Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# Comprehensive Host-Based Malicious BehaviourDetection in VANETs

**HAMIDEH BAHARLOUEI**

hm729953@dal.ca

Dalhousie University

**Adetokunbo Makanju**

New York Institute of Technology

**Nur Zincir-Heywood**

Dalhousie University

**Research Article**

**Keywords:** VANET, anomaly detection, malicious behaviour detection, federated learning, AI/ML, security, privacy

**Posted Date:** March 4th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-8468681/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

**Version of Record:** A version of this preprint was published at Journal of Network and Systems Management on May 23rd, 2026. See the published version at https://doi.org/10.1007/s10922-026-10067-0.

# Comprehensive Host-Based Malicious Behaviour Detection in VANETs

Hamideh Baharlouei[1*], Adetokunbo Makanju[2] and Nur Zincir-Heywood[1]

[1]Department of Computer Science, Dalhousie University, Halifax, NS, Canada.
[2]Department of Computer Science, New York Institute of Technology – Vancouver, Vancouver, BC, Canada.

*Corresponding author(s). E-mail(s): hm729953@dal.ca;
Contributing authors: amakanju@nyit.edu; zincir@cs.dal.ca;

**Abstract**

Given the safety-critical nature of Vehicular Ad Hoc Networks (VANETs), ensuring real-time security requires both prompt detection of attacks and accurate identification of the malicious nodes involved. In this context, we introduce ADVENT (Attack/Anomaly Detection in VANETs), a comprehensive system that addresses both tasks simultaneously, bridging a critical gap in prior work, which often treats these components in isolation. Through its use of Federated Learning (FL), ADVENT also addresses privacy concerns for data coming from each node, a requirement that is often overlooked in prior work. No prior work contains all of these elements together. A key strength of ADVENT lies in its lightweight feature engineering module, which reduces computational complexity to $\mathcal{O}(n)$ using a single operation strategy, unlike the multi-step and resource-intensive approaches commonly found in related systems. In addition, the federated design of ADVENT minimizes communication overhead and protects sensitive data, improving its applicability in real-world VANET environments.

**Keywords:** VANET, anomaly detection, malicious behaviour detection, federated learning, AI/ML, security, privacy

# 1 Introduction

Vehicular Ad Hoc Networks (VANETs) are a specialized type of mobile ad-hoc network commonly used in transportation systems. The dynamic nature of VANETs, characterized by constantly changing network topologies, varying vehicle speeds, and heterogeneous communication densities, introduces new security challenges and vulnerabilities [1].

In this paper, we present ADVENT (Attack/Anomaly Detection in VANETs), a real-time malicious behavior detection system composed of three primary components: (i) feature engineering, (ii) attack onset detection, and (iii) malicious node identification. ADVENT also integrates Federated Learning (FL) [2] to ensure critical data privacy while leveraging both machine learning (ML) and statistical approaches for attack detection.

The effectiveness of ML models in ADVENT relies on high-quality training data, feature extraction, and feature selection. We propose an innovative feature engineering method that captures the number of packets received by each node every second within the VANET network. To our knowledge, ADVENT is the most comprehensive detector in the literature, combining attack onset detection, malicious node detection, FL, and lightweight feature engineering in a unified and efficient framework. Prior work often addresses these aspects in isolation or only partially.

Because deploying real VANETs is expensive and labor-intensive, simulation tools are commonly used to evaluate detection strategies before real-world implementation [3]. In this study, we utilize two datasets: the publicly available VeReMi-Extension [4] and the *VANET-FourCities Dataset*, generated by the authors via simulations as described in [5, 6] (details in Section 4.2).

**In summary, this research introduces four key contributions:**

1. A novel three-tier real-time detection system, ADVENT, that simultaneously detects attack onset and identifies malicious nodes, enabling timely and precise responses.
2. Integration of FL to preserve data privacy while reducing false negative rates by leveraging distributed node information without sharing local data.
3. An efficient feature engineering methodology that minimizes computational overhead and enhances interpretability, making ADVENT suitable for resource-constrained vehicular environments.
4. Comprehensive evaluation on diverse datasets, including realistic simulation-based datasets and publicly available benchmarks, demonstrating robustness across different network settings and attack scenarios.

# 2 Related Work

Cavalcanti et al. [7] reviewed a decade of VANET research and highlighted that simulation is the primary validation tool. Simulation frameworks typically combine network and mobility simulation tools. They observed a transition from NS-2 [8] to OMNET++ [9] for network simulation, with SUMO [10] and VanetMobiSim [11] commonly used for

vehicular mobility. In our work, we employ Veins [12], which integrates OMNET++ and SUMO.

This section summarizes prior studies focusing on three aspects: statistical/ML techniques for attack detection, FL applications, and research gaps.

### 2.1 Statistical and Machine Learning Techniques

Valentini et al. [13] proposed MDASTI, a host-based attack detection system using the Mean Absolute Deviation (MAD) statistic, validated with data from Sao Paulo, Brazil.

Alheeti et al. [14] developed an intelligent IDS based on backpropagation neural networks for DDoS detection. Rani et al. [15] designed an IDS using tree-based models and ensemble learning for IoV traffic. Aloqaily et al. [16] applied deep belief networks and decision trees for IoV IDS. Gao et al. [17] employed a Random Forest-based classification for real-time network traffic detection. Injadat et al. [18] used multistage optimized ML with hyperparameter tuning for attack detection.

Li et al. [19] introduced a hybrid multitier IDS (MTH-IDS) for IoV environments, demonstrating high detection rates with low false alarms. In our previous work [5, 6], we employed MAD and tree-based ensemble ML models to detect both attack onset and malicious nodes in VANET simulations, optimized via BO-TPE (Bayesian Optimization with Tree-structured Parzen Estimator).

### 2.2 Federated Learning (FL)

Cheng et al. [20] proposed SecureBoost, a privacy-preserving FL system for vertically partitioned data using encrypted boosting trees. Ma et al. [2] applied horizontal FL for XGBoost, improving privacy and communication efficiency while handling tabular data.

### 2.3 Research Gaps in Prior Work

Yang et al. [19, 21] focus on intrusion detection in IoV but do not address Malicious Node Detection (MND). Their evaluations rely on datasets (CICIDS2017, Car-Hacking) lacking VANET mobility, limiting real-world applicability. Valentini et al. [13] addressed malicious nodes but not attack onset detection. None of the prior works integrated FL for collaborative learning while preserving privacy.

In contrast, ADVENT leverages shared experiences across vehicles without compromising privacy and incorporates SMOTE-based augmentation to address class imbalance, improving detection performance. Table 1 summarizes the comparison.

## 3 Methodology/System Overview

ADVENT aims to identify the onset of DDoS attacks, and the malicious vehicles involved. Figure 1 shows ADVENT's architecture. In the following, we present the details of this architecture.

**Table 1** Feature Comparison of VANET Intrusion Detection Methods. Legend: VD: VANET Dataset, AD: Attack Detection, DOA: Detecting Onset of Attacks, MND: Malicious Node Detection, FE: Feature Engineering, SL: Supervised Learning, FL: Federated Learning, DP: Data Privacy, AM: Aggregation Method, OT: Optimization Techniques, ID: Imbalanced Data Handling.

| **Authors** | **VD** | **AD** | **DOA** | **MND** | **FE** | **SL** | **FL** | **DP** | **AM** | **OT** | **ID** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Valentini et al. [13] | * | | | * | * | | | | | | |
| Yang et al. [19] | | * | | | * | | | | | | |
| Yang et al. [21] | | * | | | * | * | | | | * | * |
| Cheng et al. [20] | | | | | | | * | * | * | | |
| Ma et al. [2] | | | | | | | * | * | * | | |
| Liu et al. [22] | * | | | | | | | * | | | |
| Liu et al. [23] | * | | | | | | | * | | | |
| Liu et al. [24] | * | | | | | | | * | | | |
| Baharlouei et al. [6] | * | * | * | * | * | * | | | | * | |
| **ADVENT (Ours)** | * | * | * | * | * | * | * | * | * | * | * |

## 3.1 Components of ADVENT

ADVENT includes the following components.

