Computer Standards & Interfaces 90 (2024) 103845

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Detection of DDoS attacks in SDN-based VANET using optimized TabNet

Mohamed Ali Setitra *, Mingyu Fan

School of Computer Science and Engineering (Cyberspace Security), University of Electronic Science and Technology of China (UESTC), No. 2006, Xiyuan Ave., West Hi-Tech.Zone, 611731, Chengdu, China

ARTICLE INFO

Keywords:
SDN networks
VANET
DDoS detection
TabNet
Deep learning
Adam optimization

ABSTRACT

Vehicular Ad Hoc Network (VANET) serves as a crucial component in developing the Intelligent Transport System (ITS), which provides a range of services expected to increase road safety and improve the global driving experience. At the same time, Software Defined Network (SDN) is a promising solution for VANET communication security due to the risk related to the dynamic nature of the vehicular network. However, the centralized structure of SDN-based VANET exposes vulnerabilities to Distributed Denial of Service (DDoS) attacks, which can significantly impact the network's performance. This work presents a deep learning technique for identifying DDoS attacks in SDN-based VANET, commonly called TabNet, a cutting-edge deep learning model for tabular data that generally surpasses traditional machine learning models regarding crucial performance metrics. The model underwent hyperparameter tuning and employed Adam optimization to enhance its performance. Comparative evaluations against other machine learning algorithms demonstrated the proposed model's robustness, achieving an overall accuracy of 99.42%. Our suggested method presents a potential solution for detecting DDoS attacks in SDN-based VANET, outperforming conventional techniques in terms of accuracy and efficiency.

1. Introduction

VANET is a mobile ad hoc network (MANET) that enables vehicles to communicate with each other and with roadside units (RSUs) and base stations (BS). It provides various services to enhance road safety, improve the driving experience, and reduce traffic congestion [1,2]. VANETs use wireless communication technologies such as Wi-Fi, Bluetooth, or Dedicated Short Range Communications (DSRC) to exchange data between vehicles and RSUs. The vehicle communication can be either direct or indirect through RSUs, allowing for a highly dynamic and flexible network architecture [3,4]. VANETs are an essential component of the Intelligent Transport System (ITS) and play a crucial role in creating innovative, connected, and safe roads [5].

SDN is a networking architecture that separates the control plane (the decision-making process) from the data plane (the forwarding process). In SDN, the control plane is managed by a centralized controller, which communicates with the data plane (network devices) using APIs [6,7]. This architecture allows network administrators to manage and control the network programmatically, leading to increased network visibility, flexibility, and programmability [8]. Centralized management in SDN enables the network to be more agile, easier to manage and maintain, and more cost-effective [9].

As described in Fig. 1, SDN-based VANET refers to using SDN technology to manage and control the communication network in VANET [10,11]. The SDN technology offers a centralized architecture to manage the network, making it easier to configure and monitor [12]. This centralized architecture provides communication services between vehicles and roadside units, enabling information sharing between vehicles and the road infrastructure to improve road safety and driving experience [13]. The integration of SDN and VANET technology enhances the security and reliability of the communication network in VANET, making it a promising solution for developing ITS [14].

Distributed Denial of Service (DDoS) attacks are a type of cyber attack in which multiple compromised systems are used to flood a target system with a large amount of traffic, overwhelming its resources and rendering it unavailable [15–17]. In an SDN-based VANET infrastructure, these attacks can be hazardous, as they can disrupt the communication and performance of the network [18,19]. Machine learning algorithms can analyze large amounts of network traffic data and identify patterns and anomalies that may indicate an attack, providing a more proactive approach to network [20–22]. The use of machine learning techniques to detect DDoS attacks is a crucial step towards improving the safety of SDN-based VANET systems due to their increasing complexity and scale, where the traditional signature-based or rule-based security methods become insufficient [23,24].

Machine learning algorithms can also reduce the manual workload and time needed to detect and mitigate DDoS attacks, making the

* Corresponding author.
E-mail addresses: setitra@outlook.com (M.A. Setitra), ff98@163.com (M. Fan).

https://doi.org/10.1016/j.csi.2024.103845
Received 7 August 2023; Received in revised form 1 February 2024; Accepted 12 February 2024
Available online 18 February 2024

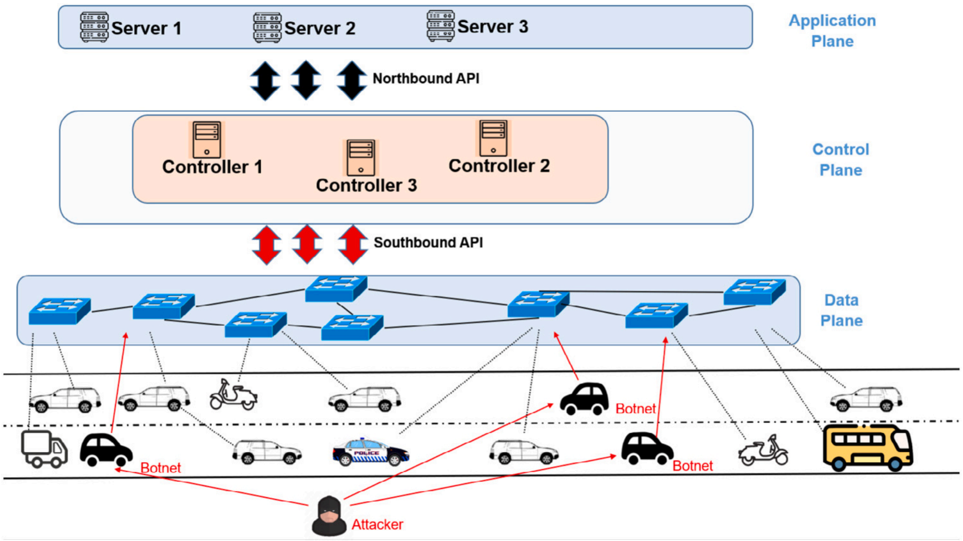


Fig. 1. DDoS attacks in SDN-based VANET [10].

process more efficient [25–27]. Additionally, as the attack patterns evolve, machine learning algorithms can adapt and learn to detect new attacks, making them more resilient to evolving security threats [28, 29]. This can lead to severe consequences in a VANET, as it is a crucial component in developing ITS, which provides many services to enhance road safety and improve the driving experience. Getting practical and efficient methods for detecting DDoS attacks in VANET is vital [30].

This article discusses using a deep learning technique called TabNet to DDoS attacks in SDN-based VANET. The dynamic nature of the vehicular network creates security risks, and SDN is seen as a promising solution for VANET communication security. TabNet is a cutting-edge deep learning model for tabular data that outperforms traditional machine learning models in accuracy and efficiency. In this research, the hyperparameters of TabNet were tuned and optimized using Adam optimization to enhance its performance further. The proposed approach successfully identified DDoS attacks and revealed the key features responsible for the detection, and it outperformed conventional techniques regarding accuracy and efficiency. The overall goal is to provide a solution that effectively detects DDoS attacks in SDN-based VANET while offering insights into the key features responsible for the detection. The motivations for using these techniques stem from the need to address the vulnerability of the centralized structure of SDN-based VANET to DDoS attacks and the limitations of traditional methods for detecting such attacks. The contributions of the article are described through the following points:

- **Innovative Application of Adam Optimization and TabNet:** We introduce an application of Adam optimization coupled with the TabNet deep learning classifier for detecting DDoS attacks in SDN-based VANETs. This approach is grounded in theoretical frameworks that underpin VANET security, ensuring a robust fusion of theoretical insights with practical solutions using two publicly available datasets (described later in the article) [31,32].
- **Thorough Comparative Analysis of DDoS Detectors:** This paper comprehensively analyzes DDoS detectors, leveraging machine learning (ML) and deep learning (DL) approaches. Our selection of reviewed papers is deliberate, focusing on their use of similar ML techniques and providing a nuanced evaluation and comparison concerning DDoS detection. We delve into the intricacies of datasets, optimization methods, and targeted systems, such as SDN or SDN-VANET, contributing to both theoretical understanding and practical insights.
- **Efficient TabNet architecture enhancement and significance for an automotive approach:** Our research achieves a notable milestone in enhancing the efficiency of the TabNet architecture, specifically tailored for automotive applications, employing Adam optimization and an exhaustive Grid Search Cross-Validation (GSCV) during training. We ensure that TabNet is well-equipped to address the unique challenges posed by vehicular environments. This enhancement not only advances the capabilities of TabNet for robust DDoS attack detection but also contributes meaningfully to the broader field of automotive security by providing a model intricately designed to meet the specific demands of securing communication systems in automotive settings.
- **Performance Evaluation Against State-of-the-Art Techniques:** Our proposed approach undergoes rigorous evaluation against four supervised machine learning techniques and two deep learning techniques to detect DDoS attacks in SDN-based VANETs. This comparison is contextualized within relevant studies, providing a benchmark for performance assessment. This contributes to the theoretical discourse by highlighting the strengths and limitations of existing models in the specific context of VANET security.
- **Promising Solution for VANET Communication Security:** The culmination of our contributions provides a promising solution for enhancing VANET communication security through accurate and interpretable DDoS attack prediction. This advances theoretical understanding and offers a practical pathway for addressing security challenges in dynamic vehicular network environments.

The rest of the paper is structured as follows: Section 2 reviews previous work on detecting DDoS attacks in VANET and SDN. Section 3 delves into the field's comprehensive background and core concepts. Section 4 presents the proposed methodology for detecting DDoS attacks in SDN-based VANET based on an optimized version of TabNet. Section 5 showcases the performance analysis and comparison of the results obtained from the experiments. Lastly, the paper concludes in Section 6 with a summary of the findings and future work.

It is helpful to mention that Table 1 describes the definition of acronyms and notations used in this paper.

1.1. Motivation

Ensuring the security and resilience of modern network infrastructures against DDoS attacks stands as a cornerstone for maintaining the reliability and functionality of these complex systems. Mainly, SDN

Table 1
Definition of notations.

Abbreviation	Meaning	Abbreviation	Meaning
ABCA	Artificial Bee Colony Algorithm	LSTM	Long Short-Term Memory
BAT	Bat algorithm	MANET	Mobile Ad hoc NETwork
BS	Base Stations	ML	Machine Learning
CNN	Convolutional Neural Network	MLP	Multilayer perceptron
DDoS	Distributed Denial of Service	MRMR	Maximum Relevance Minimum Redundancy
DL	Deep Learning	MSOM	Multilayer Self-Organizing Maps
DNN	Deep Neural Network	PSO	Particle Swarm Optimization
DRL	Deep Reinforcement Learning	PCA	Principal Component Analysis
DSRC	Dedicated Short-Range Communications	RBF	Radial Basis Function Kernel
DT	Decision Tree	RF	Random Forest
FFNN	Feed-Forward Neural Network	RMSProp	Root Mean Square Propagation
GBT	Gradient boosting	RSUs	RoadSide Units
GLU	Gated Liner Unit	SDN	Software Defined Network
GSCV	Grid Search Cross-Validation	SGD	Stochastic Gradient Descent
ITS	Intelligent Transport System	SVM	Support vector machine
KNN	K Nearest Neighbor	VANET	Vehicular Ad Hoc Network
LR	Logistic Regression		

leveraged within VANETs has introduced innovative communication paradigms but concurrently raised the vulnerability to DDoS threats. The motivation behind this study arises from the urgent necessity to fortify these network systems against the evolving and multifaceted landscape of cyber threats.

One of the primary motivations is rooted in addressing the augmented cybersecurity risks within SDN-based VANETs. The wireless and dynamic nature of VANETs amplifies the susceptibility to DDoS attacks, necessitating comprehensive defensive measures. These attacks' escalating frequency and complexity demand more sophisticated detection methods, prompting the exploration of cutting-edge technologies to enhance network security.

The deployment of advanced machine learning models, specifically deep learning approaches, stands out as an avenue to fortify the defense mechanisms against DDoS attacks in these network infrastructures. TabNet, an optimized deep learning model, promises advanced capabilities in efficiently detecting these threats, thus becoming a focal point of interest in this study. Leveraging such high-performing algorithms is essential to proactively identify and combat the sophisticated nature of DDoS attacks that challenge the security of SDN-based VANETs.

The integration challenges of merging SDN into VANETs heighten the vulnerability to these cyber threats. This creates an impetus for developing and deploying tailored security measures to safeguard the system's integrity. Optimized models, like TabNet, present a strategic solution that might substantially reinforce DDoS detection in this intricate network environment, thereby enhancing these integrated systems' overall resilience and robustness.

The adoption of GSCV and the ADAM optimizer is motivated by the pursuit of optimal model performance in DDoS attack detection. GSCV enables exploring a vast parameter space to identify the most effective hyperparameter combinations to enhance the TabNet model's detection capabilities. On the other hand, the ADAM optimizer, known for its efficiency in large-scale optimization problems, dynamically adjusts learning rates and model parameters, expediting convergence and providing stability. Integrating GSCV and ADAM aims to fortify TabNet's effectiveness in countering complex DDoS threats.

By enhancing network security in SDN-based VANETs according to implementing advanced methodologies, this research seeks to pave the way for evolving proactive defense strategies. The goal is to fortify the defense mechanisms against DDoS attacks and to set a benchmark for future network security in similarly dynamic and evolving network ecosystems.

