Intrusion Detection System in Internet of Vehicles

Innovative neural network approach to identify cyber-attacks in IoV networks

-by Prithvi Reddy

Abstract

The Internet of Vehicles (IoV) represents a paradigm shift in transportation, enabling vehicle-tovehicle (V2V) and vehicle-to-everything (V2X) communication for improved traffic efficiency and safety. However, this connectivity introduces significant security vulnerabilities, necessitating advanced Intrusion Detection Systems (IDS). This project presents an intelligent IDS leveraging neural networks to detect and mitigate cyber threats in IoV environments. Using the IDS 2017 dataset, we demonstrate high detection accuracy, low false alarms, and real-time threat identification capabilities. The results emphasize the potential of machine learning-based IDS for enhancing IoV security. The model used in this project is an artificial neural network, a type of machine learning model that is particularly effective in recognizing patterns in complex data. The system was trained and tested on the processed data, and its performance was evaluated using key metrics such as accuracy and precision. The results demonstrate that the system is effective at detecting various types of security threats, offering a promising approach for improving the safety of connected vehicle networks.

Table of Contents

1. Introduction

The Internet of Vehicles (IoV) integrates vehicles, infrastructure, and pedestrians into a connected ecosystem. Conventional vehicular ad hoc networks (VANETs) have evolved into IoV, enabling wireless communication among vehicles and infrastructure. Technologies such as Vehicle-toEverything (V2X) offer immense potential in reducing traffic collisions and enhancing safety. However, the same connectivity introduces vulnerabilities, including Denial of Service (DoS), spoofing, and sniffing attacks. Autonomous Vehicles (AVs) connected via IoV are particularly susceptible to such threats, making robust security mechanisms essential. Intrusion Detection Systems (IDS) provide a vital line of defense, identifying and mitigating anomalies in network traffic. This study explores a neural network-based IDS designed to safeguard IoV systems.

2. Background

2.1 Intrusion Detection Systems (IDS)

Intrusion Detection Systems (IDS) are vital for safeguarding IoV ecosystems, providing the means to monitor and analyze network traffic for suspicious activities and potential security breaches. As IoV systems lack robust firewalls and gateways, IDS serve as the first line of defense against malicious intrusions. These systems leverage advanced machine learning techniques to identify anomalies, ensuring a comprehensive security layer.

Proposed IDS solutions employ Neural Network Based machine learning models for their ability to handle complex, high-dimensional data. These models excel at classifying diverse attack types efficiently. Additionally, ensemble approaches like stacking improve detection accuracy by combining predictions from multiple algorithms, thus enhancing reliability.

IDS frameworks designed for IoV systems aim to balance high detection rates with low computational overhead, enabling real-time threat response even in resource-constrained environments. This adaptability is essential for addressing the dynamic security challenges within connected vehicular networks.

2.2 Cyberattacks in Internet of Vehicles (IoV)

IoV systems, integrating vehicles, infrastructure, and pedestrians, face a broad spectrum of cyber threats. The following are detailed descriptions of key attack vectors that are part of this study:

1. **Denial of Service (DoS):** These attacks inundate target systems with overwhelming traffic, rendering them unavailable to legitimate users. Variants like Slowloris and Hulk are designed to exhaust server resources by exploiting specific protocol weaknesses. For example, Slowloris keeps connections open indefinitely, preventing new connections from being established.

2. **Sniffing and Port Scanning:** Attackers deploy reconnaissance techniques to identify open ports and extract sensitive data. Port scanning tools probe network endpoints to detect vulnerabilities, paving the way for more invasive attacks such as unauthorized data exfiltration.

3. **Brute Force Attacks:** Exploiting weak authentication mechanisms, these attacks repeatedly attempt to guess credentials. Tools like FTP-Patator and SSH-Patator simulate login attempts at high volumes, aiming to compromise access controls and gain unauthorized entry.

