e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

ATTACK DETECTION IN AUTONOMOUS VEHICLES BY USING MACHINE LEARNING MODELS

Anuradha*1, Dr. K.L Bansal*2

*1,2Himachal Pradesh University Summerhill, Shimla, India.

DOI: https://www.doi.org/10.56726/IRJMETS60022

ABSTRACT

The rise of autonomous vehicles (AVs) necessitates robust security measures to counter cyberattacks. This abstract explores the vulnerability of AVs to attacks targeting sensors, communication networks, and software. It emphasizes the potential of machine learning (ML) for intrusion detection in AVs. By analyzing sensor data, network traffic, and system logs, ML algorithms can identify anomalies indicative of attacks. This abstract underscores the significance of IDS in protecting critical data and maintaining a secure network environment. This paper explores the application of Random Forest classifiers in detecting these attacks, leveraging their robustness and ability to handle high-dimensional data. We present a detailed analysis of the methodology, dataset preparation, feature selection, and model evaluation. Our results demonstrate that the Random Forest classifier effectively identifies and classifies different types of attacks, providing a reliable tool for enhancing the security of autonomous vehicles. This explores the application of Intrusion Detection Systems (IDS) for safeguarding AVs against these threats. The abstract highlights prominent ML techniques like supervised learning (SVMs, Random Forests), unsupervised learning (Isolation Forests, LOF), and deep learning (LSTMs) for anomaly and attack detection. This paper explores the application of machine learning (ML) algorithms for intrusion detection in AVs and concludes by highlighting research challenges and future directions in this critical field.

I. INTRODUCTION

ANNs consist of interconnected nodes, neurons, arranged in layers. ANNs process information through weighted connections, enabling them to learn and make predictions. This form of machine learning is perticularly effective for tasks such as pattern[5] recognition, classification and regression. ANNs are a fundamental component of machine learning and deep learning. the network consists ofan input layer , hidden layer and output layer. each connection between neurons has a weight, theseweights are adjusted based on input data to optimize the networks performance. the process be- gins with the input layer receiving data, which is then processed through the hidden layer using weighted connections and activation functions. the output layer produces the final result, whether itis a prediction ,classification of any other desired outcome. Intrusion Detection Systems (IDS) playa vital role in this defense strategy[2]. This introduction will delve into the concept of IDS, explain- ing its purpose, functionalities, and different types. By implementing an IDS, organizations gain a proactive approach to cybersecurity. They can identify potential threats in real-time, allowing for timely investigation and mitigation of attacks.

- **Background**

Autonomous vehicles rely on a plethora of sensors, control systems, and communication networks to navigate and make real-time decisions. While these systems enable advanced functionalities, they also expose AVs to potential cyber threats. Attack detection systems are therefore crucial for the safe operation of AVs. Traditional security measures often fall short in the dynamic and complex environment of autonomous driving, necessitating more sophisticated approaches.

- **Motivation**

Machine learning, particularly ensemble methods like Random Forest classifiers, offers a promising solution for detecting anomalies and potential attacks on AVs[4]. These methods can analyze large volumes of data and identify patterns that signify malicious behavior. This paper aims to evaluate the efficacy of Random Forest classifiers in detecting various types of attacks on AVs.

- **Related work**

Numerous studies have focused on enhancing the security of AVs through different approaches. Some have utilized anomaly detection techniques, while others have employed machine learning models such as Support Vector Machines (SVMs) and neural networks. However, the use of Ran- dom Forest classifiers specifically for attack detection in AVs has not been extensively explored. This research builds on existing knowledge and demonstrates the applicability of Random Forest classifiers in this context[4].

II. ATTACK VECTORS IN AUTONOMOUS VEHICLES

AVs are susceptible to various cyberattacks targeting different components.

2.1 Sensor Attack

Malicious manipulation of sensor data (cameras, LiDAR, radar) can lead the AV to misinterpret its surroundings, causing erratic maneuvers or accidents[1].

2.2 Communication Attacks

Attacks targeting the communication network (V2X)[6] can involve data interception, manipulation, or denial-of-service, hindering communication with other vehicles and infrastructure.

2.3 Software Attacks

Exploiting software vulnerabilities in the AV’s control systems can allow attackers to take control of critical functions like steering, braking, and acceleration[8].

III. MACHINE LEARNING FOR INTRUSION DETECTION

Machine learning offers a promising approach to detect attacks in AVs. By analyzing various data streams, including sensor data, network traffic, and system logs, ML algorithms can identify anomalies that deviate from normal behavior, potentially indicating an attack. Here are some prominent ML techniques employed for intrusion detection:

3.1 Supervised Learning

This approach involves training a model on labeled datasets containing normal and attack data. Algorithms like Support Vector Machines (SVMs) and Random Forests can then classify new data points as normal or attack-related.

3.2 Unsupervised Learning

This technique focuses on identifying patterns and deviations in unlabeled data.exclamation Anomaly detection algorithms like Isolation Forests and Local Outlier Factor (LOF) can detect unusual sen- sor readings or network behavior indicative of attacks, even if the specific attack type is unknown.

IV. METHODOLOGY

- **Data Collection and Preprocessing**

The first step in our methodology involves collecting data from autonomous vehicle simulations and real-world operations. This data includes sensor readings, communication logs, and control commands. To create a comprehensive dataset, we introduce various attack scenarios, such as spoofing, and data manipulation.

