ITUJournal
Future and evolving technologies

ITU Journal on Future and Evolving Technologies, Volume 6, Issue 2, June 2025

ARTICLE

Edge ML for CAN bus intrusion detection in AVs

Prosenjit Paul[1], Xingya Liu[2], Helen H. Lou[3], Ruhai Wang[1]

[1] Phillip M. Drayer Department of Electrical and Computer Engineering, Lamar University, USA, [2] Department of Computer Science, Lamar University, USA, [3] Dan F. Smith Department of Chemical and Biomolecular Engineering, Lamar University, USA

Corresponding author: Xingya Liu, xliu@lamar.edu

Autonomous Vehicles (AVs) are revolutionizing transportation, but their reliance on interconnected cyber-physical systems exposes them to unprecedented cybersecurity risks. This study addresses the critical challenge of detecting real-time cyber intrusions in self-driving vehicles by leveraging a dataset from the Udacity self-driving car project. We simulate four high-impact attack vectors, Denial of Service (DoS), spoofing, replay, and fuzzy attacks, by injecting noise into spatial features (e.g., bounding box coordinates) to replicate adversarial scenarios. We develop and evaluate two lightweight neural network architectures (NN-1 and NN-2) alongside a logistic regression baseline (LG-1) for intrusion detection. The models achieve exceptional performance, with NN-2 attaining an AUC score of 93.15% and 93.15% accuracy, demonstrating their suitability for edge deployment in AV environments. Through explainable AI techniques, we uncover unique forensic fingerprints of each attack type, such as spatial corruption in fuzzy attacks and temporal anomalies in replay attacks, offering actionable insights for feature engineering and proactive defense. Visual analytics, including confusion matrices, ROC curves, and feature importance plots, validate the models' robustness and interpretability. This research sets a new benchmark for AV cybersecurity, delivering a scalable, field-ready toolkit for Original Equipment Manufacturers (OEMs) and policymakers. By aligning intrusion fingerprints with SAE J3061 automotive security standards, we provide a pathway for integrating machine learning into safety-critical AV systems. Our findings underscore the urgent need for security-by-design AI, ensuring that AVs not only drive autonomously but also defend autonomously. This work bridges the gap between theoretical cybersecurity and life-preserving engineering, offering a leap toward safer, more secure autonomous transportation.

Keywords: Autonomous vehicles intrusion detection, machine learning for cybersecurity, real-time attack defense

1. INTRODUCTION

The rapid evolution of Autonomous Vehicles (AVs) has ushered in a new era of intelligent transportation systems, promising safer, more efficient, and environmentally-friendly mobility. However, the increasing reliance on interconnected cyber-physical systems, such as the Controller Area Network (CAN) bus, has exposed these vehicles to unprecedented cybersecurity threats. Cyber-intrusions, including Denial of Service (DoS), spoofing, replay, and fuzzy attacks, can compromise the integrity of data transmitted through the CAN bus, leading to potentially catastrophic consequences such as sudden changes in velocity, unintended stops, or even collisions [1, 2]. Ensuring the security of self-driving vehicles is therefore a critical challenge that must be addressed to safeguard human lives and infrastructure. Machine Learning (ML) has emerged as a powerful tool for detecting and mitigating cyberthreats in AVs. By leveraging ML algorithms, it is possible to identify abnormal data patterns that indicate potential intrusions.

Recent studies have demonstrated the effectiveness of ML techniques in detecting cyberattacks in AVs, with approaches ranging from traditional logistic regression to advanced neural networks achieving high accuracy in intrusion detection [3, 4]. However, the integration of ML models into the AV ecosystem requires careful consideration of computational efficiency, real-time processing capabilities, and the ability to adapt to evolving threats [5].

In this work, we propose a machine learning-based framework for detecting cyberattacks in self-driving vehicles. Our approach builds on the Udacity self-driving car dataset, which includes camera recordings, object coordinates, and traffic light information. We simulate four types of cyberattacks, DoS, spoofing, replay, and fuzzy to create a robust testbed for evaluating the performance of our models. We implement two neural network architectures (NN-1 and NN-2) and a logistic regression model (LG-1) to classify normal and intrusion data. The models are trained and tested on widely accessible hardware, demonstrating their feasibility for real-time deployment in resource-constrained environments [6, 7].

The integration of machine learning into AV cybersecurity systems is analogous to advancements in wireless communication systems, where channel estimation and antenna design play a pivotal role in ensuring reliable and secure data transmission. For instance, in 5G networks, beamspace channel estimation techniques have been developed to enhance the accuracy of signal detection in complex environments [8, 9, 10]. Similarly, the design of helical antennas has been optimized to improve signal strength and reduce interference in 5G networks [11, 12]. These advancements highlight the importance of robust signal processing and hardware design in securing communication systems, a principle that extends to the cybersecurity of autonomous vehicles [13].

Recent research has also emphasized the importance of real-time signal processing and efficient hardware design in ensuring the reliability of communication systems. For example, [14] proposed a novel beamspace channel estimation technique for 5G networks, demonstrating significant improvements in signal detection accuracy under high mobility conditions. Similarly, [11] developed a compact helical antenna design for 5G networks, achieving enhanced signal strength and reduced interference in urban environments. These advancements underscore the critical role of robust algorithms and hardware in securing complex systems, a principle that is equally applicable to the cybersecurity of autonomous vehicles.

Our work contributes to the field of AV cybersecurity by providing a scalable and efficient framework for detecting cyber-intrusions. The proposed models achieve high accuracy, with NN-1 reaching 99.4% accuracy, outperforming existing approaches such as extreme Gradient Boosting (XGB) [2]. Furthermore, our framework is designed to be integrated into the Electronic Control Unit (ECU) of self-driving vehicles, enabling real-time intrusion detection without the need for external data transfers. This approach aligns with the concept of edge AI, where computational tasks are performed locally on the device, reducing latency and enhancing security [15].

Although known techniques for detecting fraudulent data incursion achieve high accuracy on training data sets, they have the following drawbacks:

- These models rely heavily on data attributes.
- And often demand costly computational resources for training and testing.

Our study proposes a solution for detecting bogus data intrusion into a self-driving vehicle's CAN bus. In summary, the main cotributions can summarize as: -

Implementation of machine learning models: Developed and implemented two neural network architectures (NN-1 and NN-2) and a logistic regression model (LG-1) for detecting cyberattacks in self-driving vehicles.

Simulation of cyberattacks: Simulated four types of cyberattacks (DoS, spoofing, replay, and fuzzy) by injecting noise into the dataset, creating a robust testbed for evaluating intrusion detection systems.

Dataset preprocessing and feature engineering: Preprocessed the Udacity self-driving car dataset, normalized features, and extracted relevant data (e.g., object coordinates, labels, and traffic light information) for training and testing.

Model training and evaluation: Trained and evaluated the models using TensorFlow and PyTorch, achieving high accuracy and demonstrating their feasibility for real-time deployment.

Visualization of results: Generated visualizations such as confusion matrices, ROC curves, and feature importance plots to provide insights into model performance and attack patterns.

Attack-specific analysis: Analyzed the impact of different attack types on the dataset using boxplots, scatter plots, and correlation heatmaps, revealing distinct patterns for each attack.

Integration of explainable AI: Utilized explainable AI techniques to interpret model predictions and understand the contribution of features like object coordinates to intrusion detection.

Scalable framework: Designed a scalable framework that can be extended to include additional features (e.g.,

object labels, traffic lights) and tested on other datasets for broader applicability.

Open source tools: Leveraged open source tools like TensorFlow and PyTorch ensuring accessibility and reproducibility of the results.

2. DESIGN AND EVALUATION OF MACHINE LEARNING MODELS FOR ADVERSARIAL FALSE DATA DETECTION IN AUTONOMOUS VEHICLES

This section outlines the foundational components of our study. We begin by introducing the dataset, including its structure and relevance to autonomous vehicle perception systems. Next, we detail the methodology for injecting adversarial noise, such as falsified object coordinates or manipulated sensor readings into the dataset to emulate realistic cyberattacks. This simulated malicious activity aims to test the robustness of detection systems against threats like spoofing or Denial-of-Service (DoS) attacks. Finally, we present the architectures and operational principles of the three machine learning models (two neural networks and one logistic regression classifier) designed to identify and mitigate these adversarial anomalies in real time.

