




Intrusion detection in vehicular networks using machine learning

Heitor Tonel Ventura  [Universidade Tecnológica Federal do Paraná | heitorventura@alunos.utfpr.edu.br]

Raian de Almeida Moretti  [Universidade Tecnológica Federal do Paraná | raianmoretti@alunos.utfpr.edu.br]

Ana Cristina Barreiras Kochem Vendramin  [Universidade Tecnológica Federal do Paraná | criskochem@utfpr.edu.br]

Daniel Fernando Pigatto   [Universidade Tecnológica Federal do Paraná | pigatto@utfpr.edu.br]

 Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Av. Sete de Setembro, 3165, Rebouças, Curitiba, PR, 80230-901, Brazil.

Received: 30 September 2024 • **Accepted:** 16 March 2025 • **Published:** 16 May 2025

Abstract Vehicular networks and intelligent transport systems play a critical role in modern urban mobility. In order to improve urban transportation in smart cities, vehicles and fixed stations exchange information about traffic, road conditions, and accidents, allowing better decision-making and ensuring greater safety for the population. However, to provide security, a vehicular network must be resilient to attacks. Anomaly detection models are a potential solution to the reduced effectiveness of signature-based intrusion detection systems, which struggle to detect new attacks due to the absence of previous signatures. Leveraging artificial intelligence in intrusion detection systems becomes relevant, as it allows learning from a vast amount of data. However, many models proposed for anomaly detection based on machine learning lack validation and application in vehicular networks, thus lacking evidence of promising results in these specific contexts. Therefore, this work aims to address this gap by comparing two models used in anomaly detection in the context of vehicular networks: the CNN-LSTM model that has already been applied in the area of vehicular networks and the TranAD model that needed to be adapted for this type of network. The results demonstrate that the CNN-LSTM model provides superior performance, presenting an F1 of 0.9585 against 0.8839 of TranAD in the scenario in which both models obtained the best result.

Keywords: Vehicular Networks; Intelligent Transport System; Urban Mobility; Intrusion Detection Systems; Machine Learning.

1 Introduction

Urban mobility is a critical component of modern city planning, encompassing the strategies and technologies designed to manage transportation efficiently and sustainably [Borchers *et al.*, 2024; Gopal *et al.*, 2023]. Smart transportation systems are essential to the development of smart cities [Aldhanhani *et al.*, 2024; Kumar and Mikkili, 2024]. Recent advancements in smart transportation systems, such as autonomous vehicles, electric scooters, and integrated public transit solutions, are reshaping urban mobility by providing more flexible and eco-friendly options. These innovations aim to reduce traffic congestion, lower carbon emissions, and enhance overall transit efficiency [Kumar and Mikkili, 2024; Aldhanhani *et al.*, 2024; Gopal *et al.*, 2023].

Advancements in Vehicular Ad-hoc Networks (VANETs), including Vehicle-to-Everything (V2X) communication, and the integration of Intelligent Transportation Systems (ITS) technologies are revolutionizing urban mobility. Integrating VANETs into urban mobility systems can enhance traffic efficiency and safety by enabling real-time communication between vehicles and infrastructure [Kumar and Mikkili, 2024].

A VANET is a decentralized, self-organizing system with a highly dynamic topology, as its nodes (vehicles) are moving most of the time. These characteristics enable

a message to reach its intended destination even if there is no direct connection between the source and destination, and also contribute to traffic management [Kumar and Mikkili, 2024].

There are two types of devices that play specific roles in a VANET [Al-Absi *et al.*, 2021]: On-Board Unit (OBU) and Road Side Unit (RSU). OBUs are devices installed in vehicles that generally function to collect and transmit important vehicle information, such as speed, direction, and location, to other nodes in the network. Vehicle-to-Vehicle (V2V) communication is purely ad hoc, with no connection to fixed infrastructures [Chang *et al.*, 2019]. On the other hand, RSUs are devices installed on the sides of roads and other infrastructure along the way. They collect traffic information and other important data, such as weather and road conditions, and transmit them to the nodes in the network. Additionally, RSUs can also be used to provide connectivity services to vehicles passing through their coverage area. These communications with RSUs are commonly referred to as Vehicle-to-Infrastructure (V2I) [Guerna *et al.*, 2022]. RSUs can also be interconnected through Infrastructure-to-Infrastructure (I2I) communication, extending the dissemination of traffic information, road conditions, and potential accidents [Aslam *et al.*, 2012]. Another type of communication existing in VANETs is Vehicle-to-Broadband (V2B), which allows the vehicle to communicate with broadband services, enabling

vehicle tracking and active driver assistance [Marinov *et al.*, 2017]. These information exchanges contribute to providing various services in smart environments, such as intelligent transportation systems [Gillani *et al.*, 2022; Vihurskiy, 2024; Izhari and Dhany, 2024]. With accurate and timely information, vehicles can make better decisions and react appropriately to traffic conditions, ensuring greater passenger safety [Grover *et al.*, 2018].

However, since the primary goal is to provide passenger safety, this can only be achieved when the vehicular network is resilient to attacks, a complex challenge in the absence of traditional security infrastructure [Feiri *et al.*, 2013].

Anomaly detection in urban mobility systems is crucial for maintaining the efficiency and safety of modern transportation networks. Identifying and addressing anomalies in real-time can prevent disruptions, enhance system reliability, and improve overall traffic management. This capability is particularly important in vehicular networks, where timely detection of irregularities can mitigate potential risks and ensure smoother operation of transportation systems. The ability of vehicular networks to detect anomalies, such as unexpected traffic patterns, communication failures, or cybersecurity threats directly impacts the efficiency of traffic management and the overall quality of urban mobility. In this context, machine learning technologies emerge as powerful solutions. Their ability to learn from data, identify anomalies, and make real-time decisions makes them powerful tools for enhancing urban traffic management [Izhari and Dhany, 2024] - such as traffic prediction, signal control, and congestion reduction [Vihurskiy, 2024] - as well as for improving transportation security [Bangui and Buhnova, 2021; Alqah-tani and Kumar, 2024].

The detection of an attack on a network is a task performed based on knowledge of the attack's characteristics and behavior. A network administrator, through a monitoring system, can interpret the traffic and segregate it into genuine and anomalous [Hoque *et al.*, 2014]. However, manually processing and analyzing VANET traffic becomes unfeasible, as these networks generate a large volume of traffic in a decentralized network [Deeksha *et al.*, 2017]. This presents the challenge of improving VANET security through automatic intrusion detection. Therefore, this work aims to address this gap by comparing two models used in anomaly detection in the context of vehicular networks: the CNN-LSTM model, which has already been applied in this area, and the TranAD model, which was adapted for this type of network. The objective is to validate the feasibility and performance of these models in identifying anomalies, providing a comparative analysis that highlights their strengths and limitations within vehicular network environments.

The research problem arises from the increasing number of studies on anomaly detection models applicable to Intrusion Detection Systems (IDS), which utilize artificial intelligence, particularly machine learning [Veeramreddy *et al.*, 2011; Khraisat *et al.*, 2019; Ahmad *et al.*, 2021; Bangui and Buhnova, 2021]. Many of these proposed models have not been validated and applied in VANETs, and therefore, there is little evidence of promising results in this specific context. The objective is to validate the feasibility and performance of these models in identifying anomalies, providing a compar-

ative analysis that highlights their strengths and limitations within vehicular network environments.

This work contributes to the field of anomaly detection in vehicular networks by systematically evaluating two machine learning models, CNN-LSTM and TranAD, in the context of intrusion detection. While CNN-LSTM has been previously applied to VANETs, TranAD was adapted for this scenario, and its performance was compared against a validated baseline. The key contributions of this study are as follows:

- **Comparative Evaluation of Anomaly Detection Models:** This study provides a direct comparison between CNN-LSTM and TranAD, highlighting their strengths and limitations when applied to vehicular networks. The results demonstrate that CNN-LSTM achieves superior detection performance in most cases, while TranAD shows potential as a lightweight alternative.
- **Dataset Refinement and Preprocessing Strategy:** We use the VeReMi extension dataset by applying adjustments to better reflect real-world conditions, including corrections to the labeling of the occasional stop attack. Additionally, we implement a preprocessing pipeline tailored for sequence-based anomaly detection, ensuring a fair and consistent evaluation of both models.
- **Analysis of Training Data Composition on Model Performance:** We investigate how different proportions of anomalous data in the training set impact detection performance. The findings highlight the trade-off between training models with purely normal data versus incorporating some level of anomalies, offering insights into the challenges of balancing detection sensitivity and generalization.
- **Implications for Future Research on IDS in Vehicular Networks:** The study identifies key challenges in applying machine learning to intrusion detection in VANETs, particularly the impact of noisy training data and the limitations of static training approaches. These findings serve as a foundation for future research on adaptive and scalable IDS solutions for vehicular networks.

By explicitly addressing these aspects, this work investigates the effectiveness of machine learning-based anomaly detection models applied to vehicular networks, based on the hypothesis that these approaches provide more effective and flexible solutions than traditional signature-based intrusion detection systems, particularly in identifying new attacks. To this end, two research questions are posed: (a) which anomaly detection model, TranAD or CNN-LSTM, achieves the best performance in identifying attacks in vehicular networks, and (b) how does the performance of these models behave given the particularities of this environment, including the exchange of information between vehicles and fixed stations.

2 Information Security

According to the international information security management standard, ISO 27001 [ISO/IEC 27001, 2022], information security is based on three pillars to protect information:

confidentiality, integrity, and availability.

An information security attack is an action that compromises the availability, integrity, confidentiality, and authenticity of information [Cebula *et al.*, 2014].

Confidentiality attacks allow the attacker to intercept communications, breaching the privacy of sensitive data. These attacks can be carried out by simulating a fake node or RSU. Eavesdropping or espionage is one type of attack that falls into this category, as it intercepts the conversation traffic between two nodes [Hezam Al Junaid, Mohammed Ali *et al.*, 2018].