### 3.1.1 Multi-access Edge Computing (MEC) Server

The MEC server operates as a central cloud server responsible for initializing, synchronizing, and continuously refining the attack detection model in all VANET vehicles. During the initialization phase, the MEC server acquires initial models from real-world or simulated data and provides each vehicle with the latest model upon joining the VANET and at predetermined intervals. This ensures a consistent starting point. As vehicles detect potential attacks, they report suspected malicious nodes to the MEC server, which consolidates these reports into a global list of malicious nodes. This list is then distributed to all vehicles, maintaining VANET-wide synchronization. Additionally, the MEC server aggregates model updates from vehicles, adjusts model weights, and redistributes the refined model, enhancing real-time decision-making and reducing latency.

### 3.1.2 Nodes (Vehicles)

Upon joining the VANET, each vehicle initiates data processing using ADVENT and creates its labeled data set from observed data streams. During regular operations, vehicles periodically train new attack-onset detection models, sharing them with the MEC server at predetermined intervals to maintain an updated model across the VANET. In addition, all nodes continue to engage in regular vehicular communications to exchange information. For this study, we assume the vehicles are trustworthy, and verifying the trustworthiness of nodes is beyond the scope of this work.[22, 23, 25]

### 3.1.3 Communication Links

Communication links serve as a conduit for data transfer, with vehicles actively contributing to the MEC server's repository of the latest models. This bidirectional communication ensures that both the MEC server and the vehicles maintain current and synchronized information. We anticipate leveraging cellular networks for reliable data transmission from vehicles to the MEC server. The choice of communication technology (links) depends on the specific requirements and environmental conditions. In addition, the potential inclusion of Road Side Units (RSUs) is an additional consideration to enhance communication, but ADVENT is designed to operate seamlessly with or without them. Given that ADVENT does not involve the sharing of sensitive data that could be reconstructed by attackers, conventional communication security methods are sufficient to safeguard the system.

### 3.1.4 Attack Onset Detection (AOD) Mechanism

The AOD component is an early warning system that flags suspicious activity, enabling ADVENT to identify potential threats in near real-time. It meticulously monitors network activity for aberrations, such as abrupt spikes in packet transmissions or irregular connectivity patterns, indicative of malicious intent.

### 3.1.5 Malicious Node Detection (MND) Mechanism

The MND component identifies potentially malicious vehicles within the VANET. By systematically evaluating each vehicle's behavior against predefined benchmarks, ADVENT detects anomalous actor behaviour and classifies them as malicious nodes.

### 3.1.6 Federated Learning (FL) and Adaptation

ADVENT leverages FL to enhance both attack onset detection and malicious node identification. Aggregation is performed on the MEC server, enabling dynamic adaptation of vehicles to the evolving conditions of the VANET.

### 3.1.7 Time Slicing Method

ADVENT incorporates a Time Slicing method. This technique improves the identification of malicious nodes by analyzing network behavior across discrete time intervals. As detailed in Algorithm 1, the Time Slicing algorithm systematically monitors each node within these intervals, aggregating detection results over time to achieve a more reliable classification of malicious activity.

**Algorithm 1:** Time-Slicing-Based Malicious Node Detection

```
Input: Nodes, Time Interval, Slice Duration, Threshold, Detection Function
Output: Malicious_Nodes
Initialize Suspicious_Counts and Malicious_Nodes;
for time_slice in range(0, 120, 2) do
    for each node in Nodes do
        Status = Detection_Function(node, time_slice);
        if Status == "Suspicious" then
            Suspicious_Counts[node] += 1;
            if Suspicious_Counts[node] ¿ Threshold then
                Malicious_Nodes.append(node);
return Malicious_Nodes
```

## 3.2 ADVENT's MEC Server: Model Building

ADVENT builds upon the methodology presented in [2] by leveraging FedXGBllr to update hyper-parameters and enhance privacy. The models are built locally by each vehicle, resulting in the creation of individual trees and weights. The vehicles then establish connections with the MEC server to update the tree ensemble and weight parameters for subsequent rounds. Once the server distributes the updated trees, each vehicle is able to detect the onset of an attack. Once the onset of an attack is detected, the Malicious Node Detection component is activated to identify attacking vehicles. Under normal operation, nodes actively participate in regular model building activities, contributing to the collaborative effort of maintaining an up-to-date model across the VANET. In this paper, four approaches for building and updating models are discussed. The first approach involves using one model for a static interval, initiating the model building/updating phase based on the chosen interval. The second approach involves setting a threshold based on the number of new nodes during the authentication phase. If the number of newly joined nodes exceeds this threshold, the model building/updating phase is triggered. The third approach involves building/updating models based on the number of detected attacks in the network. When a predetermined threshold of attack onsets is detected, the models are built/updated, allowing nodes to use the updated models for detecting subsequent attack onsets. The fourth approach assigns a weight to each incident, and a threshold is defined for the cumulative weight. If the combined weight exceeds the set threshold, ADVENT triggers the model building/updating process, ensuring the preparation of an up-to-date model. In the absence of ongoing attacks and if no new vehicles meet the MEC server's new arrival threshold, nodes can continue operating with the existing model, avoiding unnecessary updates.

## 3.3 ADVENT's Operation under Normal Conditions

Each vehicle, upon joining the VANET, undergoes a standardized initialization process through receiving the latest model from the MEC server, this is referred to as the 'Cold Start' phase. Using a combination of simulation-derived initialization for

the MEC server and immediate model distribution for new nodes, ADVENT guarantees a seamless integration process. The communication link facilitates the exchange of information, allowing vehicles to actively contribute to the MEC server's model repository. Thus, during regular operations, the components collaborate to monitor and secure the network. Moreover, during regular operation, each vehicle processes its data stream using ADVENT. Thus, each vehicle monitors its own observations, which in return labels its own data based on what the vehicle observes on the VANET. Vehicles also train new attack onset detection models, ensuring that both the MEC server and individual vehicles maintain up-to-date models. The attack onset detection models leverage Federated Learning [6]. Each node independently constructs its model using local data without sharing the data with the MEC server. This reinforces privacy measures within ADVENT. Regular updates contribute to the accuracy and relevance of the overall system. This synchronized operation enables ADVENT to adapt proactively to changing network dynamics. The malicious node detection mechanism is never utilized during operation under normal conditions.

### 3.4 ADVENT's Operation under Attack Conditions

When any node detects the onset of an attack during normal operations, it initiates the Malicious Node Detection component, presented in section 5.3, and figure 3. This component generates a suspicious node list based on the data it observes and shares the list with the MEC server. The server first determines the number of vehicles that report the onset of an attack. On the MEC server, multiple reports from different vehicles are aggregated, enabling the server to validate and confirm the attack. To build the final malicious list, the server considers all suspicious lists received from nodes, implementing a threshold for the number of reports for each suspicious node to prevent false positives. Subsequently, the server aggregates and distributes the malicious node list to all nodes. This collaborative approach minimizes the impact of false positives. In real-world VANET deployments, two commonly used approaches exist for managing and updating the list of malicious nodes. The first approach is a stateless approach. This includes the MEC server creating and sending the final list to the nodes without retaining it for future use. This approach is valuable considering the highly dynamic nature of VANETs. However, if an attacker remains silent for a period and then resumes the attack, both the server and nodes would not remember the previous state, thus necessitating a new detection process for each attack. The second approach is a stateful approach. This includes keeping the last malicious list and setting a timer for each malicious node in the list. If the timer expires and there is no new report for the malicious node, the server removes it from the updated list. However, if the malicious node is reported again, the server resets the timer and retains the node in the list. This approach allows ADVENT to maintain a memory of previous experiences, avoiding the need to repeat the entire detection process for a previously identified attacker.