2. Related works

2.1. Literature review

This section overviews recent approaches to detect DDoS attacks in SDN-based VANET and similar networks. A comprehensive comparison of recent related works on enhancing ML-based DDoS attack detection is presented in Table 2.

In SDN security, Tang et al. [33] introduced FTODefender, a method designed to detect Low-rate Flow Table Overflow (LFTO) attacks. This innovative approach combines eviction and cut-off strategies to identify potential threats in SDN environments. Meanwhile, Rao et al. [34] proposed a novel framework integrating Support vector machine (SVM), Multilayer perception (MLP), and Long Short-Term Memory (LSTM) to address correlated features within standard networks. This framework is evaluated using the NSL-KDD dataset, providing insights into its efficacy.

Rashid et al. [35] thoroughly investigated DDoS attack detection, employing techniques such as GBT, LR, MLP, RF, and SVM. Notably, the optimization process in this study was carried out manually, with a specific focus on VANETs. The authors assessed the adaptive approach in a simulated environment, utilizing the Kaggle dataset to evaluate its performance.

Jebur et al. [36] contribute to the field by introducing a Modified Chaotic Cellular Neural Network as a detection mechanism. Their innovative method involves a hybrid Particle Swarm Optimization (PSO)-Bat algorithm (BAT) optimization applied to VANETs. The study focuses on leveraging the CICIDS-2017 dataset, emphasizing a unique set of 10 features for a more targeted approach to threat detection.

In [38], the authors proposed using the Radial Basis Function Kernel Support Vector Machine (RBF-SVM) algorithm with Grid Search Cross-Validation (GSCV) hyperparameter tuning. The experiment was conducted using the SDN-DDoS dataset. In [39], a Feed-Forward Neural Network (FFNN) approach was used to enhance the performance of MANET in an SDN environment. A secure forwarding strategy based on Deep Reinforcement Learning was proposed for the VANET environment in [39].

The research in [40] combined Bayesian optimization and Maximum Relevance Minimum Redundancy (MRMR) feature selection to detect DDoS attacks in SDN-based VANET using decision tree machine learning and a self-generated dataset. Using a self-generated topology, an approach based on the invariant state set was proposed in [41] to detect attacks in an SDN-enabled vehicle platoon control system.

In [42], the authors used the SDN-DDoS dataset and SVM-RBF to detect DDoS attacks in SDN-based VANET. In [43], the Multilayer Self-Organizing Maps (MSOM) mechanism was used to defend against DDoS attacks in ZSN-based VANET using three different datasets (CAIDA, NSL-KDD, and DARPA). The Variant Artificial Bee Colony Algorithm was proposed in [44] to mitigate DDoS attacks in VANETs by fine-tuning the level of exploitation in the onlooker bee phase and enhancing the degree of exploration in search dimensions at the scout bee phase. A hybrid approach combining KNN SVM with stacked sparse autoencoders was proposed and implemented in [45] to detect DDoS attacks on SDN-based VANET.

Table 2
Comparison of recent literature.

Ref	Year	Technique	Optimization	Feature selection	Network	Dataset
[33]	2024	Combining eviction and cut-off attack sources	Tuned manually	NS	SDN	Self-generated
[34]	2024	SVM, MLP and LSTM	NS	Correlated features	Standard network	NSL-KDD
[35]	2023	GBT, LR, MLP, RF and SVM	Tuned manually	NS	VANET	Self-generated & Kaggle
[36]	2023	Modified Chaotic Cellular Neural Network	Hybrid PSO-BAT Optimization	10 features	VANET	CICIDS-2017
[37]	2023	Proposed DoSRT based on Direct trust and Indirect Trust	Tuned manually	NS	VANET	Simulation
[38]	2023	RBF-SVM	Hyperparameter tuning using GSCV	Analyze with PCA	SDN-based VANET	SDN-DDoS
[24]	2023	FFNN	NS	NS	MANET	NS
[39]	2023	DRL	Tuned manually	NS	VANET	Self-generated
[40]	2022	DT	Bayesian optimization	MRMR	SDN-based VANET	Self-generated
[41]	2022	Proposed Distributed Information System	Convex optimization	Minkowski sum	SDN-based vehicle platform	Self-generated
[42]	2022	RBF-SVM	NS	NS	SDN-based VANET	SDN-DDoS
[43]	2021	MSOMs	NS	NS	SDN-based VANET	CAIDA, NSL-KDD, DARPA
[44]	2021	ABCA	Differential evolution and Chaotic systems	NS	VANET	Simulation
[45]	2020	Hybrid based on KNN SVM	NS	NS	SDN-based VANET	Simulation
Proposed model	**2024**	**OptTabNet**	**Hyperparameter tuning using GSCV**	**TabNet**	**SDN-based VANET**	**SDN-DDoS, InSDN**

NS: Not Specified.

Our study aims to contribute to the existing work on detecting DDoS attacks in SDN-based VANET using machine learning techniques. This approach will utilize a state-of-the-art optimized deep learning model, TabNet, to detect DDoS attacks.

2.2. Key findings

In the evolving landscape of detecting DDoS attacks in SDN-based VANETs, extensive research has unearthed key insights:

1 **Machine Learning Adoption in VANETs:** The research underscores the increased reliance on machine learning methodologies, particularly deep learning, as an effective tool for DDoS attack identification in VANETs. This utilization demonstrates enhanced pattern recognition and advanced analysis for threat detection.
2 **SDN Integration in VANETs:** The integration of SDN into VANETs has emerged as a prominent trend, promoting centralized network control. This convergence is crucial, as it centralizes network control, fostering proactive DDoS mitigation and more efficient network management.
3 **Optimization for Robust Models:** The importance of optimization techniques is paramount. Crafting resilient machine learning models engineered for detecting DDoS attacks requires advanced optimization strategies to ensure the accuracy and efficiency of threat identification.
4 **Trend Towards Advanced Technologies:** The collective findings underscore a prevalent shift towards leveraging sophisticated technologies and methodologies to fortify threat detection mechanisms in vehicular networks. This shift reflects a progressive movement in bolstering security protocols for VANETs.

Our research aims to significantly advance the field by introducing state-of-the-art methodologies for detecting DDoS attacks within the complex structure of SDN-based VANETs. Our approach involves harnessing the exceptional capabilities of TabNet, a sophisticated and finely-tuned deep-learning model. By integrating this advanced technology, we aim to reinforce the network's defense mechanisms against DDoS threats substantially. This implementation promises to enhance the resilience and responsiveness of these networks, contributing to the broader landscape of network security and fortifying these critical communication infrastructures against potential cyber threats.

3. Background

In the context of improving the detection system's performance of DDoS attacks in SDN-based VANET, many techniques, such as deep learning methods and optimization algorithms, can be used [46]. For example, a TabNet-based model can be used to process the tabular data related to the network traffic, and a Convolutional Neural Network (CNN) or Deep Neural Network (DNN) can be used to analyze the behavior of the traffic. Finally, the ADAM optimization algorithm can be used to optimize the model weights, reducing the training time and improving the performance.

3.1. TabNet

The TabNet algorithm is a state-of-the-art deep learning method for processing tabular data. It utilizes a multi-step process for predictions and has sparse feature selection, which is performed on a per-instance basis, as well as a sequential multi-step architecture that partially determines each decision based on the selected features [47,48]. The single architecture of TabNet provides efficient feature selection and enhances the learning of high-dimensional features. During each step, a D-dimensional feature vector is processed through a Feature Transformer block, which consists of multiple layers that can be shared or unique to each decision step. The block includes fully connected layers, a batch normalization layer, and a Gated Liner Unit (GLU) activation, with the GLU also connecting to a residual normalization connection to

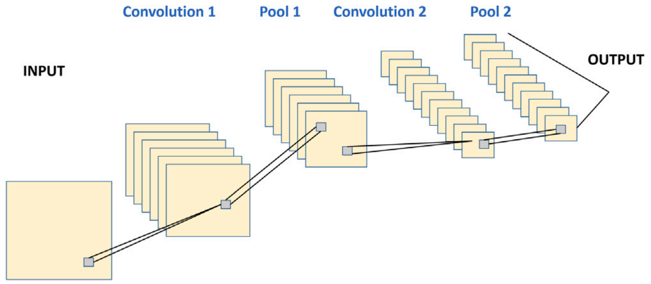


Fig. 2. CNN description.

maintain variance throughout the network. The multiple layers in the Feature Transformer block improve feature selection and the efficiency of the network's parameters.

The split module in TabNet separates the output of the initial feature transformer to create features $a[i-1]$ in the feature selection process's first step ($i = 1$). The attentive transformer in TabNet is made using a trainable function h_i that includes a fully connected layer and a batch normalization layer, generating high-dimensional features by disregarding spatial information. The masks in each step provide interpretable information by selecting features and aggregating the masks from different decision steps can achieve a global interpretation. This technique improves the spectral domain's discriminative ability by implementing local and global interpretability in selecting features. The attentive transformer in TabNet produces masks $M[i]$ as a soft selection of significant features from the processed features $a[i-1]$. The masks are generated in the form of $M[i] \in R^{Bxd}$ as:

$$a[i-1] : M[i] = \text{sparsemax}(P[i-1] * h_i(a[i-1])) \tag{1}$$

Where sparsemax layer is used for coefficient normalization resulting in sparse feature selection, and $P[i-1]$ is the prior scales item obtained as follow:

$$P[i] = \prod_{j=1}^{i} (\gamma - M[j]) \tag{2}$$

The relaxation parameter γ determines how much a feature is enforced at a single decision step or across multiple phases. When $\gamma = 1$, the feature is strongly enforced at the current stage.

TabNet provides both local and global explanations for interpretability. The contribution of each decision step towards the final result for a specific sample can be calculated by summing the output vector of each step and obtaining a scalar, which reflects the importance of that step to the final outcome as follow:

$$\eta_{b[i]} = \sum_{c=d}^{N_d} \text{ReLU}(d_{b,c}[i]) \tag{3}$$

The final decision is obtained by aggregating the output from the Feature Transformers and transforming it through a linear mapping, as described by Eq. (3).

3.2. Convolutional neural network

CNN is an artificial neural network that analyzes and manages information using grid-like data [49]. In this grid, each neuron of a layer only connects to a small number of neurons belonging to the next layer, commonly called the convolutional kernel. As shown in Fig. 2, the CNN applies this kernel to local input patches, resulting in a feature map that summarizes specific patterns or features in the input. The network learns hierarchical representations of the information by repeating this process several times with different kernels. These representations are then passed through clustering layers, which reduce the spatial dimensions of feature maps, and fully connected layers, which perform the final classification.

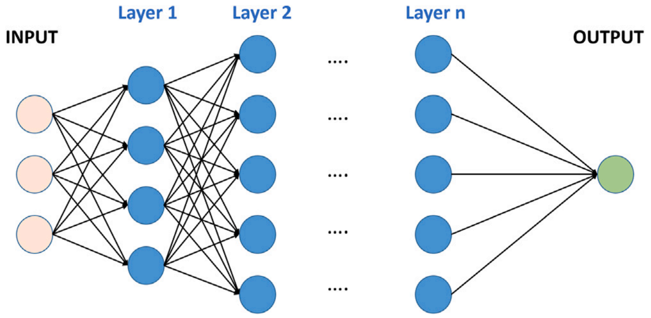


Fig. 3. DNN description.

3.3. Deep neural network

DNNs are a type of artificial neural network consisting of multiple hidden layers, allowing the model to learn complex representations of the input data [50]. This complexity is achieved through the depth of the network, where deeper networks have more layers and can learn more abstract relationships between inputs and outputs, as described in Fig. 3. The layers within the network are interconnected and contain a large number of artificial neurons or nodes, which process and transmit information. These nodes use activation functions to introduce nonlinearity into the network and allow it to model complex nonlinear relationships between the various inputs-outputs. DNNs are trained using large amounts of labeled data and optimization algorithms such as stochastic gradient descent, which helps minimize the prediction error.