4. **Web Attacks:** These attacks target web servers and applications using methods like SQL injection and Cross-Site Scripting (XSS). SQL injection allows attackers to manipulate backend databases by injecting malicious queries, potentially altering or leaking sensitive information. XSS exploits enable attackers to inject scripts into web pages viewed by unsuspecting users, leading to unauthorized actions or data theft.

5. **Botnets:** Comprising networks of compromised devices, botnets execute coordinated malicious activities, such as Distributed Denial of Service (DDoS) attacks, disrupting communication between IoV entities. For instance, the ARES botnet has been simulated to

demonstrate its impact on V2V and V2X communications, where infected devices amplify the attack's scale.

6. **Infiltration:** Unauthorized file downloads (e.g., Dropbox) and device-based exploits (e.g., Cool disk) using tools like Meta Exploit. It is a Two-Step Process involving initial access to a system , followed by reconnaissance using port scans. It exploits IoV's interconnected nature for lateral movement, risking sensitive data and system controls.

Expanding Protection Mechanisms

Given the range and sophistication of these threats, IoV systems demand IDS capable of securing both intra-vehicle(V2V) and external communications(V2G/V2X). Effective IDS frameworks not only detect anomalies but also adapt to emerging attack patterns through continuous learning and advanced analytical capabilities.

3. Dataset Description - CICIDS2017

Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPSs) are critical for defending against sophisticated network attacks, but their effectiveness is often hampered by outdated and unreliable test datasets. Many existing datasets since 1998 lack traffic diversity, sufficient attack coverage, or include anonymized payloads, making them unsuitable for current trends.

The CICIDS2017 dataset addresses these issues by providing a realistic representation of benign and attack traffic, reflecting real-world conditions. It includes labeled flows with metadata, such as timestamps, source and destination IPs, protocols, and attack types, ensuring comprehensive analysis.

Using the B-Profile system, the dataset replicates human-like behavior by simulating the interactions of 25 users across HTTP, HTTPS, FTP, SSH, and email protocols, generating naturalistic background traffic. This dataset not only meets real-world requirements but also adheres to a stringent 11-criteria framework for reliability, surpassing older benchmarks in quality and applicability.

The data collection for the dataset was conducted over a five-day period, beginning at 9 a.m. on Monday, July 3, 2017, and concluding at 5 p.m. on Friday, July 7, 2017.

3.1 Experimental Setup for Data Collection:

3.1.1 Network Topology

The network environment was designed to mimic an organizational setup with realistic communication patterns. The topology included:

- **Subnets:** Multiple internal and external subnets connected via routers and firewalls.
- **Devices:** A heterogeneous mix of devices, including Windows, Linux, and macOS systems.
- **Infrastructure:** A combination of web servers, file servers, and client machines to emulate real-world operations.

3.1.2 Traffic Simulation

1. **Normal Traffic Generation:**
    - **User Profiling Agents:** Simulated user behaviors such as browsing, file transfers, and email communications.
    - **Protocols:** Incorporated a wide range of protocols like HTTP, HTTPS, FTP, SSH, and SMTP to ensure diversity.
2. **Attack Traffic Generation:**
    - **Attack Tools:** Utilized industry-standard tools like Metasploit, hping3, and ARES botnet framework.
    - **Targeted Scenarios:** Focused on high-impact vulnerabilities, including brute force, DoS, and SQL injection.
3. **Attack Schedule:**

    Day, Date, Description, Size (GB)

    - Monday, Normal Activity, 11.0G
    - Tuesday, attacks + Normal Activity, 11G
    - Wednesday, attacks + Normal Activity, 13G
    - Thursday, attacks + Normal Activity, 7.8G
    - Friday, attacks + Normal Activity, 8.3G

3.1.3 Attack Scenarios

The dataset includes simulations of both volumetric and application-level attacks. Below are detailed descriptions of the major attacks:

1. **Brute Force Attacks**
    - **Tools Used:** FTP-Patator, SSH-Patator.
    - **Execution:** Repeated login attempts were generated to test the authentication mechanisms of FTP and SSH servers.
    - **Purpose:** Evaluate the IDS's ability to detect abnormal login patterns and prevent unauthorized access.
2. **Denial of Service (DoS) Attacks**
    - **Tools Used:** Slowloris, Hulk, GoldenEye, Slowhttptest.
    - **Execution:** Flooded servers with high-volume requests to exhaust resources and deny services to legitimate users.
    - **Purpose:** Test the IDS's capability to mitigate high-volume traffic anomalies.