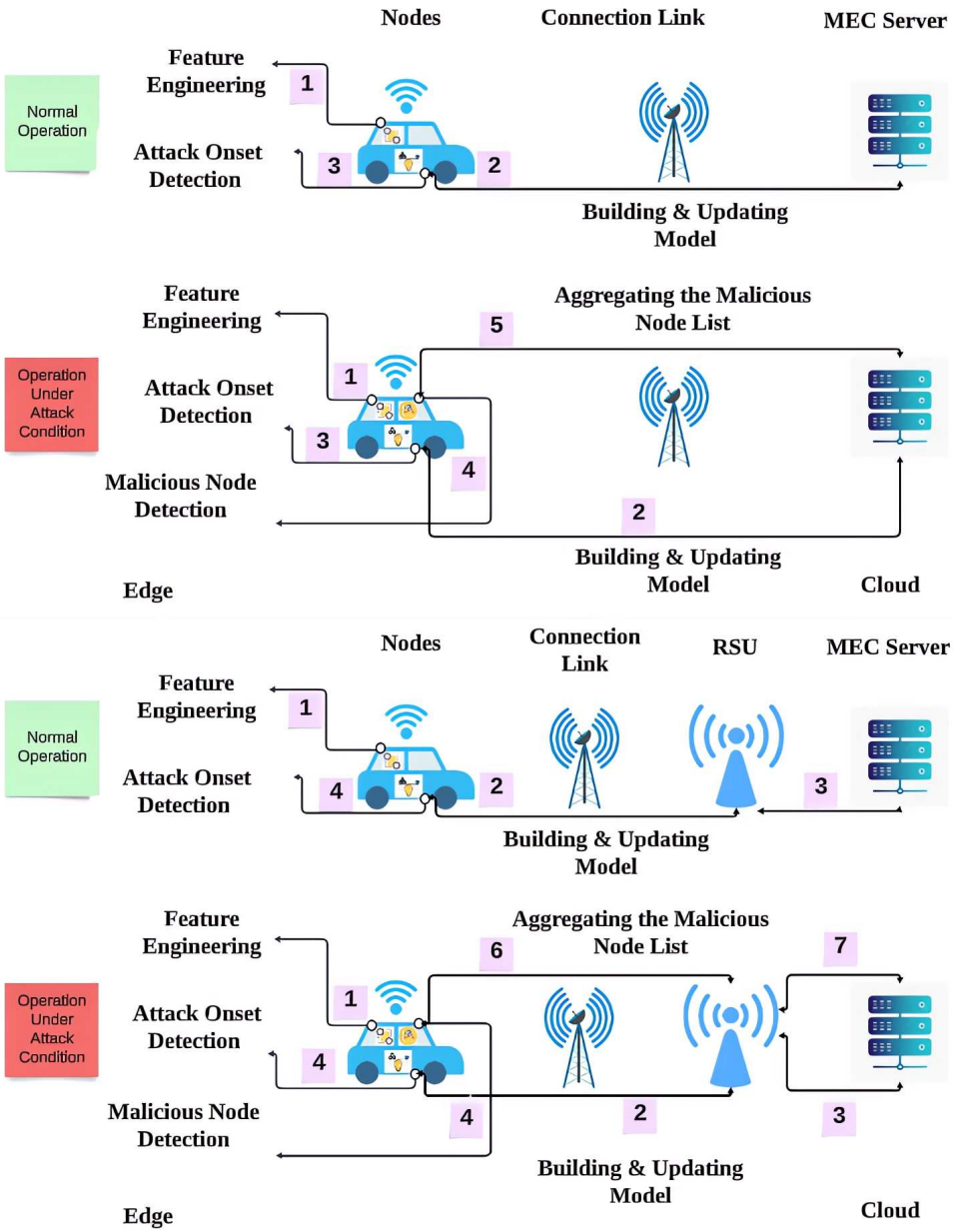



**Fig. 1** Proposed system: Attack/Anomaly Detection in Vehicular Networks (ADVENT)

# 4 Datasets Employed

In this work, we use publicly available VANET datasets as well as datasets[1] created on our own testbed. This ensures the reproducibility of our research. In the following, we detail the datasets used in our evaluations. Our dataset and the publicly available dataset complement each other to demonstrate the robustness of our approach. The public dataset covers many versions of DDoS attack in realistic traffic simulations on the road network of a single city, while our dataset covers a single type of DDoS attack in different scenarios in realistic traffic simulations on road networks of multiple cities.

## 4.1 VANET - VeReMi Extension Datasets

The VeReMi-Extension datasets are a widely recognized benchmark for misbehavior detection in VANETs. They were generated using the Framework for Misbehavior Detection (F2MD) [26], an extension of the open-source VEINS simulator [12]. The simulation scenarios in the VeReMi-Extension dataset are based in Luxembourg City, incorporating a subsection of the Luxembourg SUMO Traffic (LuST) VANET. They span an area of 1.61 $km^2$, with a maximum vehicle density of 67.4 vehicles per $km^2$. To enhance realism, sensor error models are incorporated to account for variations in position, velocity, acceleration, and heading. Due to their extensive coverage and realistic simulation parameters, VeReMi datasets have been widely used in several previous works [27–29] as summarized in Section 2.

The VeReMi-Extension datasets consist of 39 publicly available subsets, each designed to capture distinct types of misbehavior in VANETs. In addition, they include an attacker penetration rate of 30%, offering a controlled environment to evaluate misbehavior detection techniques. As an open-source resource, it can be replicated and extended using the F2MD tool, which is publicly available on GitHub. They include multiple attack scenarios, providing a solid basis for evaluating ADVENT's performance. All attack types are simulated at different times of the day to capture rush hour and low traffic scenarios. This provides two subsets for each type of misbehavior, one subset during rush hour (7h-9h) and another during low traffic time (14h-16h). These datasets are differentiated with a suffix of 0709 and 1416 respectively in our evaluations. The specific attack types present in the VeReMi-Extension datasets are as follows:

- **(D)DoS**: Attackers generate an excessive volume of messages, exceeding frequency limits defined by IEEE 1609.2 or ETSI ITS-G5, thus inducing network congestion and disrupting critical communications necessary for road safety and traffic management.
- **(D)DoS Disruptive**: Attackers replay messages received from neighboring vehicles, flooding the network with redundant data and obstructing legitimate broadcasts, making it challenging to distinguish between genuine and malicious transmissions.

[1] Will be made publicly available upon acceptance of the paper

- **(D)DoS Disruptive Sybil**: A combination of disruptive DoS and Sybil attacks in which attackers replay messages while frequently altering their identities, overwhelming the network with false messages, and complicating detection mechanisms.
- **(D)DoS Random**: Attackers transmit randomly generated messages, disrupting VANET operations. When combined with Sybil tactics, attackers also continuously alter their identities, further hindering detection efforts.
- **(D)DoS Random Sybil**: A hybrid attack that merges random flooding and Sybil tactics, where attackers send random messages while frequently changing identities, exploiting VANET security vulnerabilities and posing a severe risk to network reliability.

By evaluating ADVENT against these diverse attack types, we ensure a comprehensive assessment of its robustness and effectiveness in detecting malicious behaviour and safeguarding VANETs from a broad spectrum of threats.

## 4.2 VANET - FourCities Datasets

Following the methodology adopted by the creators of the VeReMi dataset, we used the Instant Veins5.1-i2 simulation environment [30] to generate a comprehensive set of vehicular network data. Veins integrates OMNeT++ 5.6.2 [9] and SUMO 1.8.0 [10], providing a powerful and flexible simulation framework to model realistic mobility and communication behavior in VANETs.

Our dataset incorporates real-world traffic patterns by leveraging four diverse urban scenarios: Sao Paulo (Brazil), Manhattan (USA), Valencia (Spain), and Teruel (Spain). For each city, we performed six simulation runs, varying the proportion of attacker nodes to assess the robustness of the system under different adversarial intensities. The complete simulation configuration details are summarized in Table 2.

In these simulations, the initial ten vehicles were introduced within the first ten seconds of all scenarios, resulting in a range of 10 to 30 vehicles being present in the VANET during the simulations at any given time. Throughout each simulation, we executed six attacks, each spaced 600 simulation seconds apart. Each attack had a duration of 25 simulation seconds, equivalent to 25 real-world minutes. When attackers were present in the simulations, they engaged in attacks against other vehicles for the entire 25-second period or for the duration of the attacks. At all other times, malicious nodes behaved normally within the simulation. We collected data from all vehicles during the simulation. The simulations provide data that can be used to evaluate ADVENT's performance under varying attack scenarios and geographical conditions.

Based on the details discussed in [5], the four cities are selected for their distinct topologies. This enables us to assess the impact of the VANET topology on ADVENT's performance. In addition, we tested six different attacker distributions, considering the interplay between the attacker distribution, vehicle speed (which varies based on selected road networks and maps), and VANET topology. This comprehensive evaluation allows us to analyze ADVENT's performance across diverse scenarios and deployment conditions. In total, we generated 24 datasets [5].