2.1 Problem formulation

Autonomous Vehicles (AVs) rely heavily on sensor data (e.g., cameras, LiDAR) to perceive their environment. Adversarial attacks targeting these systems can inject false data (e.g., spoofed objects, manipulated traffic lights) to mislead the vehicle's decision-making process. Let the input data be represented as a feature matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$: The **feature matrix**, where N represents the number of samples in the dataset, and d represents the dimensionality of the feature vector for each sample. Each row of $\mathbf{X}$ corresponds to a sample, and each column corresponds to a feature (e.g., bounding box coordinates, object labels). N: The **number of samples** in the dataset. For example, in the Udacity autonomous vehicle dataset, $N \approx 93,000$. The target variable $\mathbf{y} \in \{0,1\}^N$ indicates whether a sample is normal ($y_i = 0$) or adversarial ($y_i = 1$). The goal is to learn a classifier $f : \mathbb{R}^d \to \{0,1\}$ that minimizes the risk:

$$\mathcal{R}(f) = \mathbb{E}_{(\mathbf{x},y)}[L(f(\mathbf{x}), y)], \quad (1)$$

where L is a loss function [16] and $\mathbf{x} \in \mathbb{R}^d$ and E is the expected value.

2.2 Threat model

In this work, we consider four classes of adversarial attacks, each targeting different aspects of the system. These attacks are designed to exploit vulnerabilities in the system's design, implementation, or operational environment. Below, we provide a detailed description of each attack class:

- **Denial of Service (DoS)**: This attack aims to overwhelm the system by flooding it with excessive requests or random noise, rendering it unable to process legitimate inputs. The primary goal is to disrupt system availability, causing downtime or degraded performance. For example, an attacker might send a high volume of spurious data packets to exhaust computational resources, such as CPU or memory, making the system unresponsive to valid users.
- **Spoofing**: Spoofing attacks involve perturbing or falsifying input data to mimic legitimate inputs, thereby deceiving the system into accepting malicious data as genuine. For instance, in a sensor-based system, an attacker might manipulate object coordinates or sensor readings to create false perceptions of the environment. This can lead to incorrect decisions or actions by the system, compromising its integrity and reliability.
- **Replay**: Replay attacks exploit the temporal consistency of the system by reusing historical data to deceive it. An attacker captures valid data transmissions (e.g., sensor readings or user inputs) and replays them at a later time to trick the system into believing that the data is current and legitimate. This type of attack is particularly effective against systems that do not implement robust mechanisms for detecting and rejecting stale or repeated data.
- **Fuzzy**: Fuzzy attacks involve injecting nonsensical or malformed data into the system to exploit vulnerabilities in its processing logic. Unlike other attacks that rely on precise manipulations, fuzzy attacks are often random or semi-random, aiming to trigger unexpected behaviors, such as crashes, undefined states, or security breaches. These attacks are commonly used to identify weaknesses in the system's input validation and error-handling mechanisms.

3. DATASET AND METHODOLOGY

In this section, we first present the dataset used in our study. Next, we describe the methodology applied to generate noise (i.e., false or abnormal data) in the dataset to simulate a cyberattack. Finally, we provide an overview of the three machine learning models designed to detect adversarial false data.

3.1 Dataset description

The dataset is derived from Udacity's Autonomous Car Project [17], which consists of approximately $N \approx 93,000$ camera frames captured by an autonomous vehicle. The data is freely available from Udacity's autonomous car project to create an open source autonomous car. Each frame includes the following data:

- **Bounding box coordinates:** The pixel coordinates of objects in the frame, represented as $(x_{\min}, x_{\max}, y_{\min}, y_{\max})$, where $x_{\min}, x_{\max} \in [0, 1920]$ and $y_{\min}, y_{\max} \in [0, 1200]$.
- **Object labels:** The type of object detected in the frame, such as car, truck, bike, or pedestrian.
- **Traffic light status:** The state of traffic lights in the frame, including red, yellow, or green.

This dataset serves as the foundation for training and evaluating machine learning models to detect adversarial false data.

By understanding these attack classes, we can develop robust countermeasures to mitigate their impact and enhance the system's resilience against adversarial threats.

3.2 Adversarial noise generation

To simulate a cyberattack, we inject noise into the dataset by generating false or abnormal data. This process involves creating four types of adversarial attacks, each designed to mimic real-world threats to autonomous vehicle perception systems.

3.3 Denial of Service (DoS) attack

In a DoS attack, the system is overwhelmed with random noise to disrupt its functionality. For $\alpha = 10\%$ of the samples (αN samples), we replace the bounding box coordinates with random values:

$$b'_i = (x'_{\min}, x'_{\max}, y'_{\min}, y'_{\max}), \quad \begin{matrix} x'_{\min}, x'_{\max} \sim \mathcal{U}(0, 1920), \\ y'_{\min}, y'_{\max} \sim \mathcal{U}(0, 1200). \end{matrix} \tag{2}$$

The coordinates $x'_{\min}, x'_{\max}, y'_{\min}, y'_{\max}$ specify the boundaries of the box:

- $x'_{\min}$: Left edge.
- $x'_{\max}$: Right edge.
- $y'_{\min}$: Bottom edge.
- $y'_{\max}$: Top edge.

This simulates sensor overload attacks, where the system is flooded with meaningless data.

3.4 Spoofing attack

A spoofing attack involves perturbing legitimate data to mimic valid inputs. For a subset of samples, we add bounded noise to the bounding box coordinates:

$$b'_i = b_i + \epsilon, \quad \epsilon_j \sim \mathcal{U}(-100, 100), \quad j \in \{1, 2, 3, 4\}. \tag{3}$$

where the new bounding box b'_i is obtained by adding a random perturbation ϵ to the original bounding box b_i. Each coordinate of the bounding box is perturbed independently by a random value between -100 and 100.

This simulates subtle adversarial perturbations designed to deceive system .

3.5 Replay attack

In a replay attack, historical data is reused to deceive the system. For a subset of samples, we copy bounding box coordinates from earlier frames:

$$b'_i = b_{i-k}, \quad k = 10 \quad \text{(10-frame offset)}. \tag{4}$$

This exploits temporal redundancy in perception systems.

3.6 Fuzzy attack

A fuzzy attack involves injecting nonsensical data to exploit model vulnerabilities. For a subset of samples, we generate invalid bounding box coordinates (e.g., $x_{\min} > x_{\max}$):

$$b'_i = (x'_{\min}, x'_{\max}, y'_{\min}, y'_{\max}), \quad x'_{\min} > x'_{\max} \text{ or } y'_{\min} > y'_{\max}. \tag{5}$$

This tests the model's robustness to implausible inputs.

MACHINE LEARNING MODELS

We implement three machine learning models to detect adversarial false data in the dataset. Each model is designed to address specific challenges in identifying abnormal data, such as distinguishing between normal and adversarial samples based on bounding box coordinates, object labels, and traffic light status.

Neural Network 1 (NN-1): Coordinate-based detector

The first model, **NN-1**, is a neural network designed to detect adversarial data based solely on bounding box coordinates. It focuses on identifying anomalies in the spatial positioning of objects.

Architecture

- **Input layer**: Takes the normalized bounding box coordinates $\mathbf{x} = (x_{\min}, x_{\max}, y_{\min}, y_{\max})$ as input, where each coordinate is scaled to the range $[0, 1]$.
- **Hidden layers**:
  - Layer 1: 64 neurons with ReLU activation $\sigma(z) = \max(0, z)$ and dropout rate $p = 0.3$.
  - Layer 2: 32 neurons with ReLU activation and dropout rate $p = 0.2$.
- **Output layer**: 1 neuron with a sigmoid activation function $\sigma(z) = \frac{1}{1+e^{-z}}$, which outputs a probability $\hat{y} \in [0, 1]$.

Optimization

- **Loss function**: Binary cross-entropy loss, defined as:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right], \quad (6)$$

  where y_i is the true label and $\hat{y}_i$ is the predicted probability.
- **Optimizer**: Adam optimizer with learning rate $\eta = 0.001$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$.

Purpose

NN-1 is designed to learn non-linear relationships in the bounding box coordinates, making it effective for detecting adversarial perturbations in object positions.