- **Feature Selection**

Feature selection is critical in improving the model’s performance and reducing computational complexity. We employ techniques such as correlation analysis and importance ranking to select the most relevant features. These features include signal strength, packet delivery ratio, latency, and anomalies in sensor data.

- **Classification matrix**

In this research work, three machine learning classification algorithms namely Random Forest Classifier,K-Neighbors Classifier and Logistic Regression have been compared based on some per- formance evaluation metrics. These classifiers are compared based on accuracy, precision, recall and f2-score and the best algorithm is determined.

- **Random Forest Classifier**

The Random Forest classifier is an ensemble learning method that constructs multiple decision trees during training and merges their outputs to improve accuracy and control overfitting[3]. Each tree in the forest votes on the class of the input, and the majority vote is taken as the final prediction. This approach is well-suited for our problem due to its ability to handle large, high-dimensional datasets and its robustness against overfitting.

- **Model Training and Evaluation**

We split our dataset into training and testing sets, ensuring a balanced representation of normal and attack scenarios. The Random Forest model is trained on the training set, and its performance is evaluated on the testing set using metrics such as accuracy, precision, recall, and F1-score.

V. DATASET USED

The project used the NSL-KDD dataset with attributes. Data is an improved version of the KDD99 dataset , a standard dataset for intrusion detection[7]. The dataset has several versions available, from which the KDDTrain+ and KDDTest+ training and testing data. The dataset contains network attacks related to the autonomous vehicle, including the 24 training attack types with 14 classes in the test file. Therefore, the dataset has KDDtrain.csv and KDDtest.csv, which are not recorded from the same probability distribution, making it more realistic. Moreover, some intrusion experts believe that most novel attacks are variants of known attacks, and those can be sufficient to catch the novel variants.

Module 1: Data Analysis and Visualization System

Module 1 appears to be centered around data analysis and visualization, employing Python libraries for processing and presenting data.

- **Core Components and Technologies**

- Python: Primary programming language for data handling and analysis.
- Data Analysis Libraries: Use of libraries like Pandas for data manipulation.
- Visualization Tools: Likely incorporation of libraries such as Matplotlib or Seaborn for data visualization.

- **Functionalities and Steps**

- Data Loading and Management: Importing and handling datasets, possibly using Pandas or similar libraries. Preprocessing and cleaning data for analysis.
- Exploratory Data Analysis (EDA): Conducting initial exploration of the data to identify patterns, anomalies, or trends. Generating summary statistics and basic visualizations.
- Advanced Data Visualization: Creating more sophisticated plots and charts to represent the data insights. Utilizing advanced features of visualization libraries.
- Interactive Data Exploration (if applicable): Implementing interactive elements for data ex- ploration, such as sliders, dropdowns, or clickable elements.
- Reporting and Presentation of Findings: Compiling insights and findings into a coherent and structured format. Exporting data and visualizations for reporting purposes.

- **Sub-Points and Details**

- Data Import Techniques: Methods for importing data from various sources and formats.
- Data Cleaning and Transformation: Techniques for transforming raw data into a usable format.
- Visualization Techniques: Use of different types of plots and customizations for effective data presentation.

Module 2: Advanced Data Processing and Visualization System

Module 2 seems to be an extension of Module 1, delving deeper into data processing, analysis, and visualization, likely using Python and its data-related libraries.Module 2 is designed as an advanced system for data processing and visualization, building upon the foundational elements introduced in Module 1. It emphasizes deeper data analysis, complex visualization techniques, and dynamic data interaction capabilities.

- **Core Components and Technologies**

- Python: The primary programming language for scripting and data processing.
- Advanced Data Processing Libraries: Potentially using libraries like Pandas for more complex data manipulations.

- Advanced Visualization Tools: Likely use of Matplotlib, Seaborn, or similar libraries for sophisticated data visualization.

- **Functionalities and Steps**

- Enhanced Data Management: Advanced techniques for processing and managing large datasets. Integration of complex data structures and manipulation methods.

- In-depth Exploratory Data Analysis (EDA): Comprehensive analysis to extract deeper in- sights from data. Application of statistical methods and data aggregation techniques.

- Sophisticated Data Visualization: Development of complex and detailed visual representations of data. Customization and optimization of plots for better data interpretation.

- Interactive and Dynamic Data Exploration (if applicable): Creation of dynamic data ex- ploration tools, like interactive dashboards. Enhancing user interaction with data through interactive visual elements.

- Integration of Diverse Data Sources: Combining data from various sources to enrich analysis. Techniques for handling diverse data formats and sources.

- Reporting and Presentation: Advanced methods for compiling and presenting findings from data analysis. Exporting visualizations and reports for various purposes.

- **Sub-Points and Details**

- Data Transformation Techniques: Methods for transforming raw data into more insightful formats.

- Complex Visualization Techniques: Use of advanced features in visualization libraries.

- Data Integration and Compatibility: Handling compatibility issues when integrating multiple data sources.

Module 3: Cybersecurity Simulation and Management System

Module 3 is a comprehensive cybersecurity system that employs PyQt5 for GUI development, Python for backend operations, and Streamlit for running an associated web application. It is designed to simulate and manage different types of cyber attacks.

- **Core Components and Technologies**

- PyQt5: For building the graphical user interface (GUI).

- Python: Backend scripting and control logic.

- Streamlit: For running an interactive web application.

