Machine Learning-Based Intrusion Detection System for Big Data Analytics in VANET

Mingyuan Zang, Ying Yan
Department of Photonics Engineering
Technical University of Denmark
Kgs. Lyngby, Denmark
Email: minza@dtu.dk, yiya@fotonik.dtu.dk

2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring) | 978-1-7281-8964-2/20/$31.00 ©2021 IEEE | DOI: 10.1109/VTC2021-Spring51267.2021.9448878

***Abstract*—Attacks as Distributed Denial of Service (DDoS) are ones of the most frequent vehicle cybersecurity threats. In this paper, we propose a Machine Learning-based Intrusion Detection System (IDS) for monitoring network traffic and detecting abnormal activities. This IDS framework integrates streaming engines for big data analytics, management and visualization. A Vehicular ad-hoc network (VANET) topology of multiple connected nodes with mobility capability is simulated in the Mininet-Wifi environment. Real-time data is collected using the sFlow technology and transmitted from the simulator to our proposed IDS framework. We have achieved high detection accuracy results by training the Random Forest as the classifier to label out the anomalous flows. Additionally, the network throughput has been evaluated and compared with and without deploying the proposed IDS. The results verify the system is a lightweight solution by bringing little burden to the network.**

***Index Terms*—Intrusion Detection System (IDS), VANET, Machine Learning, Distributed Denial of Service (DDoS), Big Data analytics, Mininet-Wifi**

I. Introduction

The Intelligent Transportation System (ITS) empowers the streamlining of communications and operations among vehicles. With a variety of sensors embedded, vehicles are able to collect surrounding information of road environment to realize the safe driving, auto piloting, emergency warning, etc. Two types of communications may get involved in this process, vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). V2V indicates the instant data exchange between vehicles, while V2I refers to the data transmission to the cloud or remote service providers. Since vehicles can dynamically switch between the moving and stationary state, the Vehicular ad-hoc network (VANET) is proposed to support a spontaneous network for V2V and V2I. In VANET, Road-Side Unit (RSU) is the stationary node deployed alongside the road connecting the moving vehicles to the cloud or remote servers.

In VANET, a huge volume of data with multiple types of protocols and heterogeneous formats will be exchanged. Since the vehicles would rely on this data to make decisions and take actions, the security issues are ones of the main concerns in VANET. The Distributed Denial of Service (DDoS) is a type of rampant and rapidly growing attack, which is predicted by Cisco that the total quantity of DDoS attacks will be doubled to 15.4 million by 2023. [1] DDoS attack targeting at availability can cause the depreciation of service by exhausting victim's resource. Attackers can aim at both the RSU (as Fig. 1 (a)) and the vehicle (as Fig. 1 (b)), for example to block the connectivity of the roadside communication or overload the in-vehicle communication. The consequences can cause road accidents and human death. Intrusion Detection System (IDS) is proposed to be an efficient solution to mitigate this type of attacks by detecting and reporting the malicious traffics to administrators to take countermeasures. The design of IDS has attracted considerable attention from researchers as it is a challenging task to deploy IDS in the real-time scenarios in ITS. According to the study in [2], the unstable wireless access, dynamic topology, lack of a central controller pose obstacles on the traffic collection, detection and management for IDS design. To meet the requirements of high throughput, low latency and high reliability for communications in VANET, the IDS needs to possess the following features:

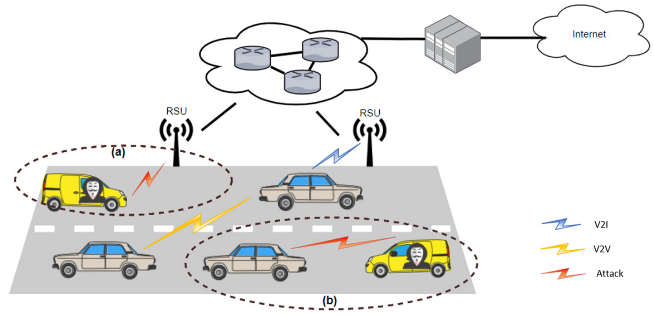


Fig. 1. VANET scenario with attack points.