Integrity attacks enable the attacker to manipulate data. These attacks can be executed by delaying or altering critical information, causing changes in the target's decision-making process [Hezam Al Junaid, Mohammed Ali *et al.*, 2018]. Some examples of this type of attack are: the Sybil attack, which sends multiple messages with fake senders to a specific node to flood it. In VANETs, network flooding with fictitious devices can result in route diversion of the target nodes, severely affecting nearby routes and providing the attacker with an unoccupied path; the Replay Attack, which sends back received location packets, altering the current view of node traffic; and the Position Faking Attack, which changes the position of a node, either hiding it from the network or placing it in another region, preventing this node from receiving critical information from vehicles that are physically close but virtually distant. This type of attack aids in executing attacks like the Sybil.

Availability attacks aim to completely or partially prevent the target system from performing its functions, which can be done by flooding the system with packets or damaging it [Hezam Al Junaid, Mohammed Ali *et al.*, 2018]. The DoS attack falls into this category as it floods the target node with multiple packets, causing an overload. This overload can slow down the device's services or even shut them down entirely. There are DoS attacks with different modes of operation, which may involve sending malformed packets or creating multiple deliberately slow connections by sending small packets without ever completing a request [Black and Kim, 2022].

Other attacks considered in this work include: random DoS, which also randomizes the data of each message; occasional stop, where the node behaves legitimately until it starts sending messages with a speed of zero to simulate a stopping vehicle; and disruptive attacks that aim to send previously received messages from other nodes, creating a higher level of plausibility. Disruptive and random attacks can be combined with DoS or Sybil, which will increase the number of messages or modify the sender's pseudonym of the message, respectively [Kamel *et al.*, 2020].

2.1 Intrusion and Anomaly Detection

Intrusion refers to the attempt of unauthorized access to a computer or a network, potentially compromising the reliability, integrity, or availability of its data [Ahmad *et al.*, 2021]. A detection system is the software or hardware responsible for detecting malicious activity, helping maintain the system [Khraisat *et al.*, 2019].

Regarding network availability, IDSs can be subdivided

into two categories: HIDS and NIDS. HIDS are deployed at the device level, while NIDS are deployed at the network level and are responsible for detecting traffic across multiple devices [Ahmad *et al.*, 2021]. The operation of an IDS consists of three stages: data collection, data analysis, and incident response, which aims to mitigate the impact of attacks on the system, also known as IDPS [Gonçalves *et al.*, 2021].

IDSs can also be divided into two categories of intrusion detection [Veeramreddy *et al.*, 2011]:

- The SIDS operates by comparing known network traffic patterns with real-time data to identify malicious activity. To do this, a SIDS has a database with the behavior of various attacks, which may use a system of hashes, malicious bytes, or packet sequences to verify if the behavior being received matches the behavior cataloged in its database. Its main advantage is its high efficiency for previously known attacks, given the availability of signatures. However, with the increase in zero-day attacks (i.e., threats that have not been previously identified by defense systems and security administrators [Sarhan *et al.*, 2023]), the effectiveness of this technique is reduced due to the lack of a previously recorded signature for these attacks. Some examples of applications of this type are Snort [Cisco Systems, 2023] and Suricata [Open Information Security Foundation, 2023];
- The AIDS assumes that any deviation from expected behavior can be considered malicious. This can be done through knowledge bases, statistical models, or machine learning. Its main advantage is its ability to detect new threats, such as zero-day attacks. However, defining the parameters that govern the fine line of abnormality becomes a challenge, as the efficiency of this system depends solely on adapting this curve to the proposed scenario [Khraisat *et al.*, 2019].

An anomaly, or outlier, can be defined as an observation that deviates significantly from others to the point that there is a suspicion that it was not generated by the same mechanisms [Hawkins, 1980]. The existence of anomalies may be linked to the unusual behavior of the processes that generate the data, which may contain information characterizing the anomalies present in these processes, making their detection useful for various purposes, such as in intrusion detection systems in computer networks [Aggarwal, 2017].

According to Ahmed *et al.* [2016], the main limitations in building an anomaly detection system are: there is no universal outlier detection technique, making systems for one specific type of data potentially useless for another; the analyzed data contains noise that can be identified as an anomaly; there are not enough publicly available datasets for most tasks, and the usual behavior of a distribution may change over time, meaning that something considered normal at one point may become anomalous after a certain discovery. The latter occurs frequently in IDSs for network security, given the discovery of a zero-day attack. Another problem more present in the supervised approach is the natural imbalance of classes. There is much more ordinary behavior than anomalous behavior in the data, which is a known problem in the data mining area, making it difficult to distinguish between classes [Bhuyan *et al.*, 2014].

3 Machine Learning

Machine learning (ML) plays a fundamental role in intrusion detection systems for vehicular networks, as it enables models to learn from large-scale data and identify patterns indicative of anomalies. Given the challenges of real-time detection in decentralized environments like VANETs, selecting appropriate learning paradigms – supervised, unsupervised, or semi-supervised – is crucial for enhancing detection performance. This section contextualizes the machine learning techniques applied in this study, emphasizing their relevance to anomaly detection in vehicular networks and justifying the choice of CNN-LSTM and TranAD models for evaluation.

Machine Learning arises from the need for artificial intelligence systems to acquire knowledge independently from patterns detected in data and past experiences [Goodfellow *et al.*, 2016]. A program learns from experience when its performance on a task improves based on an objective metric [Murphy, 2022; Mitchell, 1997]. In the task of anomaly detection, three types of machine learning are primarily used: supervised learning, unsupervised learning, and semi-supervised learning [Nassif *et al.*, 2021].

Supervised learning is based on improving a model so that its inputs map to a defined output. There is a target outcome that can be compared and minimized during the process; therefore, annotations in the data used are necessary to describe this outcome. Tasks that use this learning are usually classification, regression, or a combination of these.

In contrast, unsupervised learning seeks to understand the relationships among the studied data. Since it does not attempt to condition the model to a specific response, annotated data is unnecessary, simplifying its application in practical environments. The main tasks that use this learning include data clustering, which creates smaller data sets that are related in some way [Murphy, 2022]. Clustering organizes unlabeled data into groups based on some similarity measure, making it widely applied in anomaly detection due to its unsupervised nature, not requiring explicit descriptions of attacks [Buczak and Guven, 2016].

To address the limitation of annotated data in real-world applications, semi-supervised learning is employed. It combines supervised and unsupervised learning techniques [Chapelle *et al.*, 2009]. In anomaly detection, semi-supervised learning can be applied using partial data annotation for the most frequent and ordinary classes, with the remaining unannotated data assumed to be part of an anomalous class [Nassif *et al.*, 2021].

3.1 Neural Networks

An Artificial Neural Network (ANN) is a computational model inspired by the structure and functionality of biological neural networks, such as the human brain [Krogh, 2008]. ANNs are composed of interconnected artificial neurons organized into layers. Each neuron receives inputs, performs a weighted sum of these inputs, applies an activation function, and produces an output. The connections between neurons, represented by weights, determine the strength and influence of the inputs on the output. These weights are adjusted during training based on a specific learning algorithm, such as

back-propagation, to minimize the difference between the predicted and desired results. By iteratively adjusting the weights, ANNs can approximate complex nonlinear mappings and make predictions or classifications based on new inputs [Dongare *et al.*, 2012].

A Convolutional Neural Network (CNN) is a type of ANN that can capture features through convolutional filters. These filters aim to represent receptors that respond to a set of features. Compared to fully connected ANNs, not all neurons in subsequent layers are connected. Their weights are based on a group of previous neurons, reducing the number of trainable parameters in the model and speeding up training convergence [Li *et al.*, 2022].

Another type of ANN with significant impact on anomaly detection is known as Recurrent Neural Network (RNN). It is an ANN that allows loops or feedback connections within its architecture, which are continuously activated to process a data sequence. This structure enables the network to have a "memory," allowing later inputs to be influenced by the context provided by earlier inputs [Salehinejad *et al.*, 2017]. RNNs are designed to process sequential data, such as time series, which include network traffic.

Although traditional RNNs are theoretically simple, they face significant challenges during training. In these networks, the computed and propagated error tends to either vanish or explode over time, resulting in sharp weight oscillations and impairing the model's learning process. This phenomenon is known as "vanishing gradient" or "exploding gradient" [Ribeiro *et al.*, 2020]. One way to mitigate this problem is by using the Long Short-Term Memory (LSTM) architecture, a more robust recurrent neural network. LSTMs have gating mechanisms that control data flow, allowing them to retain and forget information more efficiently, making them effective in modeling longer-term sequences [Hochreiter and Schmidhuber, 1997].

RNNs have a fundamental limitation on a large scale, as they need to process inputs sequentially, which limits efficient training on extensive sequential data. The transformer model is a neural network architecture designed to address this limitation. Instead of using recurrent layers, the transformer relies primarily on attention mechanisms to capture dependency relationships in input sequences, allowing the model to parallelize its training. The model consists of an encoder and a decoder, both composed of several stacked layers of attention blocks [Vaswani *et al.*, 2017].

The attention mechanism is the key component of the transformer. It allows the model to relate different positions in a sequence to capture the dependency between them. It is computed through three linear operations, projecting the input sequences into queries, keys, and values. Then, the similarity between all combinations of input pairs is computed and used to weigh the corresponding values. The final result is obtained through a linear combination of the weighted values. The attention mechanism enables the transformer to assign different weights to various parts of the sequence, capturing important relationships and helping to efficiently model the global context [Vaswani *et al.*, 2017].

