

A Hybrid Trust-and-Learning Approach for Real-Time DDoS Detection in VANETs

Abstract

Vehicular Ad-Hoc Networks (VANETs) allow cars and roadside units to send safety and traffic messages to each other. These messages should arrive quickly and without delay, because they may warn about accidents or road problems. When a Distributed Denial-of-Service (DDoS) attack happens, the network can be filled with many useless packets, and the important safety messages may be delayed or even lost.

In many earlier works, the authors used either simple trust rules that are fast but easy to trick, or heavy machine learning models that are more accurate but slow and hard to run on real devices in cars. In this work, I suggest a simple hybrid idea: a fast trust check at the edge, and a small learning model behind it. These two parts share some small updates so they can follow what is really happening on the road. The goal is to react quickly to attacks but still keep good detection quality.

Keywords: VANET, DDoS, trust, machine learning, hybrid detection

1. Introduction

Vehicular Ad-Hoc Networks (VANETs) let cars talk to each other and to roadside units. They share messages about traffic, road conditions, and safety alerts. Because the network is open and cars keep moving from place to place, attackers can send fake or junk packets into the system. These extra packets can slow down or hide the real safety messages, which is dangerous.

When I read earlier research, I noticed that many solutions follow one of two ways. Some solutions only use quick and simple checks, which are fast but cannot see deeper patterns. Other solutions use more complex analysis, which is stronger but needs more time and more computing power. In my project, I want to combine both ideas: a fast part that reacts first to the traffic, and a small learning part that looks again only at traffic that seems strange or suspicious. In this way, the system can stay fast but still give decisions that I can trust.

2. Background and Related Work

Trust-based systems watch how each node in the network behaves. For example, they can count how many messages a node sends, or look at the timing between its messages. If a node sends too many messages or behaves in a strange way, its trust score goes down. These systems are not very complex, and in many test scenarios they work well. However, if an attacker sends packets slowly and carefully, or several attackers work together, they may still pass these checks.

On the other side, some methods use machine learning or deep learning. These methods try to learn patterns from network traffic data. They can catch more advanced or hidden attacks, which is good. But they usually need a lot of training examples and stronger hardware, and they can make the system slower if used directly in a real VANET.

Traffic in VANETs does not stay the same. Sometimes the road is almost empty; at other times it is very busy. Because of this, we still need an approach that can answer quickly but does not lose too much accuracy when road and network conditions change.

3. Problem Statement

Domain

This work is about real-time protection of safety messages in VANETs against DDoS attacks.

Problem

Many existing systems use only one style of defense. Simple rule-based systems are fast, but smart attackers can trick them. Heavy learning models are more accurate, but they are slow and may not fit well on real VANET hardware, such as on-board units in cars. It is hard to get both speed and good accuracy at the same time.

Significance

If safety messages are delayed or dropped because of attack traffic, drivers may miss important warnings about accidents, sudden stops, or road hazards. This can increase the chance of crashes and can reduce trust in VANET technologies in general. A better detection method can help keep roads safer.

Gap

Trust-based filters can react quickly, but slow or coordinated attackers can sometimes bypass them. Learning-based models can detect complex attacks, but they need more data and computing power, and they can add delay to the system. There is still no simple and balanced solution that combines both speed and smart detection in a way that fits real VANET conditions.

Need

We need a hybrid design that uses fast checks at the edge of the network and a small learning model in the back end. The two parts should share light feedback, so that their thresholds and decisions follow the current state of the road and the traffic, instead of using fixed values all the time.

4. Research Objectives

The main goal of this project is to design and evaluate a hybrid DDoS detection approach for VANETs that is both fast and effective. To reach this goal, I define the following specific objectives:

1. To design a small trust-based edge filter that checks short-term traffic behavior and keeps processing time low.
2. To build a compact learning module, for example using a Random Forest classifier and a small autoencoder, that can correctly separate normal and attack traffic in collected datasets.
3. To connect both parts so that unclear or suspicious traffic goes to the learning module, and the learning module can update the trust thresholds at the edge from time to time to reduce overall delay.

4. To compare the hybrid system with three baselines: a trust-only method, a simple threshold method, and a learning-only method, all tested under the same scenarios.
5. To measure accuracy, false positives, detection delay, packet delivery, and resource usage (CPU, memory, and communication cost), and to show that the hybrid design performs better than using only one type of method.
6. To study how the system behaves under different traffic conditions (city, suburban, highway) and different attack patterns, and to see how much the feedback mechanism helps the system stay up to date.

5. Proposed Method

5.1 Edge Filter (OBU/RSU)

The first part of the system is a simple filter that runs close to the cars, for example on an on-board unit (OBU) or a roadside unit (RSU). This filter looks at basic features such as how many packets a sender sends in a short time, how much time passes between packets, and whether the sender repeats the same message too often. Using this information, the edge filter gives each sender a basic label, such as “normal”, “suspicious”, or “blocked”. It can also change its own limits (for example, allowed packet rate) based on how busy the road is and what type of area it is (city, suburban, or highway).