3.4. ADAM optimizer

ADAM is a commonly utilized optimization algorithm in the field of deep learning. Adam optimizes models by combining the benefits of Stochastic Gradient Descent (SGD) and Root Mean Square Propagation (RMSProp) [51]. Using moving averages of the gradient and the squared gradient calculates adaptive learning rates for each weight, resulting in faster convergence and less noise compared to SGD. Additionally, ADAM uses bias correction to avoid overestimating the learning rates early in the training phase. This makes ADAM well-suited for training large and complex neural networks, where the convergence can be slow and noisy [9]. Adam's parameter update is given by:

$$m_w{}^{(t+1)} \leftarrow \beta_1 m_w{}^{(t)} + (1-\beta_1)\nabla_w L^{(t)} \tag{4}$$

$$v_w{}^{(t+1)} \leftarrow \beta_2 v_w{}^{(t)} + (1-\beta_1)(\nabla_w L^{(t)})^2 \tag{5}$$

$$\hat{m}_w = \frac{m_w{}^{(t+1)}}{1-\beta_1{}^t} \tag{6}$$

$$\hat{v}_w = \frac{v_w{}^{(t+1)}}{1-\beta_2{}^t} \tag{7}$$

$$m_w{}^{(t+1)} \leftarrow w^t - \eta \frac{\hat{m}_w}{\sqrt{\hat{v}_w} + \varepsilon} \tag{8}$$

$$\left\{ \begin{array}{l} Where \\ w^{(t+1)} : the\ given\ parameters. \\ L^{(t)} : the\ loss\ function. \\ t : indexes\ the\ current\ training\ iteration. \\ \varepsilon : a\ small\ scalar. \\ \beta_1 : the\ forgetting\ factors\ for\ gradients. \\ \beta_2 : the\ second\ moments\ of\ gradients. \end{array} \right.$$

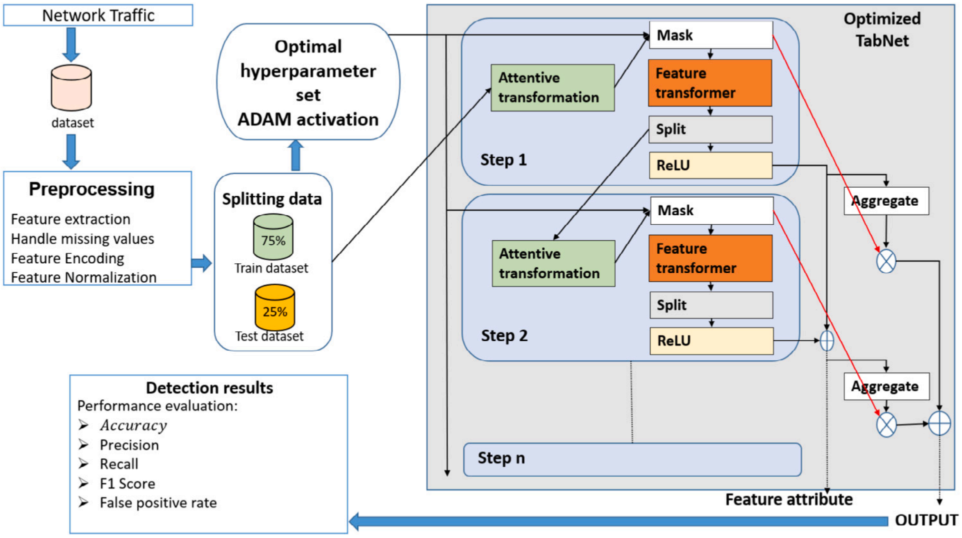


Fig. 4. Schematic of the proposed model.

3.5. Supervised machine learning

Supervised machine learning is a type of machine learning where the algorithms are trained on labeled data, and the goal is to learn a mapping between input features and a specific target output [52]. In supervised learning, the algorithm receives input data along with the corresponding correct output, and the algorithm uses this data to make predictions on new, unseen data. Examples of supervised learning algorithms include Logistic Regression (LR), Support vector machine (SVM), Decision Tree (DT), and Random Forest (RF).

LR is a commonly used machine learning technique in classification and binary problems. The weights are multiplied by the inputs and then sent to the LR sigmoid part [53].

SVM seeks to find the optimum hyperplane to separate data into different classes. It operates based on the maximum margin principle, allowing for nonlinear data separation. In SVM, input data (z) is calculated using vectors (Ψ), weights (v), and a bias (c) [54]. The calculation of SVM is as follows:

$$f(z) = \sin\left(\sum_{i=1}^{N} v_i \Psi(z_i) + c\right) \tag{9}$$

DT uses a tree-like model to make decisions based on potential consequences. It represents a program using only conditional check statements. For a probability distribution of the dataset (p), the root node being determined using information theory [54]:

$$H(x) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{10}$$

Random Forest is a machine learning method that creates multiple decision trees during training for regression and other tasks. The selected class by the majority of the trees represents the output of the Random Forest. The mean or average estimate of the individual trees is returned for regression [53].

4. Proposed methodology

The proposed methodology seeks to deploy cutting-edge technologies and techniques to bolster SDN-based VANETs' resilience against DDoS attacks. This introduction section outlines the key strategies and methods incorporated into the study to fortify network security in the face of these sophisticated and evolving cyber threats. Critical elements discussed in this segment will encompass data collection, model architecture, integration of SDN in VANETs, optimization methodologies using GSCV and ADAM, masking mechanisms, and training/validation protocols. The following sections will delve deeper into each facet, providing a comprehensive understanding of the proposed approach's structure and relevance in enhancing DDoS attack detection within these dynamic vehicular networks. Fig. 4 illustrates a conceptual overview of our model.

4.1. Data preprocessing

In the data preprocessing stage, The raw data is cleaned, filtered, and transformed into an appropriate format for machine learning algorithms. This process involves cleaning and transforming the data, including removing irrelevant or redundant features, handling missing or inconsistent values, and converting categorical data into numerical representations. The feature extraction step involves selecting relevant components from the preprocessed data useful in making predictions. Missing values are handled by identifying, removing, or imputing the missing data in the dataset. The features are then encoded by converting categorical features into numerical representations. Finally, feature normalization scales the features to a standard range, such as between 0 and 1, to ensure that they are on a similar scale and can be treated equally by the machine learning model. This step helps to improve model performance and avoid issues such as skewed weightings or bias towards certain features.

4.2. Optimized TabNet

Our proposed approach uses ADAM optimization to train the TabNet model. This optimization algorithm updates the model parameters based on the gradients of the loss function computed during each iteration's forward and backward pass. The learning rate and other hyperparameters are typically determined through a combination of manual tuning and hyperparameter optimization techniques such as the GSCV algorithm. The proposed Opt-TabNet methodology focuses on the “attentive transformation” concept, which assigns importance to each feature in making predictions. This is achieved by computing attention scores for each feature and weighting them according to these scores before using them. The attention scores are learned during training and are used to give more importance to relevant features and less importance to irrelevant ones. A mask is applied before the attention mechanism to handle missing or unavailable data. Feature transformations are also applied to the input data to capture the interactions between different features and to help the network learn a more robust

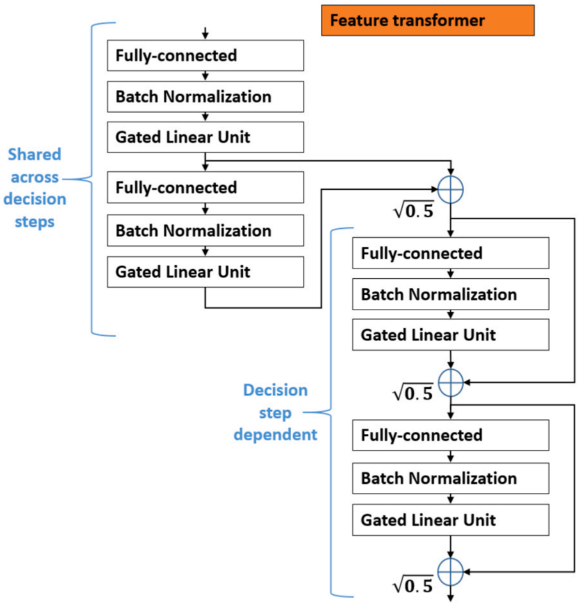


Fig. 5. Feature transformer for Opt-TabNet.

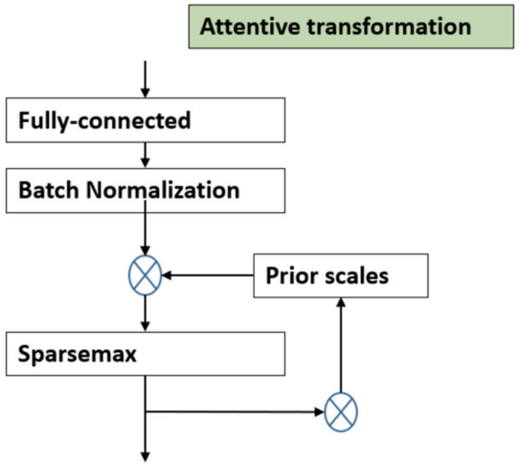


Fig. 6. Attentive transformation.

Table 3
Optimization of our model by trial-and-error approach.

Parameter	Value
Tuning	Optimizer_fn=torch.optim.Adam, n_d=8, n_a=8, n_steps=3, optimizer_params=dict(lr=2e-2), scheduler_params={“step_size”:10, “gamma”:0.9}, mask_type=’entmax’, scheduler_fn=torch.optim.lr_scheduler.StepLR
Learning	Max_epochs=50, patience=15, weights=1, batch_size=256, virtual_batch_size=128, num_workers=0, drop_last=False

representation of the data. The split-and-aggregate mechanism used in TabNet splits the input data into multiple parts and aggregates the results, allowing the network to learn fine-grained data representations and combine these representations into a single output. ReLU activation function is used as it helps the network learn complex relationships between the input and output. The Opt-TabNet handles missing or unavailable data, captures complex relationships between features and targets, and allows the network to learn a robust data representation.

As described in Fig. 5, the feature transformer in TabNet transforms the input features into a more expressive and informative representation. It consists of several layers shared across different decision steps, allowing the same features to be used for different decision-making processes. Additionally, it includes decision step-dependent layers, where the output from the previous decision step influences the features in the current step [55,56]. The feature transformer architecture typically involves concatenating shared layers and decision step-dependent layers. Each fully connected layer is followed by batch normalization and a gated linear unit [57,58]. The gated linear unit activation function enables feature selection and nonlinearity in the model. The normalization process with a factor of $\sqrt{0.5}$ helps stabilize learning throughout the network. Ghost batch normalization is applied to enhance computational efficiency, which involves selecting only a portion of samples rather than using the entire batch at once. This approach utilizes a virtual or small batch size and momentum instead of the full batch, reducing computational costs while maintaining performance. The feature transformer’s output is a compact representation of the input features that captures the relevant information for the subsequent decision-making steps. It enables the model to learn complex patterns and relationships within the tabular data.

As described in Fig. 6, the attentive transformer in TabNet focuses on feature selection and interoperability [59]. The attentive transformer in TabNet employs masks as attention mechanisms to focus on salient features during the decision-making process selectively. These masks play a crucial role in determining the relevance and importance of each feature. The attentive transformer generates masks in a soft selection manner by utilizing processed features from the previous step, as described earlier in Eq. (1). These masks represent the probabilities of selecting each feature, allowing for a flexible and adaptive selection process. The masks are computed using a trainable function, which typically consists of fully connected and batch normalization layers. This function generates high-dimensional features that contribute to the selection process. The generated masks, are used to update the prior in the attentive transformer. This updating process incorporates soft feature selection based on the importance assigned by the masks. By selectively attending to relevant features, the model achieves interpretability and can focus on the most informative aspects of the data.

Algorithm 1 outlines the steps for building the Opt-TabNet detector. It begins by initializing the TabNet model and splitting the data into training and validation sets. Data preprocessing is then applied to prepare the data for model training. The binary cross-entropy loss function and ADAM optimizer are defined, followed by the definition of hyperparameters to be tuned. A GSCV object is created, combining the TabNet model, hyperparameter grid, and scoring metric to find the best hyperparameter configuration. The model is trained using the ADAM optimizer, with updates performed for each epoch and batch in the training data. The model’s performance is evaluated on the validation data, and the average training and validation losses are calculated. Finally, the model’s performance is assessed on the testing data using specified metrics, and the trained Opt-TabNet model is returned as the algorithm’s output.

4.3. Detailed architecture and hyperparameters

The Opt-TabNet model architecture is tailored to address the intricacies of identifying DDoS attacks within SDN-based VANETs. This model is composed of critical hyperparameters and components aimed at effective feature extraction, attention mechanisms, and dynamic learning as described in Table 3.

Algorithm 1 Building the proposed Opt-TabNet detector

```
Input: Training data $(X_{train}, y_{train})$, Validation data $(X_{val}, y_{val})$, Hyperparameter grid $params$, Scoring metric $scoring$
Output: Trained Opt-TabNet model
Initialize the TabNet model architecture
Split the data into training and validation sets
Perform data preprocessing
Define the binary cross-entropy loss function and ADAM optimizer
Define the hyperparameters to tune
Create a GSCV object with the TabNet model, hyperparameter grid, and scoring metric
Fit the GSCV on the training data
Get the best hyperparameters from the GSCV
Initialize the TabNet model with the best hyperparameters
Train the TabNet model with ADAM optimizer:
for each epoch do
    Set the model to training mode
    Initialize the running loss
    for each batch in the training data do
        Perform a forward pass through the model
        Compute the loss using the predicted outputs and the ground truth labels
        Compute the gradients of the loss concerning the model parameters
        Update the model parameters using the ADAM optimizer and the gradients
        Update the running loss
    end for
    Set the model to evaluation mode
    Initialize the validation loss
    for each batch in the validation data do
        Perform a forward pass through the model
        Compute the loss using the predicted outputs and the ground truth labels
        Update the validation loss
    end for
    Calculate the average training and validation loss for the epoch
    Evaluate the model performance on the testing data using the specified metrics.
end for
Return the trained Opt-TabNet model
```

4.3.1. Architectural significance: An automotive approach

Feature Transformers (n_d = 8)

For instance, consider a scenario where the Feature Transformers (n_d = 8) act as the "sensory processing" mechanism, akin to the vehicular sensors within an SDN-based VANET. These transformers meticulously analyze and process various input features such as traffic flow data, communication patterns, and network behavior, akin to how sensors perceive and interpret environmental cues in real-time.