Neural networks are applied in a supervised manner to various tasks but can also be used in unsupervised problems. One way is by using autoencoders [Hinton and Salakhutdinov, 2006].

nov, 2006]. An autoencoder is a neural network trained to reproduce its input. It can be seen as a two-part network: an encoder and a decoder. The network's goal is to learn a model capable of creating a compressed representation in the encoder and then decoding it as accurately as possible in the decoder. A "bottleneck" is created at the end of the encoder, so for the network to succeed in training, it must learn to store the most important information from the input [Goodfellow et al., 2016]. Due to its ability to train without requiring data annotations and seek a reduced-dimensional representation of data, it is often applied in anomaly detection tasks, directly or indirectly [Alladi et al., 2021; Tuli et al., 2022; Jiao et al., 2022].

In the context of anomaly detection in vehicular networks, where real-time detection and large-scale data analysis are critical, CNN-LSTM and TranAD models stand out for their ability to handle both temporal and spatial features of data. In CNN-LSTM, the CNN layers are responsible for extracting local spatial features from the data, while the LSTM layers capture the temporal dependencies that are essential for identifying anomalies in sequential data, such as network traffic.

On the other hand, TranAD, based on the Transformer architecture, is capable of identifying anomalies in large-scale sequential data by leveraging the attention mechanism to focus on the most relevant parts of the data. This is especially useful in VANETs, where attacks or faults may not always be well defined or known in advance.

The choice of CNN-LSTM and TranAD models for evaluation in this study is justified by their ability to address the specific challenges of anomaly detection in VANETs. Together, these models cover a wide range of learning paradigms, making them well-suited for the dynamic and decentralized nature of vehicular networks. Moreover, the way CNN-LSTM and TranAD encodes information are different, enabling them to handle data patterns in distinct ways. This allows our study to analyze and compare different forms of reconstruction using the same framework.

4 Related Work

This section presents works that propose anomaly detection models. Some of these models were specifically designed and applied directly to VANETs, while others were designed as generic anomaly detectors.

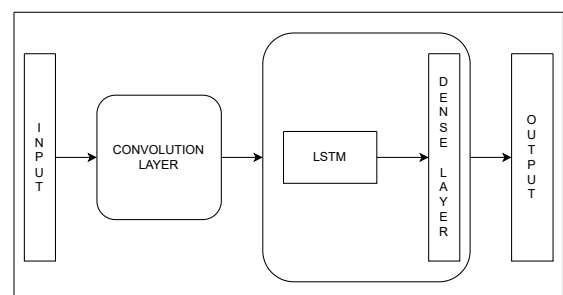
Tan et al. [2018] present an authentication scheme that uses an unsupervised anomaly detection system in VANETs using hierarchical clustering. The authors argue that traditional security techniques can create a heavy computational load on RSUs, making them vulnerable to Denial of Service (DoS) attacks. To mitigate this issue, they proposed using the agglomerative hierarchical clustering algorithm to identify anomalous traffic patterns in RSUs. By analyzing the inherent patterns of traffic flows using the Dynamic Time Warping (DTW) algorithm as a distance metric, the method filters abnormal messages without revealing their content. The DTW allows the model to consider temporal distortions, improving its ability to compare time sequences and thus increasing its robustness. This approach seeks to reduce the computational overhead of the authentication process and enhance

the overall security of VANETs. The metric used to measure the proposed model's capability was only the detection rate, synonymous with recall.

Nie et al. [2019] propose an anomaly detection approach for VANETs based on CNNs. The approach consists of two phases. In the first phase, a CNN architecture is designed to estimate network traffic based on spatiotemporal and sparse features. In the second phase, a decision-making approach is used to identify normal and anomalous traffic inputs based on traffic matrix estimates. A loss function is defined using Mahalanobis distance, a generalization of Euclidean distance that accounts for correlations and variances of variables, to measure the difference between the estimated and actual traffic. As the complete traffic matrix is required, anomaly detection is designed to be executed centrally in the RSU. Experiments are conducted on a testbed including an RSU and 12 OBU, and the proposed approach's performance is evaluated using a real traffic dataset collected from the testbed conducted by the authors. The experimental results presented metrics such as the true positive rate or recall, Spatial Relative Error (SRE), and Temporal Relative Error (TRE).

Alladi et al. [2021] propose an anomaly detection system for VANETs based on a sequence reconstruction and thresholding algorithm. The proposed structure consists of: extracting all types of messages from the data exchanged in the network, including failures, attacks, and genuine messages; preprocessing the data and converting it into sequences; passing these sequences to a reconstruction unit and classifying the sequences as anomalous or genuine using a threshold based on reconstruction results. The authors employ six different Deep Neural Network (DNN) architectures in the proposed structure, and these were evaluated using precision, recall, accuracy, and F_1 -score metrics. The modeled scenario considers the broadcast of messages to the RSUs, which will perform the anomaly detection. The open dataset called Vehicular Reference Misbehavior (VeReMi) [van der Heijden et al., 2018; Kamel et al., 2020] was used. The results showed that the proposed system achieved high precision in anomaly detection, with the model utilizing a CNN combined with LSTM presenting the best performance among the tested architectures. This model is composed of a convolutional layer with 20 filters and four layers with 256 LSTM units each. Afterward, a dense layer with a Sigmoid function is used to adjust the network output to the data normalization performed. This model sums up 1,864,968 trainable parameters. An overview of the architecture can be seen in Figure 1.

Figure 1. CNN-LSTM Model Architecture

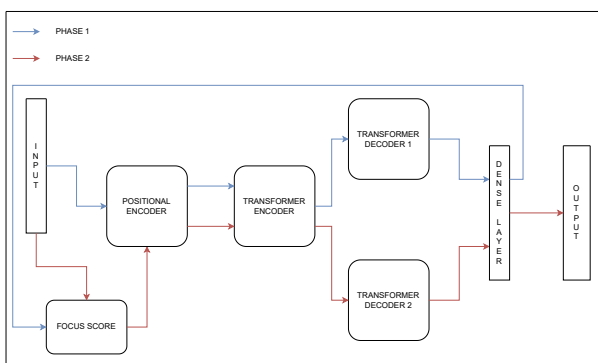


Source: Adapted from [Alladi et al., 2021]

Ramadan *et al.* [2021] propose a real-time intrusion detection system for Flying Ad hoc Network (FANET) based on deep learning. The system uses LSTM models for anomaly detection. The architecture consists of drones and a base station. Each drone is equipped with a simplified version of an LSTM model to detect attacks on itself, while the base station analyzes the drones' traffic using a more robust LSTM model to confirm detection. The decision-making module analyzes the results from the modules and generates alerts based on a voting process, categorizing the certainty level of the intrusion. Experiments were conducted using several datasets, though none specific to ad hoc networks. The metrics used for evaluation were precision, accuracy, recall, and F_1 -score.

Tuli *et al.* [2022] propose TranAD, a deep transformer network for anomaly detection in multivariate time series data. The proposal is aimed at generic anomaly detection, not specifically focused on VANETs. The model performs a two-phase inference, where the first phase generates an approximate reconstruction of the input window, and the second phase focuses on discrepancies concerning the first phase inference. An overview of the architecture is presented in Figure 2. The first pass begins with the positional encoding of elements using trigonometric functions, a process widely used in transformer architectures. After that, the data with positional information is submitted to the transformer encoder, then to the first transformer decoder, and finally to the dense prediction layer, which uses a Sigmoid function to limit the values between 0 and 1. Finally, the mean squared error between the first pass output and the initial input is calculated, generating the focus score. This focus score facilitates the attention mechanism to extract temporal trends. The second pass starts by concatenating the focus score with the initial input. Then, positional encoding is added to the concatenation, and it is submitted to the same transformer encoder used in the first pass, followed by a second transformer decoder and, finally, the same dense layer. The authors demonstrate, through tests on publicly available datasets, that TranAD outperforms reference methods in detection performance while significantly reducing training times. Precision, recall (Area Under the Curve (AUC)), and F_1 -score metrics were used.

Figure 2. TranAD Model Architecture



Source: Adapted from [Tuli *et al.*, 2022]

Jiao *et al.* [2022] present a technique called TimeAutoAD for anomaly detection in multivariate time series data within network systems. This is also a generic anomaly detection

system that was not specifically applied to VANETs. The technique involves an automatic anomaly detection process that automatically optimizes the model configuration and hyperparameters. It is composed of three main parts: automatic representation learning, automatic anomaly score calculation, and automatic negative sample generation. These parts are divided into nine modules, each with its own options and hyperparameters. In the automatic representation learning part, three options are considered for the encoder and decoder: RNNs, LSTMs, and Gated Recurrent Units (GRUs). The process is evaluated on real-world datasets, and the metric used for validation was the AUC.

The choice of models to be used in this work was made based on qualitative criteria. Recent articles that use different machine learning techniques were preferred to maximize the quality of the tests.

This study proposes to analyze the performance of the models proposed by Alladi *et al.* [2021], referred to in this work as CNN-LSTM, and Tuli *et al.* [2022], called TranAD. These models are adapted for anomaly detection in vehicular networks, and their performance is analyzed through simulations. Several factors guided the choice of these models. Firstly, both models follow an unsupervised approach based on sequence reconstruction, so their evaluation methods are similar. Secondly, the CNN-LSTM model was the one that achieved the best performance, serving as a baseline for this study, also surpassing the baseline obtained by Kamel *et al.* [2020].

Finally, the model presented in [Tuli *et al.*, 2022] was applied in various anomaly detection tasks, often surpassing the state of the art. Its selection in this work aims to represent the advances achieved in the general area of anomaly detection. This study adapts and tests this model in the context of attacks in VANETs.

5 Methodology

To analyze the performance of the CNN-LSTM and TranAD models in relation to anomaly detection in vehicular networks, the Vehicles in Network Simulation (VEINS) simulator [Sommer *et al.*, 2011] was used within the Luxembourg SUMO Traffic (LUST) scenario. This scenario simulates vehicle traffic in the city of Luxembourg [Codeca *et al.*, 2015].