a) precise detection. The detection algorithms need to be sensitive to the malicious flows. The false positive detection will lead to a block of the normal packet, while the false negative result fails to alert the threat. Paper [3] presents a Decision Tress-based IDS to achieve an accurate detection with low false positive rate. Although VANET is simulated in network simulator 3 (NS-3), another part of the dataset is from NSL-KDD which is an outdated dataset.

b) high-throughput process. The heterogeneous data is transmitted at high speed in VANET. The IDS framework need to support a high-throughput process to enable a low-latency data flow collection and analyzation. Specifically, the management of the collected flows and detection results has to sustain the massive amount of logs. The work in [4]

978-1-7281-8964-2/21/$31.00 ©2021 IEEE

deploys Spark and HDFS to process high-speed data and store massive data records. However, the solution is only tested with public dataset, no real-time traffic is simulated to evaluate the solution's performance.

c) lightweight system. The dynamic status and limited resource in VANET demand a lightweight system which will not burden the network much for the security mechanisms as the resources are reserved for main ITS functions. The researchers in paper [5] propose a lightweight detection mechanism by building a vehicle's behavior evaluation protocol to reduce the detection overhead.

The system design and performance evaluation can also be an issue for studies on VANET. Since it is not realistic to build a real VANET with a large amount of true vehicle modules, researchers have used various simulation tools to mimic the real-case scenarios. Simluators like OMNET++, used in research [6], are capable of simulating large scale vehicular networks. Compared with simulators, emulators like NS-3 and Mininet-Wifi [7] can provide a more realistic network simulations by supporting the real network interfaces. NS-3 is featuring for its high scalability and used by research work like paper [3] to build the VANET communication. Similar to NS-3 but with a user-friendly graphical interface, Mininet-Wifi supports the mobility and propagation model of network nodes and the wireless connection in IEEE 802.11p, which is defined for VANET communication. For instance, paper [8] utilizes the advantages of Mininet-Wifi to build a vehicular network topology in ITS.

In this paper, we propose an IDS framework for the VANET network by deploying the big data streaming and analytics engines to provide an efficient end-to-end detection solution. To reduce the impact on the network performance from the system, the sFlow technology [9] is used for compatible and lightweight traffic flow collection. Apache Kafa is deployed for message queueing. The open-source ELK stack (Elasticsearch, Logstash, Kibana) is applied for data storing, indexing and visualization. Random Forest algorithm is trained and run on Apache Spark for the Machine Learning-based anomaly detection method. A VANET network scenario with moving vehicles is emulated in Mininet-Wifi for system model training and performance evaluation.

Our contributions are summarized as follows:

- First, we propose an IDS framework to realize efficient and lightweight traffic flow collection, detection, management and visualization based on big data streaming engines for massive DoS/DDoS attack detection in high-throughput VANET scenario.
- Second, to the best of our knowledge, little work has been done to utilize Mininet-Wifi to emulate the real-time traffic for IDS evaluation in VANET. By building a VANET topology and connecting it to the IDS framework, we train and evaluate our IDS under real-time mobility scenarios with heterogeneous volume of traffic.
- Third, we train the Random Forest model with six basic flow features for anomaly detection on DoS/DDoS attacks. The model reaches significant low false positive rate under VANET scenarios with different network traffic loads.

The remainder of this paper has the following structure. Section II describes our proposed IDS framework for VANET network. Section III explains the details of the experimental setup for the system evaluation. Section IV presents the experimental results and related discussions. Section V summarizes our work and lists the future work.

II. Proposed Method

A. Proposed Framework

The IDS framework proposed in this paper is aiming at monitoring and detecting the abnormal activities across the vehicular network. Traffic exchanged in between the network elements is collected and analyzed. If any sign indicating an anomalous activity is censored, the system will log and report this activity. The whole system consists of four main modules: traffic collector, anomaly detector, traffic logger and traffic visualizer. Fig. 2 illustrates the proposed system. The function of each component in the system is listed below.