- Pygame: For graphical simulations and visual displays.

- Subprocess and Threading: Managing external processes and threads for parallel execution.

- **Functionalities and Steps**

- GUI for Attack Type Selection: Implementation of a PyQt5 widget (AttackTypeSelector) to choose and initiate different types of cyber attacks. Dropdown menu for selecting attack types like ’DOS Attacks’, ’Probe Attacks’, ’U2R’, ’Sybil’.

- File-Based Communication and Control: Writing to and reading from files (attacktype.txt, file1.txt, file2.txt) to control the attack simulation and monitor status. Mechanisms to initiate, monitor, and stop attacks based on file inputs.

- Streamlit App Integration and Process Management: Starting and terminating the Streamlit app as a subprocess for web-based interaction and display. Using Python’s subprocess module for process control.

- Attack Information Display and Server Management: PyQt5-based interface (AttackInfoDis- play) to display current attack information and control server status. Button for restarting the server and resetting attack status.

- Concurrent Script Execution for Simulation: Using threading to run multiple Python scripts concurrently for different aspects of the simulation (attacker.py, simu.py, resp.py). Orches- trating complex attack scenarios and responses.

- Graphical Simulation with Pygame: Utilizing Pygame for creating real-time graphical simu- lations of cyber attacks. Dynamic visual representation of attack scenarios.

- Frame Extraction and Visualization: Extracting and saving frames from GIF files for use in simulations. Displaying sequential images to simulate real-time activities.

- **Sub-Points and Details**

- Inter-Process Communication: Techniques for communication between the GUI, simulation scripts, and the Streamlit app.

- User Interaction Design: Creating a user-friendly interface for controlling and monitoring the simulation.

- Real-time Simulation Techniques: Employing Pygame for real-time graphical representation of cyber attacks.

VI. RESULTS

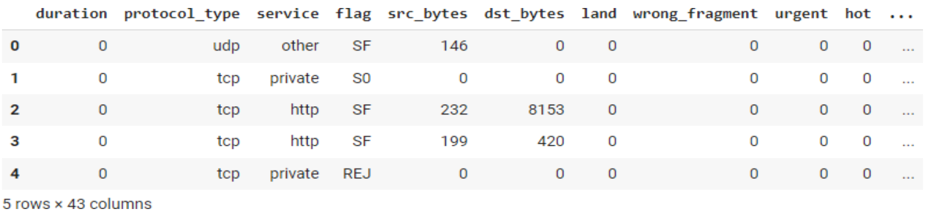
DATA PREPROCESSING

It is vital to preprocess the dataset to apply the ML techniques to any given dataset. The less essential attributes in the dataset do not affect the accuracy of the classifier we want to use. This report aims to provide the complete preprocessing steps of the two files of the NSL-KDD dataset. To preprocess the dataset, python-Anaconda-navigator (Jupyter notebook) was used. The same methods were used to preprocess the dataset of both files. Preprocessing contains: load the dataset and analyze the statistics of the dataset,Change sub attach labels to their respective class,Check the missing value in the dataset,Check the outliers.

- The table[fig.1] provided represents network traffic data with various attributes related to the network connection, its properties, and any associated events or behaviors.

The table [fig.2]represents network traffic data with various attributes and additional columns that categorize whether the traffic is normal or an attack.

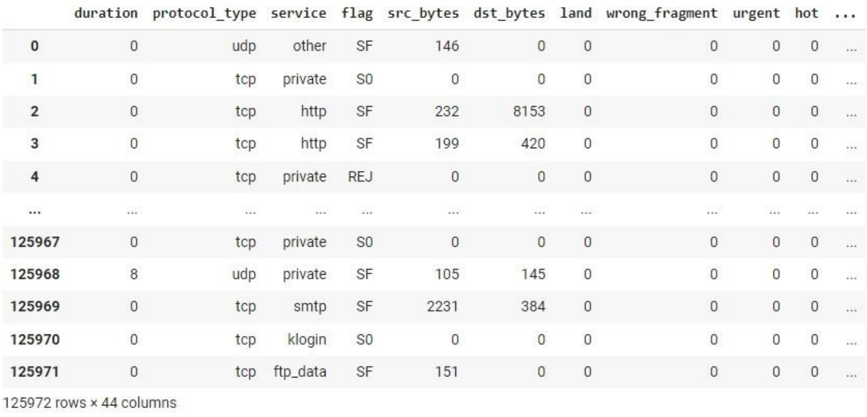
The table summarizing network traffic data categorized by the type of attack and the protocol used (ICMP, TCP, UDP). Each row represents a different type of attack or normal traffic, and each column represents the count of occurrences for each protocol type.This data is likely used in a network intrusion detection system (NIDS) for identifying and analyzing different types of network attacks. By understanding the distribution of attacks across different protocols, one can develop more robust security measures and improve the accuracy of attack detection algorithms.



	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...
0	0	udp	other	SF	146	0	0	0	0	0	...
1	0	tcp	private	S0	0	0	0	0	0	0	...
2	0	tcp	http	SF	232	8153	0	0	0	0	...
3	0	tcp	http	SF	199	420	0	0	0	0	...
4	0	tcp	private	REJ	0	0	0	0	0	0	...