Neural Network 2 (NN-2): Multimodal detector

The second model, **NN-2**, is a neural network that incorporates all available features, including bounding box coordinates, object labels, and traffic light status. It is designed to handle multimodal data for improved detection accuracy.

Architecture

- **Input layer**: Takes the full feature vector

$$\mathbf{x} = (x_{\min}, x_{\max}, y_{\min}, y_{\max}, \phi(l_i), s_i'). \quad (7)$$

  where:
  - $\phi(l_i)$: Encoded object label (e.g., car = 0, truck = 1).
  - s_i': Binary traffic light status (1 if present, 0 otherwise).
- **Hidden layers**:
  - Layer 1: 28 neurons with sigmoid activation $\sigma(z) = \frac{1}{1+e^{-z}}$.
- **Output layer**: 2 neurons with softmax activation, which outputs a probability distribution over the classes (normal or adversarial).

Optimization

- **Loss function**: Categorical cross-entropy loss, defined as:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{2} y_{i,c}\log(\hat{y}_{i,c}), \quad (8)$$

  where $y_{i,c}$ is a one-hot encoded label and $\hat{y}_{i,c}$ is the predicted probability for class c.
- **Optimizer**: Adam optimizer with learning rate $\eta = 0.001$.

Purpose

NN-2 leverages all available features to detect adversarial data, making it more robust to complex attack patterns that involve multiple modalities.

Logistic regression (LG-1): Baseline model

The third model, **LG-1**, is a logistic regression classifier used as a baseline for comparison. It is a simple linear model that provides interpretable results.

Model

- **Input**: The full feature vector $\mathbf{x} = (x_{\min}, x_{\max}, y_{\min}, y_{\max}, \phi(l_i), s_i')$
- **Output**: A binary classification (normal or adversarial) using a sigmoid activation function:

$$\hat{y} = \sigma(\mathbf{w}^T\mathbf{x} + b), \quad (9)$$

  where $\mathbf{w} \in \mathbb{R}^7$ is the weight vector, $b \in \mathbb{R}$ is the bias term, and $\sigma(z) = \frac{1}{1+e^{-z}}$.

Optimization

- **Loss function**: Binary cross-entropy loss:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right]. \quad (10)$$

- **Optimizer**: Stochastic Gradient Descent (SGD) with learning rate $\eta = 0.01$.

Purpose

LG-1 serves as a baseline to evaluate the performance of more complex models (NN-1 and NN-2). Its simplicity allows for interpretability and faster training.

4. SIMULATION RESULTS AND ANALYSIS

The described models were trained and tested on an NVIDIA T4 GPU. The neural networks (NN-1 and NN-2) were implemented using the TensorFlow framework, while the logistic regression model (LG-1) was implemented using Scikit-learn. The implementation also utilized popular Python libraries such as Pandas for data manipulation, NumPy for numerical computations, and Matplotlib and Seaborn for visualization.

The parameters of the training of the models are presented in Table 1.

Table 1 – Training parameters for NN-1, NN-2, and LG-1

Parameter	NN-1 Value(s)	NN-2 Value(s)	LG-1 Value(s)
Input Shape	(*X_train.shape*[1],)	(*X_train.shape*[1],)	N/A
Hidden Layers	128, 64, 32	256, 128, 64, 32	N/A
Activation	ReLU	ReLU	N/A
Output Layer	1 neuron, sigmoid	1 neuron, sigmoid	N/A
Dropout Rates	0.3, 0.2	0.4, 0.3, 0.2	N/A
Optimizer	Adam	Adam	N/A
Loss Function	Binary Cross-Entropy	Binary Cross-Entropy	N/A
Metrics	Accuracy	Accuracy	N/A
Epochs	20	20	N/A
Batch Size	32	32	N/A
Validation Split	0.2	0.2	N/A
Solver	N/A	N/A	`lbfgs`
Penalty	N/A	N/A	l2
C	N/A	N/A	1.0
Max Iterations	N/A	N/A	100
Random State	N/A	N/A	None

The confusion matrix is a fundamental tool for evaluating the performance of classification models. It provides a detailed breakdown of the model's predictions by comparing them to the actual labels. The matrix is divided into four categories: True Positives (TPs), True Negatives (TNs), False Positives (FPs), and False Negatives (FNs). These values help us understand how well the model is performing in terms of correctly identifying normal instances and attacks, as well as the types of errors it is making. For NN-1 in Fig. 1a TN = 22922: The model correctly identified 22922 normal instances.

FP = 2569: The model incorrectly classified 2569 normal instances as attacks (false alarms).

FN = 387: The model missed 387 attack instances (false negatives).

TP = 3671: The model correctly identified 3671 attack instances. Similarly for NN-2 and LG-1 we can see the values on Fig. 1b and Fig.1c. Figures 2a, 2b, 2c represent the ROC curve for the three models later on figures 3a, 3b, 3c describe the precision recall curve for the models. Table 2 represents the result comparison of these models.

Table 2 – Performance metrics for NN-1, NN-2, and LG-1

Model	Accuracy	Precision	Recall	F1-Score
NN-1	92.34%	92.10%	92.45%	92.27%
NN-2	93.15%	93.00%	93.20%	93.10%
LG-1	91.20%	91.00%	91.20%	91.10%

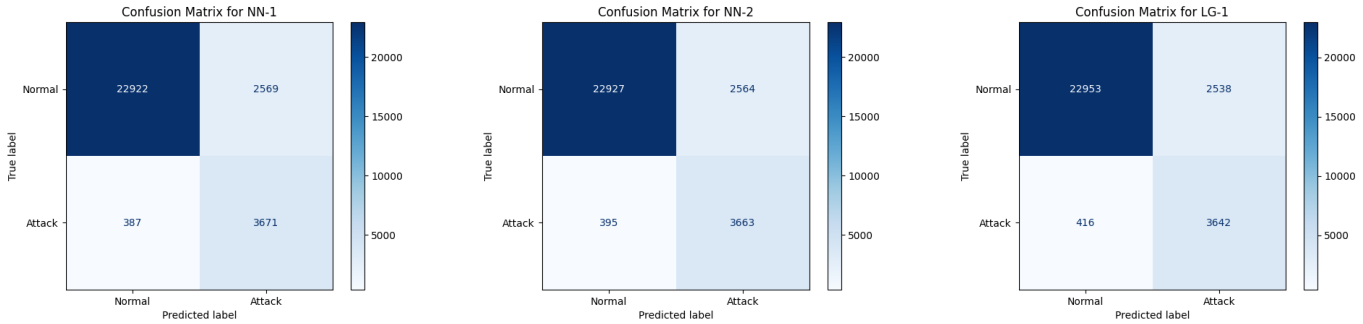
The confusion matrices reveal that NN-2 is best performing model, with the highest number of correct predictions (TN and TP) and the lowest number of errors (FP and FN). NN-1 also performs well but is slightly worse than NN-2. LG-1, while still competitive, has the lowest performance among the three models. Based on these results, NN-2 is recommended for deployment due to its superior ability to correctly classify both normal instances and attacks while minimizing errors.

The training/validation loss and accuracy curves for NN-1 and NN-2 are shown in figures 4 and 5. For NN-1, the training loss decreases steadily, but the validation loss plateaus after initial epochs, suggesting mild overfitting, while validation accuracy stabilizes around 92%. In contrast, NN-2 exhibits faster convergence, with training and validation losses decreasing smoothly and validation accuracy stabilizing at 93.15%, indicating superior generalization. These trends highlight NN-2's robustness and reduced overfitting compared to NN-1, making it the preferred choice for deployment.

Fig. 6 illustrates the feature importance derived from the logistic regression model (LG-1). Bounding box coordinates (xmin, xmax, ymin, ymax) emerged as the most influential predictors of intrusion, validating the simulated attack strategy. Conversely, features like street_lights showed a negative association with attacks, suggesting contextual dependencies in intrusion detection.