- Data Acquisition: the traffic collector is based on sFlow technology. Its packet sampling mechanism enhances the collection process by relieving the congestion and reducing the burden on CPU. The sFlow agents configured alongside with RSU in the network collect the data from the network nodes and forward it to a centralized sFlow collector. The collector will process the traffic to generate statistics and encapsulate them in UDP packets.
- Anomaly Detection: Apache Kafka and Apache Spark are deployed and collaborated to do the data streaming analytic to achieve a high-throughput traffic processing. Apache Kafka is a scalable messaging system which is deployed here as a message queue to transmit the collected data from the sFlow collector to Apache Spark. Apache Spark is an analytic engine for big data streaming processing. Spark MLlib is utilized in our work to implement Machine Learning algorithm for traffic analysis and abnormal activity detection. With micro-batch mechanism in Spark, such analysis can be done in low latency under the high-throughput scenario.
- Traffic Logging: By recording the collected traffic flows and detection results, logs are made for administrator's prospective investigation. Elasticsearch acts as a database by indexing and storing the input data. Logstash is a log analysis platform based on pipeline structure. The captured traffics consumed from Kafka are parsed in Logstash filters to pull out the specific fields and then output towards the target index in Elasticsearch for log managing. The anomaly detection and traffic logging stages are done in parallel to speed up the whole process.
- Data Visualization: Kibana is deployed as the endpoint of the system for data visualization. It sends queries to Elasticsearch by specifying the index to extract data and displays it on browser-based dashboard in the form of charts and metrics.

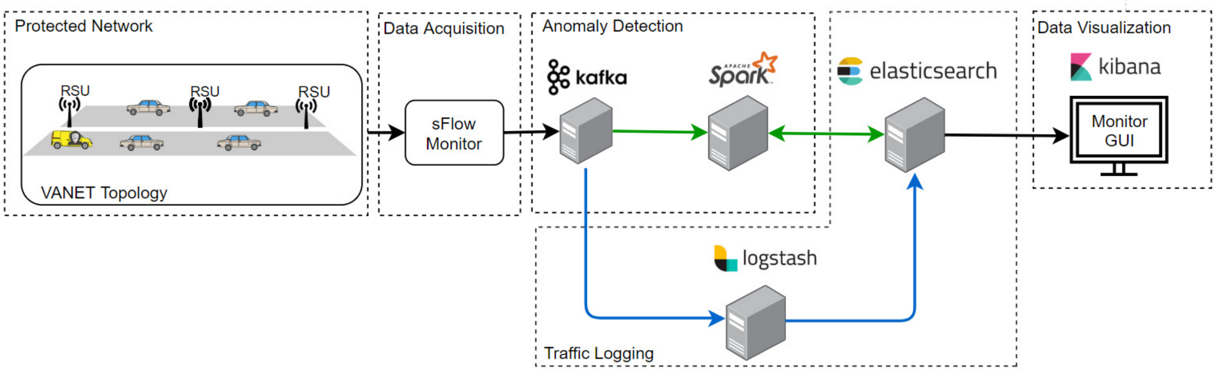


Fig. 2. Block diagram of the proposed IDS framework.

The data streaming process in Fig. 2 can be described as the following. When the traffic flows in the VANET network topology are captured by the sFlow-based monitor, they are then parsed and forwarded to Apache Kafka for streaming and queueing. After that, the workflow is split into two sets: real-time detection and data logging. The main stream will continue the traffic analyzation and anomaly detection. Apache Spark works as a data processing engine with the MLlib library to run Machine Learning-based detection model. The model will analyze each captured flow and label it as normal or abnormal. The labeled results will thereafter be saved to Elasticsearch. The other workflow set aims at logging for record backup. In case of the false alerts, the administrators can check the flows with these backup logs. The captured flows consumed from Kafka will be saved to Elasticsearch via Logstash at the same time when they're processed in Spark. Both the labeled and logged traffics stored in Elasticsearch can be searched and visualized from a Kibana dashboard.

B. Attacks and Dataset

DoS/DDoS attack is one of the main threats in vehicular communication that will cause the victim node no longer being available to respond to requests. In V2V communication, DoS/DDoS attack may exhaust the wireless bandwidth so that the messages can't be exchanged between vehicles. In V2I scenario, the server suffered with DoS/DDoS attack may end up with exhausted resources and incapability to provide services.