Attention Transformers (n_a = 8)

Simultaneously, the Attention Transformers (n_a = 8) play a role analogous to the cognitive focus of a driver in a VANET scenario. They strategically allocate attention to specific features within the network data, prioritizing critical elements much like a driver prioritizes hazards or essential road signs while navigating through traffic.

Decision Steps (n_steps = 3)

Furthermore, the Decision Steps (n_steps = 3) within each TabNet layer represent the sequential decision-making process, mirroring the multiple layers of interpretation and action in a vehicular network's decision-making hierarchy. This sequential processing enables the model to discern and react to complex patterns and potential threats within the network.

4.3.2. Hyperparameter optimization strategy for vehicle navigation

The optimization strategy involves fine-tuning hyperparameters using the Adam optimizer (Optimizer_fn=torch.optim.Adam). The optimizer's learning rate (lr) is set to 2×10^{-2}, influencing convergence and model performance during training, akin to optimizing vehicle speed to navigate traffic conditions effectively. For example, when facing heavy traffic congestion (analogous to challenging network conditions), the learning rate adjustment helps the model converge faster or slower, resembling how a vehicle adapts its speed in congested or open-road scenarios.

Further enhancing optimization, a StepLR scheduler (scheduler_fn= torch.optim.lr_scheduler.StepLR) dynamically adjusts the learning rate during training. Parameters such as "step_size" and "gamma" (scheduler_params="step_size":10, "gamma":0.9) aid in stabilizing the training process and fine-tuning performance. In an SDN-based VANET context, this scheduler can be metaphorically linked to adaptive cruise control, which automatically adjusts the vehicle's speed based on changing road conditions. For instance, when encountering a sudden change in traffic density, analogous to network variations, the StepLR scheduler fine-tunes the learning rate to ensure stable and efficient training, similar to how adaptive cruise control maintains a consistent and safe speed.

4.3.3. Analogy to vehicle operations

Critical training settings can be likened to preparing a vehicle for a specific road condition or environment. For instance:

- Maximum epochs (Max_epochs=50) can be compared to planning a journey's maximum duration, ensuring that the vehicle does not continue driving indefinitely, analogous to setting a time limit for training.
- Early stopping patience (patience=15) is akin to an alert system in a vehicle that pauses when there is an unexpected roadblock or obstacle, preventing unnecessary continuation in the training process when the model's improvement plateaus.
- Class weights (weights=1) are similar to balancing the load within a vehicle to ensure stability and optimized performance.
- Batch size (256) and virtual batch size (128) relate to the number of passengers the vehicle can comfortably accommodate, optimizing efficiency.
- The worker parameter (0 workers) is analogous to the number of people coordinating and operating the vehicle's controls during the journey.
- Retaining the last batch (drop_last=False) can be compared to ensuring that all available resources or remaining space in the vehicle is utilized effectively before concluding the training process.

In summary, these architectural elements within the Opt-TabNet model can be metaphorically compared to the intricate workings and adaptive responses within an SDN-based VANET. This analogy helps to provide a more tangible understanding of how the model operates within the context of vehicular networks, elucidating its potential to identify DDoS threats effectively.

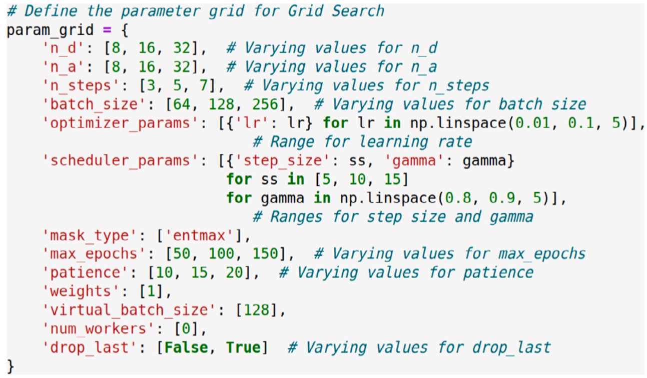
5. Experiments, results, and discussion

In this section, we present the results of our experiments using the Opt-TabNet detector. We evaluate the model's performance and discuss the findings.

5.1. Implementation environment

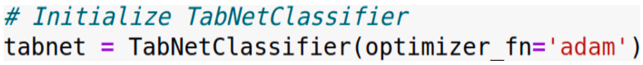
5.1.1. Hardware configuration

The hardware infrastructure employed for the experiments encompassed a 3.6 GHz Intel Core i7 computer equipped with 16 GB of memory. This configuration provided the computational power and memory resources necessary to run the experiments effectively and ensure reliable performance evaluations.



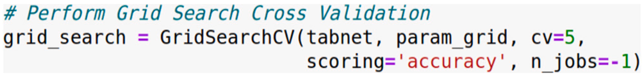
```python
# Define the parameter grid for Grid Search
param_grid = {
    'n_d': [8, 16, 32],  # Varying values for n_d
    'n_a': [8, 16, 32],  # Varying values for n_a
    'n_steps': [3, 5, 7],  # Varying values for n_steps
    'batch_size': [64, 128, 256],  # Varying values for batch size
    'optimizer_params': [{'lr': lr} for lr in np.linspace(0.01, 0.1, 5)],
                            # Range for learning rate
    'scheduler_params': [{'step_size': ss, 'gamma': gamma}
                         for ss in [5, 10, 15]
                         for gamma in np.linspace(0.8, 0.9, 5)],
                            # Ranges for step size and gamma
    'mask_type': ['entmax'],
    'max_epochs': [50, 100, 150],  # Varying values for max_epochs
    'patience': [10, 15, 20],  # Varying values for patience
    'weights': [1],
    'virtual_batch_size': [128],
    'num_workers': [0],
    'drop_last': [False, True]  # Varying values for drop_last
}
```

Fig. 7. GSCV parameters definition.



```python
# Initialize TabNetClassifier
tabnet = TabNetClassifier(optimizer_fn='adam')
```

Fig. 8. Opt-TabNet initialization.



```python
# Perform Grid Search Cross Validation
grid_search = GridSearchCV(tabnet, param_grid, cv=5,
                           scoring='accuracy', n_jobs=-1)
```

Fig. 9. Finding the best model configuration.

5.1.2. Software framework

The experimentation setup was thoughtfully designed for a robust software foundation, comprising a meticulously selected set of software components.

- Operating System: The operating system forming the foundation of the experimentation environment was Ubuntu 20.04. Renowned for its stability and versatility, Ubuntu 20.04 served as the fundamental backbone upon which the entire software stack was built [60].
- Programming Language: Python, at Version 3.7.9 [61], was chosen as the primary programming language for its versatility and extensive libraries, providing an ideal platform for implementing and executing the DDoS detection mechanism.
- Key Software Versions: The experimentation setup was further fortified with specific software versions strategically selected for their pivotal roles:
    - NumPy (Version 1.19.5): NumPy was employed extensively for its comprehensive capabilities in numerical computations and efficient handling of arrays, contributing significantly to data processing and manipulation [61,62].
    - PyTorch (Version 1.9.0): PyTorch, a prominent deep learning framework, was utilized to develop and train neural network models, forming the core engine for our Opt-TabNet detector [61,63].
    - scikit-learn (Version 0.24.2): scikit-learn played a crucial role, offering an extensive toolkit for machine learning algorithms, aiding in data preprocessing, evaluation metrics, and model validation [61,64].

5.1.3. Major libraries utilized

To complement the core software components, the experimentation environment was enriched with the inclusion of the pytorch-tabnet library (Version 3.1.1) [47]. Specifically tailored for implementing the TabNet model, this library provided essential functionalities encapsulated within the TabNetClassifier. This strategic selection of software components and libraries laid the groundwork for a cohesive and robust experimentation environment, facilitating in-depth evaluations of the Opt-TabNet detector's efficacy in detecting DDoS attacks.

5.1.4. Primary code functions and operations

As described in Fig. 7, the parameter grid outlined for Grid Search encompasses a comprehensive range of hyperparameters meticulously chosen for optimizing the Opt-TabNet detector's performance. This parameter grid, designed to explore various configurations, includes essential hyperparameters crucial for the TabNet model's architecture and training dynamics. It incorporates a wide array of values for critical parameters, such as 'n_d', 'n_a', and 'n_steps', denoting the dimensions and steps within the TabNet layers. Additionally, it explores different batch sizes ('batch_size') and learning rates ('optimizer_params') to understand their influence on model convergence and performance.

The 'scheduler_params' grid encompasses diverse combinations of 'step_size' and 'gamma' for the learning rate scheduler, exploring their impact on training stability and convergence.

Furthermore, it involves settings for 'max_epochs', 'patience', 'weights', 'virtual_batch_size', 'num_workers', and 'drop_last', varying these parameters to gauge their effect on model training, regularization, and performance enhancement. The 'mask_type' parameter remains constant, employing the 'entmax' variant, focusing on its impact on attention mechanisms.

Overall, this exhaustive grid is crafted to comprehensively explore the hyperparameter space, aiding in identifying the optimal configuration for maximizing the Opt-TabNet detector's efficacy in DDoS attack detection. The fine-tuned optimization results are detailed in Table 3 for comprehensive insights.

Fig. 8 initializes a TabNetClassifier, an instance of the TabNet model used for classification tasks. Within the parentheses, the 'optimizer_fn' parameter is specified as 'adam', indicating the choice of the Adam optimizer to be used during the training of this TabNetClassifier model.

The code in Fig. 9 executes GridSearchCV to find the optimal configuration for the TabNetClassifier model by systematically exploring various hyperparameter combinations. It splits the dataset into 5 folds for cross-validation ('cv=5') and evaluates the model's accuracy using different parameter sets. The 'n_jobs=-1' parameter maximizes CPU usage, accelerating the grid search by parallelizing computations. The resulting 'grid_search' object holds the optimized parameters, facilitating model fitting, parameter identification, and performance evaluation.

The source code utilized for implementing OptTabNet can be accessed and reviewed through the provided Ref. [65]. This comprehensive code repository includes the implementation details of OptTabNet and encompasses the necessary scripts, modules, and configurations required for the successful deployment and utilization of the model. Researchers, practitioners, and enthusiasts can benefit from exploring this repository to gain deeper insights into the architecture, optimization strategies, and overall functionality of OptTabNet. Additionally, the availability of the source code fosters transparency and reproducibility, allowing stakeholders to replicate experiments, validate results, and further contribute to the advancement of DDoS detection methodologies in SDN-based VANETs.

5.1.5. Datasets description

The dataset containing network traffic data records is accessed and imported into the Python environment using the Comma-Separated Values (CSV) file format. This format is a standard file type used to store tabular data where each line represents commas or other delimiters separate from a record and columns within the file.

The IEEE DataPort (SDN-DDoS) dataset [31] is utilized for testing and evaluating the performance of the DDoS detection system in SDN-based VANETs. This dataset was created to evaluate the effectiveness of DDoS attacks on SDN and IoT networks and was generated using the

Mininet network emulator. The simulation captures normal and malicious network traffic, including TCP, UDP, ICMP traffic, and various DDoS attacks like TCP Syn, UDP Flood, and ICMP attacks. The resulting dataset comprises 104,335 samples, including 40,784 malicious and 63,561 benign traffic. The features used in the analysis are 22 in total and include various network traffic characteristics. The dependent variable is a label indicating the presence or absence of a DDoS attack.

To gauge the adaptability and robustness of Opt-TabNet across diverse scenarios, we aim to leverage the InSDN dataset as an additional benchmark. The InSDN dataset, introduced by Elsayed et al. in their research [32], is purpose-built for studying intrusion detection within SDN environments. This curated dataset incorporates four virtual machines and spans a broad spectrum of attack classes from internal and external sources within the SDN network. Notably, it also encompasses representations of normal or benign network traffic like HTTPS, HTTP, DNS, Email, FTP, and SSH, reflecting the multifaceted nature of various application services. The InSDN dataset employs a virtualized setup using four virtual machines, each representing different elements of an SDN network. This setup includes an attacker server, an SDN controller, and various network hosts and switches, thereby creating a controlled yet complex and diverse SDN environment. With a substantial collection of 361,317 instances, this dataset comprises 68,424 instances depicting typical network activity and 292,893 instances simulating a range of attack scenarios. Featuring an array of 84 distinct features, the dataset offers a comprehensive panorama of network traffic attributes and trends within SDN setups.

Both datasets are generated using virtualized environments, with the SDN-DDoS dataset using the Ryu SDN controller and the InSDN dataset using the ONOS (Open Network Operating System) controller. The virtualized environments include multiple virtual machines representing different components of the SDN network, such as switches, controllers, and hosts. The communication between virtual hosts is facilitated using L3 switching connectivity. Additionally, Mininet is used to create a realistic virtual network with virtual switches, hosts, and links, enabling the generation of both legitimate and malicious network traffic.