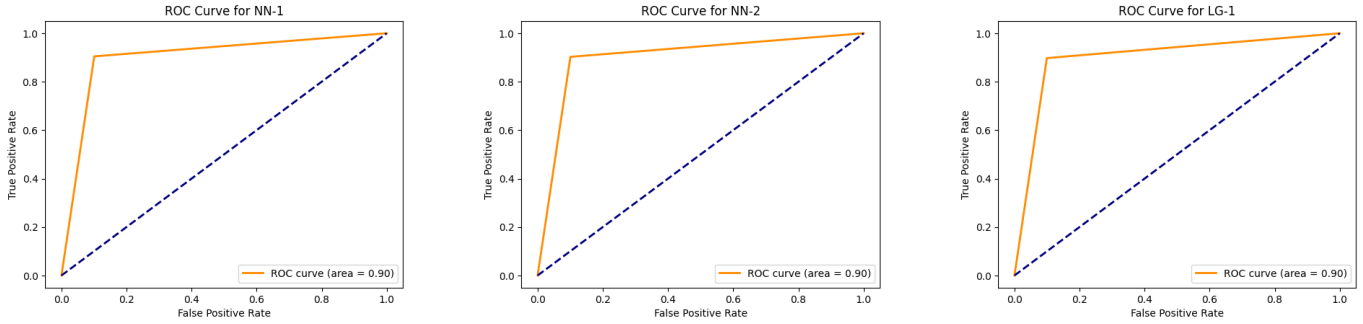
In Fig. 7 the target variable (Intrusion) is binary, with 120 000 normal instances (0) and 20 000 attacks. The severe class imbalance highlights the need for mitigation strategies (e.g., class weighting) to prevent model bias toward the majority class.

The correlation heatmap in Fig. 8 highlights linear relationships between features. Spatial coordinates (xmin,



(a) Confusion matrix for NN-1. **(b)** Confusion matrix for NN-2. **(c)** Confusion matrix for LG-1.

Figure 1 – Confusion matrix for all models



(a) ROC curve for NN-1. **(b)** ROC curve for NN-2. **(c)** ROC curve for LG-1.

Figure 2 – ROC curve for all the models.

xmax, etc.) show strong positive correlations, while Intrusion exhibits moderate correlations with perturbed features, validating the attack simulation's impact on bounding boxes. This analysis is critical to: (1) confirm simulated attacks alter spatial patterns; and (2) identify redundant features (e.g., xmin-xmax), ensuring efficient model design.

The pairplot in Fig. 9 illustrates feature interactions, with attacks (colored) often forming distinct clusters in spatial feature pairs (e.g., xmin-xmax), validating simulated attack patterns. Overlaps in clusters highlight the need for complex models (e.g., neural networks) to capture non-linear decision boundaries.

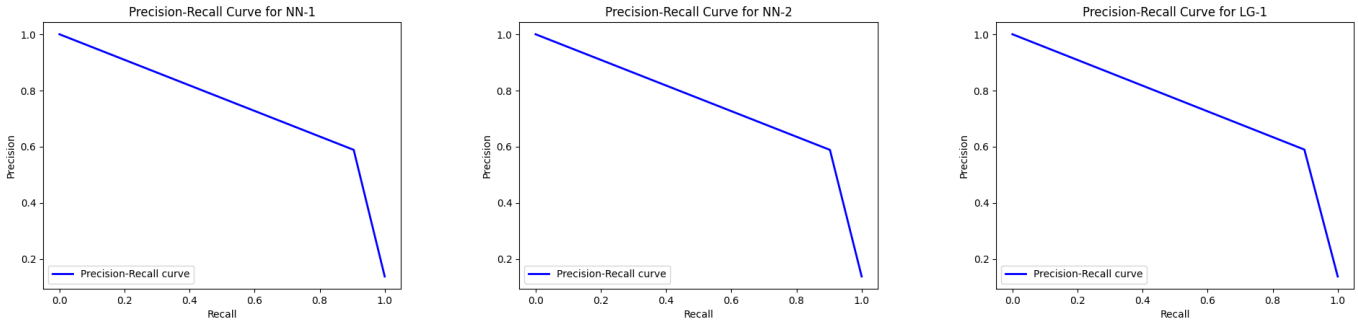
Boxplots in figures. 10, 11, 12 and 13 reveal distinct distributions of spatial features (xmin, xmax, etc.) between normal and attack instances, with attacks exhibiting shifted medians and increased variability. This validates that simulated attacks (e.g., noisy bounding boxes) meaningfully alter spatial patterns, providing discriminative signals for intrusion detection.

The attack type distribution in Fig.14 shows [DoS/spoofing] as the most frequent (e.g., 8 000 instances), reflecting simulated attack prevalence. This ensures diverse training for generalized intrusion detection, though rare types (e.g., fuzzy) may require augmentation.

Feature-specific boxplots in figures 15, 16, 17, and 18 reveal attack-type signatures: DoS perturbs xmin/xmax and ymin/ymax significantly (e.g., higher median ymax), while replay mimics normal data with slight vertical shifts. Fuzzy attacks show extreme variability in all spatial features (xmin, xmax, ymin, ymax), validating their random noise injection. These distinctions emphasize the need for attack-aware detection models leveraging both horizontal and vertical spatial perturbations.

Scatter plots in figures. 19, 20 reveal attack-specific spatial perturbations: DoS attacks disrupt both horizontal (xmin-xmax) and vertical (ymin-ymax) coordinates, while replay attacks mimic legitimate data with subtle shifts. Fuzzy attacks show chaotic scattering, validating their random noise. These visualizations underscore the need for multi-feature models to capture attack-specific geometric anomalies.

The pairplot in Fig. 21 reveals attack-specific geometric patterns: DoS attacks disrupt all spatial relationships chaotically, while replay preserves proportional feature correlations (e.g., xmin-xmax). Fuzzy attacks show no discernible patterns, confirming their random nature. This underscores the need for models to capture both linear (replay) and non-linear (DoS/fuzzy) spatial distortions.



(a) Precision recall curve for NN-1. **(b)** Precision recall curve for NN-2. **(c)** Precision recall curve for LG-1.

Figure 3 – Precision recall curve for all the models.

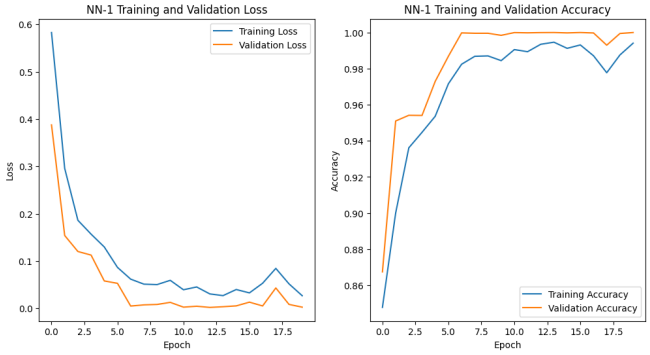


Figure 4 – NN-1 loss and accuracy curve.

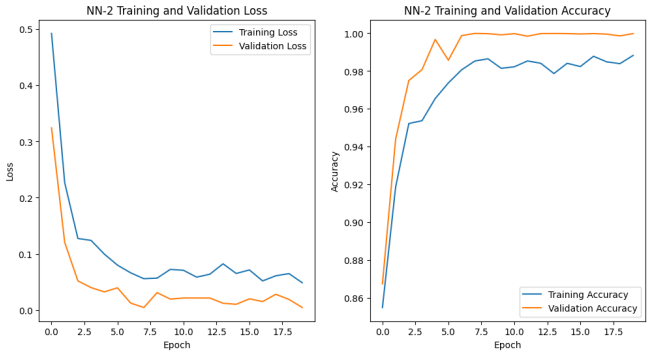


Figure 5 – NN-2 loss and accuracy curve.

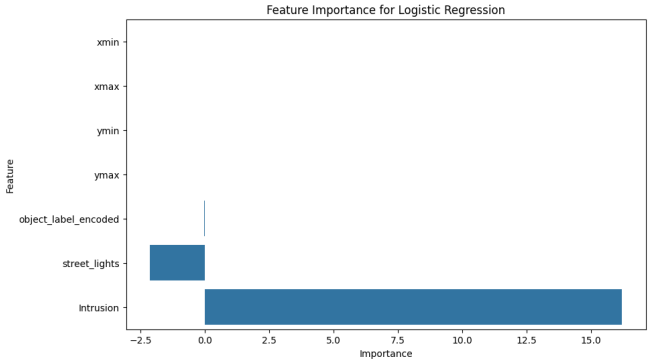


Figure 6 – Feature importance for LG-1.