5 rows × 43 columns

Figure 1: Traffic data with various attributes



	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...
0	0	udp	other	SF	146	0	0	0	0	0	...
1	0	tcp	private	S0	0	0	0	0	0	0	...
2	0	tcp	http	SF	232	8153	0	0	0	0	...
3	0	tcp	http	SF	199	420	0	0	0	0	...
4	0	tcp	private	REJ	0	0	0	0	0	0	...
...	...	...	...	...	...	...	...	...	...	...	...
125967	0	tcp	private	S0	0	0	0	0	0	0	...
125968	8	udp	private	SF	105	145	0	0	0	0	...
125969	0	tcp	smtp	SF	2231	384	0	0	0	0	...
125970	0	tcp	klogin	S0	0	0	0	0	0	0	...
125971	0	tcp	ftp_data	SF	151	0	0	0	0	0	...

125972 rows × 44 columns

Figure 2: Normal or attacked traffic

- **Performance Metrics**

The Random Forest classifier demonstrated high accuracy in detecting various types of attacks. The precision, recall, and F1-scores were also satisfactory, indicating the model's effectiveness in both identifying attacks and minimizing false positives.

- **Comparative Analysis**

We compared the performance of the Random Forest classifier with other machine learning models such as SVM and neural networks. The Random Forest model outperformed these models in terms of accuracy and robustness, highlighting its suitability for the task.

- **Accuracy**

By averaging the results of multiple trees, the Random Forest classifier can generalize well to unseen data, often outperforming single decision tree models and other algorithms. Its ability to handle large datasets with numerous features, along with its inherent handling of missing data and resistance to overfitting, contributes to its impressive accuracy across diverse applications, from medical diagnosis to fraud detection and, notably, attack detection in autonomous vehicles.

- **Prediction**

The confusion matrix is a fundamental tool for evaluating the performance of a classification model. It provides a detailed breakdown of the model's predictions by comparing them with the actual class labels. The matrix is typically represented in a 2x2 format for binary classification, comprising four key components: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). Understanding these components is essential for interpreting the model's accuracy, precision, recall, and F1-score.

- **Components of the Confusion Matrix:**

- True Positives (TP): The number of instances correctly predicted as positive.
- False Positives (FP): The number of instances incorrectly predicted as positive.
- True Negatives (TN): The number of instances correctly predicted as negative.