5.2. Metrics for evaluating the models

Assessing the proposed DDoS attack detection method's effectiveness is crucial in determining its performance and accuracy. Typically, the following evaluation metrics are employed in the field to gauge the performance of classification algorithms. These metrics are based on four elements: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN), such as accuracy, precision, recall, and F1-Score :

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{14}$$

The loss function used in the TabNet model, especially for binary classification problems like detecting DDoS attacks, is the binary cross-entropy loss. Mathematically, the binary cross-entropy loss is formulated as follows:

$$\text{Loss}(y, \hat{y}) = -y \cdot \log(\hat{y}) - (1 - y) \cdot \log(1 - \hat{y}) \tag{15}$$

In this equation:

* y represents the true binary label (0 or 1) indicating the actual class (0 for normal traffic, 1 for DDoS attacks).

* $\hat{y}$ represents the predicted probability that the data point belongs to class 1 (indicating a DDoS attack).

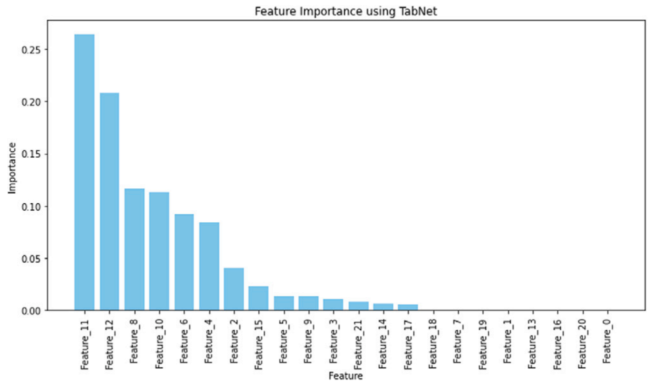


Fig. 10. Feature selection results using Opt-TabNet in SDN-DDoS dataset.

The binary cross-entropy loss, as defined in Eq. (15), measures the discrepancy between the predicted probability $\hat{y}$ and the actual label y. This loss function is commonly used for binary classification problems and is crucial in training some machine learning models to predict DDoS attacks in SDN-based VANETs accurately.

5.3. Implementation process

OptTabNet incorporates several optimization techniques to enhance its performance in detecting DDoS attacks. Optimization involves careful tuning of hyperparameters, dynamic learning rate adjustment, and extensive evaluations. Here is a more detailed clarification of the optimization techniques used in OptTabNet:

5.3.1. Hyperparameter tuning

OptTabNet fine-tunes its architecture by systematically searching for the optimal set of hyperparameters, such as the number of decision steps (n_d), the number of attention blocks (n_a), and the type of masking (mask_type). This tuning process helps optimize the model's architecture for improved accuracy and efficiency.

5.3.2. Dynamic learning rate adjustment

OptTabNet dynamically adapts its learning rate during training using the ADAM optimizer, accelerating convergence and elevating overall model performance. By adjusting the learning rate based on the model's performance on the training data, OptTabNet can effectively navigate the optimization landscape and avoid issues like slow convergence or overshooting.

5.3.3. Sparse attention mechanisms

OptTabNet leverages sparse attention mechanisms to focus on relevant features while disregarding irrelevant ones. This attention mechanism enables the model to attend to only the most informative features in the input data, improving its ability to detect DDoS attacks while reducing computational overhead.

5.3.4. Feature selection

OptTabNet incorporates feature selection techniques to identify the most discriminative features for detecting DDoS attacks. By selecting only the most relevant features, the model can improve its accuracy and efficiency in distinguishing between normal network traffic and DDoS attacks.

Fig. 10 illustrates the relevance of each feature in detecting DDoS attacks using the SDN-DDoS dataset. It provides insights into the discriminative power of different features and their contribution to the overall classification performance. The figure typically displays a horizontal bar chart or a heatmap, where each feature is represented along the vertical axis, and the corresponding importance or relevance score

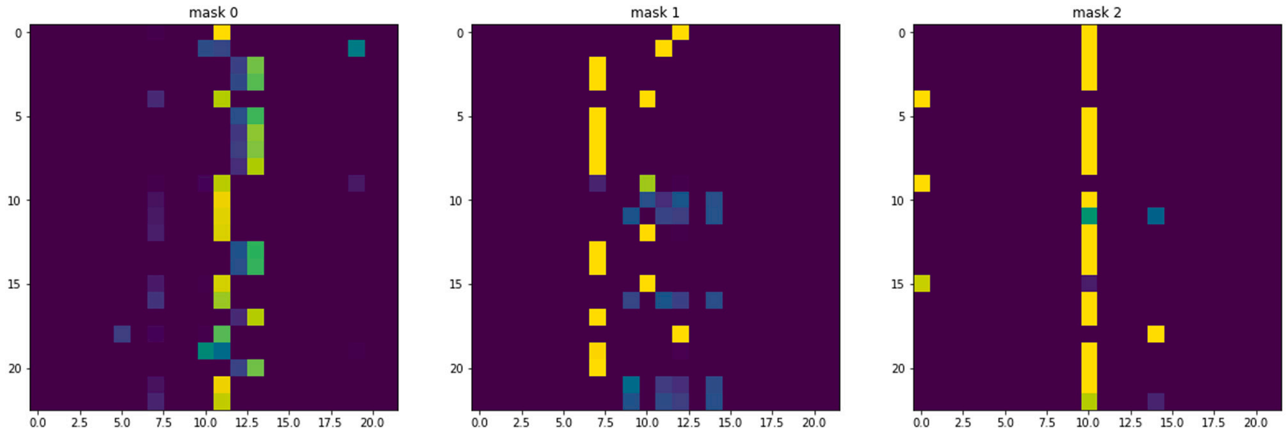


Fig. 11. Opt-TabNet mask using SDN-DDoS dataset.

Table 4
Evaluation of Opt-TabNet using SDN-DDoS dataset.

Metric	Value
best Accuracy	99.4194658448895%
Precision	98.5045999352121%
Recall	99.9767055729081%
F1-score	99.2351935623166%

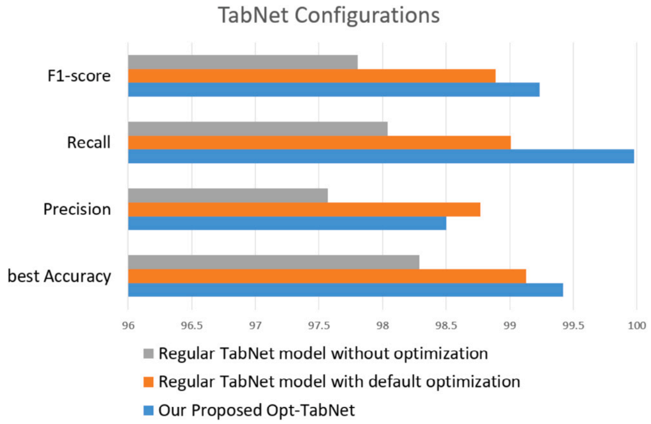


Fig. 12. Diverse TabNet configurations result using SDN-DDoS dataset.

is shown on the horizontal axis. The length or color intensity of the bars or cells indicates the magnitude of importance, with longer bars or darker cells representing more influential features. By examining the figure, one can identify the top-ranked features with the highest impact distinguishing between normal traffic and DDoS attacks. These features are considered the most informative for accurately classifying network traffic.

This visualization aids in understanding the underlying patterns and characteristics of DDoS attacks, enabling researchers and practitioners to gain insights into the critical features and design more effective detection systems. It also helps in feature selection or dimensionality reduction, as less relevant or redundant features can be eliminated based on their low importance scores, thereby simplifying the detection model and reducing computational complexity.

5.3.5. Regularization techniques

OptTabNet utilizes regularization techniques such as dropout and L2 regularization to prevent overfitting and improve generalization performance. These techniques help the model generalize well to unseen data and mitigate the risk of memorizing noise in the training data, enhancing its robustness against DDoS attacks.

5.4. Opt-TabNet detector

As shown in Fig. 11 and described in Fig. 4, we use masks to handle missing data: mask0, mask1, and mask2. Mask0 masks the missing values in the input data, which is applied before feeding the data into the network. Mask1 is involved in the attentive transformation to the attention scores, allowing the network to assign more importance to the relevant features and less significance to the irrelevant ones. In the split-and-aggregate phase, the input data is divided into multiple parts and combined to form the final output. Before the split operation, Mask2 is applied to the data, ensuring that the network considers only the available data. This step reduces the risk of overfitting. From the above, Opt-TabNet can deal effectively with missing or unavailable data, allowing it to learn a more robust representation of the data.

The Opt-TabNet was evaluated on the SDN-DDoS dataset, and its performance was measured using various evaluation metrics such as accuracy, precision, recall, and F1-score as described in Table 4. The best accuracy score obtained was 99.42%, which indicates that the model correctly predicted the class of 99.42% of the data samples in the dataset. Precision, which measures the number of true positive results among the positive predictions, was 98.50%. Recall, which measures the number of true positive results compared to the actual number of positive samples, was 99.98%. Finally, the F1-score, which is the harmonic mean of precision and recall, was 99.24%. These results indicate that the Opt-TabNet model has high accuracy and good performance in detecting DDoS attacks.

5.5. Comparative analysis

5.5.1. Comparative analysis based on experiments

(a) Diverse TabNet Configurations:

The evaluation results depicted in Fig. 12 offer a comparative analysis between our proposed Opt-TabNet algorithm and a range of TabNet configurations applied to a specific dataset. These configurations include the standard TabNet deep learning model without any optimization and a version with default optimization settings. The results illustrate the superiority of the Opt-TabNet algorithm in terms of accuracy over other configurations, signifying its increased significance. This outcome highlights the potential additional value the proposed Opt-TabNet approach brings, underscoring the need to delve deeper into its specific advantages and attributes for further development.

(b) Opt-TabNet and supervised machine learning:

Fig. 13 shows the evaluation results of our proposed Opt-TabNet and four different supervised machine learning algorithms on a given

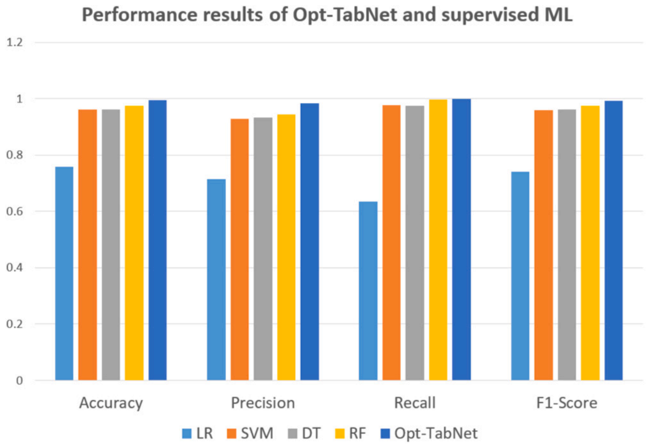


Fig. 13. Opt-TabNet and supervised machine learning results using SDN-DDoS dataset.

dataset. These algorithms are LR, SVM, DT, and RF. The evaluation metrics used are Accuracy, Precision, Recall, and F1-Score. The values in the columns show the average performance of the algorithms across different runs. The results demonstrate that the Opt-TabNet algorithm has the highest importance in all evaluation metrics compared to the other algorithms. This indicates that the Opt-TabNet algorithm has the best overall accuracy, precision, recall, and F1-Score performance.

(c) Opt-TabNet and deep learning models:

The figure depicted in Fig. 14 illustrates the accuracy results obtained after 50 epochs of testing using Opt-TabNet, CNN, and DNN algorithms. The accuracy is calculated by comparing the predicted values to the actual labels and determining the percentage of correct predictions. The graph presents the performance of the proposed Opt-TabNet model. In this same context relating to deep learning, the loss function measures the difference between the model predictions and the actual target values. It helps the model understand how to perform and provides tips for updating weights and biases during training. Fig. 15 shows the loss function of different deep learning methods. The Opt-TabNet model loss combines sparse attention loss and prediction loss. In the CNN and DNN models, the loss function is cross-entropy. In summary, the loss functions of the TabNet, CNN, and DNN models may differ, but we can observe that the loss is minimized in the three cases, especially with Opt-TabNet, which improves their performance.

(d) Experiments summary:

The radar plot in Fig. 16 shows the accuracy of different machine learning models applied to the SDN-DDoS dataset. The accuracy of the models is expressed as a percentage. The Opt-TabNet model has an accuracy of 99.42%, while the regular TabNet model has an accuracy of 98.29%. The DNN model has an accuracy of 98.72%, while the CNN model has an accuracy of 95.74%. The accuracy of the LR model is 75.94%, while the accuracy of the SVM model is 96.23%, and the accuracy of the DT model is 96.28%. The accuracy of the RF model is 97.58%.

5.5.2. Comparative analysis with existing models

The comparative analysis with existing models, as presented in Table 5, provides a detailed examination of the proposed model's performance in relation to several state-of-the-art techniques for DDoS attack detection. Each referenced model is scrutinized based on its year of publication, underlying technique, and achieved accuracy.

In the realm of DDoS attack detection, FTODefender, proposed by Tang et al. [33] in 2024, combines eviction and cut-off attack sources, yielding an accuracy of 97.59%. This technique showcases a competitive performance in comparison to traditional methods. Rao et al.'s framework [34], introduced in the same year, employs SVM, MLP, and LSTM, achieving an accuracy of 97.80%. While effective, this approach slightly trails behind the performance of the proposed model.