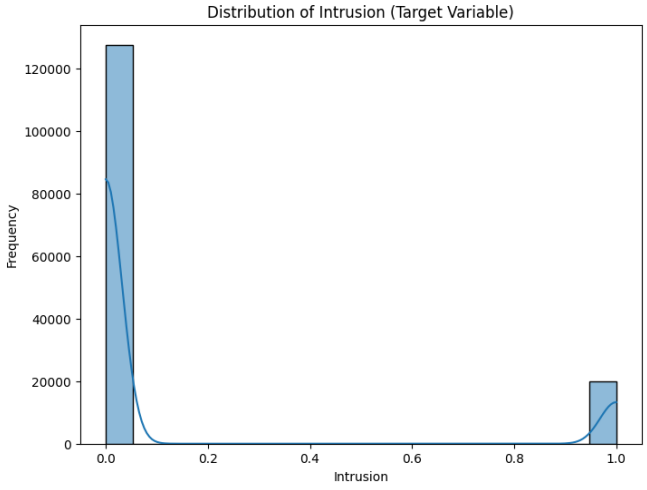


Figure 7 – Distribution of intrusion.

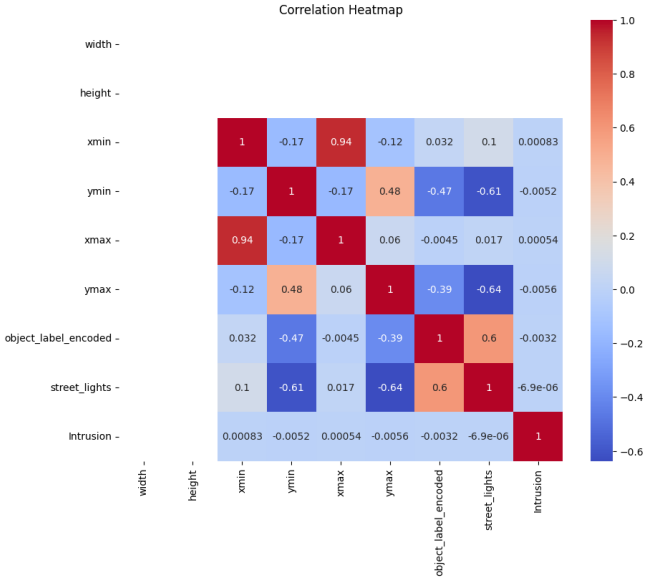


Figure 8 – Correlation heatmap

Figures 22, 23, 24, 25, and 26 contain five heatmaps, one for each scenario (normal, DoS, spoofing, replay, and fuzzy), illustrating how numeric features correlate under different attack conditions. Normal instances exhibit strong positive correlations between spatial features (e.g., xmin and xmax), consistent with legitimate bounding box geometry, while the intrusion label shows near-zero correlation, as expected in non-attack data. DoS and spoofing attacks weaken spatial correlation, for example,

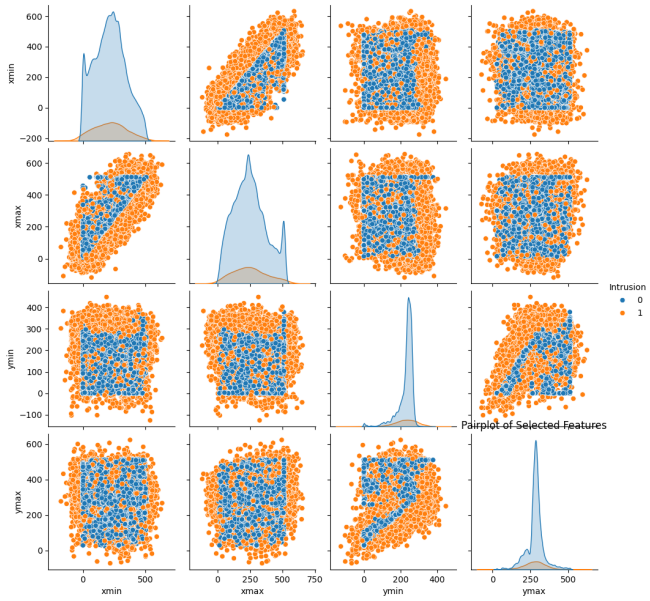


Figure 9 – Pairplot of selected features.

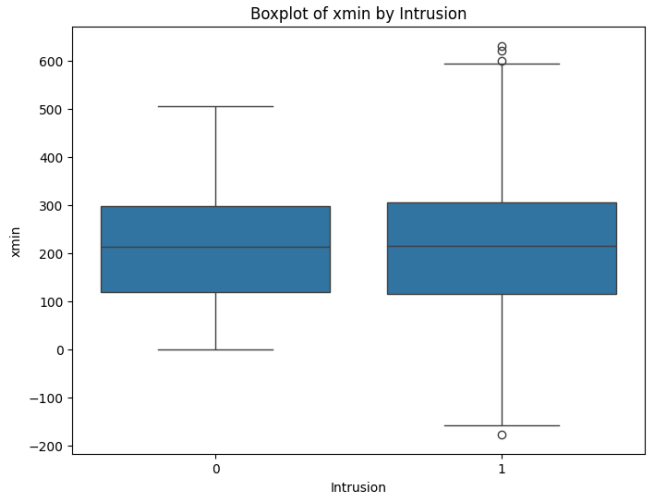


Figure 10 – Boxplot of xmin by intrusion.

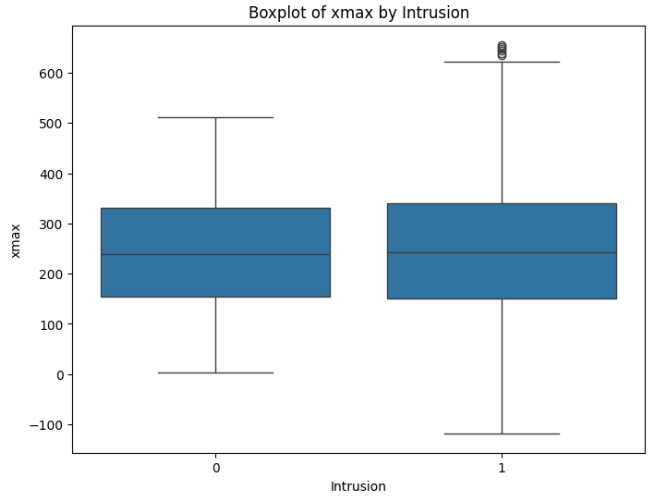


Figure 11 – Boxplot of xmax by intrusion.

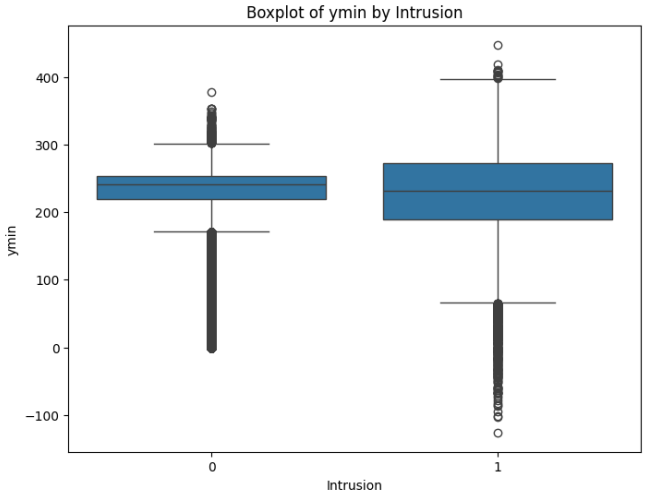


Figure 12 – Boxplot of ymin by intrusion.

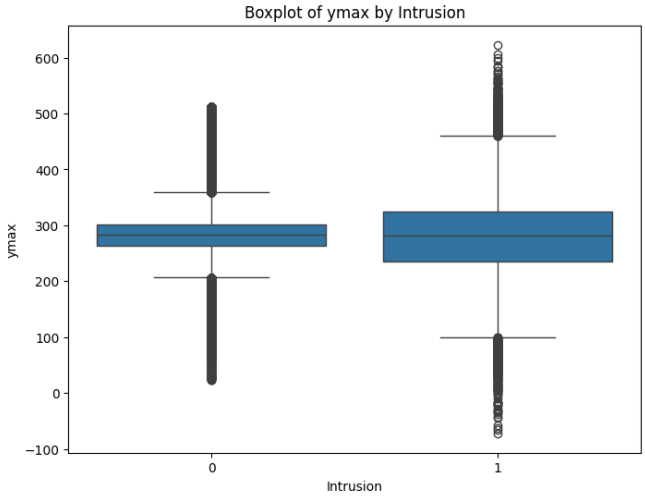


Figure 13 – Boxplot of ymax by intrusion.