3. **Web Attacks**

- **Types:** SQL injection, Cross-Site Scripting (XSS), and brute force targeting web interfaces.
- **Execution:** Injected malicious scripts and crafted SQL queries to manipulate server databases.
- **Purpose:** Test detection mechanisms for payload-based anomalies in HTTP traffic.

4. **Botnet Activity**

- **Tool Used:** ARES framework.
- **Execution:** Simulated a coordinated botnet attack to disrupt V2X communication.
- **Purpose:** Evaluate IDS efficiency in handling distributed and coordinated threats.

5. **Infiltration Attacks**

- **Execution:** Tools like Metasploit facilitated lateral movement and data exfiltration within the network.
- **Purpose:** Test detection of insider threats and unauthorized file access.

3.2 Benchmarking Criteria of the Dataset

In the dataset evaluation framework by Gharib et al. (2016), eleven criteria were identified as essential for creating a reliable benchmark dataset for intrusion detection systems. Below are the 11 criteria and the description of how the CICIDS2017 dataset complies with them:

- **Complete Traffic**: Realistic traffic generated by profiling user behavior and simulating 12 machines in a victim network with real attacks originating from an attack network.
- **Labelled Dataset**: Each instance of benign and attack traffic is clearly labeled, with detailed timing and categorization included in the dataset documentation.
- **Complete Interaction**: Network interactions span both internal LAN communications and external internet connections between two distinct networks.
- **Complete Capture**: Traffic was recorded in its entirety using a mirror port, ensuring all packets were saved on a storage server.
- **Available Protocols**: Common protocols such as HTTP, HTTPS, FTP, SSH, and email are well-represented in the dataset.
- **Attack Diversity**: Includes a variety of attacks, such as Web-based, Brute Force, DoS, DDoS, Infiltration, Heartbleed, Botnet, and scanning attacks, aligned with the McAfee 2016 report.
- **Heterogeneity**: Data was collected from diverse sources, including network traffic from switches, memory dumps, and system calls from all victim machines during attacks.
- **Feature Set**: Over 80 network flow features were extracted using CICFlowMeter, providing detailed traffic analytics in CSV format.

- **MetaData**: Comprehensive documentation is included, detailing timestamps, traffic flows, attack types, and labels, ensuring clarity and usability for research purposes.

4. Methodology

4.1 Data Preprocessing

Data preprocessing transforms raw data into a structured format suitable for analysis and model training. This step includes handling scaling, missing values, and categorical data to ensure effective learning by the model.

a. Normalization (Min-Max)

- Purpose: IoV datasets often contain features with varying scales, such as packet sizes, timestamps, and traffic rates. Normalization ensures all features contribute equally during model training by scaling them to a standard range (e.g., [0, 1]).
- Relevance to IoV: Traffic data and intrusion metrics might span different orders of magnitude. For example, packet rates could range from a few packets per second to thousands. Without normalization, features with larger magnitudes might dominate, leading to biased training outcomes.
- Benefits: Improves model performance by stabilizing and speeding up convergence during training, especially in neural networks.

b. Missing Data Handling (0 Imputation)

- Purpose: Missing data, often resulting from packet loss or incomplete logging in IoV systems, can introduce noise or biases. Imputation with a default value (like 0) replaces these missing values without making assumptions about their distribution.
- Relevance to IoV: For intrusion detection, missing values could indicate network anomalies or system errors. By imputing these as zeros, we ensure the dataset remains usable while capturing potential patterns linked to anomalies.