Table 5
Performance metrics comparison of the proposed model with existing models.

Ref	Year	Technique	Accuracy
[33]	2024	Combining eviction and cut-off attack sources	97.59%
[34]	2024	SVM, MLP and LSTM	97.80%
[38]	2023	RBF-SVM with hyperparameter tuning using GSCV	99.40%
[35]	2023	GBT, LR, MLP, RF and SVM	96%
[36]	2023	Hybrid PSO-BAT Optimization based on Modified Chaotic Cellular Neural Network	98%
[37]	2023	DoSRT based on Direct trust and Indirect Trust	87.36%
[40]	2022	DT with Bayesian optimization	99.35%
[42]	2022	RBF-SVM	99.40%
[45]	2020	Hybrid based on KNN SVM	96.90%
Proposed method	**2024**	**Optimization of TabNet**	**99.42%**

Anyanwu et al.'s model [38], which utilizes RBF-SVM with hyperparameter tuning using GSCV, demonstrates a robust accuracy of 99.40%. This represents a commendable benchmark within the landscape of DDoS detection models. Rashid et al. [35] present an adaptive approach in 2023, employing GBT, LR, MLP, RF, and SVM techniques, achieving an accuracy of 96%. Although proficient, it falls short of the proposed model's accuracy.

Jebur et al. [36] introduce a Hybrid PSO-BAT Optimization based on Modified Chaotic Cellular Neural Network in 2023, achieving a noteworthy accuracy of 98%. Keshari et al. [37], also in 2023, propose a DoSRT based on Direct trust and Indirect Trust, with an accuracy of 87.36%. While innovative, this approach exhibits a lower accuracy compared to the proposed model.

Turkouglu et al. [40] and Anyanwu et al. [42] present models utilizing DT with Bayesian optimization and RBF-SVM, achieving accuracies of 99.35% and 99.40%, respectively. These models perform exceptionally well, yet the proposed OptTabNet surpasses them marginally with an accuracy of 99.42%.

The Hybrid approach based on KNN SVM by Polat et al. [45] achieves an accuracy of 96.90%. Finally, the proposed method, Optimization of TabNet, emerges as the front-runner in terms of accuracy, boasting an impressive 99.42%.

The OptTabNet's superior performance highlights its effectiveness in DDoS attack detection. This heightened accuracy demonstrates the model's robustness and sophistication in identifying complex attack patterns. The individual components within the OptTabNet, such as the attention mechanisms and feature selection, contribute synergistically, enabling a comprehensive understanding of network behavior and more accurate threat identification.

This analysis underlines the OptTabNet model's capability to outperform other methods, showcasing its potential as an efficient DDoS detector system within SDN-based VANETs. The model's accuracy, attributed to its components and synergistic effect, signifies OptTabNet's advancements in enhancing network security and addressing the complexities of modern-day DDoS attacks within vehicular environments.

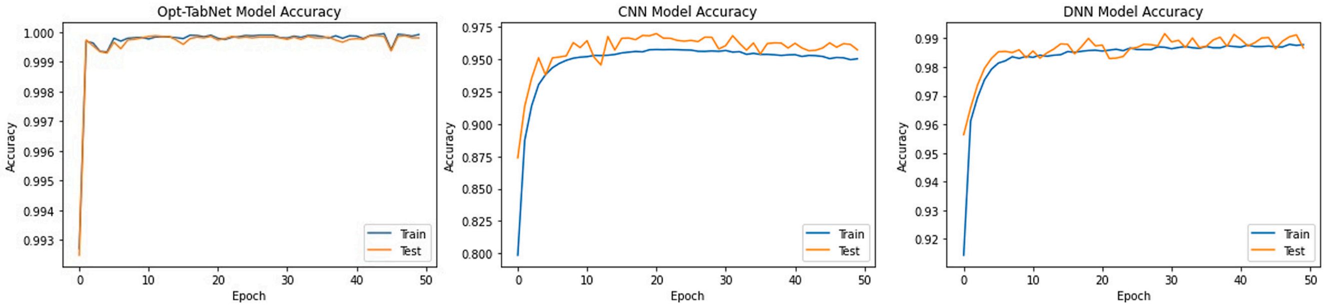


Fig. 14. Accuracy in deep learning using SDN-DDoS dataset.

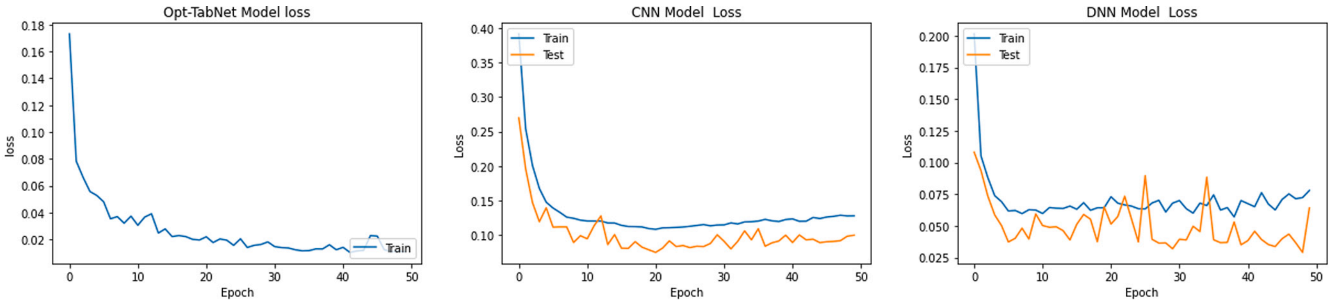


Fig. 15. Loss in deep learning using SDN-DDoS dataset.

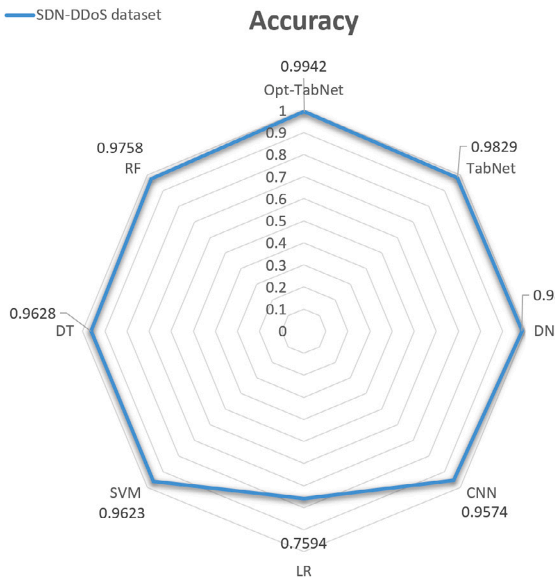


Fig. 16. Radar plots of accuracy using SDN-DDoS dataset.

5.6. OptTabNet evaluation across datasets in SDN-VANETs

The assessment of machine learning models on various datasets is essential to comprehend their adaptability across different scenarios and ascertain their robustness. In SDN-based VANETs, such evaluations are pivotal due to network traffic scenarios' complexities and diverse nature. The OptTabNet model, theoricly recognized for its efficacy in DDoS attack detection, necessitates validation on multiple datasets to verify its generalizability.

While it has shown promising results on the SDN-DDoS dataset, assessing its performance on distinct real-world datasets reflecting high network traffic scenarios common in SDN-based VANETs becomes imperative. However, suitable SDN datasets, especially large-scale real-world ones, remain limited. Hence, selecting the InSDN dataset for secondary experiments holds significant merit.

Table 6
Evaluation of Opt-TabNet using InSDN dataset.

Metric	Value
best Accuracy	99.4115055594642%
Precision	99.1478134039119%
Recall	99.38397851317 05%
F1-score	99.2657554924771%

The InSDN dataset, chosen purposefully for its characteristics mirroring high network traffic instances, is an apt choice to test further and validate the OptTabNet model's adaptability and performance in diversified SDN environments. This choice addresses the necessity for evaluating the model's efficacy beyond a single dataset and fills the gap in available SDN datasets by providing an opportunity to scrutinize the model's behavior under varied real-world traffic scenarios.

Table 6 summarizes the performance metrics achieved by the Opt-TabNet model when evaluated on the InSDN dataset. It presents key metrics for assessing the model's effectiveness in detecting DDoS attacks. The metrics include the best accuracy attained, which reaches 99.41%, indicating the proportion of correctly predicted instances. The precision stands at 99.15%, signifying the accuracy of positive predictions made by the model. Additionally, the model demonstrates a recall of 99.38%, indicating its ability to identify actual positive instances correctly. The F1-score, representing the harmonic mean of precision and recall, registers at 99.27%, portraying a balanced measure between the model's precision and recall.

The comparison between the two experiments, where the OptTabNet model was tested using the SDN-DDoS dataset and the larger InSDN dataset, showcases remarkable stability in performance across the two datasets. Both experiments demonstrate high accuracy, precision, recall, and F1-score values as shown in Fig. 17. Despite the shift to a larger dataset, the overall performance metrics remain consistently high, indicating the stability and robustness of the OptTabNet model. The minor fluctuations observed in the metrics between the two datasets are marginal and within an acceptable range.

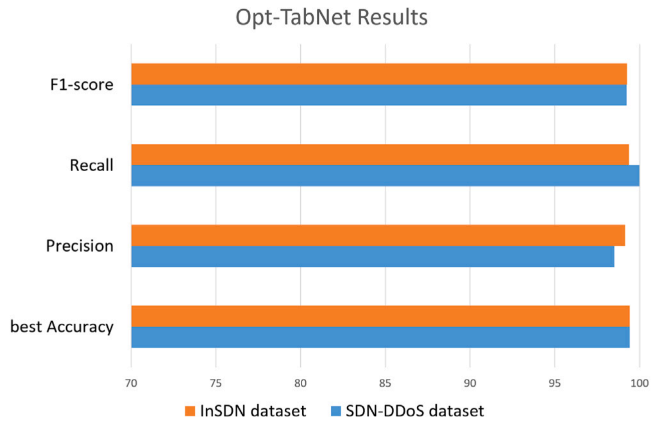


Fig. 17. Opt-TabNet across two datasets.

The results indicate a consistent and stable performance across both datasets, reaffirming the model's capability to maintain high accuracy, precision, recall, and F1-score values, even with variations in dataset sizes. These findings underscore OptTabNet's reliability, generalizability, and consistent performance across diverse SDN-based VANET scenarios, highlighting its possible effectiveness in detecting DDoS attacks amidst varied real-world traffic scenarios.

5.7. Discussion

In this section, we delve into the multifaceted aspects of our research, highlighting the efficacy of the Optimized TabNet in detecting DDoS attacks, extending its application beyond VANETs, and emphasizing its significant contribution to advancing network security.

5.7.1. Enhanced efficacy in DDoS detection with optimized TabNet

The optimization journey commenced with an in-depth examination of critical hyperparameters governing the TabNet model's behavior. Parameters such as n_d, n_a, n_steps, and mask_type were systematically fine-tuned to align the model with the intricacies of SDN-VANETs. The results of this hyperparameter optimization were illuminating, revealing the profound influence these settings have on the model's overall performance. Furthermore, a dynamic learning rate optimization strategy was employed, harnessing the power of a step scheduler. This adaptive approach allowed the model to adjust its learning rate dynamically in response to evolving network conditions. The significance of this dynamic adjustment mechanism cannot be overstated, as it played a pivotal role in enhancing the model's convergence and overall performance. The ultimate litmus test for the optimized TabNet was its effectiveness in detecting DDoS attacks. Through the rigorous optimization process, the model exhibited a newfound capability to identify the presence of DDoS attacks within the dynamic landscape of SDN-VANETs.

To firmly establish and validate the enhanced efficacy of the optimized TabNet, a comprehensive evaluation was conducted across multiple dimensions:

- Diverse TabNet Configurations: The performance of the Opt-TabNet algorithm was systematically compared against that of standard TabNet configurations. The results resoundingly underscored the algorithm's superiority in terms of accuracy. This solidified its potential to contribute substantially to the DDoS attack detection field within SDN-VANETs.
- Opt-TabNet and Supervised Machine Learning: A meticulous evaluation pitted Opt-TabNet against four distinct supervised machine learning algorithms, including LR, SVM, DT, and RF. Across all critical evaluation metrics, including accuracy, precision, recall, and F1-Score, Opt-TabNet consistently emerged as the top performer, cementing its position as the premier choice for DDoS detection.
- Opt-TabNet and Deep Learning Models: The model's prowess in deep learning was tested by meticulously comparing its accuracy and loss to traditional deep learning algorithms such as CNN and DNN. The results were nothing short of striking, with Opt-TabNet showcasing marked improvements in accuracy and minimized loss. This reaffirmed its effectiveness in deep learning for DDoS attack detection.
- Comparative Analysis with Existing Models: A comprehensive comparative analysis was conducted, positioning the proposed Opt-TabNet model against recently developed models. The outcome was unequivocal, Opt-TabNet emerged as the top-performing model, achieving an impressive accuracy rate of 99.42%. This compelling result validated its effectiveness and highlighted its substantial contributions to enhancing network security.
- Evaluation Across Datasets: The evaluation of OptTabNet extended beyond the confines of the initial SDN-DDoS dataset to include the InSDN dataset—a distinct dataset designed to mirror high network traffic scenarios prevalent in SDN-based VANETs. What emerged from this extended evaluation was a clear demonstration of Opt-TabNet's remarkable stability and consistency in performance. Across both datasets, it maintained consistently high accuracy, precision, recall, and F1-score values, showcasing its robustness and generalizability.