The dataset used to train and validate the models is an extension of the VeReMi dataset [van der Heijden *et al.*, 2018; Kamel *et al.*, 2020], which was originally designed for evaluating misbehavior detection in vehicular networks. This extension enhances the dataset by increasing the volume of vehicle-generated messages, introducing variations in attack patterns, and refining the labeling of certain behaviors to improve classification accuracy. Additionally, in our approach, a correction was applied to the occasional stop attack, as suggested by Dutra *et al.* [2022], ensuring that normal behavior within this attack type was correctly labeled as genuine.

The dataset is split into training and testing data to be applied to the selected models. It consists of 101,347,646 log messages from each vehicle in the simulation, containing two types of messages: messages sent by other vehicles (V2V) and messages sent by internal sensors (Vehicle-to-

Table 1. Vehicle behavior scenarios

Type	Class	Name
0	Genuine	Genuine behavior
10	Attack	Disruptive
11	Attack	Data replay
12	Attack	Eventual stop
13	Attack	DoS
14	Attack	Random DoS
15	Attack	Disruptive DoS
16	Attack	Congestion <i>sybil</i>
17	Attack	Data replay <i>sybil</i>
18	Attack	Random DoS <i>sybil</i>
19	Attack	Disruptive DoS <i>sybil</i>

Source: Adapted from [Alladi et al., 2021]

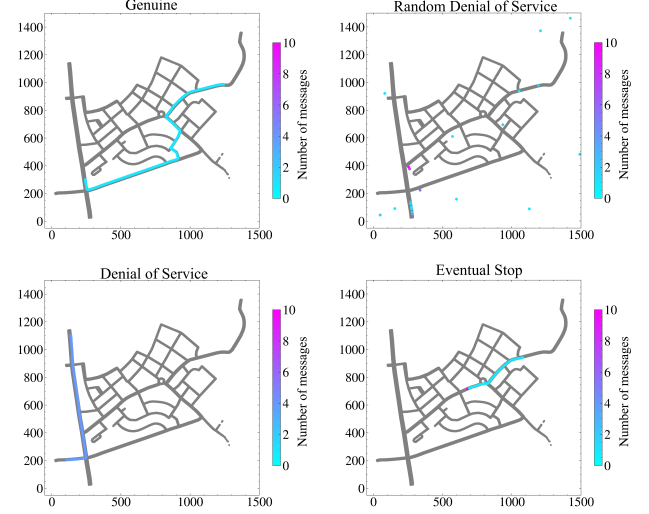
Sensor (V2S)). The V2V messages follow the Basic Safety Message (BSM) standard defined by Society of Automotive Engineers (SAE). These messages contain information such as the position, speed, and direction of each vehicle, allowing the verification of its current state. The V2S messages are used to represent the periodic behavior of the vehicle's Global Positioning System (GPS) module [van der Heijden et al., 2018; V2X Core Technical Committee, 2023].

The data fields present in the V2V message set are encoded in JavaScript Object Notation (JSON) format and include the following simulation-related information: type of attack, time of receipt and transmission of the message, sender, sender pseudonym, message identifier, sender position, sender position with noise, sender speed, sender speed with noise, sender acceleration, sender acceleration with noise, and sender direction with noise. The V2S messages do not include the fields for time of receipt and transmission of the message, sender, and sender pseudonym, as they are transmitted internally [Kamel et al., 2020].

The VeReMi dataset also includes a variety of vehicle behavior scenarios, which can be divided into genuine behaviors, failures, and attacks. Among the behaviors present in the dataset, this study will only use the attacks, which are presented in Table 1.

Figure 3 presents four maps of a region in Luxembourg. Each map illustrates one of the 20 possible behaviors that make up the dataset used. The x and y axes represent the position of the message-sending vehicle within the region, and the color of the points indicates the concentration of messages at that location, with cooler colors representing a lower number of messages and warmer colors representing a higher number of messages. It is possible to observe that genuine behavior shows cooler colors on the color scale, along with a set of points within a road, which are characteristics of normal behavior. In contrast, the other points deviate from these criteria, being outside the road, having a higher concentration of messages, or both. For example, warmer colors signify a high concentration of messages in a particular location, which could indicate a vehicle that has been stationary for a long time or is attempting to flood the network. Meanwhile, vehicles outside the road indicate potentially spoofed message positions.

The dataset proposed for model evaluation follows the data split standard used in other publications [Alladi et al., 2021; Dutra et al., 2022] that utilize the VeReMi (*Vehicular Reference Misbehavior*), dividing the dataset into 80%

Figure 3. Attack types distributed on a map of Luxembourg

for training and 20% for testing. The complete dataset contains approximately 28% of interactions initiated by attackers. This proportion is maintained in both the training and testing splits. Additionally, to approximate real-world situations where it is not feasible to annotate and completely clean the collected data, three training sets are utilized: one set composed exclusively of genuine data, where attacks are completely removed from the training set, and two other sets containing additions of attack sequences comprising 5% and 10% of the final constructed dataset. This allows the observation of each model's performance in the ideal case where it was trained only with genuine sequences and in the case where anomalies are present during training. The test dataset is the same for all training scenarios and contains both genuine sequences and attack sequences. In the test set, interactions initiated by attackers represent 46% of the total messages, rather than the expected 28% based on the total interaction count. This occurs because not all interactions have the same number of messages, and many of the attacks are of the denial-of-service type, where there is a higher number of messages sent compared to genuine behavior. Therefore, in the test set, the distribution of genuine or anomalous behavior classes by message is balanced.

A correction made to the dataset was guided by the work of Dutra et al. [2022], which points out a deficiency in the occasional stop attack. The attack exhibits legitimate behavior until the anomalous behavior begins. However, since they belong to the same group, all data are classified as attacks, potentially confusing the detection model. To address this issue, all legitimate behavior within the occasional stop group was labeled as genuine. This adjustment has a much greater impact on the work of Dutra et al. [2022], which uses supervised learning and classification and therefore uses labels during training. In this work, however, it affects the addition of attacks to the training sets, result evaluation, and threshold selection, where labels are utilized.

5.1 Performance Metrics

To evaluate the performance of the CNN-LSTM and *TranAD* models, the metrics precision, recall, and F_1 score were used. The definitions of these metrics are presented below, where

TP represents the number of true positives, TN represents the number of true negatives, FP represents the number of false positives, and FN represents the number of false negatives.

Precision (P) or true positive rate is defined in Equation 1:

$$P = \frac{TP}{TP + FP} \quad (1)$$

Recall (R) is defined in Equation 2:

$$R = \frac{TP}{TP + FN} \quad (2)$$

The F_1 score is defined as the harmonic mean of R and P , as formulated in Equation 3:

$$F_1 = 2 \times \frac{P \times R}{P + R} \quad (3)$$

5.2 Model Training

To make it possible to use the dataset and apply it directly to the models, the data must first be pre-processed into sequences so they can be used for training and inference of the ML algorithms. A sequence is defined by the messages sent by a sender, ordered chronologically.

For model training, the position and speed fields of the sender were selected. The attack type is only used in testing to calculate the model's performance metrics. This selection is based on the work of [Alladi et al., 2021].

The selected models work by analyzing fixed-size windows; therefore, the sequences must be divided according to this size. This implies that in the process of creating the message windows, there may be sequences where a window is smaller than the defined fixed size. When this occurs, the window is discarded, truncating the original sequence. This decision simplifies implementation and does not significantly impact the results, as the number of discarded messages is low.

The window size selection is based on the articles of the selected models. In [Tuli et al., 2022], fixed windows of 10 messages are used with a sliding step of 1. This scheme aims to obtain an autoregressive inference, as all elements will be analyzed as the end of the window and will be reconstructed based on the previous messages. In [Alladi et al., 2021], fixed windows of 20 messages are used with a sliding step of 10. This second scheme seeks to increase the analysis window in each processing and reduce the amount of redundancy of messages between windows. Consequently, the number of windows to be processed is reduced. In this work, the scheme based on [Tuli et al., 2022] is used, with autoregressive inference, utilizing a fixed window of size 10 and a sliding step of 1.

The final step of processing is data normalization. The same normalization proposed in [Tuli et al., 2022] is used, as defined in Equation 4, where T is the training dataset, ϵ' is a small value to avoid division by zero, and x_t is each record in question. This normalization ensures that all training values are between 0 and 1.

$$x'_t = \frac{x_t - \min(T)}{\max(T) - \min(T) + \epsilon'} \quad (4)$$

In the post-processing after inference, it is necessary to threshold the sequences. This is calculated from the mean prediction error of all features, defined by the mean squared error of the reconstruction. To choose the best threshold capable of distinguishing anomalies, an adaptation of Peaks-Over-Threshold (POT) is used, proposed in [Su et al., 2019] and utilized in [Tuli et al., 2022]. This technique computes an anomaly score for each observation and reconstruction tuple and uses this information to create a threshold based on a probabilistic distribution called Extreme Value Theory (EVT), which aims to find the rule of extreme values positioned at the end of the distribution. The classic POT is a theorem of this statistical theory, and its goal is to fit the end of this distribution to a Generalized Pareto Distribution (GPD).

Specifically, for each model and dataset configuration, we used a percentile that best balanced precision and recall in our tests. For the CNN-LSTM model, the threshold was set at the 99th percentile for the dataset with 0% attacks, 75th percentile for 5% attacks, and 30th percentile for 10% attacks. For the TranAD model, we applied the 95th percentile for the dataset with 0% attacks, 64th percentile for 5% attacks, and 25th percentile for 10% attacks. This dynamic threshold selection reflects the different reconstruction behaviors of each model and dataset scenario.