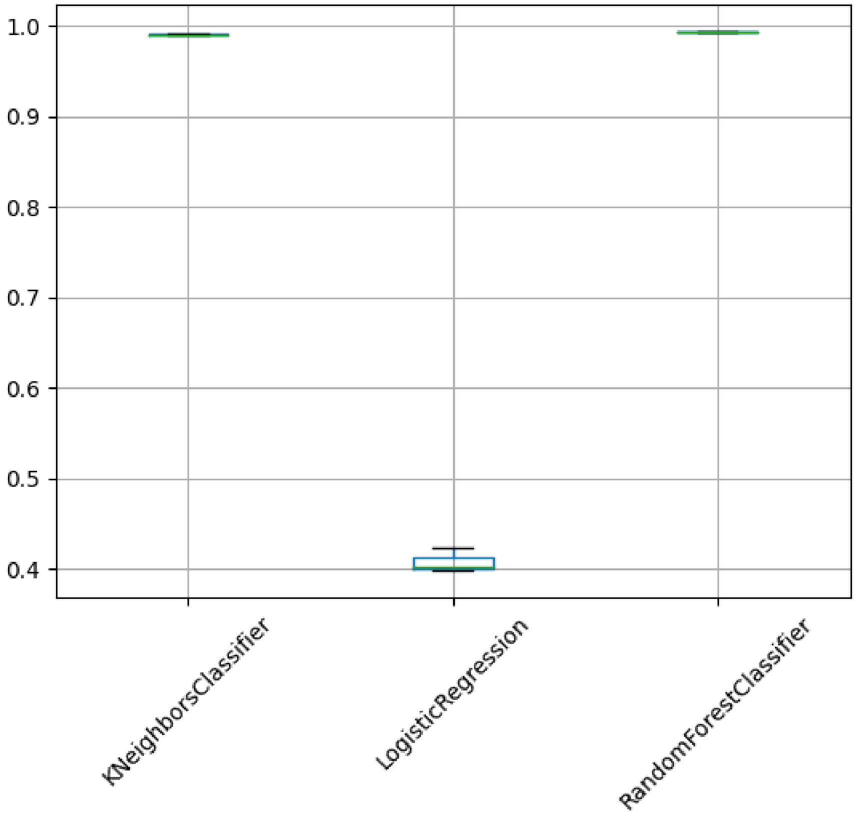


Figure 3: Performance and Variations

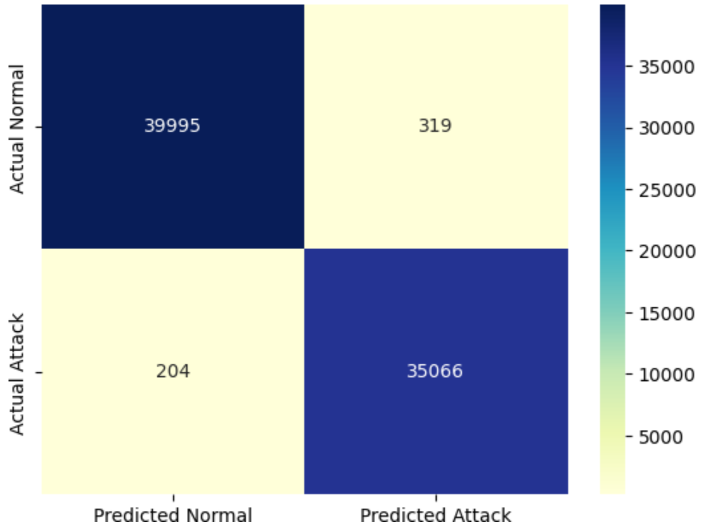
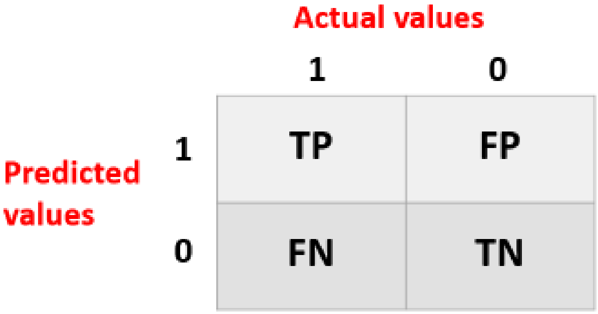


Figure 4: Data prediction



Predicted values	Actual values	1	0
	1	TP	FP
	0	FN	TN

Figure 5: Confusion Matrix

- False Negatives (FN): The number of instances incorrectly predicted as negative. Key Metrics Derived from the Confusion Matrix:

- Accuracy: The overall correctness of the model, calculated as the ratio of correctly predicted instances to the total instances.

Accuracy= TP+TN/TP+FP+TN+FN

- Precision: The ratio of correctly predicted positive instances to the total predicted positives, indicating the model's ability to avoid false positives.

Precision= TP/TP+FP

- Recall (Sensitivity): The ratio of correctly predicted positive instances to the actual positives, reflecting the model's ability to capture all positive instances.

Recall= TP/TP+FN

- F1-Score: The harmonic mean of precision and recall, providing a single metric that balances both concerns.

F1-Score = 2 × Precision × Recall Precision + Recall F1-Score=2× Precision+Recall Preci- sion×Recall

The [fig.7] box plots reveal the distribution of accuracy scores across different cross-validation folds for each model. The Decision Tree Classifier, Gradient Boosting Classifier, K-Neighbors Classifier, and Random Forest Classifier exhibit high and consistent performance, with median ac- curacies nearing 1 and minimal variability. In contrast, the Logistic Regression model demonstrates significantly lower accuracy, with a median around 0.55 and a wider interquartile range, indicating greater inconsistency in its performance. The SVC model, while slightly less consistent than the top performers, still maintains a high median accuracy. This analysis highlights the superior and stable performance of ensemble methods and SVC over the more variable and lower-performing LogisticRegression in this context.