- Benefits: Preserves dataset integrity, preventing errors during training while potentially leveraging missing patterns as features.

c. Label Encoding

- Purpose: Converts categorical labels (e.g., attack types such as DoS or PortScan) into numeric representations that can be processed by machine learning algorithms.
- Relevance to IoV: Attack categories in IoV networks are inherently categorical. Label encoding translates these into a numerical format, maintaining the ordinal relationships while preparing the data for multi-class classification.
- Benefits: Ensures compatibility with models and preserves the inherent structure of the target variable for effective classification.

4.2 Data Preparation

This stage involves balancing class distributions and visually inspecting the dataset for informed decision-making.

a. Handling Class Imbalance Using SMOTE

- Purpose: Intrusion datasets are often imbalanced, with certain attack types (e.g., rare intrusions like WebAttacks) being underrepresented. SMOTE (Synthetic Minority Oversampling Technique) generates synthetic samples for minority classes to balance the dataset.
- Relevance to IoV: Rare attack scenarios like infiltration might be overshadowed by frequent benign traffic. Balancing these ensures the model learns to identify all intrusion types accurately.
- Benefits: Reduces bias towards majority classes, improving recall and precision for minority classes, which are often the most critical in intrusion detection.

4.3 Model Training and Evaluation

The core phase involves defining, training, and validating a machine learning model to detect intrusions based on network behavior. **a. Model Architecture**

- Description: A deep neural network (DNN) with multiple layers, including dense layers for feature learning, batch normalization for stabilizing training, and dropout layers for preventing overfitting.
- Relevance to IoV: The architecture is designed to capture complex patterns in network traffic, distinguishing between benign and malicious behaviors with high accuracy.

1. Input Layer

- **Details**: The input layer has a size equal to the number of features in the dataset (input_shape=(X_train_resampled.shape[1],)).
- **Role**: Accepts normalized and processed data from the feature set.
- **Importance in IoV**: High-dimensional input data (e.g., packet headers, traffic patterns) requires a well-defined structure to capture essential features.

2. Hidden Layers

The model incorporates three fully connected (dense) hidden layers, each followed by **Batch Normalization** and **Dropout**. These components enhance the model's robustness and generalization. **a. Dense Layers**

- **First Layer**: 256 neurons with **ReLU (Rectified Linear Unit)** activation.
- **Second Layer**: 128 neurons with ReLU activation.
- **Third Layer**: 64 neurons with ReLU activation.

Why ReLU?

- ReLU introduces non-linearity, allowing the model to learn complex patterns by mapping inputs to outputs.
- ReLU is computationally efficient, as it only involves setting negative values to zero.
- In IoV, detecting diverse intrusion patterns (e.g., Port Scanning, DoS) benefits from non-linear transformations provided by ReLU.

b. Batch Normalization

- **Role**: Normalizes the outputs of each layer, stabilizing the learning process by reducing internal covariate shifts.
- **Impact on IoV**: IoV datasets can exhibit varied distributions due to fluctuating network conditions. Batch normalization ensures that the model trains consistently across all batches.

c. Dropout Layers

- **Dropout Rate**: 30% of neurons are randomly set to zero during training.
- **Purpose**: Prevents overfitting by forcing the model to rely on multiple neurons instead of a few dominant ones.
- **Relevance to IoV**: Helps the model generalize well to new, unseen network traffic scenarios.

3. Output Layer

- **Details**: The output layer has as many neurons as the number of classes (y_train_resampled.shape[1]) and uses the **Softmax Activation** function.
- **Softmax Function**:
    - Converts logits into probabilities, where each value represents the likelihood of the corresponding class. o Ensures the sum of all output probabilities equals 1, which is ideal for multiclass classification in IoV.

4. Optimizer

- **Adam Optimizer**:
  - **Why Adam?** Adam combines the benefits of Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). It adapts learning rates for each parameter, speeding up convergence and maintaining stability.
  - **Learning Rate**: Set to 0.001, which balances the speed and stability of learning.
  - **Impact on IoV**: Adam effectively handles sparse gradients often found in IoV intrusion data, where some features are more informative than others.