For model training, the open-source code provided by Tuli et al. [2022] is used as a base. It contains the model implementations using the *PyTorch* library [Paszke et al., 2019]. In this codebase, the CNN-LSTM model is implemented, and adaptations are made to the *transformer* model, as well as adding the created dataset. The *AdamW* optimizer [Loshchilov and Hutter, 2019] is used. A batch size of 2048 is chosen, and all models are trained with a learning rate of 0.0001. The models are trained for 150 epochs, as the results started to converge after 140 epochs. Each training session takes about 8 hours per CNN-LSTM model and 6 hours per TranAD model.

5.3 Scenario

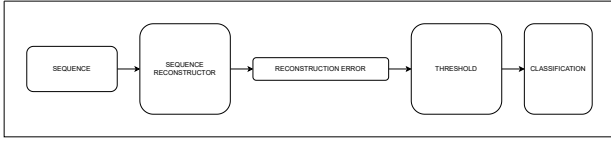
It is necessary to establish a scenario that represents an IDS architecture, providing a foundation for the experiments and tests so that the stages of collection, processing, and evaluation can be plausibly adapted to a real-world situation. This scenario should describe the behavior of network traffic and the means for implementing these stages. The architecture underpinning the experiments in this work was based on the scenario described in [Alladi et al., 2021].

The model training is conducted *offline*, using previously collected messages as the data source for learning. Only when the training is complete is the model loaded onto the RSU, which will then be capable of making the decisions presented in the following process.

Data is constantly transmitted via broadcast between the network nodes in message format. Each vehicle, through its installed OBU, stores the received messages and forwards them to the nearest RSU, where they are converted into sequences. From these sequences, signal reconstruction will be performed at the RSU. Based on the reconstruction error, each point in the sequence is classified into genuine or anomalous categories using a thresholding algorithm. This

flow can be visualized in Figure 4.

Figure 4. Generic Scenario Implementation on RSU



It is expected that the models will have difficulty reconstructing the anomalous signal because the amount of anomalies is dominated by the genuine signal in the training dataset. The RSUs are loaded with models trained on previously collected data.

5.4 Model Modifications

In [Alladi *et al.*, 2021], the architecture was initially proposed to reconstruct the entire sequence window in one iteration. However, in this work, it was adapted to predict only the last element to fit an auto-regressive inference similar to that of [Tuli *et al.*, 2022] and simplify the comparison.

In the TranAD model, 512 neurons were used in the dense layer of the encoders, 256 neurons for the dense layer of the decoders, 2 transformer layers in one encoder, and 1 transformer layer for each decoder. In this configuration, the model has 27,908 trainable parameters. The dropout layers presented in [Tuli *et al.*, 2022] were removed because they negatively affected the model's performance during the initial test cycles.

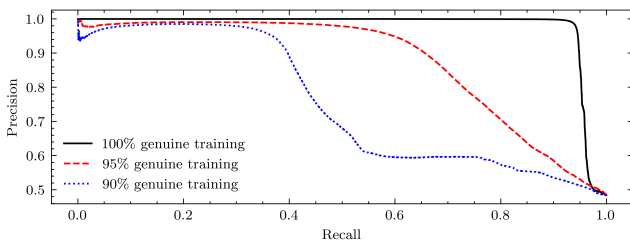
6 Results

In this section, the results obtained by the CNN-LSTM and *TranAD* models are highlighted, and the main differences and similarities between them are presented.

6.1 CNN-LSTM Model

First, it is observed that the model trained only with genuine data performs better than the other models in almost all possible threshold settings. This can be seen in Figure 5.

Figure 5. Precision-recall curve of the CNN-LSTM model



The precision and recall metric results are presented in Tables 2 and 3.

It is observed that the threshold found prioritizes higher recall at the expense of precision. Under these conditions, this implies that false positives are more frequent than false negatives. Thus, when analyzing Table 3, it can be seen that, with the exception of occasional stop, DoS, and Sybil congestion attacks, all training datasets achieved average recall. Of

Table 2. Precision of the CNN-LSTM model varying the percentage of attacks

Attack type	0% attacks	5% attacks	10% attacks
Disruptive	0.9774	0.6896	0.3588
Data replay	0.9791	0.6619	0.3557
Occasional stop	0.0103	0.1129	0.3296
DoS	0.9922	0.6013	0.5937
Random DoS	0.9936	0.8807	0.6607
Disruptive DoS	0.9938	0.8748	0.6404
Sybil congestion	0.9947	0.8593	0.7300
Sybil data replay	0.9757	0.6799	0.3716
Sybil random DoS	0.9897	0.8223	0.5715
Sybil disruptive DoS	0.9889	0.8298	0.5675
Average	0.9893	0.7893	0.5582

Table 3. Recall of the CNN-LSTM model varying the percentage of attacks

Attack type	0% attacks	5% attacks	10% attacks
Disruptive	0.9999	0.9862	0.8825
Data replay	0.9860	0.8510	0.8186
Occasional stop	0.0003	0.0726	0.9845
DoS	0.8565	0.1910	0.6481
Random DoS	1.0000	0.9998	0.9966
Disruptive DoS	0.9992	0.9816	0.8775
Sybil congestion	0.9474	0.5374	0.8098
Sybil data replay	0.9863	0.8766	0.8241
Sybil random DoS	1.0000	0.9998	0.9963
Sybil disruptive DoS	0.9996	0.9823	0.9004
Average	0.9295	0.7273	0.8559

those that achieved above-average performance, the data replay attack, for example, has a recall of 0.9860, 0.8510, and 0.8186 in the training sets with 0% attackers, 5%, and 10%, respectively. The random DoS attack, for example, achieved recalls of 1, 0.9998, and 0.9966 in these datasets.

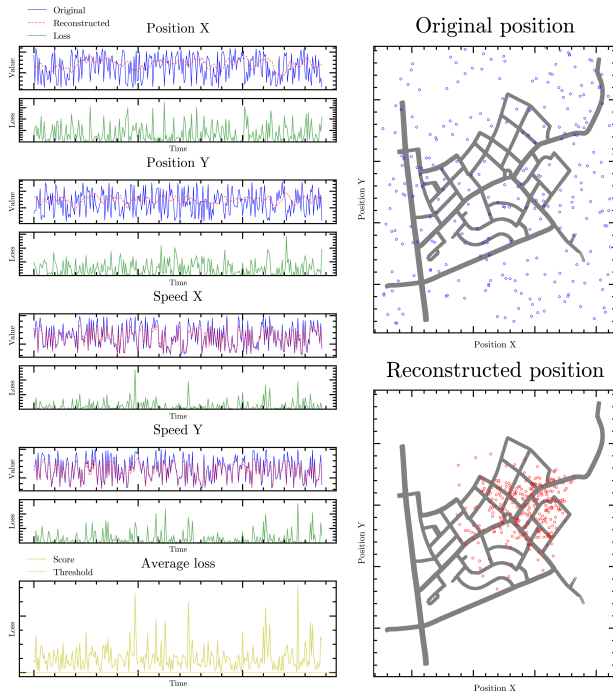
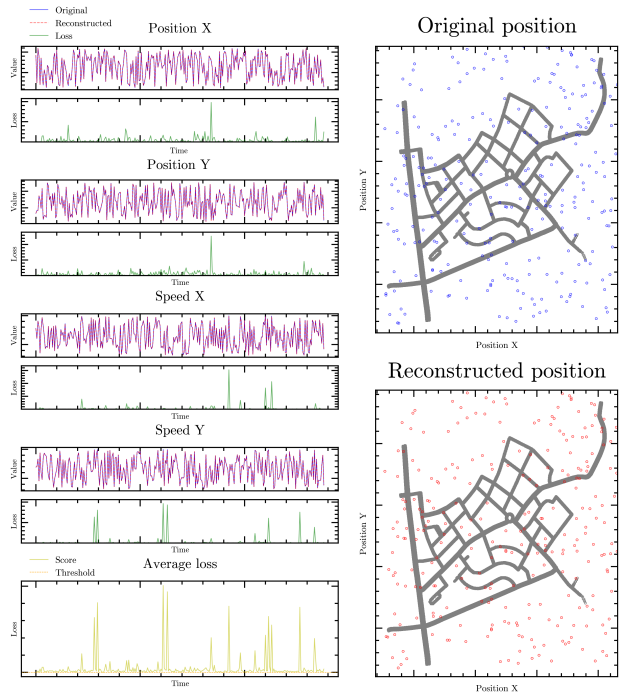
To better understand the results obtained, graphs were generated that present the relationship between the original sequences and the sequences reconstructed by the CNN-LSTM model. These graphs also include the loss that the model returns in reconstruction for each of the fields, as well as the average reconstruction loss and the variation in the position of the vehicles as they progress in the simulation.

In Figure 6, it is possible to observe the variation in vehicle position and speed generated by the random DoS attack with the training set with 0% attackers. Since this is an attack where the model had difficulty in reconstruction, good performance in distinguishing anomalies from genuine behaviors is noted.

In Figure 7, which has 5% attackers in the training base, there was a greater ease in reconstructing the characteristics. This occurs because these attack data impact the training of the reconstructor, making it a better reconstructor of anomalous signals, which implies lower loss and, consequently, poorer performance in distinguishing between anomalous and genuine behavior.

6.2 TranAD Model

As observed in the results of the CNN-LSTM model, Figure 8 shows that when the TranAD model is trained with a completely genuine dataset, the performance achieved in

Figure 6. CNN-LSTM Model - Random DoS - Genuine data**Figure 7.** CNN-LSTM Model - Random DoS - 95% genuine data

terms of precision and recall metrics is higher or equal across all threshold values. The dataset with 5% attackers performs worse than the dataset with 10% attackers only when recall is between 0.83 and 0.87 and precision is between 0.5 and 0.54.