In this paper, to enable the proposed IDS to identify DoS/DDoS attacks, the training dataset is composed of two parts: flow records collected from Mininet-Wifi and CIC-IDS2017 dataset [10]. Traffics collected from the network topology in Mininet-Wifi are labeled and included in the training dataset to construct vehicular network's normal behaviors. To augment the training dataset, an IDS evaluation dataset CIC-IDS2017 is added. The dataset consists of traffics and network events via multiple protocols as well as common attcks. In our work, the flow record on Wednesday of CIC-IDS2017 is studied since it mainly includes the DoS/DDoS attacks. Specifically, the following three types of attack are investigated for detection:

- DoS Hulk: an attack with high-frequency flooding of HTTP GET requests towards the target server.
- DoS slowloris: an attack that generates the slow-rate attack traffics by exploiting the flaw of HTTP protocol to create legitimate TCP connection. [11]
- DDoS UDP flood: a rate-based attack sending fixed-length UDP packets in a large volume within a short time.

To pinpoint these attacks from the normal traffics, features are selected to promote the learning process and to avoid the possible impact brought by the irrelevant features. The selection based on rank of Gini Feature Importance score is performed on the flow records. The score of feature θ is computed as the reduction of the Gini impurity for all nodes in T trees as:

$$I_G(\theta) = \sum_{T} \sum_{\tau} \Delta i_\theta(\tau, T), \tag{1}$$

where $\Delta i(\tau) = i(\tau) - p_l i(\tau_l) - p_r i(\tau_r)$ represents the reduction of Gini impurity of node τ after samples split to its left and right sub-nodes. [12]

According to the importance ranking results of features in the training dataset for each attack (DDoS, DoS Hulk and DoS slowloris), six flow features are selected for model training: *source IP*, *destination IP*, *source port*, *destination port*, *protocol*, *packet length*. These features mark the basic attributes and direction of a flow.

C. Anomaly Detection Method

The anomaly detection process is modeled as a classification problem in this paper. Random Forest is trained as the classifier to classify data into normal/abnormal. It is based on tree structure. With the input samples to each tree in the forest, the output classifications depend on the scores indicating how much each tree voted. [13] Fig. 3 presents a schematic diagram of the whole detection process. Both the labeled dataset and real-time traffics are preprocessed firstly. The preprocess includes removing the flow records with missing value, extracting the selected features and doing the data

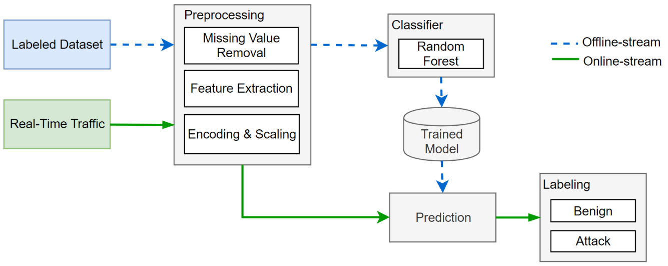


Fig. 3. Block diagram of the anomaly detection process.

encoding and scaling to have the records prepared for the model prediction. The labeled dataset is used for training the Random Forest classifier. With a well-trained model imported in system, the real-time traffic is fed into the system for detection and finally labeled as benign or attack.

III. Experimental Setup

The experimental setup consists of two modules: Intrusion Detection module and Network module. Intrusion Detection module contains all IDS units in the proposed framework as in Fig. 2 to provide traffic detection, logging and visualization. The Network module runs a VANET topology in Mininet-Wifi as the protected network of the IDS.

The VANET topology built for the experiments consists of three RSUs and N number of vehicles. The RSU is modeled as a wireless Access Point and vehicle is modeled as a station. A host is connected to RSU as a server via switches to provide HTTP services to vehicles. To simulate the malicious scenario in V2I and V2V communication, vehicles are exploited to launch attacks to the server via RSU (as Fig. 1 (a)) or directly to another vehicle (as Fig. 1 (b)). The details of the emulated topology is listed as in table I.

We train the anomaly detection algorithm, Random Forest, on a computer with the processor of Intel Xeon E5-1650 v4 @3.60GHz and the GPU of GeForce RTX 2080 Ti to speed up the training process. As mentioned in section II, we use both flow records collected from Mininet-Wifi and public dataset CIC-IDS2017 to train the model. In total, the dataset consists of 675517 samples of benign flows and 406075 samples of DoS/DDoS flows. The parameters of well-trained Random Forest classifier are listed in table II, which are obtained by Grid Search and cross validation method from *sklearn* library.