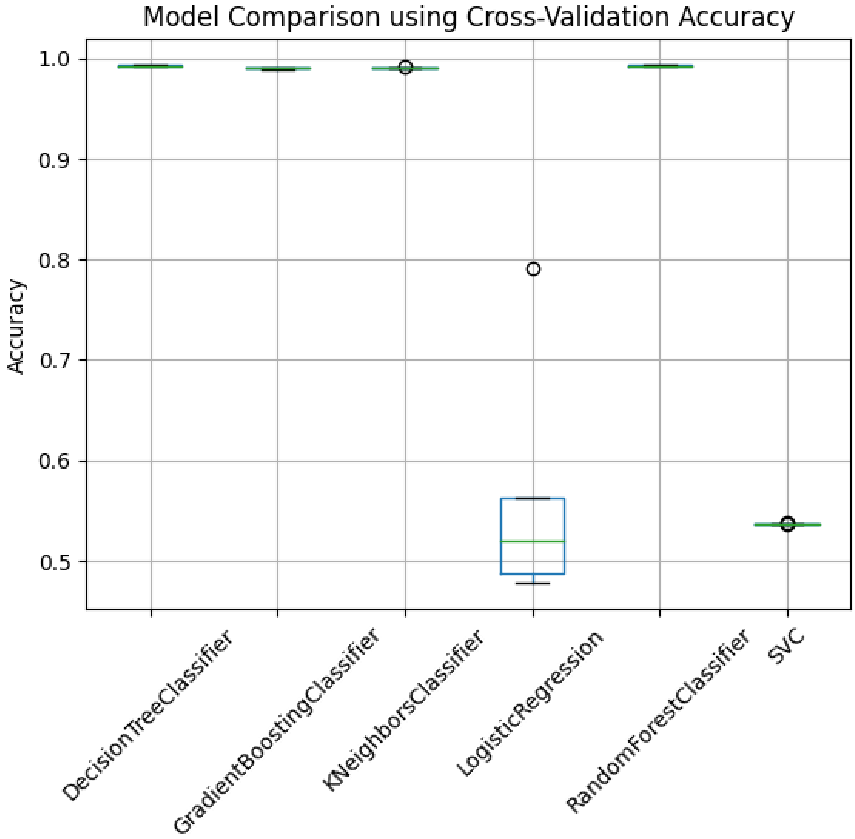


Figure 6: Model Comparison

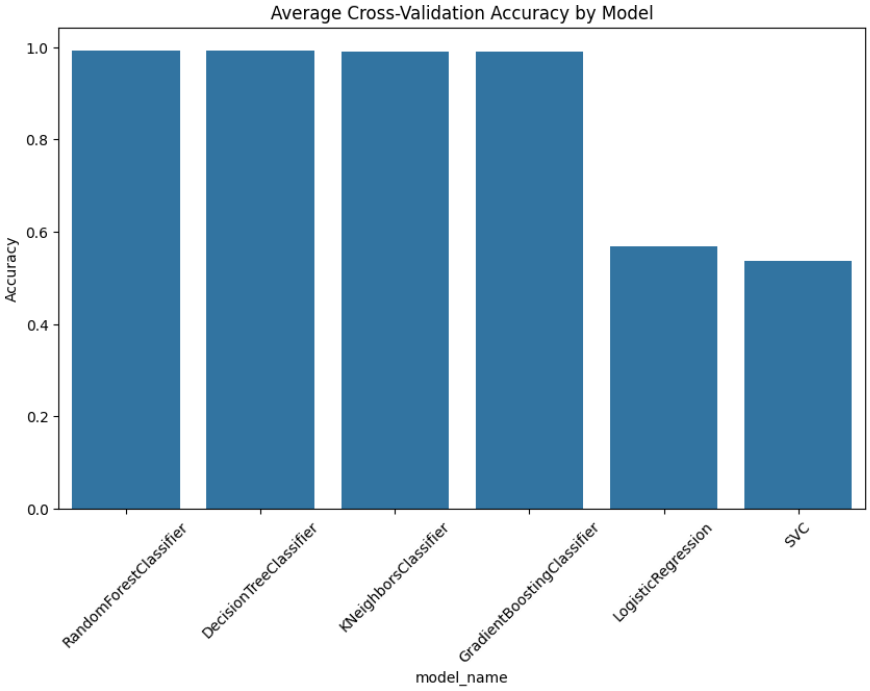


Figure 7: Models Accuracy

VII. MODEL DEVELOPMENT

- **SMART CAR SECURITY DASHBOARD**

This is a user interface designed to monitor and manage the security status of various systems in a smart car. It likely provides real-time information and options to set or view the status of different car components.

- **Components and Attack Status**

The dashboard allows users to select the attack status for different car systems, indicating whether they are experiencing any security issues or are functioning normally. The components listed are:

- Engine

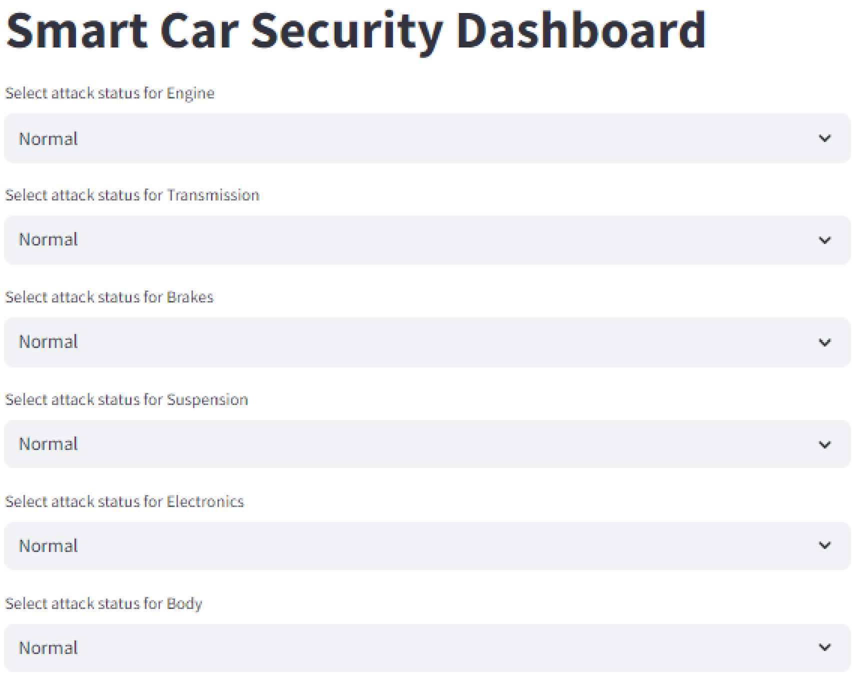
Attack Status: (Normal) The engine system is functioning properly without any detected security threats.

- Transmission

Attack Status: (Normal) The transmission system is operating normally with no security issues detected.

- Brakes

Attack Status:(Normal) The braking system is secure and functioning as expected.



Smart Car Security Dashboard

Select attack status for Engine

Normal

Select attack status for Transmission

Normal

Select attack status for Brakes

Normal

Select attack status for Suspension

Normal

Select attack status for Electronics

Normal

Select attack status for Body

Normal

Figure 8: Normal Status

- Suspension

Attack Status: (Normal) The suspension system shows no signs of security threats and is operating normally.

- Electronics

Attack Status: (Normal) The electronic systems (which could include navigation, and other onboard electronics) are secure and functioning correctly.

- Body

Attack Status:(Normal) The body of the car, potentially including sensors and external communication systems, is secure with no detected issues.

- Purpose and Usage The primary purpose of this dashboard is to allow the car owner or a security professional to:

Monitor the security status of critical car systems in real-time, Identify and address potential security threats, Ensure that each component of the car is functioning securely.

- Implications: It Indicates that the system is operating as expected with no security threats detected(Normal Status). If the status were anything other than ”Normal,” it would imply a security issue or an attack on that particular system, prompting further investigation or immediate action(Attack Detected).