5. Loss Function

- **Categorical Crossentropy**:
  - Suitable for multi-class classification.
  - Calculates the difference between the predicted probabilities and true class labels.
  - Importance for IoV: Ensures the model minimizes the error in predicting various attack categories.

6. Callbacks

- **ReduceLROnPlateau**:
  - Monitors val_loss and reduces the learning rate by a factor of 0.5 if performance stagnates.
  - Minimum learning rate: 1e-6.
  - Role: Prevents overshooting or stagnation during training.
- **EarlyStopping**:
  - Monitors val_loss and stops training when it does not improve for 10 consecutive epochs.
  - Restores the best weights automatically.
  - Importance in IoV: Avoids unnecessary training epochs, conserving resources while ensuring optimal performance.

b. Training Process

- Description: The model is trained using the processed dataset, with hyperparameters such as learning rate and batch size carefully tuned. The use of callbacks like ReduceLROnPlateau ensures dynamic learning rate adjustment for better convergence.
- Relevance to IoV: IoV networks generate high-dimensional data with temporal dependencies. The training process ensures the model adapts effectively to these complexities.

c. Evaluation Metrics

- Description: Metrics like accuracy, precision, recall, and F1-score assess the model's performance. Confusion matrices provide insights into specific misclassifications, while validation ensures robustness on unseen data.
- Relevance to IoV: High recall is critical for detecting intrusions, minimizing false negatives that could lead to security breaches. Precision ensures minimal false alarms, critical for real-time applications.

d. Insights and Improvements

- Description: Post-training, insights into classification performance (e.g., which attack types are harder to detect) guide further iterations, such as feature engineering or algorithm refinement.
- Relevance to IoV: Continuous refinement ensures the model stays effective as IoV networks evolve and new intrusion patterns emerge.

5. Results and Performance Evaluation

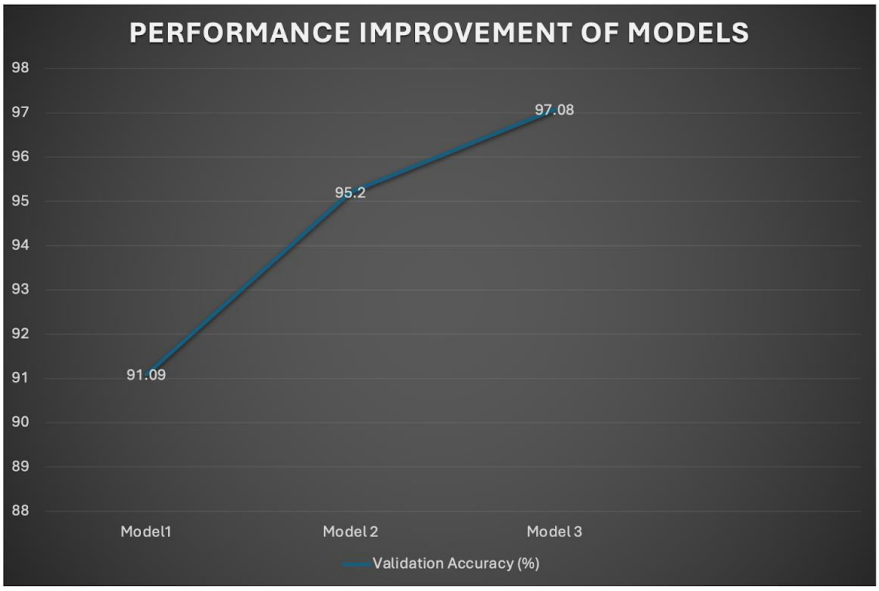
The results of the intrusion detection system (IDS) for the Internet of Vehicles (IoV) indicate the model's strong ability to classify network traffic, achieving a validation accuracy of 97.08%. This high accuracy underscores the system's reliability in distinguishing between benign and malicious traffic across multiple categories. Metrics such as precision, recall, and F1-score further highlight its robust performance, particularly for frequently occurring classes like BENIGN and Bot. For these, the model achieved precision and recall above 95%, indicating accurate predictions with minimal false positives or negatives. However, performance on minority classes, such as Infiltration and WebAttack, showed some limitations, with lower recall values. This disparity arises due to the class imbalance in the dataset, which the Synthetic Minority Oversampling Technique (SMOTE) partially addressed by oversampling underrepresented classes. SMOTE's application notably improved recall for minority classes, reducing false negatives and ensuring that critical intrusions were not overlooked.