**Table 2** Simulation Parameters

| | |
|---|---|
| Simulation Time (Seconds) | 3600 |
| Route Scenario | Realistic urban |
| Mobility Model | SUMO |
| Propagation Model | Two-Ray Ground |
| Physical and MAC Layer | IEEE 802.11p |
| Routing Protocol | AODV |
| Transmission Rate | 6Mbps BW10MHz |
| Radio Coverage (Meters) | 400 |
| Transmission Power (dBm) | 33.8 – 60.0 |
| Transport Protocol | UDP |
| Total Vehicles | 360 |
| Total Serving Vehicles | 10 - 30 |
| Total Malicious Vehicles | 5% - 30% |
| Attack Duration Time (Seconds) | 25 |
| Number Of Attacks | 6 |

# 5 Operation of ADVENT

ADVENT's operation process starts with each node (vehicle) training an XGBoost model using its local data. Subsequently, the node sends the resulting trees, accompanied by a unique Client ID (CID), to the MEC server. The server then consolidates these trees and updates the global model weights (W) to improve efficiency while safeguarding the privacy of vehicles' data. Once an attack onset is detected, ADVENT triggers the MND component. To identify potential threats, each vehicle calculates a threshold based on the count of connections inbound from neighboring nodes, labeling those nodes that exceed this threshold as *Suspected*. The lists of *Suspected* nodes are then collectively shared with the MEC server to create an aggregated Malicious Node list which is then distributed to all nodes. To achieve this, the following operations are triggered by ADVENT.

## 5.1 Feature Engineering and Extraction Component

The proposed feature engineering approach used in ADVENT focuses on the last 10 seconds of traffic reception history. This results in 10 distinct features. These features encapsulate the packet counts received from different senders over time, supplying valuable information for training and evaluating tree-based models. In essence, if there is an anomalous change in the number of packets received from each node per second, this alteration is highly likely to be detected within the initial second of observation (as shown in results). In contrast to traditional feature engineering methods, which can introduce additional overhead and complexity, our approach is simplified and expedited. Given this, each vehicle (node) creates and stores its own data as follows:

- Let *line(*$t_i$*)* be the row created for time period $t_i$.
- Let $a$ be a randomly chosen integer, $a$ represents the number of features for the dataset. We set $a = 10$ in our work.
- Then Line($t_i$) contains *Count*($t_i$), *Count*($t_{(i-1)}$), *Count*($t_{(i-2)}$), ..., *Count*($t_{(i-(a-1))}$) as the features.

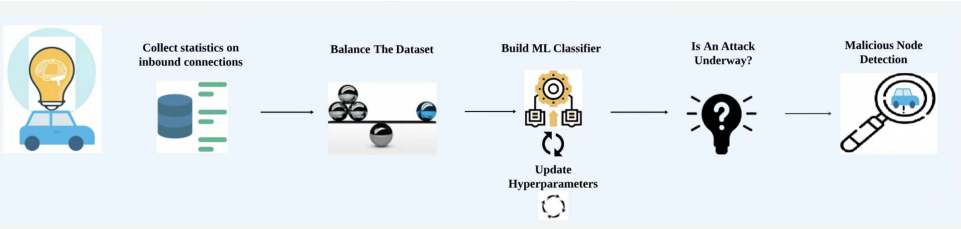



**Fig. 2** The Attack Onset Detection system

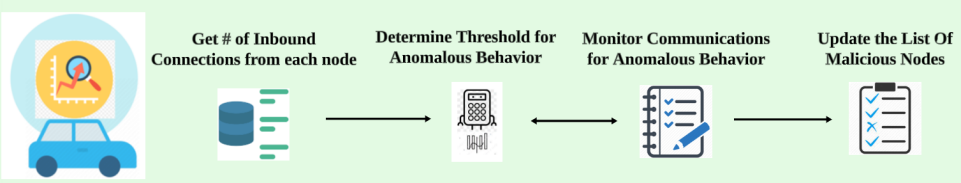



**Fig. 3** The Malicious Node Detection system

- Each line is labelled as a positive exemplar if the attack is in progress at time period $t_i$, negative otherwise.

## 5.2 AOD Component

Figure 2 presents the AOD component, in which the process begins with data acquisition, as elaborated in 5.1, where the number of inbound connections received is computed over 1-second windows during the last 10 seconds. At time t=0, all nodes train their local XGBoost tree ensemble and send the trees to the server. The server aggregates the tree ensemble and builds a Convolutional Neural Network (CNN). In particular, clients only build XGBoost models in the first round, and the aggregated tree-ensemble remains fixed after that. For federated training of the one-layer CNN, the protocol follows a standard FL algorithm, updating weights for nine rounds, finalizing the model each node uses. The evaluation results indicate that it takes around 6 seconds before nodes can start detecting the onset of an attack.

## 5.3 MND Component

Fig. 3 presents the MND component, which becomes active upon the triggering of the onset of an attack. Once the MND component gets the number of inbound connections from each node, it establishes the threshold to identify anomalous behavior using MAD [5, 6, 13]. ADVENT calculates MAD using Equation (1). It should be noted here that following Leys et al. [31], the median of each series is calculated and multiplied by 'b = 1.4826' to ensure the estimator's consistency with respect to the parameter of interest.

$$MAD = bM_i(|x_i - M_j(x_j)|) \tag{1}$$

The next critical step in MND involves establishing a rejection threshold, denoted *tr*. Values exceeding this threshold are identified as outliers or anomalies. To determine

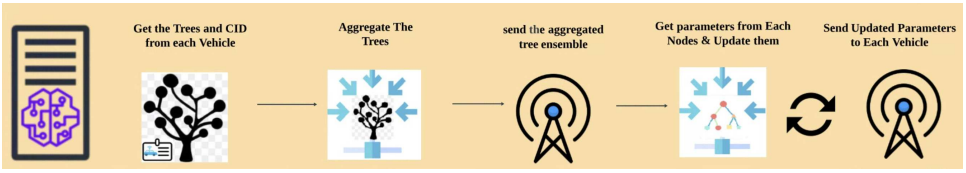



**Fig. 4** Federated Learning - Attack Onset Detection Component

'tr' and evaluate different exclusion criteria, denoted as *ce*, the value of $ce = 3.0$ is established following previous work [5]. Equation (2) is used to calculate *tr*. Once *tr* is determined, it is applied to the anomaly detection process.

$$M - ce * MAD < t_r < M + ce * MAD \tag{2}$$

where *tr* represents the rejection threshold, either negative or positive, used to identify outliers. *MAD* represents the calculated Median Absolute Deviation value. *ce* represents the exclusion criterion and *M* represents the median value of the original classified series.

Using this method, the results are heavily influenced by the chosen topology and each node's experience of encountering attackers in the VANET. Specifically, if a node is located far from an attacker or if there are obstructions that hinder packet reception, that node may not receive enough packets from the attacker to identify it as malicious. Consequently, it might not include the attacker in its suspicious list. This could result in a high false alarm and affect the overall performance of ADVENT. To address this challenge, ADVENT utilizes FL, which leverages each node's unique experience and allows vehicles to share these experiences with one another while respecting their privacy.

## 5.4 MEC Server: FL

In an FL approach, only the model updates are shared among the nodes, eliminating the need to exchange raw data to preserve the privacy of data. This decentralized approach offers multiple advantages for VANETs, including enhanced security, reduced communication overhead, and increased adaptability [32].

### FL for AOD

In this phase, nodes share their models (trees) according to FL principles. ADVENT employs FedXGBllr [2] for two key reasons: (i) the absence of gradient sharing reduces concerns related to communication security and message exchange; and (ii) minimizing per-node communication frequency further strengthens privacy protection. Figure 4 illustrates how the MEC server receives the trees and CIDs from the nodes. The process commences with the MEC server's aggregation and broadcast of the trees and associated parameters.

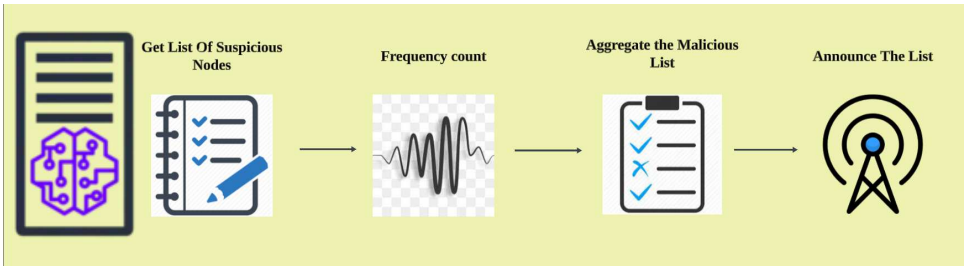



**Fig. 5** Federated Learning - Malicious Node Detection Component

### 5.4.1 In the Initial Round *(t = 0)*:

Upon receiving the trained XGBoost tree ensembles and CIDs from nodes that have trained their local XGBoost trees using available data, the MEC server aggregates these tree ensembles as follows:

- Sorting and broadcasting the aggregated XGBoost tree ensemble along with corresponding CIDs to all clients.
- Initializing *w0* for the one-layer 1D CNN.
- Each node receives the broadcasted aggregated XGBoost tree ensemble and proceeds to evaluate these ensembles with their local data. This evaluation yields prediction outcomes generated by all trees for each node.