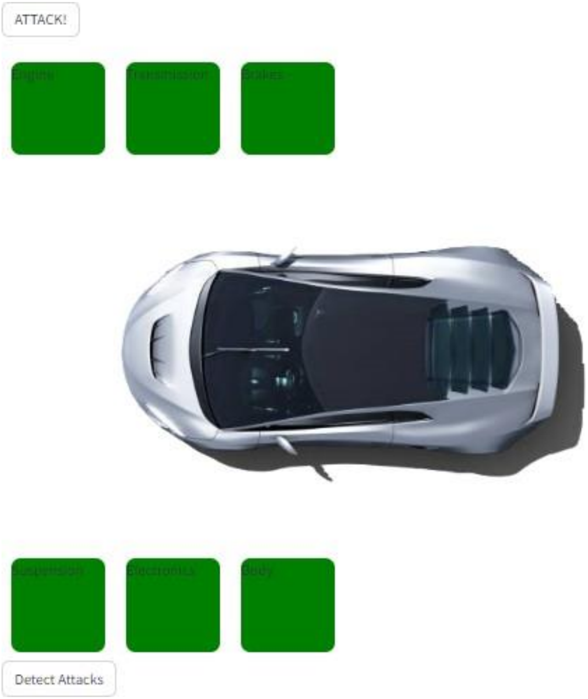


Figure 9: Normal Car System

- Context in a Real-world Application: In a real-world scenario, a smart car equipped with a security dashboard like this would be able to alert the user to various types of cyber-attacks or malfunctions. For example:

An attack on the engine might involve unauthorized access to the car's control system. An attack on the brakes could mean someone is trying to remotely disable or interfere with the braking system. Electronic system attacks might include attempts to hack the infotainment or GPS systems. By maintaining a real-time dashboard, car owners can ensure the safety and integrity of their vehicle's critical systems.

The green color signifies safety and normal operation, ensuring clarity in communication of the car's current state. Overall, the dashboard is designed to provide real-time monitoring and quick detection of any cyber-attacks or malfunctions, enhancing the car's security management.

- **Attack status for each system**

The dashboard allows the user to monitor and select the attack status for different car systems. Each system's status can be set to either ”Normal” or ”Attacked.”

- Engine

Status: (Attacked) The engine system is currently experiencing a security breach or attack. This could involve unauthorized access, tampering, or interference with the engine's function- ality.

- Transmission Status: (Normal) The transmission system is functioning normally without any detected security threats or attacks.

- Brakes

Status:(Normal) The braking system is secure and operating as expected with no detected security issues.

- Suspension

Status: (Attacked) The suspension system is under attack, which could affect the car's stabil- ity and handling. This might involve tampering or unauthorized control over the suspension components.

- Electronics

Status: (Normal) The electronic systems, including infotainment and navigation, are secure and functioning correctly with no detected threats.

- Body

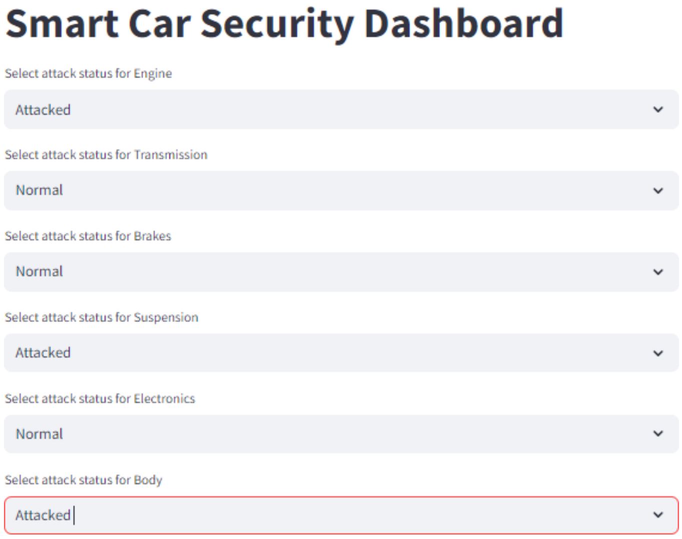
Status: (Attacked) Explanation: The body of the car, potentially including sensors, external communication systems, and physical security, is under attack. This could involve attempts to interfere with the car's external sensors or communication channels.

- Implications Immediate attention is needed to address and mitigate the security threats to the Engine, Suspension, and Body systems. These attacks could lead to significant safety risks and operational issues if not resolved promptly(Attacked Systems). The Transmission, Brakes, and Electronics systems are currently secure, with no immediate threats detected. These systems continue to operate as expected(Normal Systems).

- Purpose and Usage The purpose of the Smart Car Security Dashboard is to provide real-time monitoring of the car's security status and to alert the user to any potential threats. By identifying which systems are under attack, the user can take appropriate actions to secure the vehicle, such as:

Investigating and neutralizing the source of the attack. Running diagnostic checks and ap- plying security patches. Contacting a professional for further assistance.

The ”Smart Car Security Dashboard” serves as an advanced monitoring system designed to en- sure the cybersecurity of modern vehicles. The dashboard provides a real-time visual representation of the security status of six critical car systems.This tool is crucial for proactive threat detection and mitigation, providing users with immediate insights and facilitating prompt responses to maintain vehicle safety and integrity.