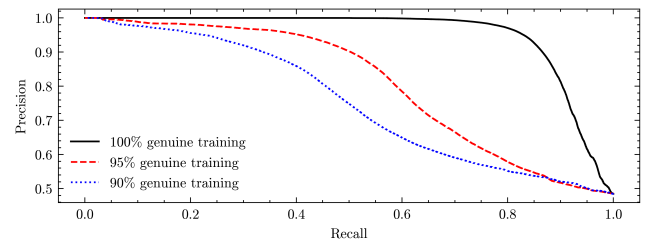
In Table 4, the precision of the TranAD model is presented, where a high performance was achieved, reaching an average of 0.9431.

Table 4. Precision of the TranAD model varying the percentage of attacks

Attack type	0% attacks	5% attacks	10% attacks
Disruptive	0.8977	0.5494	0.3597
Data replay	0.8880	0.5025	0.3472
Eventual stopping	0.0271	0.0855	0.2623
DoS	0.9561	0.5807	0.6005
Random DoS	0.9696	0.8047	0.6458
Disruptive DoS	0.9653	0.7903	0.6344
Sybil congestion	0.9621	0.7106	0.6797
Sybil data replay	0.9011	0.5333	0.3679
Sybil random DoS	0.9461	0.7319	0.5485
Sybil disruptive DoS	0.9474	0.7373	0.5623
Average	0.9431	0.6729	0.5381

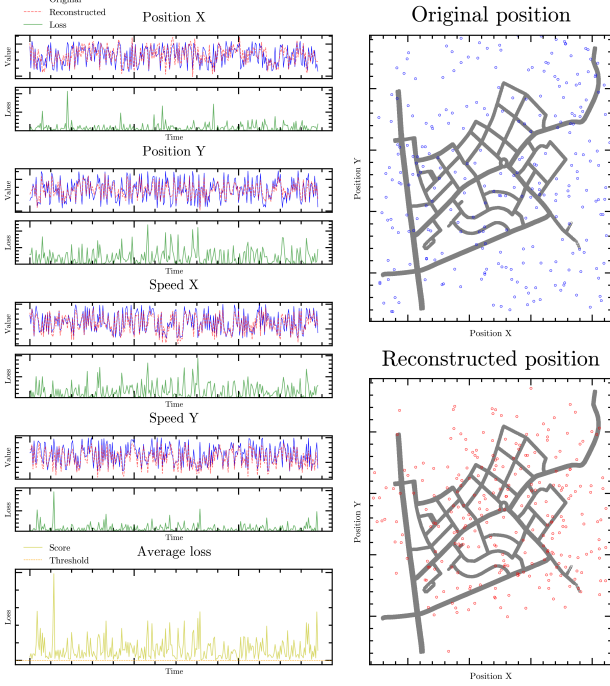
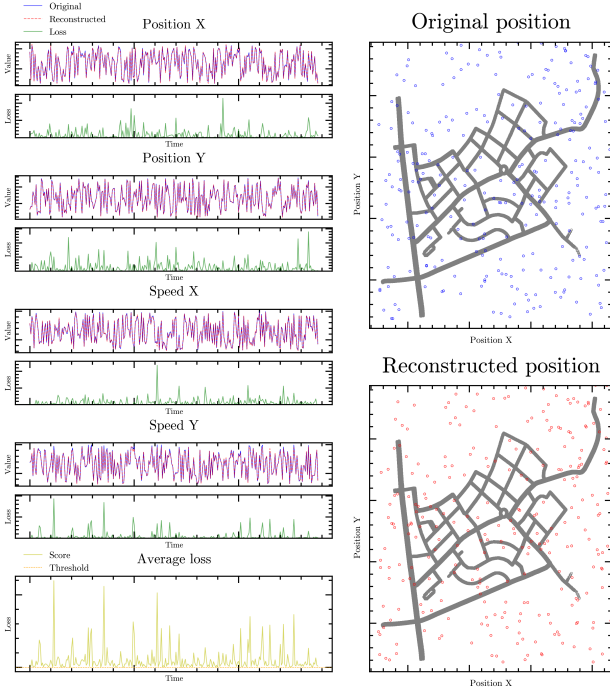
Similar to the CNN-LSTM, when analyzing Table 5, it is noticeable that the eventual stopping, DoS, and Sybil congestion attacks showed the worst performance, with their best results being in the training set with 10% attackers at 0.7768, 0.7272, and 0.7175, respectively. Although the recall for these attacks was above 0.7, the low precision of 0.2623, 0.6005, and 0.6797 for the dataset in question indicates a significant number of false positives.

To better understand the results obtained, graphs were generated showing the relationship between the original sequences and the sequences reconstructed by the TranAD model. For later comparison, the same behaviors and datasets presented in the CNN-LSTM model are used.

Figure 8. Precision-recall curve for the TranAD model**Table 5.** Recall of the TranAD model varying the percentage of attacks

Attack type	0% attacks	5% attacks	10% attacks
Disruptive	0.9929	0.9215	0.9645
Data replay	0.9011	0.7686	0.8917
Eventual stopping	0.0042	0.0926	0.7768
DoS	0.7266	0.3028	0.7272
Random DoS	1.0000	0.9986	0.9991
Disruptive DoS	0.9903	0.9161	0.9617
Sybil congestion	0.6171	0.3612	0.7151
Sybil data replay	0.9187	0.7981	0.9085
Sybil random DoS	1.0000	0.9986	0.9993
Sybil disruptive DoS	0.9919	0.9409	0.9761
Average	0.8318	0.6853	0.8718

Figure 9 shows the reconstruction of the random DoS attack using the training set with 0% attackers. Since this is an attack that uses data with large variations, the TranAD model struggles to reconstruct the sequence. Although it achieved a similar approximation, the threshold was adjusted so that the vast majority of the anomalous behavior was detected. Figure 10 shows the same attack, but using 5% attackers in the training dataset. In this dataset, the TranAD adapted the reconstruction very close to the original line, meaning that even though it was able to identify the vast majority of threats, it lost the ability to identify anomalous behaviors.

Figure 9. TranAD Model - Random DoS - Genuine data**Figure 10.** TranAD Model - Random DoS - 95% genuine data

6.3 Comparison between TranAD and CNN-LSTM models

This section presents a comparison between the results obtained with the CNN-LSTM and TranAD models. The metric primarily used is the F_1 score, as it represents both precision and recall, which are crucial metrics.

As seen in Figures 6 and 9, in vehicle position reconstruction, the CNN-LSTM model learns to condense the reconstruction of random behavior much more centrally within the simulated area, whereas TranAD, despite achieving numerically similar results, does not exhibit this same behavior. This indicates fundamental differences in how the evaluated models operate, with the CNN-LSTM learning to reconstruct random DoS more stably and TranAD doing so in a more chaotic manner, farther from the original.

Table 6 contains the results for the dataset with 0% attackers. It is observed that the CNN-LSTM model outperforms TranAD in almost all presented attacks. Notably, for random and disruptive denial-of-service attacks, the difference is small, totaling 0.0122 and 0.0189 respectively in the F_1 metric. In contrast, for the Sybil congestion attack, the difference was the largest, with CNN-LSTM achieving an F_1 score 0.2186 higher. For the occasional stop attack, both models performed poorly.

Table 6. F_1 Metric for Both Models Trained with 0% Attackers

Attack type	CNN-LSTM	TranAD
Disruptive	0.9885	0.9429
Data Repetition	0.9825	0.8945
Occasional Stop	0.0006	0.0072
DoS	0.9194	0.8257
Random DoS	0.9968	0.9846
Disruptive DoS	0.9965	0.9776
Sybil Congestion	0.9705	0.7519
Sybil Data Repetition	0.9810	0.9098
Random Sybil DoS	0.9948	0.9723
Disruptive Sybil DoS	0.9942	0.9691
Average	0.9585	0.8839

The results for the dataset with 5% attackers are presented in Table 7. The superiority of the CNN-LSTM model is maintained in this dataset as well, except for the occasional stop and denial-of-service attacks. Both models perform poorly in denial-of-service attacks, with CNN-LSTM showing lower performance compared to TranAD.

Table 7. F_1 Metric for Both Models Trained with 5% Attackers

Attack type	CNN-LSTM	TranAD
Disruptive	0.8116	0.6884
Data Repetition	0.7446	0.6077
Occasional Stop	0.0883	0.0889
DoS	0.2899	0.3980
Random DoS	0.9365	0.8912
Disruptive DoS	0.9251	0.8486
Sybil Congestion	0.6612	0.4790
Sybil Data Repetition	0.7658	0.6394
Random Sybil DoS	0.9024	0.8447
Disruptive Sybil DoS	0.8996	0.8267
Average	0.7570	0.6791

Finally, Table 8 shows the comparison for datasets with 10% attackers. In this table, both models show poor results, with an average F_1 score of 0.6757 for CNN-LSTM and 0.6655 for TranAD. It is observed that results are more balanced, with TranAD achieving better performance in 6 attacks and CNN-LSTM in 4 attacks.

Table 8. F_1 Metric for Both Models Trained with 10% Attackers

Attack type	CNN-LSTM	TranAD
Disruptive	0.5101	0.5240
Data Repetition	0.4959	0.4998
Occasional Stop	0.4938	0.3922
DoS	0.6197	0.6578
Random DoS	0.7946	0.7845
Disruptive DoS	0.7404	0.7645
Sybil Congestion	0.7678	0.6970
Sybil Data Repetition	0.5122	0.5238
Random Sybil DoS	0.7264	0.7082
Disruptive Sybil DoS	0.6962	0.7135
Average	0.6757	0.6655

The results show that the CNN-LSTM model outperforms TranAD in all attacks where both models achieve good detection performance. It is noted that in the occasional stop attack, both models fail to differentiate between genuine and anomalous messages, indicating that the features used are insufficient for its identification. Given that TranAD is configured with fewer parameters, it shows potential for improvement and possibly achieving the performance of the CNN-LSTM model. Also, it is observed that the dataset with 10% attackers is practically unfeasible, as the presence of attacks in the training had a significant negative impact on model performance. The dataset with 5% attackers achieves acceptable performance, especially in random and disruptive attacks, though still far below that trained with entirely genuine data.