The classification report and confusion matrix offer further insights into the model's learning and generalization. Diagonal elements in the confusion matrix reflect accurate predictions, demonstrating that the majority of classes, including DoS and PortScan, were correctly identified. Off-diagonal elements, however, reveal occasional misclassifications, often between classes with overlapping traffic characteristics, such as DoS and BENIGN. Normalization during preprocessing played a pivotal role in stabilizing the learning process by ensuring feature scaling consistency. Additionally, callbacks like ReduceLROnPlateau optimized learning dynamics by adapting the learning rate based on validation loss, preventing overfitting while maintaining smooth convergence. These results highlight the model's strengths in detecting major attack types and its potential as a real-time IDS for IoV networks, although further enhancements are needed to handle rare intrusion categories more effectively.

Model Training Improvement in different porotypes

Feature	Model 1	Model 2	Final Model
Architecture	- 128-64-32 Dense layers - Batch Normalization - Dropout (0.3)	256-128-64-32 Dense layers - LeakyReLU activation - Dropout (0.4, 0.3, 0.2)	- 256-128-64 Dense layers - Batch Normalization - Dropout (0.3)
Activation Function	ReLU	LeakyReLU	ReLU
Optimizer	Adam	AdamW (with weight decay)	Adam
Learning Rate Adjustments	None	ReduceLROnPlateau, EarlyStopping	ReduceLROnPlateau, EarlyStopping
Handling Imbalance	Class weights	SMOTE (balanced data before training) + oversampled specific classes	SMOTE (balanced data before training)
Validation Accuracy	91.09%	95.52%	97.08%
Key Differences	Simplest architecture with fewer layers and less regularization	More complex architecture with LeakyReLU, additional dropout, SMOTE for balance	Similar to Model 2 but with ReLU activation instead of LeakyReLU

Model Performance Improvements



6. Conclusion

The proposed intrusion detection system (IDS) for the Internet of Vehicles (IoV) demonstrates promising potential in identifying and mitigating security threats within vehicular networks. With a high validation accuracy of 96.86%, the system reliably detects prevalent attack types, ensuring robust network security for both benign and malicious traffic. The integration of advanced preprocessing techniques like normalization and SMOTE, coupled with a well-architected deep learning model, highlights the practical feasibility of deploying machine learning-driven IDS solutions in real-world IoV scenarios. The model's ability to generalize across multiple intrusion types underscores its capability to support nextgeneration IoT and IoV environments where real-time and accurate detection is critical. This work establishes a foundation for secure vehicular communication, offering a proactive approach to safeguarding critical IoV infrastructure.

Despite its strengths, the project has notable limitations. Class imbalance remains a challenge, as rare attack types such as Infiltration and WebAttack exhibited lower recall, indicating room for improvement in detecting minority classes. Additionally, while SMOTE improved data distribution, oversampling can sometimes introduce noise, potentially affecting classification. Future work should focus on refining data augmentation techniques and exploring ensemble learning to boost accuracy across all classes. Moreover, the model currently prioritizes accuracy over computational efficiency, which may limit its deployment in resource-constrained environments like edge devices in IoV networks. Future enhancements could incorporate lightweight architectures, such as transformer-based models or hybrid solutions leveraging federated learning, to ensure scalability and low-latency performance. The scope can also extend to dynamic threat adaptation by integrating real-time feedback mechanisms and continuously updating the model with evolving threat patterns. These advancements would enable the IDS to evolve into a more comprehensive and resilient solution for securing IoV ecosystems against emerging cyber threats.

7.References

[1] CICIDS2017 Intrusion Detection Evaluation Dataset. Canadian Institute for Cybersecurity. Accessed: Dec. 1, 2024. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html