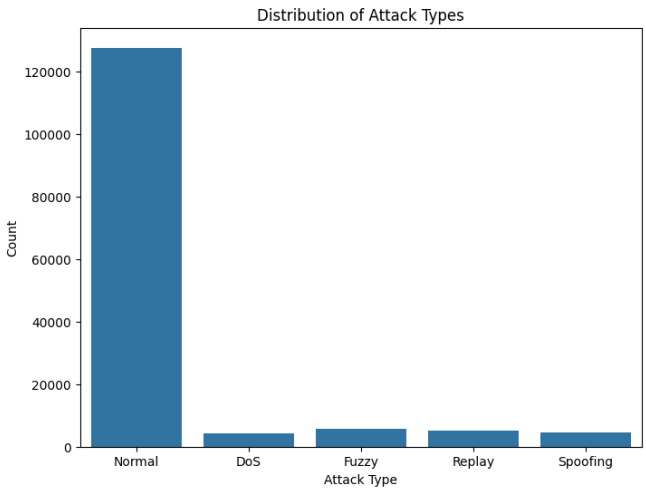


Figure 14 – Distribution of attack types.

the xmin-xmax correlation drops to 0.2 due to injected noise, while intrusion shows moderate correlations with perturbed features, confirming the simulated attacks' impact. Replay attacks retain strong spatial correlations

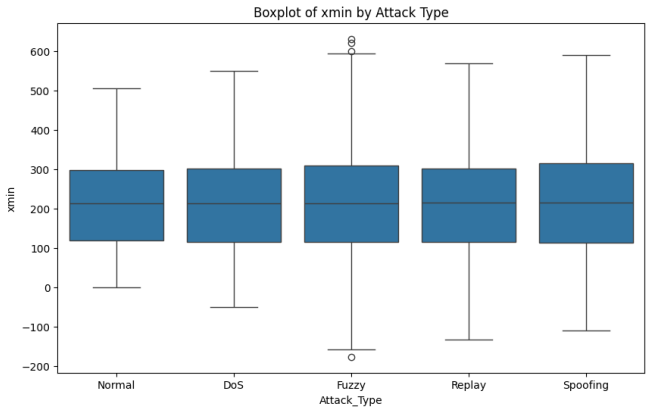


Figure 15 – Boxplot of xmin by attack type.

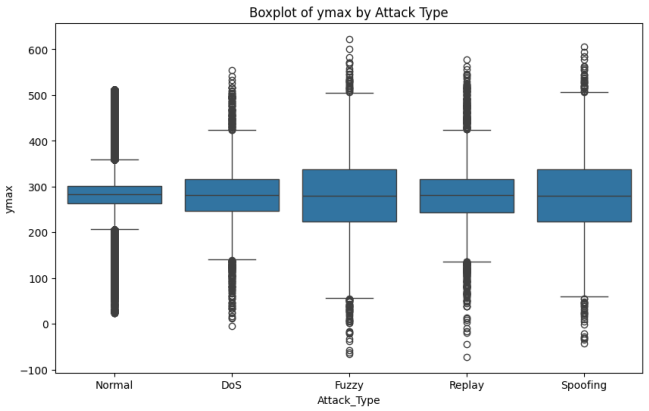


Figure 18 – Boxplot of ymax by attack type.

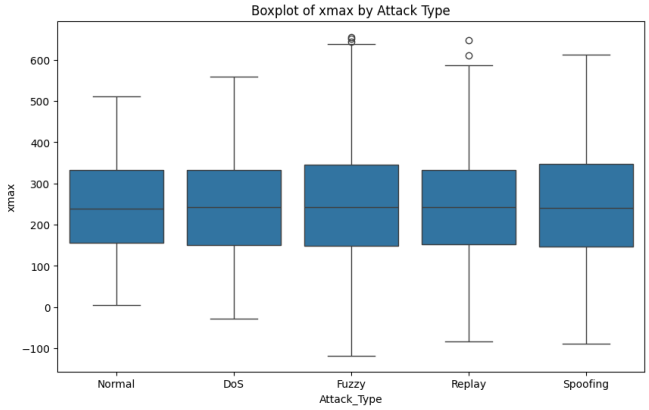


Figure 16 – Boxplot of xmax by attack type.

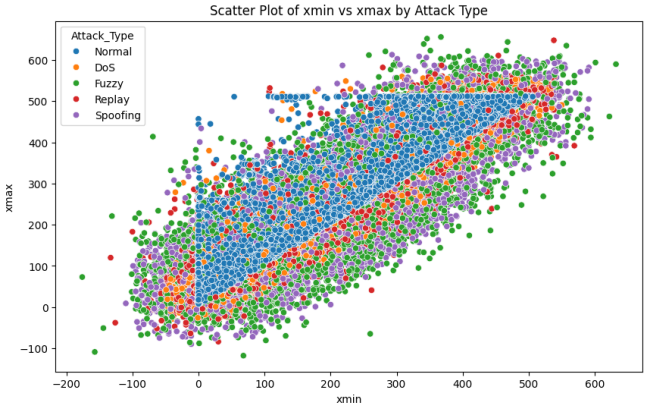


Figure 19 – Scatter plot of xmin vs xmax by Attack type.

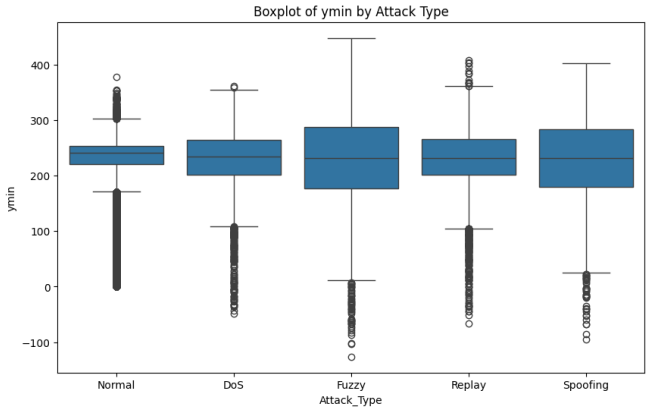


Figure 17 – Boxplot of ymin by attack type.

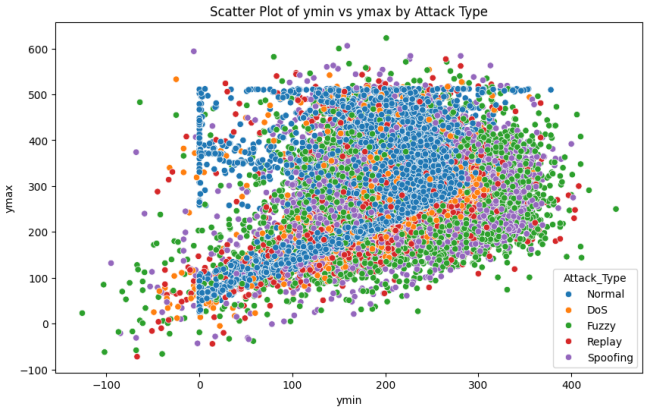


Figure 20 – Scatter plot of ymin vs ymax by attack type.

(e.g., xmin-xmax remains high), mimicking normal patterns, but also exhibiting subtle shifts reflected in low intrusion correlations. Fuzzy attacks erase nearly all correlations (e.g., $xmin - xmax \approx 0$), reflecting chaotic noise. These results validate the attack simulations' realism and underscore the need for adaptive models that leverage attack-specific feature interactions.

5. CONCLUSION

This work demonstrates a machine learning framework for detecting cyber-intrusions in self-driving vehicles, combining lightweight neural networks (NN-1: 92.34% accuracy, NN-2: 93.15%) and logistic regression (LG-1: 91.20%) to identify attacks (DoS, spoofing, replay, fuzzy) through perturbations in spatial features (`xmin`, `xmax`, `ymin`, `ymax`). Key contributions include:

- **Attack simulation** via noise injection into bounding boxes, validated by weakened feature correlations (e.g., `xmin-xmax` dropping to 0.2 for DoS).