6.4 Discussion

The results obtained in this study highlight the effectiveness of machine learning models for anomaly detection in vehicular networks, particularly in distinguishing between normal and attack behaviors. The CNN-LSTM model demonstrated superior performance across most attack scenarios, achieving an F_1 score of 0.9585 in the dataset trained exclusively with genuine data. The TranAD model, despite not reaching the same level of accuracy, showed promise due to its smaller architecture and potential for optimization.

One of the key challenges identified was the sensitivity of both models to the presence of attacks in the training dataset. As observed, when the models were trained with 10% attack sequences, their performance significantly degraded, indicating that the presence of anomalous data in the training phase can reduce their ability to generalize. This suggests a limitation in unsupervised learning approaches, which rely on clean datasets for optimal performance.

The results also indicate that the models struggled to detect certain attacks, particularly occasional stop, which had a significantly lower F_1 -score compared to other attack types. This poor performance can be attributed to the fact that oc-

casional stop combines periods of both normal and anomalous behavior, making it difficult to distinguish between legitimate traffic and an actual attack. Since the model relies on reconstructing temporal patterns, the presence of long periods of normal behavior within the sequence may have reduced its sensitivity to detecting the transition to anomalous behavior.

The inclusion of attack samples in the training set negatively impacted overall performance, as observed in the experiments with 5% and 10% of anomalous data. This suggests that the model may have learned anomalous patterns as part of normal behavior, reducing its generalization capability. This effect is particularly evident in the degradation of precision scores for attacks such as disruptive and data replay, which showed significant performance drops as the proportion of attack samples in training increased.

Another limitation concerns the applicability of the models in real-world VANET environments. While our study employed a simulation-based dataset, real vehicular networks may introduce additional complexities, such as data noise, sensor inaccuracies, and evolving attack strategies. Future research should explore model adaptability to dynamic environments and investigate strategies for continuous retraining to mitigate concept drift.

Additionally, while the POT-based threshold selection method proved effective in identifying anomalies, the chosen threshold values may not be optimal for all deployment scenarios. Fine-tuning these parameters for specific vehicular network conditions remains an open challenge.

Despite these limitations, our findings reinforce the importance of anomaly detection techniques for enhancing security in intelligent transportation systems.

7 Conclusions

Vehicular networks are crucial for modern urban mobility, as they enhance traffic management, reduce congestion, and improve safety. This research aimed to identify and evaluate two anomaly detection models and their architectures within the context of VANETs. The CNN-LSTM model, which had already been validated as an attack detector in VANETs [Alladi *et al.*, 2021], was compared with the TranAD model, which achieved strong performance in state-of-the-art generic contexts [Tuli *et al.*, 2022].

The research contributes a preprocessing proposal for model comparison in the Veremi dataset within a self-regressive scenario. The validation of the CNN-LSTM model was also achieved, yielding consistent results even with the variations implemented to better facilitate a direct comparison between models. The research also presents experiments with the TranAD model using the Veremi dataset, showing that the model did not achieve the performance of the CNN-LSTM model in the tested VANET context; however, it showed potential due to its reduced size. The analysis was also extended for both models with variations in the number of attacks present in the training, demonstrating the superiority of training with only genuine data and limitations when using this technique with a dataset containing anomalies, as training with 10% attackers resulted in unsatisfactory

performance for both models.

Based on the results obtained, the hypothesis and research questions were verified. Machine learning-based models offer a significant advantage over traditional systems, as they can detect new attacks beyond previously known threats with specific signatures. The results of this study clearly show that the CNN-LSTM model outperforms TranAD, achieving an F1 score of 0.9585 compared to 0.8839 in the best-case scenario for both models. Furthermore, the findings indicate that real-time information exchange between vehicles and fixed stations enhances anomaly detection, as it integrates up-to-date and diverse data that reflects the dynamic nature of the vehicular network.

7.1 Future Work

This work represents just the beginning of a broader field with substantial room for exploration. Many aspects of the proposed models, techniques, and methodologies still require further investigation and development. In sequence, we highlight some potential directions for future studies, providing a foundation for other researchers to build upon, refine, and extend these ideas.

1. **Machine Learning for Intrusion Detection.** Future work should focus on exploring more robust scenarios using the machine learning models studied in this work for intrusion detection. This includes addressing aspects of encryption and infrastructure that were not covered in this study, as well as developing systems that are better suited to real-world environments. These systems should:
 - Provide comprehensive notification mechanisms, enabling data analysis and auditing.
 - Implement continuous retraining policies for models.
 - Leverage signature-based servers to take preventive measures and assist in decision-making based on anomaly models.
2. **VANET Attack Datasets.** The evolution of these models will require the availability of a real, publicly accessible VANET attack dataset to enable more concrete studies.
3. **Continuous Learning in Real-World Environments.** Additionally, implementing continuous learning in real-world vehicular networks could be explored, especially in scenarios where models need to adapt to emerging traffic patterns and evolving cyber threats. This opens up an opportunity for work in the area of data collection and the development of non-simulated datasets.
4. **Preprocessing and Thresholding Techniques.** Another area for future research is the refinement of preprocessing techniques to reduce noise in the training data, such as removing extreme outliers or adjusting the balance between genuine and anomalous samples. Furthermore, employing advanced thresholding techniques to fine-tune detection sensitivity could improve the model's ability to distinguish between legitimate and anomalous patterns, thereby reducing the false positive rate for more challenging attacks.

5. **Efficient Machine Learning Models for Real-World Scenarios.** We understand that deploying machine learning models that require heavy computation can be challenging in real-world scenarios where computational resources are constrained. However, several strategies can help make ML models more efficient and adaptable to these limitations, while still providing practical value in various applications. These strategies may include:
 - **Model Compression:** Techniques such as quantization and pruning can significantly reduce the size and computational demands of models, making them more suitable for resource-constrained environments.
 - **Edge Computing:** In cases where local computation is limited, edge computing allows heavy computational tasks to be offloaded to nearby cloud servers or more powerful edge nodes, thus balancing the load between local and remote resources.
 - **Use of Specialized Hardware:** Many modern edge devices come with specialized AI hardware accelerators, such as Tensor Processing Unit (TPU), Graphics Processing Unit (GPU), or Neural Processing Unit (NPU), which are designed to run ML models efficiently. These accelerators can perform inference at lower power consumption and faster speeds compared to general-purpose CPUs.
 - **Federated Learning:** Federated learning enables distributed training of ML models across edge devices while keeping data localized. This approach allows devices with limited resources to contribute to the global model without the need for heavy computation on the device itself.
6. **Real-Time Feasibility.** Finally, future studies could explore the real-time feasibility of the proposed models by analyzing their computational efficiency, inference speed, and adaptability to large-scale vehicular networks.

Declarations

Acknowledgements

Funding

Authors' Contributions

All authors were involved in every stage of the work and contributed equally to its completion.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The source code and other materials are available at https://github.com/h8rtv/vanets_tranad_cnn_lstm.