5.2 Back-End Learner (Edge Server / Cloud)

The second part is a small machine learning model that runs on an edge server or a nearby machine with more resources. This learner looks at longer periods of traffic and uses more features than the simple edge filter. For example, it can use average rate over a longer window, standard deviation, or other simple statistics. In this work, I plan to test a Random Forest classifier and a small autoencoder-based model. The learner takes data from the edge filter about suspicious flows or nodes and decides if they are likely to be attacks or not. It then sends its decisions and some suggested threshold updates back to the edge filter.

5.3 Feedback Loop

To keep both parts aligned, there is a simple feedback loop. From time to time, or when traffic patterns clearly change, the back-end learner sends new hints or thresholds to the

edge filter. Safety-related messages, such as emergency warnings, are never fully blocked. Instead, traffic that is unclear is slowed down or limited and checked more carefully. In this way, the edge and the learner stay in sync and can adapt to changing road conditions.

6. Experimental Plan

Scenarios

I plan to test the system in several simulated road settings, such as city roads during rush hour, city roads during off-peak times, suburban roads, and highways. In each setting, I will include different attack styles, such as short, fast bursts of packets, slow attacks that send packets at a low rate, on-off attacks, and attacks from several nodes that work together.

Baselines

The hybrid method will be compared against three simpler designs: a trust-only method (only edge filter, no learning), a simple threshold method (fixed rate limits, no trust score, no learning), and a learning-only method (machine learning model without trust filter).

Metrics

To judge the system, I will measure precision, recall, F1 score, false positive rate, detection delay, packet delivery ratio for safety messages, link latency, CPU usage, memory usage, and extra communication caused by sending data to the learner.

Targets

The targets are high accuracy (around 95% F1 if the data allows it), low false positive rate so normal traffic is not blocked too much, low detection delay so attacks are found quickly, and low load on devices so the system can run on real VANET hardware.

7. Expected Outcomes

I expect that the hybrid design will react quickly because of the edge filter, while the learning module will help improve the quality of the decisions. By sending small updates from the learner to the edge filter, the system can follow changes in traffic patterns. The

expected result is that the hybrid system will have fewer false alarms than a trust-only system and less delay than a learning-only system. It should also keep the delivery of safety messages at a high level, even when there is DDoS attack traffic in the network.

8. Limitations and Scope

At the beginning, all tests will be done in a simulator. This is because it is easier to control vehicle movement and attack patterns in a simulation than in the real world. Real traffic data may be added later if the basic design works well.

The learning part is kept small on purpose. This is to keep CPU and memory needs low, so that the model can run on normal hardware without special GPUs. Only general traffic features are used, not personal data, so there are no direct privacy concerns in this design.

9. Conclusion

In this work, I aim to build and test a simple hybrid system for DDoS detection in VANETs. The main idea is to mix a fast trust-based filter at the edge with a small learning model in the back end. The trust filter gives quick first decisions, while the learning model checks the more difficult cases and sends updates back to the edge.

By combining these two parts, the system tries to balance speed and accuracy. It should be more practical for real VANET conditions than using only simple rules or only heavy learning models. In the end, the goal is to keep important safety messages moving through the network, even during attacks, and to reduce the risk for drivers on the road.

Readable Notes (Summary of the Six Papers)

- Paper 1 – Uses a leader car in each group to watch other cars and update trust. Good in many cases, but if the leader fails or is attacked, the group can suffer.
- Paper 2 – Uses fuzzy rules and trust scores. More flexible than fixed rules, but needs careful tuning to avoid false alarms.
- Paper 3 – Uses deep learning with autoencoders. Very strong detection, but needs clean data and more computing power.

- Paper 4 – Combines trust and opinions from several cars. Helpful in many situations, but extra reports can add noise.
- Paper 5 – Uses simple monitoring of bandwidth and packet rates. Easy to implement, but may miss slow or low-rate attacks.
- Paper 6 – Uses cluster-based scoring of nodes based on behavior. Works well in tests, but thresholds must be adjusted when conditions change.

Overall, these papers show that no single method is perfect. Trust-based systems are fast but can be fooled; learning systems are accurate but heavy. This is why a mixed, hybrid design may be a good next step.

References

- [1] A Distributed Network Intrusion Detection System for Distributed Denial of Service Attacks in Vehicular Ad Hoc Network.
- [2] DDoS Detection Mechanism Using Trust-Based Evaluation System in VANET.
- [3] Polat et al., “Deep Network Approach with Stacked Sparse Autoencoders in Detection of DDoS Attacks on VANETs,” IET Communications, 2021.
- [4] 2018 CCAA Study on DDoS Detection/Mitigation in VANET (NS-2).
- [5] SPPA-Style Cluster Analysis for VANET DDoS (2020).
- [6] Trust/Intelligent DDoS Detection in VANET (Alternate Trust Model Paper).