Optimizing the TabNet model has ushered in a transformative era of efficacy in DDoS detection within the challenging domain of SDN-VANETs. The enhanced OptTabNet model has firmly established itself as a robust and potent solution for detecting DDoS attacks through meticulous hyperparameter tuning, dynamic learning rate optimization, and comprehensive evaluations. Its capacity to outshine existing models and sustain exceptional performance across diverse datasets is a testament to its potential to elevate network security across a spectrum of scenarios significantly.

5.7.2. Performance under varying network conditions and attack scenarios

The OptTabNet model, designed for efficient DDoS attack detection in diverse network environments, was rigorously tested using datasets that represent a wide range of network conditions and attack scenarios. This evaluation aimed to assess the model's adaptability, accuracy, and scalability in varying contexts.

- Diverse TabNet Configurations: Performance under Varying Network Conditions and Attack Scenarios The OptTabNet model, designed for efficient DDoS attack detection in diverse network environments, was rigorously tested using datasets that represent a wide range of network conditions and attack scenarios. This evaluation aimed to assess the model's adaptability, accuracy, and scalability in varying contexts.
- Adaptability to Different Network Topologies: The datasets used, including SDN-DDoS and InSDN, encompassed multiple network topologies, ranging from simple point-to-point configurations to complex multi-layered architectures. OptTabNet was particularly effective in simpler setups, where its architecture, leveraging a decision-making process guided by n_steps, which represent the number of steps in the decision-making process, efficiently identified attack patterns.
- Handling Diverse Traffic Patterns: Both datasets presented a mix of traffic types — from standard TCP/UDP flows to more irregular patterns often used in sophisticated cyberattacks. OptTabNet's performance was robust in environments with predictable traffic, where its feature selection process, governed by n_d and n_a, effectively discriminated between benign and malicious traffic.

- Response to Various Attack Scenarios: In terms of attack detection, OptTabNet excelled in identifying conventional DDoS attacks, such as TCP SYN floods, which are well-represented in the datasets. The model's dynamic learning rate optimization, enabled by a sophisticated scheduler, allowed for quick adaptation to these attack types.
- Scalability and Efficiency in High-Traffic Environments: One of the key strengths of OptTabNet was its scalability, a crucial factor given the diverse network sizes and loads represented in the datasets. The model maintained high levels of detection accuracy even as network load and complexity increased, without a corresponding surge in computational demand. This scalability is attributed to the efficient architecture of TabNet, which balances computational complexity with predictive power.
- Generalization Across Datasets: A notable aspect of OptTabNet's performance was its generalization capability. The model consistently achieved high accuracy rates across the SDN-DDoS and InSDN datasets, which simulate different network environments and attack types. This demonstrates its robustness and adaptability to various real-world network scenarios, a critical requirement for any practical network security solution.

5.7.3. Broadening the scope beyond VANETs

Beyond its application in VANETs, the Optimized TabNet model holds promise for broader use in diverse network environments. The model's ability to discern intricate patterns in network traffic data and its adaptability to dynamic scenarios position it as a valuable tool for enhancing security in various network architectures. The extensibility of our approach to conventional wired networks, wireless sensor networks, and beyond demonstrates its potential to contribute to the overarching field of network security.

5.7.4. Contribution to the advancement of network security

Our research significantly contributes to advancing network security by introducing an innovative and efficient approach to DDoS detection. The application of deep learning, specifically the Optimized TabNet, serves as a paradigm shift in enhancing the precision and effectiveness of security measures. This contribution goes beyond immediate applications, laying a foundation for more robust and adaptive network security protocols.

5.7.5. Real-world deployment challenges

In deploying our optimized TabNet model for DDoS attack detection in SDN-based VANET environments, several real-world deployment challenges need to be addressed to ensure the model's effectiveness and sustainability.

Computational overhead and scalability challenges: The computational overhead involves a comprehensive evaluation encompassing the processing power, memory, and storage capacities required for the model to function optimally. This evaluation is critical as it dictates the hardware and software dependencies and identifies potential limitations. A particular focus is placed on the model's algorithmic efficiency and any optimizations incorporated to enhance its performance while minimizing resource consumption.

Resource requirements are a fundamental aspect of this evaluation. We need to clearly specify the necessary CPU power, RAM, and storage to ensure the model operates effectively. Additionally, the efficiency considerations, including time complexity (how computation time increases with input size) and space complexity (how memory usage scales with input size), are vital for gauging the model's efficiency. The algorithmic efficiency and any optimizations employed are also crucial components of this assessment, ensuring the model's performance is optimized for the intended environment.

Scalability is another crucial factor, particularly in the context of VANETs, which are characterized by large, diverse datasets and dynamic network conditions. Additionally, integrating the model into existing network infrastructures requires seamless incorporation without disrupting current systems or creating performance issues, necessitating technical compatibility and alignment with existing network protocols and security policies.

- Handling large and diverse datasets: The model must be capable of efficiently processing vast amounts of data that are characteristic of VANET environments.
- Adaptability to dynamic network conditions: The model should be able to adapt to changes in network topology and traffic patterns without a significant loss in performance.
- Potential limitations: Awareness of potential scalability limitations and challenges is vital for ensuring long-term effectiveness.

Trade-offs, practical viability, and maintenance challenges: The implementation of the TabNet model involves navigating various trade-offs, particularly between accuracy, speed, and resource consumption. Achieving higher accuracy might necessitate increased computational resources, potentially leading to slower processing times. Conversely, prioritizing faster inference could require compromises in accuracy or an increase in memory usage. These trade-offs are integral to the model's practical viability and must be meticulously evaluated in real-world scenarios. We consider evaluation metrics such as inference time per sample, throughput, memory footprint, and RAM consumption to understand the model's performance relative to dataset size and complexity.

Maintaining the model over time presents its own set of challenges, which must be addressed to ensure sustained effectiveness and compliance.

- Regular dpdates and learning: The model must be regularly updated to recognize new DDoS attack patterns.
- Software updates and compliance: Ensuring the model and associated systems are up-to-date and compliant with relevant standards and regulations is crucial.
- Managing False Positives/Negatives: Continuous monitoring and adjustment are required to minimize erroneous detections.
- User privacy and data security: Maintenance protocols must ensure adherence to privacy laws and data security standards.

From above, understanding and addressing these real-world deployment challenges is crucial for successfully implementing the optimized TabNet model in SDN-based VANET environments. A thorough consideration of computational overhead, scalability, trade-offs, and maintenance challenges will give stakeholders the insights to make informed decisions, ensuring the model's deployment is effective and aligned with specific network environments and operational needs.

5.7.6. Limitations

Our study has provided valuable insights into detecting DDoS attacks in SDN-based VANETs using the Optimized TabNet model, but it is important to acknowledge some limitations. First, the datasets used in our study, including the SDN-DDoS and InSDN datasets, may not fully represent the diversity of network environments and attack scenarios encountered in real-world deployments. These datasets were generated under controlled conditions, which may not fully capture the complexities and variability of network traffic and attack patterns. Our reliance on specific SDN controllers, namely Ryu and ONOS, for implementing the testbed and datasets introduces a level of controller dependency. While these controllers are commonly used in SDN research, they represent only a subset of the diverse range of SDN controller platforms available. Different controllers may have unique functionalities and security models, potentially affecting the SDN behavior and adversarial scenarios to assess the robustness and scalability.

Additionally, our study primarily focused on evaluating the model's performance under benign and known attack scenarios, while adversarial testing and the computational complexity, which involves probing the model's vulnerabilities using adversarial examples or attacks, were not extensively conducted. Future research should incorporate adversarial testing and computational complexity to assess the model's resilience against sophisticated attacks and evasion techniques without compromising detection performance. Finally, while our research contributes to the theoretical understanding of DDoS detection in SDN-based VANETs, real-world deployment of detection systems may encounter additional challenges related to network scalability and integration with existing infrastructure.

In summary, while our study provides valuable insights into DDoS detection in SDN-based VANETs, it is important to recognize these limitations and consider them in interpreting our findings and designing future research efforts.

5.7.7. Best practices

A collaborative detection system, where multiple SDN controllers work together, can enhance scalability and distribute the computational load effectively. This approach aims to optimize performance and address challenges related to the vast network size and diversity inherent in our proposed DDoS detection model. Continuous security audits and performance assessments are advocated as crucial practices to identify areas in need of maintenance or improvement, contributing to the overall robustness and reliability of the model. Furthermore, active engagement with the broader community and utilizing open-source solutions are emphasized as essential practices for continuously enhancing and updating the DDoS detection model. This collaborative approach ensures the model's adaptability and effectiveness in the face of evolving security challenges.

In summary, this section thoroughly analyzes the Optimized TabNet's capabilities and its implications for network security. By addressing these key areas, we aim to offer a comprehensive understanding of the model's potential, applications, and deployment challenges, guiding stakeholders in making informed decisions tailored to their specific network environments and operational needs.

6. Conclusion and future work

Integrating VANETs into transport systems aims to improve road safety and driver comfort, which presumably involves challenges related to managing and scaling VANETs. SDN has been suggested as a potential remedy for specific issues. However, it also brings about novel security vulnerabilities, including DDoS attacks, which can seriously jeopardize the security of SDN. In this paper, we proposed a deep learning method based on optimizing TabNet to detect DDoS attacks in SDN-based VANET. This optimization considered tuning hyperparameters and the ADAM technique training step. The experiments conducted on the SDN-DDoS dataset showed that the proposed method outperforms other existing methods in terms of accuracy. The optimization process improved the accuracy of the optimized TabNet model from 98.29% to 99.42%, making it one of the best-performing models for detecting DDoS attacks in SDN-based VANETs. The results of the proposed approach suggest that the optimized TabNet algorithm effectively detects DDoS attacks in SDN-based VANET environments.

Future research directions should aim to enhance the generalizability of the proposed methodology across diverse network settings. Validating the model's performance in adversarial conditions, particularly against sophisticated attacks, would provide critical insights into its effectiveness and resilience. For example, investigating hypothetical scenarios such as stealthy attacks, evasion attacks, data poisoning, or model-based attacks within SDN-based VANET networks could offer valuable insights into the model's vulnerabilities and strengths. This exploration will contribute to a deeper understanding of the model's ability to deal with adversarial threats effectively.

Furthermore, addressing the computational overhead, cost, and complexities associated with deep learning approaches for DDoS attack detection in SDN-based VANET environments is essential. Extending the adaptability of the approach to various network scenarios could significantly augment its practicality and robustness.

CRediT authorship contribution statement

Mohamed Ali Setitra: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Mingyu Fan:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Project administration, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Authors in this article have used publicly available datasets.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.csi.2024.103845.

References

[1] J. Jiang, W. Susilo, J. Baek, Security analysis of "SMAKA: Secure many-to-many authentication and key agreement scheme for vehicular networks", IEEE Trans. Inf. Forensics Secur. 17 (2022) 3006–3007.

[2] K. Mohammed, M. Abdelhafid, K. Kamal, N. Ismail, A. Ilias, Intelligent driver monitoring system: An internet of things-based system for tracking and identifying the driving behavior, Comput. Stand. Interfaces 84 (2023) 103704.

[3] G. López-Millán, R. Marín-López, F. Pereñíguez-García, O. Canovas, J.A.P. Espín, Analysis and practical validation of a standard SDN-based framework for ipsec management, Comput. Stand. Interfaces 83 (2023) 103665.

[4] G. Lopez-Millan, R. Marin-Lopez, F. Pereniguez-Garcia, Towards a standard SDN-based ipsec management framework, Comput. Stand. Interfaces 66 (2019) 103357.

[5] P.K. Pandey, V. Kansal, A. Swaroop, Security challenges and solutions for next-generation VANETs: An exploratory study, in: Role of Data-Intensive Distributed Computing Systems in Designing Data Solutions, Springer, 2023, pp. 183–201.

[6] M.A. Setitra, M. Fan, Z.E.A. Bensalem, An efficient approach to detect distributed denial of service attacks for software defined internet of things combining autoencoder and extreme gradient boosting with feature selection and hyperparameter tuning optimization, Trans. Emerg. Telecommun. Technol. 34 (9) (2023) e4827.

[7] B. Ilyas, A. Kumar, M.A. Setitra, Z.A. Bensalem, H. Lei, Prevention of DDoS attacks using an optimized deep learning approach in blockchain technology, Trans. Emerg. Telecommun. Technol. (2023) e4729.

[8] M.A. Setitra, B.L.Y. Agbley, Z.E.A. Bensalem, M. Fan, Combination of hybrid feature selection and LSTM-AE neural network for enhancing DDOS detection in SDN, in: 2023 20th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP, IEEE, 2023, pp. 1–6.