### 5.4.2 In Subsequent Rounds (R) *(t = 1 to R)*:

- The server designates the nodes who will participate.
- Global model parameters, represented as *wt*, are broadcasted to all nodes by the server.
- In parallel, each node updates its local model parameters to *wk(t+1)* by performing a Client Update operation, utilizing the received parameters.
- The server aggregates the updated models from all nodes to calculate the new global model parameters, *wk(t+1)*.

This iterative process also offers the advantage of faster training with local data for each node [21].

### FL for MND

ADVENT aims to combine the collective experience of all nodes, allowing vehicles to view the network from the perspective of other vehicles, Fig. 5. Once the MND component has been executed by a node, a list of suspicious actors is generated and sent to the MEC server. Then, the MEC server proceeds as follows:

- Count the frequency of each reported vehicle (node) *i*, denoted as *Fi*.
- If the frequency of a reported vehicle, Fi, is higher than the predefined threshold "TH" set for the network, then include that vehicle in the list of malicious nodes.
- Transmit the malicious node list to all vehicles in the network.

This leads to several achievements: (i) a notable reduction in false alarms, coupled with an increase in detection; (ii) enabling nodes to predict attacks before the influx of inbound connections, effectively avoiding interactions with malicious nodes; and (iii) minimizing the impact of differing geographical patterns and varying numbers of attackers in the VANET.

# 6 Results

This section presents the performance of ADVENT on all datasets employed in this work.

## 6.1 Evaluation Metrics

Given the above operation of ADVENT, results are presented using standard evaluation metrics, namely Detection Rate (DR), False Positive Rate (FPR), False Negative Rate (FNR), and F1-score. DR measures the proportion of attack behaviours correctly identified, FPR measures the proportion of benign behaviours detected as attacks, FNR measures the missed attack instances, and the F1-score is a harmonic mean of precision (positive predictive value) and recall (sensitivity). To calculate these metrics, we used the following equations, where TP denotes the number of correctly identified positive instances, TN denotes the number of correctly identified negative instances, FN indicates the number of instances mistakenly classified as negative, and FP signifies the number of instances mistakenly classified as positive. These metrics are used to evaluate ADVENT's performance in the context of AOD and MND.

$$DR = TP/(TP + FN) \tag{3}$$

$$FPR = FP/(FP + TN) \tag{4}$$

$$FNR = FN/(TP + FN) \tag{5}$$

$$Recall(Rec) = TP/(TP + FN) \tag{6}$$

$$Precision(Prec) = TP/(TP + FP) \tag{7}$$

$$F1\text{-}score = 2 * (Prec * Rec)/(Prec + Rec) \tag{8}$$

## 6.2 AOD Results

In ADVENT, once each vehicle has trained its local model and shared the trees with the server, the server aggregates these trees and distributes them to all vehicles. Subsequently, only the model weights are updated. To maintain consistency with the state of the art [2], in our simulations, we have set the number of rounds at 10 and the epochs at 100, with a batch size of 64. After training, the remaining data is used for testing purposes for each vehicle (node).

### 6.2.1 Results on VereMi-Extention Data

Table 3 shows the results of ADVENT on the VereMi-Extention datasets. These results indicate that ADVENT integrating FL is able to achieve high performance without centralized data collection. Furthermore, the findings underscore ADVENT's reliability in identifying the onset of all attacks under evaluation, F1-scores ranging between 96% to almost 98%. Table 3 also presents the FPR and FNR for different attacks. These indicate that ADVENT exhibits an average FNR of approximately 6%. This is expected because ADVENT trains models on local data rather than on all the data utilized by all nodes.

The final metric that we examined is the attack detection speed, which highlights ADVENT's effectiveness in identifying threats in real-time. As shown in Figure 6, an average of 97% of attacks are detected within the first second after the onset. This demonstrates ADVENT's ability to quickly detect the onset of an attack.

**Table 3** VeReMi-Extension Datasets: AOD Results

| Attack-Time Window | FNR | FPR | DR | F1-Score |
|---|---|---|---|---|
| DoS-0709 | 5.10 | 1.39 | 94.90 | 97.72 |
| DoS-1416 | 6.98 | 0.76 | 92.98 | 96.29 |
| DoSDistruptive-0709 | 5.15 | 1.48 | 94.80 | 97.64 |
| DoSDistruptive-1416 | 6.58 | 0.78 | 93.43 | 96.28 |
| DoSSybil-0709 | 5.46 | 1.63 | 94.51 | 97.44 |
| DoSSybil-1416 | 6.98 | 0.95 | 92.97 | 95.99 |
| DoSRandom-0709 | 5.36 | 1.46 | 94.65 | 97.65 |
| DoSRandom-1416 | 6.83 | 0.82 | 93.06 | 96.18 |
| DoSRandomSybil-0709 | 5.19 | 1.62 | 94.79 | 97.51 |
| DoSRandomSybil-1416 | 6.96 | 0.88 | 92.99 | 96.06 |

### 6.2.2 Results on VANET-FourCities Data

Table 4 presents the results on four city maps. The findings demonstrate that ADVENT achieves reliable and consistent performance on all datasets and scenarios. Using SMOTE at each node [33], ADVENT effectively compensates for limited training samples. Additionally, it detects attack onsets in real-time while preserving privacy. In general, ADVENT delivers strong performance, achieving an average DR of 99.6%, FNR of 0.4%, FPR of 0%, and F1-Score of 99.7%.

Again, the final metric we explore is how quickly attacks are detected to provide insight into the effectiveness of ADVENT. Figure 7 shows the results, indicating an average of 99.6% of all attacks detected within the first second of commencement.

## 6.3 MND Results

To ensure the efficiency of ADVENT across different network conditions, we define two key parameters for the MND component: (i) Consensus Threshold, and (ii) Adaptive Time Slicing. Consensus Threshold (CT) determines how many vehicles must report a node as suspicious before it is classified as malicious. The CT is set on the MEC

