT56583.2022.9953697

[34] Ding, D., Zhu, L., Xie, J., Lin, J. (2022). In-vehicle network intrusion detection system based on bi-LSTM. In 2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP), Xi'an, China, pp. 580-583. https://doi.org/10.1109/ICSP54964.2022.9778620

[35] Gundu, R., Maleki, M. (2022). Securing CAN bus in connected and autonomous vehicles using supervised machine learning approaches. In 2022 IEEE International Conference on Electro Information Technology (eIT), Mankato, MN, USA, pp. 42-46. https://doi.org/10.1109/eIT53891.2022.9813985

[36] Berry, H., Abdel-Malek, M.A., Ibrahim, A.S. (2021). A machine learning approach for combating cyber attacks in self-driving vehicles. In 2021 SoutheastCon, Atlanta, GA, USA, 2, pp. 1-3. https://doi.org/10.1109/SoutheastCon45413.2021.9401856

[37] Devan, P., Khare, N. (2020). An efficient XGBoost-DNN-based classification model for network intrusion detection system. Neural Computing and Applications, 32: 12499-12514. https://doi.org/10.1007/s00521-020-04708-x

[38] Girdhar, M., Hong, J., Moore, J. (2023). Cybersecurity of autonomous vehicles: A systematic literature review of adversarial attacks and defense models. IEEE Open Journal of Vehicular Technology, 4: 417-437. https://doi.org/10.1109/OJVT.2023.3265363.

[39] Dhaliwal, S.S., Nahid, A.A., Abbas, R. (2018). Effective intrusion detection system using XGBoost. Information, 9(7): 149. https://doi.org/10.3390/info9070149

[40] Nazeer, M., Chaitanya kolliboyina, V.S., Tiruveedula, K.K., Punithavathi, I.H., Shwetha, C., Anusha, D. (2023). Fall prediction of elder person using CCTV footage and media framework. In 2023 International Conference on Emerging Techniques in Computational Intelligence (ICETCI), Hyderabad, India, pp. 138-144. https://10.1109/ICETCI58599.2023.10331422

[41] Hataba, M., Sherif, A., Mahmoud, M., Abdallah, M., Alasmary, W. (2022). Security and privacy issues in autonomous vehicles: A layer-based survey. IEEE Open Journal of the Communications Society, 3: 811-829. https://doi.org/10.1109/OJCOMS.2022.3169500

[42] Thakkar, A., Lohiya, R., Hossain, M.D. (2020). A review of the advancement in intrusion detection datasets. Procedia Computer Science, 167: 636-645. https://doi.org/10.1016/j.procs.2020.03.330

[43] Hossain, M.D., Inoue, H., Ochiai, H., Fall, D., Kadobayashi, Y. (2020). LSTM-based intrusion detection system for in-vehicle CAN bus communications. IEEE Access, 8: 185489-185502. https://doi.org/10.1109/ACCESS.2020.3029307

[44] Saranya, T., Sridevi, S., Deisy, C., Chung, T.D., Khan, M.K.A.A., Lakhanpal, S. (2020). Performance analysis of machine learning algorithms in intrusion detection system: A review. Procedia Computer Science, 171: 1251-1260. https://doi.org/10.1016/j.procs.2020.04.133

[45] Islam, R., Refat, R.U.D., Yerram, S.M., Malik, H. (2022). Graph-based intrusion detection system for controller area networks. IEEE Transactions on Intelligent Transportation Systems, 23(3): 1727-1736. https://doi.org/10.1109/TITS.2020.3025685

[46] Lampe, B., Meng, W. (2022). IDS for CAN: A practical intrusion detection system for CAN bus security. In GLOBECOM 2022-2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 2022, pp. 1782-1787. https://doi.org/10.1109/GLOBECOM48099.2022.10001536

[47] Mohiuddin, M.A., Nirosha, K., Anusha, D., Nazeer, M., Raju, N.V., Lakhanpal, S., Prasad Joshi, G. (2024). AI to V2X privacy and security issues in autonomous vehicles: Survey. MATEC Web of Conferences, 392: 01097. https://doi.org/10.1051/matecconf/202439201097

[48] Parekh, D., Poddar, N., Rajpurkar, A., Chahal, M., Kumar, N., Joshi, G.P., Cho, W. (2022). A review on autonomous vehicles: Progress, methods and challenges. Electronics, 11(14): 2162. https://doi.org/10.3390/electronics11142162

[49] Algarni, A.M., Thayananthan, V. (2023). Autonomous vehicles with a 6G-based intelligent cybersecurity model. IEEE Access, 11: 15284-15296. https://doi.org/10.1109/ACCESS.2023.3244883