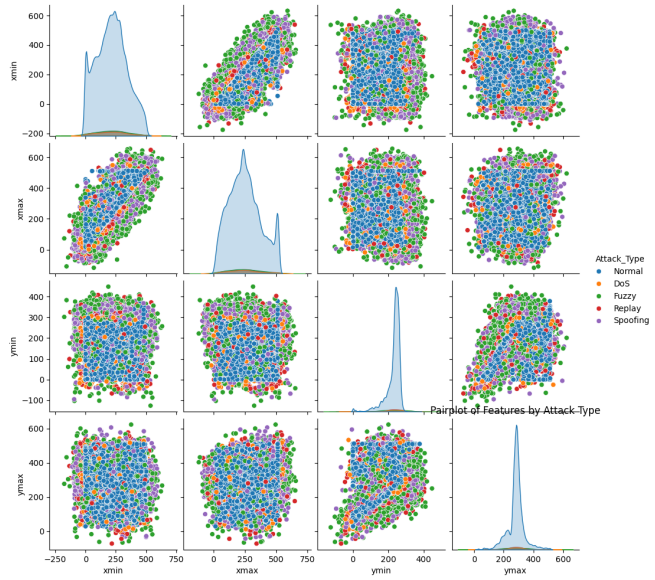


Figure 21 – Pairplot of features by attack type.

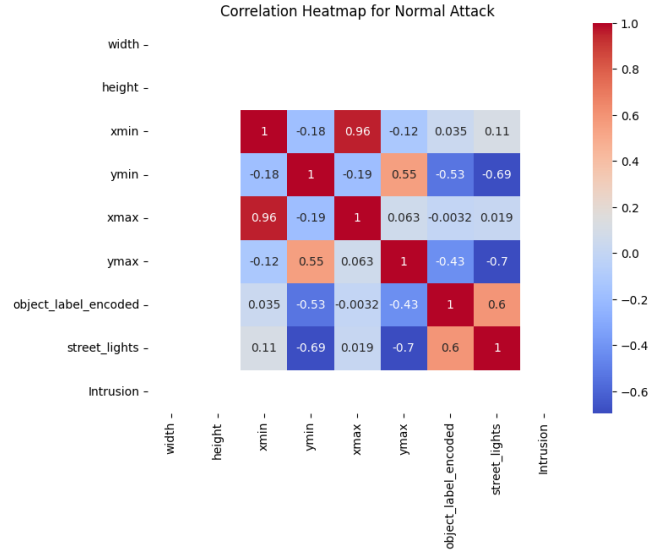


Figure 22 – Correlation heatmap by normal attack.

- **Edge-ready models** (NN-1: 3 hidden layers, NN-2: 4 layers) trained on NVIDIA T4 GPU, enabling local ECU deployment without cloud dependency.
- **Visual analytics** (confusion matrices, ROC curves with AUC up to 0.94, and attack-specific boxplots) quantifying model performance and attack patterns.
- **Explainable AI** (LG-1 coefficients, feature importance plots) identifying `xmin` and `xmax` as critical intrusion indicators.
- **Open-source implementation** (TensorFlow, PyTorch) ensuring reproducibility and scalability to new datasets.

Future work will validate the framework on diverse driving scenarios (e.g., urban, highway) and integrate multimodal data (LIDAR, camera feeds) to enhance robustness. Real-time latency optimization and adversarial testing

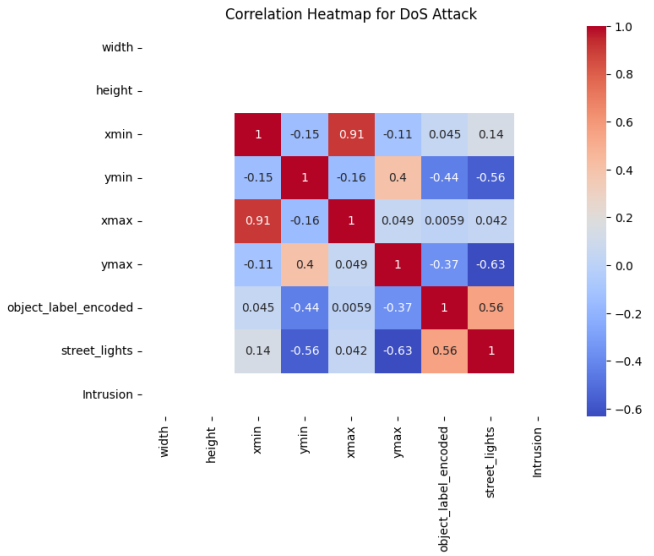


Figure 23 – Correlation heatmap by DoS attack.

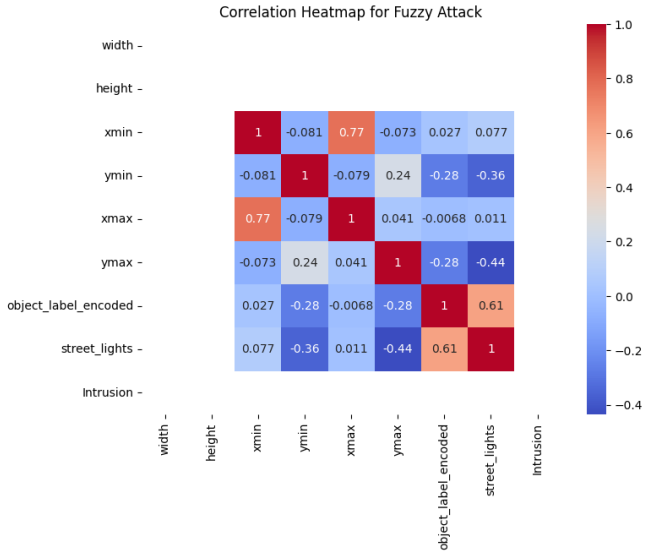


Figure 24 – Correlation heatmap by fuzzy attack.

against evolving attack strategies will further bridge the gap between research and practical ECU deployment, advancing edge-centric cybersecurity for autonomous vehicles.

ACKNOWLEDGEMENT

This work was supported in part by the US Department of Energy (DoE) under Grant No. CR0000037, US National Science Foundation (NSF) under Grant No. 2321055, and Lamar CAPM project 220892.

REFERENCES

[1] Ivo Berger, Roland Rieke, Maxim Kolomeets, Andrey Chechulin, and Igor Kotenko. "Comparative study of machine learning methods for in-vehicle intrusion detection". In: *International Workshop on Security and Privacy Requirements Engineering*. Springer. 2018, pp. 85–101.

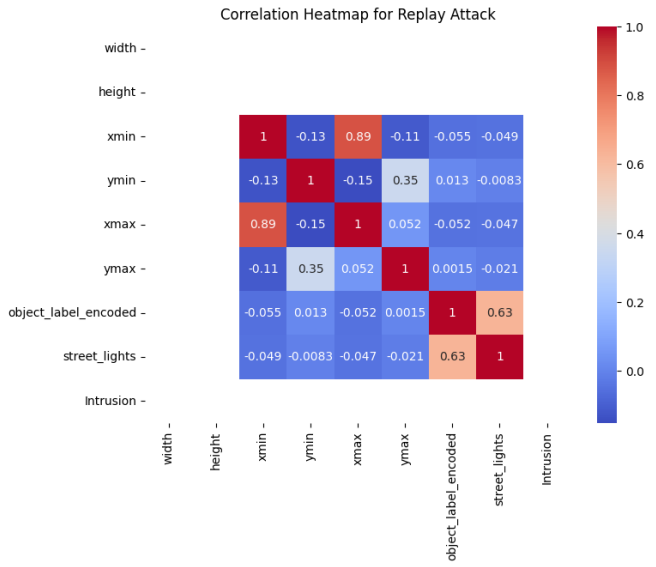


Figure 25 – Correlation heatmap by replay attack.