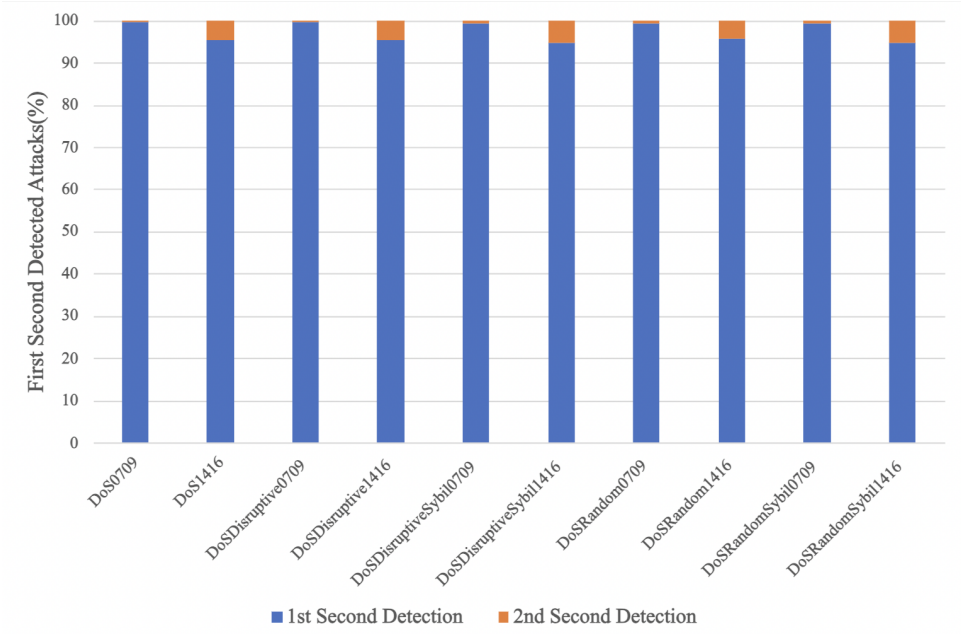



**Fig. 6** VeReMi-Extension: First Sec. Attack Detection Ratio

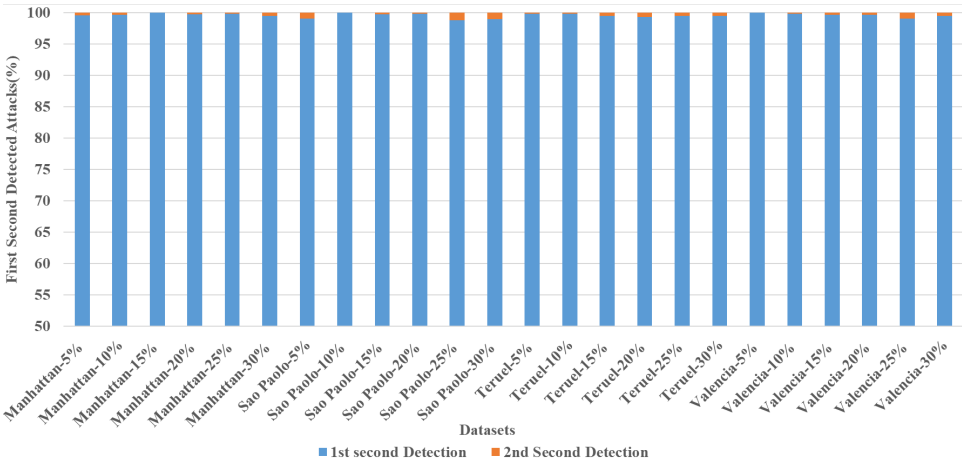



**Fig. 7** VANET-FourCities: First Sec. Attack Detection Ratio

server, and in this work it is set to 2 following [6]. After detecting the onset of an attack, each node monitors the number of inbound connections from its neighbors over a 2-minute observation window. However, processing this entire interval as a single block can lead to FNR, as normal nodes that spend an extended time with a specific node might be incorrectly flagged. To address this, we apply an Adaptive Time Slicing (ATS) mechanism, dividing a time interval (such as 2-minute) into smaller time slices

**Table 4** VANET-FourCities Datasets: AOD Results

| Dataset-Attacker% | FNR | FPR | DR | F1-Score |
|---|---|---|---|---|
| São Paulo-5 | 0.90 | 0.01 | 99.10 | 98.40 |
| São Paulo-10 | 0.00 | 0.00 | 100.00 | 99.74 |
| São Paulo-15 | 0.23 | 0.00 | 99.77 | 99.51 |
| São Paulo-20 | 0.15 | 0.00 | 99.85 | 99.81 |
| São Paulo-25 | 1.18 | 0.00 | 98.82 | 99.41 |
| São Paulo-30 | 1.03 | 0.00 | 98.97 | 99.48 |
| Manhattan-5 | 0.29 | 0.00 | 99.71 | 99.86 |
| Manhattan-10 | 0.29 | 0.00 | 99.71 | 99.86 |
| Manhattan-15 | 0.00 | 0.00 | 100.00 | 99.93 |
| Manhattan-20 | 0.13 | 0.00 | 99.87 | 98.70 |
| Manhattan-25 | 0.13 | 0.00 | 99.87 | 99.94 |
| Manhattan-30 | 0.49 | 0.00 | 99.51 | 99.75 |
| Teruel-5 | 0.15 | 0.00 | 99.85 | 99.92 |
| Teruel-10 | 0.14 | 0.00 | 99.86 | 99.93 |
| Teruel-15 | 0.51 | 0.00 | 99.49 | 99.74 |
| Teruel-20 | 0.62 | 0.00 | 99.38 | 99.69 |
| Teruel-25 | 0.47 | 0.00 | 99.53 | 99.76 |
| Teruel-30 | 0.44 | 0.00 | 99.56 | 99.78 |
| Valencia-5 | 0.00 | 0.00 | 100.00 | 100.00 |
| Valencia-10 | 0.74 | 0.00 | 99.26 | 99.63 |
| Valencia-15 | 0.28 | 0.00 | 99.72 | 99.86 |
| Valencia-20 | 0.27 | 0.00 | 99.73 | 99.86 |
| Valencia-25 | 0.91 | 0.00 | 99.09 | 99.54 |
| Valencia-30 | 0.39 | 0.00 | 99.61 | 99.80 |

(e.g., 2-second slices). Then, the MND method is executed within each slice to identify potentially malicious nodes. The reported lists of suspicious nodes across all slices of each node are then aggregated, and if it surpasses a predefined Detection Threshold (DT), it is classified as malicious and reported to the MEC server [27]. In doing so, we aim to reduce false negatives while maintaining high accuracy in identifying malicious nodes.

### 6.3.1 Results on VeReMi-Extension Data:

To evaluate ADVENT using the VeReMi-Extension datasets, we set the CT to 2, the ATS to 2 seconds and the DT to 7 following our previous work [27], in which we highlighted the positive impact of selecting the appropriate CT on the overall performance of the detection system. The results show that ADVENT maintains high detection performance in various attack scenarios, with an average DR of 95.8%, F1-Score of 96.3%, FNR of 4.7%, and FPR of 1.3%. Among the attack types evaluated, DoS0709 and DoSRandom07 achieved the highest detection performance, with DR values of 98.7% and 98.4%, respectively. These results show that ADVENT effectively identifies these DoS attacks with minimal false negatives. In contrast, DoSDisruptiveSybil14

and DoSRandomSybil14 have lower detection rates at 91.9% and 91.5%, respectively. This seems to indicate that Sybil-based attacks remain more difficult to detect. However, even in these cases, the system maintained an overall F1-Score more than 95%, demonstrating its robustness.

**Table 5** VeReMi-Extension Datasets: MND Results

| Attack-Time Window | DR | F1-Score | FNR | FPR |
|---|---|---|---|---|
| DoS-0709 | 98.69 | 96.59 | 1.31 | 2.42 |
| DoS-1416 | 93.10 | 96.03 | 6.90 | 0.34 |
| DoSDisruptive-0709 | 98.44 | 95.70 | 1.56 | 3.13 |
| DoSDisruptive-1416 | 93.88 | 96.45 | 6.11 | 0.34 |
| DoSDisruptiveSybil-0709 | 97.38 | 96.20 | 2.62 | 2.18 |
| DoSDisruptiveSybil-1416 | 91.92 | 95.30 | 8.09 | 0.42 |
| DoSRandom-0709 | 98.44 | 97.84 | 1.56 | 1.19 |
| DoSRandom-1416 | 93.49 | 95.85 | 6.51 | 0.68 |
| DoSRandomSybil-0709 | 98.38 | 96.20 | 2.62 | 2.18 |
| DoSRandomSybil-1416 | 91.52 | 95.28 | 8.48 | 0.25 |

### 6.3.2 Results on VANET-FourCities Dataset:

In this section, we present the results of MND on the FourCitites datasets, where the CT, ATS, and DT are set to 2 for Manhattan and São Paulo, 10 for Valencia, and 20 for Teruel. As seen in these results, the best performance is observed for Teruel, with an average F1-score of 99.7%, the highest among all cities. Manhattan follows closely with an F1-score of 98.8%, while Valencia and São Paulo achieve almost 97.9% and 97.6%, respectively. Furthermore, Teruel has the lowest FNR of 0.6%, compared to 1.5% for Manhattan, 1.9% for São Paulo, and the highest 2.6% for Valencia. The FPR remains extremely low for all cities, with Teruel and Valencia maintaining an FPR of 0%, Manhattan at 0.05%, and São Paulo at 0.1%. These results demonstrate that ADVENT performs robustly on different urban environments. Table 6 presents the results for all cities and all datasets. Overall, the task of detecting the list of attackers becomes more challenging as the number of attackers increases to 25% and 30%. Nevertheless, ADVENT remains efficient in detecting malicious nodes via the use of FL and Time Slicing techniques.