TABLE I
Parameters of VANET Emulation

Number of vehicles (N): 20, 40, 60	Type of packet: TCP, UDP
Number of RSUs: 3	Packet length: 64-1500 Bytes
Area: 1000m x 1000m	Type of attack: DDoS, DoS Hulk, DoS slowloris
Maximum speed: 60 km/h	Simulation time: 300 sec

TABLE II
Parameters of Detection Model

Number of trees:	100	Minimum number of samples to be at a leaf node:	3
Maximum depth of the tree:	10	Criterion for split quality:	Gini

The well-trained classifier is then loaded to the proposed IDS structure as depicted in Fig. 3 for real-case traffic detection. To evaluate the detection performance of the classifier, true positive rate (TPR) and false negative rate (FPR) are computed as

$$TPR = \frac{TP}{TP + FN}, \tag{2}$$

$$FPR = \frac{FP}{TN + FP}, \tag{3}$$

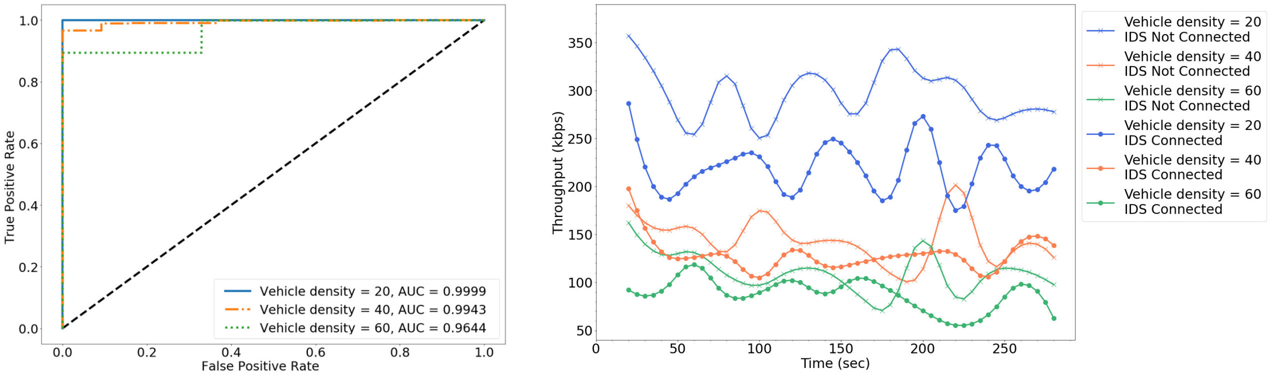
where TP is True Positive, FN is False Negative, FP is False Positive, and TN is True Negative. TPR indicates the amount of normal flows that are accurately detected as normal, while FPR means the number of normal traffic flows that are wrongly labeled as abnormal. Receiver Operating Characteristic (ROC) curves are plotted to depict the trade-off between these two parameters and area under the ROC curve (AUC) presents the degree of the separability of the model. [14]

Since the proposed IDS framework deploys sFlow and data streaming tools for a lightweight IDS solution, the network throughput is also recorded by *iperf* to evaluate if such kind of network monitoring will affect the network performance.

IV. Experimental Results and Discussions

Training time and detection time: The time it takes to train the classifier is ~2.76 seconds. The detection time of the real-time traffics takes ~0.61 seconds. The offline training and online detection scheme adopted in this work can efficiently speed up the traffic labeling process to achieve a sensitive detection system. By storing the captured real-case data in Elasticsearch, it is possible to retrain the classifier periodically to adapt the system to the protected network's environment.

Performance of anomaly detection model: Fig. 4 (a) displays the ROC curves with three kind of vehicle density. The model presents a good performance if its ROC curve is close to the top-left corner of the figure. In general, the model shows a high detection accuracy in all three type of vehicle density. By taking a closer look at the curves, with more vehicles in the network, an increment of the false positive rate can be observed as the curve is getting a bit away from the top-left corner of the graph. Also, the AUC score shows a 3% of drop from the low-density network (vehicle density = 20) to the high-density network (vehicle density = 60). This slight decrement of the detection accuracy may be caused by a more complex scenario in the vehicular network. When the vehicle density raises, the number of traffic flows exchanged in the network accumulate. Communication sessions with more sources and destinations may affect the classification results of the model. This decrement is expected to be reduced with a complex network emulation and an extended dataset.



(a)

(b)

Fig. 4. Experimental results: (a) ROC curves with respect to vehicle density (vehicles/km^2) (b) network throughput with or without IDS connected.