[9] M.A. Setitra, M. Fan, B.L.Y. Agbley, Z.E.A. Bensalem, Optimized MLP-CNN model to enhance detecting DDoS attacks in SDN environment, Network 3 (4) (2023) 538–562.

[10] M.A. Setitra, I. Benkhaddra, Z.E.A. Bensalem, M. Fan, Feature modeling and dimensionality reduction to improve ML-based ddos detection systems in SDN environment, in: 2022 19th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP, IEEE, 2022, pp. 1–7.

[11] I. Pekaric, C. Sauerwein, S. Haselwanter, M. Felderer, A taxonomy of attack mechanisms in the automotive domain, Comput. Stand. Interfaces 78 (2021) 103539.

[12] M.A. Setitra, S.A. Madoune, Z.E.A. Bensalem, M. Fan, Toward delegating the detection of DDOS attacks to the SDN data plane: A security perspective, in: 2023 20th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP, IEEE, 2023, pp. 1–5.

[13] M. Arif, G. Wang, O. Geman, V.E. Balas, P. Tao, A. Brezulianu, J. Chen, Sdn-based vanets, security attacks, applications, and challenges, Appl. Sci. 10 (9) (2020) 3217.

[14] H. Amari, W. Louati, L. Khoukhi, L.H. Belguith, Securing software-defined vehicular network architecture against ddos attack, in: 2021 IEEE 46th Conference on Local Computer Networks, LCN, IEEE, 2021, pp. 653–656.

[15] Z. Zhao, W. Susilo, F. Guo, J. Lai, B. Wang, Full black-box retrievable and accountable identity-based encryption, Comput. Stand. Interfaces 86 (2023) 103741.

[16] D.D.N. Nguyen, K. Sood, Y. Xiang, L. Gao, L. Chi, Impersonation attack detection in IoT networks, in: GLOBECOM 2022-2022 IEEE Global Communications Conference, IEEE, 2022, pp. 6061–6066.

[17] Y. Cao, J. Wei, Y. Xiang, W. Susilo, X. Chen, Abuse-resistant deniable encryption, Comput. Stand. Interfaces 87 (2024) 103761.

[18] L. Xie, B. Yuan, H. Yang, Z. Hu, L. Jiang, L. Zhang, X. Cheng, MRFM: A timely detection method for ddos attacks in IoT with multidimensional reconstruction and function mapping, Comput. Stand. Interfaces (2023) 103829.

[19] S. Han, Q. Wu, H. Zhang, B. Qin, J. Yao, W. Susilo, Sc, in: International Conference on Artificial Intelligence and Security, Springer, 2022, pp. 616–628.

[20] Y. Wu, X. Wang, W. Susilo, G. Yang, Z.L. Jiang, H. Wang, T. Wu, Efficient maliciously secure two-party mixed-protocol framework for data-driven computation tasks, Comput. Stand. Interfaces 80 (2022) 103571.

[21] Z.E.A. Bensalem, I. Benkhaddra, M.A. Setitra, M. Fan, A novel and efficient sequential learning-based malware classification model, in: 2022 19th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP, IEEE, 2022, pp. 1–8.

[22] H. Ku, W. Susilo, Y. Zhang, W. Liu, M. Zhang, Privacy-preserving federated learning in medical diagnosis with homomorphic re-encryption, Comput. Stand. Interfaces 80 (2022) 103583.

[23] A. Nitaj, W. Susilo, J. Tonien, Enhanced S-boxes for the advanced encryption standard with maximal periodicity and better avalanche property, Comput. Stand. Interfaces 87 (2024) 103769.

[24] Z.A. Abbood, D.Ç. Atilla, C. Aydin, Enhancement of the performance of MANET using machine learning approach based on SDNs, Optik 272 (2023) 170268.

[25] H. Wang, Y. Li, W. Susilo, D.H. Duong, F. Luo, A fast and flexible attribute-based searchable encryption scheme supporting multi-search mechanism in cloud computing, Comput. Stand. Interfaces 82 (2022) 103635.

[26] S. Babu, A.R.K. Parthiban, DTMR: An adaptive distributed tree-based multicast routing protocol for vehicular networks, Comput. Stand. Interfaces 79 (2022) 103551.

[27] X. Wang, D. Wang, Q. Sun, Reliable routing in IP-based VANET with network gaps, Comput. Stand. Interfaces 55 (2018) 80–94.

[28] S. Saudagar, R. Ranawat, An amalgamated novel IDS model for misbehaviour detection using VeReMiNet, Comput. Stand. Interfaces 88 (2024) 103783.

[29] I. Benkhaddra, A. Kumar, M.A. Setitra, L. Hang, Design and development of consensus activation function enabled neural network-based smart healthcare using BIoT, Wirel. Pers. Commun. (2023) 1–26.

[30] P. Thorncharoensri, W. Susilo, Y. Chow, et al., Secure and efficient communication in VANETs using level-based access control, Wirel. Commun. Mob. Comput. 2022 (2022).

[31] S. Sambangi, L. Gondi, S. Aljawarneh, S.R. Annaluri, SDN DDOS ATTACK IMAGE DATASET, 2021, http://dx.doi.org/10.21227/k06q-3t33.

[32] M.S. Elsayed, N.-A. Le-Khac, A.D. Jurcut, InSDN: A novel SDN intrusion dataset, IEEE Access 8 (2020) 165263–165284.

[33] D. Tang, Z. Zheng, C. Yin, B. Xiong, Z. Qin, Q. Yang, Ftodefender: An efficient flow table overflow attacks defending system in SDN, Expert Syst. Appl. 237 (2024) 121460.

[34] G.S. Rao, P.K. Subbarao, A novel framework for detection of dos/ddos attack using deep learning techniques, and an approach to mitigate the impact of dos/ddos attack in network environment, Int. J. Intell. Syst. Appl. Eng. 12 (1) (2024) 450–466.

[35] K. Rashid, Y. Saeed, A. Ali, F. Jamil, R. Alkanhel, A. Muthanna, An adaptive real-time malicious node detection framework using machine learning in vehicular ad-hoc networks (VANETs), Sensors 23 (5) (2023) 2594.

[36] T.K. Jebur, Proposed hybrid secured method to protect against DDOS in n vehicular adhoc network (VANET)., Int. J. Interact. Mobile Technol. 17 (11) (2023).

[37] N. Keshari, D. Singh, A.K. Maurya, Dosrt: A denial-of-service resistant trust model for VANET, Cybern. Inf. Technol. 23 (4) (2023) 165–180.

[38] G.O. Anyanwu, C.I. Nwakanma, J.-M. Lee, D.-S. Kim, RBF-SVM kernel-based model for detecting ddos attacks in SDN integrated vehicular network, Ad Hoc Netw. 140 (2023) 103026.

[39] B. Liu, G. Xu, G. Xu, C. Wang, P. Zuo, Deep reinforcement learning-based intelligent security forwarding strategy for VANET, Sensors 23 (3) (2023) 1204.

[40] M. Türkoğlu, H. Polat, C. Koçak, O. Polat, Recognition of DDoS attacks on SD-VANET based on combination of hyperparameter optimization and feature selection, Expert Syst. Appl. (2022) 117500.

[41] M. Zhou, L. Han, H. Lu, C. Fu, Y. Qian, Attack detection based on invariant state set for SDN-enabled vehicle platoon control system, Veh. Commun. 34 (2022) 100417.

[42] G.O. Anyanwu, C.I. Nwakanma, J.-M. Lee, D.-S. Kim, Appropriate SVM kernel selection for ddos attack detection in SDN-based VANET, 2022, pp. 1251–1252.

[43] M. Al-Mehdhara, N. Ruan, MSOM: efficient mechanism for defense against ddos attacks in VANET, Wirel. Commun. Mob. Comput. 2021 (2021) 1–17.

[44] K.D. Thilak, A. Amuthan, S. Rajkamal, Mitigating DDoS attacks in VANETs using a variant artificial bee colony algorithm based on cellular automata, Soft Comput. 25 (18) (2021) 12191–12201.

[45] H. Polat, M. Turkoglu, O. Polat, Deep network approach with stacked sparse autoencoders in detection of ddos attacks on SDN-based VANET, IET Commun. 14 (22) (2020) 4089–4100.

[46] C. Xu, Y. Qu, Y. Xiang, L. Gao, Asynchronous federated learning on heterogeneous devices: A survey, Comp. Sci. Rev. 50 (2023) 100595.

[47] S.Ö. Arik, T. Pfister, Tabnet: Attentive interpretable tabular learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, No. 8, 2021, pp. 6679–6687.

[48] C. Shah, Q. Du, Y. Xu, Enhanced TabNet: Attentive interpretable tabular learning for hyperspectral image classification, Remote Sens. 14 (3) (2022) 716.

[49] M.V. Valueva, N. Nagornov, P.A. Lyakhov, G.V. Valuev, N.I. Chervyakov, Application of the residue number system to reduce hardware costs of the convolutional neural network implementation, Math. Comput. Simul. 177 (2020) 232–243.

[50] Y. Bengio, et al., Learning deep architectures for AI, Found. Trends Mach. Learn. 2 (1) (2009) 1–127.

[51] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[52] X. Chen, W. Susilo, E. Bertino, Cyber Security Meets Machine Learning, Springer, 2021.

[53] M. Alduailij, Q.W. Khan, M. Tahir, M. Sardaraz, M. Alduailij, F. Malik, Machine-learning-based ddos attack detection using mutual information and random forest feature importance method, Symmetry 14 (6) (2022) 1095.

[54] Ö. Tonkal, H. Polat, E. Başaran, Z. Cömert, R. Kocaoğlu, Machine learning approach equipped with neighbourhood component analysis for ddos attack detection in software-defined networking, Electronics 10 (11) (2021) 1227.

[55] L. Zhou, Y. Zhu, T. Zong, Y. Xiang, A feature selection-based method for ddos attack flow classification, Future Gener. Comput. Syst. 132 (2022) 67–79.

[56] X. Wang, K. Tan, Q. Du, Y. Chen, P. Du, Caps-TripleGAN: GAN-assisted CapsNet for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 57 (9) (2019) 7232–7245.

[57] L. Zhou, Y. Zhu, Y. Xiang, T. Zong, A novel feature-based framework enabling multi-type ddos attacks detection, World Wide Web 26 (1) (2023) 163–185.

[58] L. Zhou, Y. Zhu, Y. Xiang, A comprehensive feature importance evaluation for ddos attacks detection, in: International Conference on Advanced Data Mining and Applications, Springer, 2022, pp. 353–367.

[59] J. Yan, T. Xu, Y. Yu, H. Xu, Rainfall forecast model based on the tabnet model, Water 13 (9) (2021) 1272.

[60] B. Choi, Creating an ubuntu server virtual machine, in: Introduction To Python Network Automation: The First Journey, Springer, 2021, pp. 169–222.

[61] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[62] D.K. Singh, Hitesh, V. Kumar, H. Pham, Decision support system for ranking of software reliability growth models, in: Applications in Reliability and Statistical Computing, Springer, 2023, pp. 227–244.

[63] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Syst. 32 (2019).

[64] R. Pryss, W. Schlee, B. Hoppenstedt, M. Reichert, M. Spiliopoulou, B. Langguth, M. Breitmayer, T. Probst, Applying machine learning to daily-life data from the trackyourtinnitus mobile health crowdsensing platform to predict the mobile operating system used with high accuracy: Longitudinal observational study, J. Med. Internet Res. 22 (6) (2020) e15547.

[65] OptTabNet Source Code, https://github.com/theacademicresearch/OptTabNet.

Mohamed Ali Setitra received the State Engineer Diploma in computer science and Master's degree in Networking and Distributed Systems from the University of Science and Technology Houari Boumediene (USTHB), Algiers, Algeria, in 2004 and 2016, respectively. He was responsible for Cybersecurity (forensic, threats intelligence, regulation), and his Master's research was predicting DDoS attacks employed on Distributed Systems. He is currently pursuing his Ph.D. degree in Cybersecurity with the School of Computer Science and Engineering at the University of Electronic Science and Technology of China (UESTC), Chengdu/China. His research interests include improving the detection of Distributed Denial of Service (DDoS) attacks in Emerging Software Defined Networks (SDN) environments.

Prof. Mingyu Fan received a B.S. degree in electronics from Sichuan University, Chengdu, Sichuan, China, in 1982, an M.S. degree in cryptography from the Chinese Academy of Sciences, Beijing, China, in 1987, and a Ph.D. degree in information control from the Department of Electrical Engineering, Southwest Jiaotong University, Chengdu, in 1996. In 1997, she was a Postdoctoral Fellow in circuits and systems with the University of Electronic Science and Technology of China; a Postdoctoral Fellow in communication and information engineering with Tsinghua University in 2002; a Visiting Scholar with the Department of Computer Science, Queen's University, U.K., in 2005; and a Visiting Scholar with the Department of Mathematics and Computer Science, Moscow University, Russia, in 2010. She is currently the Director of the Information Security Research Center at the University of Electronic Science and Technology of China, a Professor at the School of Computer/Software, and a Doctoral Tutor. Her research interests include information security application technology, chip security and its application in aerospace information networks, and internet network security technology.