An important factor in the MND of ADVENT is the ATS factor. The VeReMi-Extension Dataset exhibited significant improvements with the ATS. However, VeReMi datasets use only one topology. When four different topologies are used, as is the case in the VANET-FourCities datasets, it becomes clear that tuning of ADVENT is necessary. For example, we set the ATS to 2 seconds for Luxembourg in the VeReMi-Extension dataset. However, we discovered that higher values were needed for the VANET-FourCities datasets. Specifically, we implemented an ATS of

**Table 6** VANET-FourCities Datasets: MND Results

| Dataset-Attacker% | DR % | F1-Score % | FNR % | FPR % |
|---|---|---|---|---|
| Manhattan-5 | 100.00 | 100.00 | 0.00 | 0.00 |
| Manhattan-10 | 100.00 | 100.00 | 0.00 | 0.00 |
| Manhattan-15 | 100.00 | 98.25 | 0.00 | 0.00 |
| Manhattan-20 | 100.00 | 100.00 | 0.00 | 0.00 |
| Manhattan-25 | 100.00 | 100.00 | 0.00 | 0.00 |
| Manhattan-30 | 91.30 | 94.38 | 8.70 | 0.32 |
| São Paulo-5 | 94.12 | 96.97 | 5.88 | 0.00 |
| São Paulo-10 | 96.97 | 97.44 | 0.00 | 0.00 |
| São Paulo-15 | 96.43 | 98.18 | 3.57 | 0.00 |
| São Paulo-20 | 97.14 | 98.55 | 2.44 | 0.00 |
| São Paulo-25 | 98.76 | 97.56 | 0.00 | 0.00 |
| São Paulo-30 | 97.92 | 96.91 | 0.00 | 0.65 |
| Teruel-5 | 100.00 | 100.00 | 0.00 | 0.00 |
| Teruel-10 | 100.00 | 100.00 | 0.00 | 0.00 |
| Teruel-15 | 100.00 | 100.00 | 0.00 | 0.00 |
| Teruel-20 | 100.00 | 100.00 | 0.00 | 0.00 |
| Teruel-25 | 97.92 | 98.95 | 2.08 | 0.00 |
| Teruel-30 | 98.27 | 99.13 | 1.72 | 0.00 |
| Valencia-5 | 100.00 | 100.00 | 0.00 | 0.00 |
| Valencia-10 | 95.24 | 97.67 | 4.76 | 0.00 |
| Valencia-15 | 100.00 | 100.00 | 0.00 | 0.00 |
| Valencia-20 | 97.14 | 98.55 | 2.86 | 0.00 |
| Valencia-25 | 97.62 | 96.47 | 2.38 | 0.00 |
| Valencia-30 | 94.34 | 95.24 | 5.66 | 0.00 |

10 seconds for Valencia and 20 for Teruel. While these longer ATS produced satisfactory results for these cities, Manhattan and Sao Paulo achieved better performances with a 2-second ATS. Table 7 summarizes the characteristics we examined that influence the ATS settings and highlights the necessary adjustments. These differences can be attributed to the distinct characteristics of the road topology, vehicle density, and network connectivity characteristics observed in different cities.

1. **Sparse vs. Radial Road Network**
   Valencia has a radial-grid hybrid topology, which means that vehicles are more concentrated on major roads, with fewer alternative routes. This can cause localized congestion where attacks have a stronger impact. In contrast, Teruel has a sparse, irregular road network, which means that there are fewer vehicles and fewer communication links. This reduces the ability of ADVENT to detect anomalies in small time slices.

2. **Low Connectivity in Small Time Slices**
   In small time slices (e.g., 2s), Teruel's sparse topology results in lower message exchange, making it difficult to observe anomalies within a short window. Valencia's radial pattern means that many vehicles send packets in bursts at specific intersections, which may appear as normal fluctuations rather than anomalies in a short time window.
3. **Impact of Large Time Slices**
   When time slicing is increased (e.g., to 20s), the effects of low packet transmission density (Teruel) and bursty traffic at intersections (Valencia) are averaged over time, allowing ADVENT to recognize long-term patterns. The MAD method used detects attackers by considering deviations over longer intervals rather than reacting to momentary bursts or gaps.

Increasing the ATS parameter to 20 seconds allows ADVENT to accumulate enough data to distinguish attackers from normal behavior using MAD. In addition, topologies that have more balanced road structures, denser vehicle networks, or more stable connectivity can also result in higher performance for ADVENT.

**Table 7** Summary of Characteristics Across Different Cities

| **Characteristic** | **MHT** | **SP** | **VAL** | **TER** | **LUX** |
|---|---|---|---|---|---|
| Road Network Layout | Grid-like | Mixed | Radial-grid hybrid | Sparse | Irregular-suburban |
| Intersection Density | High | Variable | Moderate | Low | Variable |
| Vehicle Density | High | High (arterial) | High (main) | Low | Mixed |
| Packet Transmission Patterns | Regular | Irregular | Bursty | Sparse | Mixed |
| Connectivity Stability | High | Medium | Unstable (small slices) | Very low (small slices) | Medium |
| Impact of Small Time Slices (2s) | Stable | Adaptive | Poor | Poor | Stable |
| Impact of Large Time Slices (120s) | Stable | Stable | Improved | Improved | Stable |
| Poor Performance Reasons (2s) | None | None | Misclassification of spikes | Limited exchanges | None |
| Recommended Threshold Calibration | Moderate | Adaptive | Higher thresholds | Higher thresholds | Adaptive |

MHT = Manhattan, SP = São Paulo, VAL = Valencia, TER = Teruel, LUX = Luxembourg

The slight drop in F1-score and DR in higher-density scenarios such as São Paulo-30 and Valencia-30 is expected due to increased traffic complexity and congestion, yet ADVENT still performs robustly with minimal FPR and FNR.

**Table 8** Feature Engineering Complexity Comparison

| Paper Authors | FE: $\mathcal{O}$(Time) | FEO: #Operations |
|---|---|---|
| Valentini et al.[13] | $\mathcal{O}(n)$ | 1 |
| Yang et al. [19] | $\mathcal{O}(m \log m)$ | 2 |
| Yang et al. [21] | $\mathcal{O}(n^3)$ | 3 |
| Cheng et al. [20] | Missing | Missing |
| Ma et al. [2] | Missing | Missing |
| Liu et al. [22] | Missing | Missing |
| Liu et al. [23] | Missing | Missing |
| Liu et al. [24] | Missing | Missing |
| Baharlouei et al.[6] | Missing | Missing |
| ADVENT | $\mathcal{O}(n)$ | 1 |

### 6.4 Feature Engineering Complexity

In this section, we compare the computational complexity of ADVENT's feature engineering (FE) component with that of the state-of-the-art as discussed in Section 2. Valentini et al. [13] employ a lightweight feature extraction (FE) approach with a time complexity of $\mathcal{O}(n)$, making it well-suited for real-time environments. While ADVENT's FE component operates with comparable complexity, the ADVENT system as a whole provides a more comprehensive solution. In contrast, Yang et al.[19, 21] utilize a significantly more elaborate FE pipeline involving multiple operations such as Information Gain, Fast Correlation-Based Filter, and Kernel Principal Component Analysis, resulting in a cumulative complexity of up to $\mathcal{O}(n^3)$. In comparison, ADVENT adopts a streamlined and efficient FE mechanism with a single-pass operation and linear complexity $\mathcal{O}(n)$. Thus, ADVENT not only supports rapid data preprocessing but also enhances the responsiveness and feasibility of detecting both the onset of attacks and the identification of malicious nodes in real-time VANET environments.

## 7 Conclusions and Future Works

To enhance the security of VANETs, we introduced ADVENT, an innovative and comprehensive real-time malicious behavior detection system. ADVENT unifies the detection of attack onsets and identification of malicious nodes within a single framework, leveraging Federated Learning (FL) for privacy-preserving distributed learning and the statistical MAD technique for efficient feature extraction. Unlike traditional systems that rely on multi-stage preprocessing with higher computational and memory demands, ADVENT employs a lightweight, single-step transformation with linear time complexity, $\mathcal{O}(n)$, making it well-suited for real-time and resource-constrained environments.

Our experimental results demonstrate that ADVENT delivers consistently high performance. Using MAD-based features, the system achieves an average Detection Rate of 99.6% on the VANET-FourCities dataset and 97.2% on the VeReMi-Extension

dataset. Across all scenarios, attack onsets are detected within one second, and malicious node identification is achieved with a minimal average False Negative Rate of 0.41

By combining efficiency, scalability, real-time responsiveness, and privacy-preserving design, ADVENT addresses critical limitations in prior work. Future research will focus on extending ADVENT's applicability to other types of ad hoc wireless networks, such as UAV-based systems, and further evaluating its performance under a broader range of attack scenarios.