References

- Aggarwal, C. C. (2017). Outlier analysis. Springer International Publishing. DOI: 10.1007/978-3-319-47578-3.
- Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., and Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1):e4150. DOI: 10.1002/ett.4150.
- Ahmed, M., Mahmood, A. N., and Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31. DOI: 10.1016/j.jnca.2015.11.016.
- Al-Absi, M. A., Al-Absi, A. A., Sain, M., and Lee, H. (2021). Moving ad hoc networks—a comparative study. *Sustainability*, 13(11). DOI: 10.3390/su13116187.
- Aldhanhani, T., Abraham, A., Hamidouche, W., and Shaaban, M. (2024). Future trends in smart green iov: Vehicle-to-everything in the era of electric vehicles. *IEEE Open Journal of Vehicular Technology*, 5:278–297. DOI: 10.1109/OJVT.2024.3358893.
- Alladi, T., Gera, B., Agrawal, A., Chamola, V., and Yu, F. R. (2021). Deepadv: A deep neural network framework for anomaly detection in vanets. *IEEE Transactions on Vehicular Technology*, 70(11):12013–12023. DOI: 10.1109/TVT.2021.3113807.
- Alqahtani, H. and Kumar, G. (2024). Machine learning for enhancing transportation security: A comprehensive analysis of electric and flying vehicle systems. *Engineering Applications of Artificial Intelligence*, 129:107667. DOI: 10.1016/j.engappai.2023.107667.
- Aslam, B., Amjad, F., and Zou, C. C. (2012). Optimal roadside units placement in urban areas for vehicular networks. In *2012 IEEE Symposium on Computers and Communications (ISCC)*, pages 000423–000429. DOI: 10.1109/ISCC.2012.6249333.
- Bangui, H. and Buhnova, B. (2021). Recent advances in machine-learning driven intrusion detection in transportation: Survey. *Procedia Computer Science*, 184:877–886. DOI: 10.1016/j.procs.2021.04.014.
- Bhuyan, M. H., Bhattacharyya, D. K., and Kalita, J. K. (2014). Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys & Tutorials*, 16(1):303–336. DOI: 10.1109/SURV.2013.052213.00046.
- Black, S. and Kim, Y. (2022). An overview on detection and prevention of application layer ddos attacks. In *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0791–0800. DOI: 10.1109/CCWC54503.2022.9720741.
- Borchers, T., Wittowsky, D., and Fernandes, R. A. S. (2024). A comprehensive survey and future directions on optimising sustainable urban mobility. *IEEE Access*, 12:63023–63048. DOI: 10.1109/ACCESS.2024.3393470.
- Buczak, A. L. and Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2):1153–1176. DOI: 10.1109/COMST.2015.2494502.
- Cebula, J., Popeck, M., and Young, L. (2014). A taxonomy of operational cyber security risks version 2. Technical Report CMU/SEI-2014-TN-006.
- Chang, X., Li, H., Rong, J., Huang, Z., Chen, X., and Zhang, Y. (2019). Effects of on-board unit on driving behavior in connected vehicle traffic flow. *Journal of Advanced Transportation*, 2019:8591623. DOI: 10.1155/2019/8591623.
- Chapelle, O., Scholkopf, B., and Zien, Eds., A. (2009). Semi-supervised learning (chapelle, o. et al., eds.; 2006) [book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542. DOI: 10.1109/TNN.2009.2015974.
- Cisco Systems (2023). Snort - network intrusion and detection system. Available at: <https://www.snort.org>, Accessed 18 abr. 2023.
- Codeca, L., Frank, R., and Engel, T. (2015). Luxembourg sumo traffic (lust) scenario: 24 hours of mobility for vehicular networking research. In *2015 IEEE Vehicular Networking Conference (VNC)*, pages 1–8. DOI: 10.1109/VNC.2015.7385539.
- Deeksha, Kumar, A., and Bansal, M. (2017). A review on vanet security attacks and their countermeasure. In *2017 4th International Conference on Signal Processing, Computing and Control (ISPCC)*, pages 580–585. DOI: 10.1109/ISPCC.2017.8269745.
- Dongare, A., Kharde, R., Kachare, A. D., et al. (2012). Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(1):189–194. Available at: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=04d0b6952a4f0c7203577afc9476c2fcab2cba06>.
- Dutra, F., Bonfim, K., Travagini, C., Meneguette, R., Santos, A., and Pereira, L. (2022). Detecção incremental de comportamento malicioso em vanets. In *Anais do XXII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 125–138, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/sbseg.2022.225164.
- Feiri, M., Petit, J., Schmidt, R. K., and Kargl, F. (2013). The impact of security on cooperative awareness in vanet. In *2013 IEEE Vehicular Networking Conference*, pages 127–134. DOI: 10.1109/VNC.2013.6737599.
- Gillani, M., Niaz, H. A., Farooq, M. U., and Ullah, A. (2022). Data collection protocols for VANETs: a survey. *Complex & Intelligent Systems*, 8(3):2593–2622. DOI: 10.1007/s40747-021-00629-x.
- Gonçalves, F., Macedo, J., and Santos, A. (2021). Evaluation of vanet datasets in context of an intrusion detection system. In *2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–6. DOI: 10.23919/SoftCOM52868.2021.9559058.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. Available at: <http://www.deeplearningbook.org>.
- Gopal, S., Gupta, P., Sharma, M., Kaushal, D., Joshi, S., and Sharma, B. (2023). Iot enabled e-vehicles for developing smart transportation system. In *2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*, pages 1063–1068.

- DOI: 10.1109/ICAICCIT60255.2023.10466202.
- Grover, J., Jain, A., Singhal, S., and Yadav, A. (2018). Real-time vanet applications using fog computing. In Somani, A. K., Srivastava, S., Mundra, A., and Rawat, S., editors, *Proceedings of First International Conference on Smart System, Innovations and Computing*, pages 683–691, Singapore. Springer Singapore. DOI: 10.1007/978-981-10-5828-8_65.
- Guerna, A., Bitam, S., and Calafate, C. T. (2022). Roadside unit deployment in internet of vehicles systems: A survey. *Sensors*, 22(9). DOI: 10.3390/s22093190.
- Hawkins, D. M. (1980). *Identification of Outliers*. Springer Netherlands. DOI: 10.1007/978-94-015-3994-4.
- Hezam Al Junaid, Mohammed Ali, Syed, A.A., Mohd Warip, Mohd Nazri, Fazira Ku Azir, Ku Nurul, and Romli, Nurul Hidayah (2018). Classification of security attacks in vanet: A review of requirements and perspectives. *MATEC Web of Conferences*, 150:06038. DOI: 10.1051/mateconf/201815006038.
- Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507. DOI: 10.1126/science.1127647.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
- Hoque, N., Bhuyan, M. H., Baishya, R., Bhattacharyya, D., and Kalita, J. (2014). Network attacks: Taxonomy, tools and systems. *Journal of Network and Computer Applications*, 40:307–324. DOI: 10.1016/j.jnca.2013.08.001.
- ISO/IEC 27001 (2022). Information security, cybersecurity and privacy protection – Information security management systems – Requirements. Standard, International Organization for Standardization, Geneva, CH. Available at: <https://www.iso.org/standard/27001>.
- Izhari, F. and Dhany, H. W. (2024). Optimizing urban traffic management through advanced machine learning: A comprehensive study. In *Proceedings of the [Conference Name, if available]*. DOI: 10.35335/idss.v6i4.167.
- Jiao, Y., Yang, K., Song, D., and Tao, D. (2022). Timeautoad: Autonomous anomaly detection with self-supervised contrastive loss for multivariate time series. *IEEE Transactions on Network Science and Engineering*, 9(3):1604–1619. DOI: 10.1109/TNSE.2022.3148276.
- Kamel, J., Wolf, M., van der Hei, R. W., Kaiser, A., Urien, P., and Kargl, F. (2020). Veremi extension: A dataset for comparable evaluation of misbehavior detection in vanets. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pages 1–6. DOI: 10.1109/ICC40277.2020.9149132.
- Khraisat, A., Gondal, I., Vamplew, P., and Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2(1):20. DOI: 10.1186/s42400-019-0038-7.
- Krogh, A. (2008). What are artificial neural networks? *Nature biotechnology*, 26(2):195–197. Available at: <https://www.nature.com/articles/nbt1386>.
- Kumar, G. and Mikkili, S. (2024). Critical review of vehicle-to-everything (v2x) topologies: Communication, power flow characteristics, challenges, and opportunities. *CPSS Transactions on Power Electronics and Applications*, 9(1):10–26. DOI: 10.24295/CPSSSTEPA.2023.00042.
- Li, Z., Liu, F., Yang, W., Peng, S., and Zhou, J. (2022). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12):6999–7019. DOI: 10.1109/TNNLS.2021.3084827.
- Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization.
- Marinov, T., Nenova, M., and Iliev, G. (2017). Dangerous weather warning algorithm in vanet. *International Scientific Conference on Information, Communication and Energy Systems and Technologies*, Niš, Serbia. Available at: http://rcvt.tu-sofia.bg/ICEST2017_62.pdf.
- Mitchell, T. M. (1997). *Machine learning*, volume 1. McGraw-hill New York. Book.
- Murphy, K. P. (2022). *Probabilistic Machine Learning: An introduction*. MIT Press. Book.
- Nassif, A. B., Talib, M. A., Nasir, Q., and Dakalbab, F. M. (2021). Machine learning for anomaly detection: A systematic review. *IEEE Access*, 9:78658–78700. DOI: 10.1109/ACCESS.2021.3083060.
- Nie, L., Wang, H., Gong, S., Ning, Z., Obaidat, M. S., and Hsiao, K.-F. (2019). Anomaly detection based on spatio-temporal and sparse features of network traffic in vanets. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. DOI: 10.1109/GLOBECOM38437.2019.9013915.
- Open Information Security Foundation (2023). Suri-cata - observe. protect. adapt. Available at: <https://suricata.io>.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc. Available at: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- Ramadan, R. A., Emara, A.-H., Al-Sarem, M., and Elhamahmy, M. (2021). Internet of drones intrusion detection using deep learning. *Electronics*, 10(21). DOI: 10.3390/electronics10212633.
- Ribeiro, A. H., Tiels, K., Aguirre, L. A., and Schön, T. (2020). Beyond exploding and vanishing gradients: analysing rnn training using attractors and smoothness. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Available at: <https://proceedings.mlr.press/v108/ribeiro20a.html>.
- Salehinejad, H., Sankar, S., Barfett, J., Colak, E., and Valaee, S. (2017). Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078*. DOI: 10.48550/arXiv.1801.01078.
- Sarhan, M., Layeghy, S., Gallagher, M., and Portmann, M. (2023). From zero-shot machine learning to zero-day at-

- tack detection. *International Journal of Information Security*. DOI: 10.1007/s10207-023-00676-0.
- Sommer, C., German, R., and Dressler, F. (2011). Bidirectionally coupled network and road traffic simulation for improved ivc analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15. DOI: 10.1109/TMC.2010.133.
- Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., and Pei, D. (2019). Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19*, page 2828–2837, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3292500.3330672.
- Tan, H., Gui, Z., and Chung, I. (2018). A secure and efficient certificateless authentication scheme with unsupervised anomaly detection in vanets. *IEEE Access*, 6:74260–74276. DOI: 10.1109/ACCESS.2018.2883426.
- Tuli, S., Casale, G., and Jennings, N. R. (2022). Tranad: Deep transformer networks for anomaly detection in multivariate time series data.
- V2X Core Technical Committee (2023). V2X communications message set dictionary. 400 Commonwealth Drive, Warrendale, PA, United States. DOI: 10.4271/J2735_202309.
- van der Heijden, R. W., Lukaseder, T., and Kargl, F. (2018). Veremi: A dataset for comparable evaluation of misbehavior detection in vanets.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc. Available at: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
- Veeramreddy, J., Prasad, V., and Prasad, K. (2011). A review of anomaly based intrusion detection systems. *International Journal of Computer Applications*, 28:26–35. DOI: 10.5120/3399-4730.
- Vihurski, B. (2024). Optimizing urban traffic management with machine learning techniques: A systematic review. In *2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, pages 403–408. DOI: 10.1109/InCACCT61598.2024.10551137.