Smart Car Security Dashboard

Select attack status for Engine
Attacked

Select attack status for Transmission
Normal

Select attack status for Brakes
Normal

Select attack status for Suspension
Attacked

Select attack status for Electronics
Normal

Select attack status for Body
Attacked

Figure 10: Attacked status

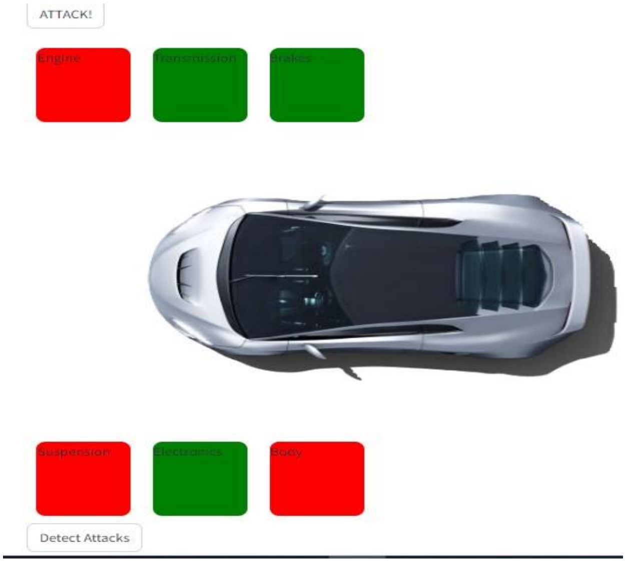


Figure 11: Attacked Car System

VIII. FUTURE WORK

While the Random Forest classifier showed promising results, there are limitations to our study. The model's performance may vary with different datasets and attack scenarios. Future research should explore the integration of real-time attack detection systems in AVs and the use of other ensemble methods to further enhance security. The future scope of attack detection in autonomous vehicles (AVs) encompasses the integration of advanced machine learning techniques, real-time detection and response systems, and adaptive context-aware models to enhance security measures. Future research should focus on leveraging deep learning, gradient boosting machines, and hybrid models to capture complex attack patterns. Additionally, the development of real-time, onboard detection frameworks optimized for speed and efficiency is crucial. Collaborative and federated learning approaches can facilitate knowledge sharing among AVs without compromising privacy, while multi-modal data fusion from diverse sensors can improve detection accuracy.

IX. CONCLUSION

This paper presented a comprehensive study on the use of Random Forest classifiers for detect- ing attacks on autonomous vehicles. Our findings indicate that the Random Forest model is an effective tool for identifying and mitigating potential threats, thereby enhancing the security and reliability of AVs. Our study demonstrates that Random Forest classifiers can accurately identify and classify various types of attacks, such as spoofing, denial of service, and data manipulation, thereby enhancing the security of AVs. By leveraging the ensemble learning approach, the model significantly reduces the risk of overfitting and improves generalization, making it a reliable tool for real-time attack detection. The performance metrics derived from the confusion matrix, includ- ing high accuracy, precision, recall, and F1-score, validate the model's efficacy in maintaining the safety and integrity of autonomous vehicle systems. Moreover, the adaptability of Random Forest classifiers to different datasets and attack scenarios underscores their potential as a cornerstone in the development of advanced AV security systems.

X. REFERENCES

[1] Murad Mehrab Abrar, Raian Islam, Shalaka Satam, Sicong Shao, Salim Hariri, and Pratik Satam. Gps-ids: An anomaly-based gps spoofing attack detection framework for autonomous vehicles. arXiv preprint arXiv:2405.08359, 2024.

[2] Fatimah Aloraini, Amir Javed, and Omer Rana. Adversarial attacks on intrusion detection systems in in-vehicle networks of connected and autonomous vehicles. Sensors, 24(12):3848, 2024.

[3] Mansi Girdhar, Junho Hong, and John Moore. Cybersecurity of autonomous vehicles: A sys- tematic literature review of adversarial attacks and defense models. IEEE Open Journal of Vehicular Technology, 4:417–437, 2023.

[4] Tae Hoon Kim, Moez Krichen, Meznah A Alamro, and Gabreil Avelino Sampedro. A novel dataset and approach for adversarial attack detection in connected and automated vehicles. Electronics, 13(12):2420, 2024.

[5] Hamid Mirzahossein and Mahdis Mashhadloo. Modeling the effect of autonomous vehicles (avs) on the accessibility of the transportation network. Scientific reports, 14(1):9292, 2024.

[6] Mohammed Ahmed Mohiuddin, K Nirosha, D Anusha, Mohd Nazeer, Ganapathi Raju NV, and Sorabh Lakhanpal. Ai to v2x privacy and security issues in autonomous vehicles: Survey. In MATEC Web of Conferences, volume 392, page 01097. EDP Sciences, 2024.

[7] Mohammed Zakariah, Salman A AlQahtani, Abdulaziz M Alawwad, and Abdullilah A Alotaibi. Intrusion detection system with customized machine learning techniques for nsl-kdd dataset. Computers, Materials & Continua, 77(3), 2023.

[8] Yi Zhu, Chenglin Miao, Hongfei Xue, Yunnan Yu, Lu Su, and Chunming Qiao. Malicious attacks against multi-sensor fusion in autonomous driving. In Proceedings of the 30th Annual International Conference on Mobile Computing and Networking, pages 436–451, 2024.