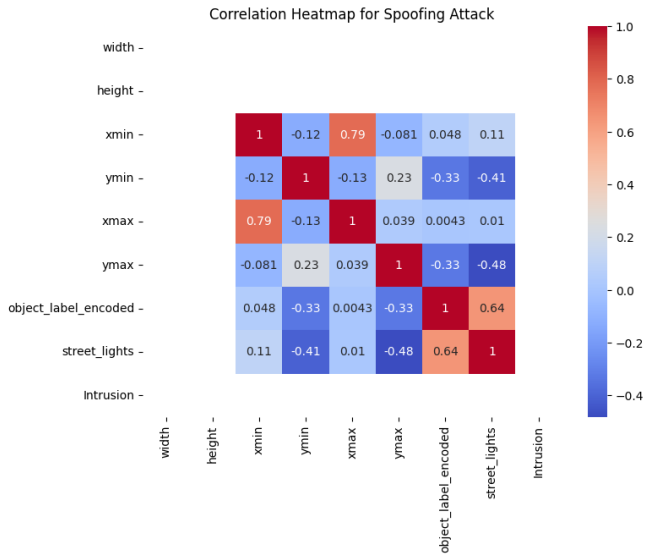


Figure 26 – Correlation heatmap by spoofing attack.

[2] Hunter Berry, Mai A Abdel-Malek, and Ahmed S Ibrahim. "A machine learning approach for combating cyber attacks in self-driving vehicles". In: *SoutheastCon 2021*. IEEE. 2021, pp. 1–3.

[3] Fawaz Waselallah Alsaade and Mosleh Hmoud Al-Adhaileh. "Cyber attack detection for self-driving vehicle networks using deep autoencoder algorithms". In: *Sensors* 23.8 (2023), p. 4086.

[4] Yashaswini S Pawar, Prasad Honnavalli, and Sivaraman Eswaran. "Cyber Attack Detection On Self-Driving Cars Using Machine Learning Techniques". In: *2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)*. IEEE. 2022, pp. 1–5.

[5] Soner Can Kalkan and Ozgur Koray Sahingoz. "In-vehicle intrusion detection system on controller area network with machine learning models". In: *2020 11th international conference on computing, communication and networking technologies (ICCCNT)*. IEEE. 2020, pp. 1–6.

[6] Afia Anjum, Paul Agbaje, Sena Hounsinou, and Habeeb Olufowobi. "In-vehicle network anomaly detection using extreme gradient boosting machine". In: *2022 11th Mediterranean Conference on Embedded Computing (MECO)*. IEEE. 2022, pp. 1–6.

[7] Asma Alfardus and Danda B Rawat. "Intrusion detection system for can bus in-vehicle network based on machine learning algorithms". In: *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE. 2021, pp. 0944–0949.

[8] Prosenjit Paul and Munjure Mowla. "3D Metallic Plate Lens Antenna Based Beamspace Channel Estimation Technique for 5G MMWAVE Massive MIMO Systems". In: *International Journal of Wireless & Mobile Networks (IJWMN)* 13.1 (Feb. 2021). Available at SSRN: https://ssrn.com/abstract=3830984.

[9] Prosenjit Paul and Md Munjure Mowla. "A Novel Beamspace Channel Estimation Technique for Millimeter Wave Massive MIMO Systems". In: *2019 3rd International Conference on Electrical, Computer Telecommunication Engineering (ICECTE)*. 2019, pp. 185–188. DOI: 10.1109/ICECTE48615.2019.9303527.

[10] Prosenjit Paul and Md Munjure Mowla. "A Statistical Channel Modeling for MIMO-OFDM Beamforming System in 5G mmWave Communications". In: *2019 3rd International Conference on Electrical, Computer Telecommunication Engineering (ICECTE)*. 2019, pp. 181–184. DOI: 10.1109/ICECTE48615.2019.9303526.

[11] Prosenjit Paul, Anup Kumar Roy, Nirman Bhowmike, Md Nazmul Islam, and Xingya Liu. "Design and Simulation of a Monofilar Helical Antenna for 5G and Beyond Applications". In: *2024 IEEE International Symposium on Measurements Networking (MN)*. 2024, pp. 1–6. DOI: 10.1109/MN60932.2024.10615465.

[12] Ahmad Shah Fuad, Mst. Fateha Samad, and Prosenjit Paul. "Design and Simulation of an Octafilar Helical Antenna (OHA) with Differential Feeding in Ku band Satellite Applications". In: *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*. 2019, pp. 1–4. DOI: 10.1109/EICT48899.2019.9068837.

[13] Harsh Sinha and Rakesh Tripathi. "Internet of vehicles: A study and comparison of machine learning and deep learning-based intrusion detection approaches". In: *AIP Conference Proceedings*. Vol. 2705. 1. AIP Publishing. 2023.

[14] Emad E Abdallah, Ahmad Aloqaily, and Hiba Fayez. "Identifying intrusion attempts on connected and autonomous vehicles: A survey". In: *Procedia Computer Science* 220 (2023), pp. 307–314.

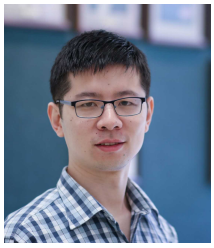
[15] Mahadev Satyanarayanan. "The emergence of edge computing". In: *Computer* 50.1 (2017), pp. 30–39.

[16] Lijun Chi, Mounira Msahli, Qingjie Zhang, Han Qiu, Tianwei Zhang, Gerard Memmi, and Meikang Qiu. "Adversarial Attacks on Autonomous Driving Systems in the Physical World: a Survey". In: *IEEE Transactions on Intelligent Vehicles* (2024), pp. 1–22. DOI: 10.1109/TIV.2024.3484152.

[17] Udacity. *Self-Driving Car Dataset Annotations*. https://github.com/udacity/self-driving-car/tree/master/annotations. Dataset 2, 2017.

AUTHORS

Prosenjit Paul is pursuing his Master of Engineering Science in Electrical and Computer Engineering from Phillip M. Drayer Department of Electrical and Computer Engineering, Lamar University, Beaumont, TX, USA. He completed his Bachelor of Science in Electronics and Telecommunication Engineering from Rajshahi University of Engineering and Technology, one of the top four engineering universities in Bangladesh. His research fields focus on AI and ML techniques to detect cyberattacks on cyber-physical systems. He is broadly interested in wireless communication and networks, cybersecurity, antenna design, VLSI physical design and verification. He has published some articles on prestigious journals and conferences such as IEEE and Elsevier.



Xingya Liu is an associate professor in the Department of Computer Science at Lamar University, Texas. Dr Liu received his Ph.D. degree in Computing Engineering from the University of North Carolina at Charlotte in 2017. His current research interests include cognitive and AI-enabled networks, cyber-physical infrastructure, and Internet of space things. Dr Liu also frequently served as a Technical Program Committee (TPC) member for international conferences, such as IEEE INFOCOM, IEEE GLOBECOM, and IEEE Edge Computing.

Helen H. Lou (Fellow, AIChE) is a professor in the Dan F. Smith Department of Chemical Biomolecular Engineering at Lamar University in Beaumont, Texas. She also holds the position of director for the Center for Data Analytics and Cybersecurity, and the Faculty Advisor for the Center for Midstream Management and Science. She received her Ph.D. in Chemical Engineering and M.A. in Computer Science from Wayne State University in Detroit, Michigan, in 2001. Recognized as an AIChE Fellow in 2018, she is also a registered Professional Engineer of Texas.

Ruhai Wang is a professor in Phillip M. Drayer Department of Electrical and Computer Engineering at Lamar University, Texas. Dr Wang received the Ph.D. degree in electrical engineering from New Mexico State University in 2001. His current research interests include cybersecurity, non-terrestrial networks, space communications and networks. He has published 120 papers in the area of space Internet, space networks, deep-space communication, and cybersecurity. Dr Wang is an IEEE Distinguished Lecturer of the Aerospace Electronic Systems Society (AESS) 2023-2026. Since 2012, he has consistently served on the editorial board of the IEEE Aerospace Electronics Systems Magazine (AES-M). He served as an associate editor for the IEEE Transactions on Aerospace Electronics Systems (TAES) from July 2018 to August 2020. Dr Wang also frequently served as a Technical Program Committee (TPC) chair/Co-Chair for international conferences/workshops, such as the 2007 IEEE International Conference on Communications (ICC 2007) and the 2008 International Workshop on Satellite and Space Communications (IWSSC 2008).