# References


[1] Garip, M.T., Gursoy, M.E., Reiher, P., Gerla, M.: Congestion attacks to autonomous cars using vehicular botnets. Proceedings 2015 Workshop on Security of Emerging Networking Technologies (2015) https://doi.org/10.14722/sent.2015.23001

[2] Ma, C., Qiu, X., Beutel, D., Lane, N.: Gradient-less federated gradient boosting tree with learnable learning rates. In: Proceedings of the 3rd Workshop on Machine Learning and Systems. EuroMLSys '23, pp. 56–63. Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3578356.3592579 . https://doi.org/10.1145/3578356.3592579

[3] Martinez, F.J., Toh, C.K., Cano, J.-C., Calafate, C.T., Manzoni, P.: A survey and comparative study of simulators for vehicular ad hoc networks (vanets). Wireless Communications and Mobile Computing **11**(7), 813–828 (2011)

[4] Kamel, J., Wolf, M., Hei, R.W., Kaiser, A., Urien, P., Kargl, F.: Veremi extension: A dataset for comparable evaluation of misbehavior detection in vanets. In: 2020 IEEE International Conference on Communications (ICC), pp. 1–6 (2020). https://doi.org/10.1109/ICC40277.2020.9149132

[5] Baharlouei, H., Makanju, A., Zincir-Heywood, N.: Exploring realistic vanet simulations for anomaly detection of ddos attacks. In: 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), pp. 1–7 (2022). https://doi.org/10.1109/VTC2022-Spring54318.2022.9860624

[6] Baharlouei, H., Makanju, A., Zincir-Heywood, N.: Exploring real-time malicious behaviour detection in VANETs. In: Proceedings of the Int'l ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet '23), pp. 1–8. Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3616392.3623412 . https://doi.org/10.1145/3616392.3623412

[7] Cavalcanti, E.R., Souza, J.A.R., Spohn, M.A., Gomes, R.C.d.M., Costa, A.F.B.F.d.: Vanets' research over the past decade: Overview, credibility, and trends. SIGCOMM Comput. Commun. Rev. **48**(2), 31–39 (2018) https://doi.org/

10.1145/3213232.3213237

[8] The ns-2 Project: The Ns-2 Network Simulator. The Network Simulator ns-2, (1995). The Network Simulator ns-2. https://www.isi.edu/nsnam/ns/

[9] Varga, A.: In: Wehrle, K., Güneş, M., Gross, J. (eds.) OMNeT++, pp. 35–59. Springer, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12331-33 . https://doi.org/10.1007/978-3-642-12331-33

[10] Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E.: Microscopic traffic simulation using sumo. In: The 21st IEEE International Conference on Intelligent Transportation Systems. IEEE, ??? (2018). https://elib.dlr.de/124092/

[11] Härri, J., Filali, F., Bonnet, C., Fiore, M.: Vanetmobisim: generating realistic mobility patterns for vanets. In: Proceedings of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks, pp. 96–97 (2006). https://doi.org/10.1145/1161064.1161084 . ACM. https://doi.org/10.1145/1161064.1161084

[12] ", I.V.V.M.: http://veins.car2x.org/ (accessed:Dec. 2021). http://veins.car2x.org/

[13] Pastori Valentini, E., Ipolito Meneguette, R., Alsuhaim, A.: An attacks detection mechanism for intelligent transport system. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 2453–2461 (2020). https://doi.org/10.1109/BigData50022.2020.9378309

[14] Yang, L., Moubayed, A., Hamieh, I., Shami, A.: Tree-based intelligent intrusion detection system in internet of vehicles. In: 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1–6. IEEE Press, ??? (2019). https://doi.org/10.1109/GLOBECOM38437.2019.9013892 . https://doi.org/10.1109/GLOBECOM38437.2019.9013892

[15] Rani, P., Sharma, R.: Intelligent transportation system for internet of vehicles based vehicular networks for smart cities. Computers and Electrical Engineering **105**, 108543 (2023) https://doi.org/10.1016/j.compeleceng.2022.108543

[16] Abd, N., Ali Alheeti, K.M., Salah, S., Al-Rawi, S.: Intelligent intrusion detection system in internal communication systems for driverless cars. Webology **17**, 376 (2020) https://doi.org/10.14704/WEB/V17I2/WEB17039

[17] Gao, Y., Wu, H., Song, B., Jin, Y., Luo, X., Zeng, X.: A distributed network intrusion detection system for distributed denial of service attacks in vehicular ad hoc network. IEEE Access **7**, 154560–154571 (2019) https://doi.org/10.1109/ACCESS.2019.2948382

[18] Injadat, M., Moubayed, A., Nassif, A.B., Shami, A.: Multi-stage optimized

machine learning framework for network intrusion detection. IEEE Transactions on Network and Service Management **18**(2), 1803–1816 (2021) https://doi.org/10.1109/tnsm.2020.3014929

[19] Yang, L., Moubayed, A., Hamieh, I., Shami, A.: Tree-based intelligent intrusion detection system in internet of vehicles. In: 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2019). https://doi.org/10.1109/GLOBECOM38437.2019.9013892

[20] Cheng, K., Fan, T., Jin, Y., Liu, Y., Chen, T., Papadopoulos, D., Yang, Q.: Secureboost: A lossless federated learning framework. IEEE Intelligent Systems **36**(06), 87–98 (2021) https://doi.org/10.1109/MIS.2021.3082561

[21] Yang, L., Moubayed, A., Shami, A.: Mth-ids: A multitiered hybrid intrusion detection system for internet of vehicles. IEEE Internet of Things Journal **9**(1), 616–632 (2022) https://doi.org/10.1109/JIOT.2021.3084796

[22] Liu, Z., Weng, J., Ma, J., Guo, J., Feng, B., Jiang, Z., Wei, K.: Tcemd: A trust cascading-based emergency message dissemination model in vanets. IEEE Internet of Things Journal **7**(5), 4028–4048 (2020) https://doi.org/10.1109/JIOT.2019.2957520

[23] Liu, Z., Weng, J., Guo, J., Ma, J., Huang, F., Sun, H., Cheng, Y.: Pptm: A privacy-preserving trust management scheme for emergency message dissemination in space–air–ground-integrated vehicular networks. IEEE Internet of Things Journal **9**(8), 5943–5956 (2022) https://doi.org/10.1109/JIOT.2021.3060751

[24] Liu, Z., Wan, L., Guo, J., Huang, F., Feng, X., Wang, L., Ma, J.: Ppru: A privacy-preserving reputation updating scheme for cloud-assisted vehicular networks. IEEE Transactions on Vehicular Technology, 1–16 (2023) https://doi.org/10.1109/TVT.2023.3340723

[25] Zhang, X., Guo, Y., Xu, X., Wu, Y., Yu, Q.: A trust model for edge-driven vehicular ad hoc networks using fuzzy logic. IEEE Transactions on Intelligent Transportation Systems (2023) https://doi.org/10.1109/TITS.2023.3305342

[26] Kamel, J., Ansari, M.R., Petit, J., Kaiser, A., Ben Jemaa, I., Urien, P.: Simulation framework for misbehavior detection in vehicular networks. IEEE Transactions on Vehicular Technology (2020)

[27] Baharlouei, H., Makanju, A., Zincir-Heywood, N.: Evaluating the robustness of advent on the veremi-extension dataset. In: Proceedings of the 20th International Conference on Network and Service Management (CNSM 2024), Halifax, NS, Canada, pp. 1–7 (2024). IFIP/IEEE

[28] Ahsan, S.I., Legg, P., Alam, S.M.I.: A Stacked Ensemble Learning IDS Model for Software-Defined VANET. In: arXiv Preprint arXiv:2312.04956 (2023).

https://arxiv.org/abs/2312.04956

[29] Slama, O., Tarhouni, M., Zidi, S., Alaya, B.: VeReMi Extension: Dataset for Misbehaviors in VANETs. In: Mendeley Data (2023). https://doi.org/10.17632/k62n4z9gdz.1 . https://data.mendeley.com/datasets/k62n4z9gdz

[30] Sommer, C., German, R., Dressler, F.: Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. IEEE Transactions on Mobile Computing (TMC) **10**(1), 3–15 (2011) https://doi.org/10.1109/TMC.2010.133

[31] Leys, C., Ley, C., Klein, O., Bernard, P., Licata, L.: Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. Journal of experimental social psychology **49**(4), 764–766 (2013)

[32] Moshawrab, M., Adda, M., Bouzouane, A., Ibrahim, H., Raad, A.: Reviewing federated learning aggregation algorithms; strategies, contributions, limitations and future perspectives. Electronics **12**(10) (2023)

[33] Baharlouei, H., Makanju, A., Zincir-Heywood, N.: ADVENT: Attack/Anomaly Detection in VANETs (2024)