Performance of proposed IDS framework: The comparison of network throughput is summarized in Fig. 4 (b). In general, the impact brought by the data collection and monitoring isn't much since the curve with cross marker and the curve with point marker indicating if the IDS is connected are close to each other. This is especially the case when there are more vehicles in the network. When there is a high traffic load with 60 vehicles in the network, the curve with cross marker and the curve with point marker intertwined with each other. In this case, around 7% of throughput reduction in average is shown when the proposed IDS structure is connected to the network. When it comes to the case of small number of vehicles in the network, the impact on throughput is more obvious with around 20% of decrease. This result can be possibly because of the effect brought by the sampling mechanism in sFlow technology. It is also noticed that the throughput is getting lower at time 300 second when there are 40 and 60 vehicles in the network. This change is probably caused by the handover process in the network when some vehicles move into the coverage of the RSU nearby.

V. Conclusions

In this paper, a Machine Learning-based IDS framework for VANET is proposed and tested using the real-time data generated from the Mininet-Wifi emulator. A vehicle network with multiple car devices and mobility is set up. The IDS framework integrates the data streaming and analytic tools to support the high-throughput flow collection, analyzation and management under big data scenario. The Random Forest is trained as the classifier to label out the anomalous flows. The classifier presents a short training time and detection time. The experimental results prove that it can detect DoS/DDoS attacks with high detection accuracy and a low false positive rate. With the traffic flow collected from vehicular network topology close to the reality, the potential impact brought by the system on the network performance is studied by comparing the network throughput with and without the IDS deployed. The results verify the system is a lightweight solution by bringing a little burden to the network.

Future work consists of two aspects: 1) deploy an SDN controller in the system to automatically mitigate the detected malicious activities; 2) train the Deep Learning algorithm for detection model based on more flow features in VANET to improve the detection capability and efficiency of IDS.

References

[1] "Cisco annual internet report (2018–2023) white paper," https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[2] S. Garg, A. Singh, K. Kaur, G. S. Aujla, S. Batra, N. Kumar, and M. S. Obaidat, "Edge computing-based security framework for big data analytics in vanets," *IEEE Network*, vol. 33, no. 2, pp. 72–81, 2019.

[3] M. Aloqaily, S. Otoum, I. A. Ridhawi, and Y. Jararweh, "An intrusion detection system for connected vehicles in smart cities," *Ad Hoc Networks*, vol. 90, p. 101842, 2019.

[4] Y. Gao, H. Wu, B. Song, Y. Jin, X. Luo, and X. Zeng, "A distributed network intrusion detection system for distributed denial of service attacks in vehicular ad hoc network," *IEEE Access*, vol. 7, pp. 154 560–154 571, 2019.

[5] H. Sedjelmaci, S. M. Senouci, and M. A. Abu-Rgheff, "An efficient and lightweight intrusion detection mechanism for service-oriented vehicular networks," *IEEE Internet of Things Journal*, vol. 1, no. 6, pp. 570–577, 2014.

[6] K. Zaidi, M. B. Milojevic, V. Rakocevic, A. Nallanathan, and M. Rajarajan, "Host-based intrusion detection for vanets: A statistical approach to rogue node detection," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6703–6714, 2016.

[7] R. R. Fontes, S. Afzal, S. H. B. Brito, M. A. S. Santos, and C. E. Rothenberg, "Mininet-wifi: Emulating software-defined wireless networks," in *2015 11th International Conference on Network and Service Management (CNSM)*, 2015, pp. 384–389.

[8] A. Raj, T. T. Huu, P. Mohan, and G. Mohan, "Crossfire attack detection using deep learning in software defined its networks," *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pp. 1–6, 2019.

[9] "Traffic monitoring using sflow®," https://sflow.org/sFlowOverview.pdf.

[10] "Intrusion detection evaluation dataset (CIC-IDS2017)," https://www.unb.ca/cic/datasets/ids-2017.html.

[11] B. B. Gupta and O. Badve, "Taxonomy of dos and ddos attacks and desirable defense mechanism in a cloud computing environment," *Neural Computing and Applications*, vol. 28, 2016.

[12] B. Menze, B. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, and F. Hamprecht, "A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data," *BMC bioinformatics*, vol. 10, p. 213, 2009.

[13] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[14] P. A. Flach, *ROC Analysis*. Boston, MA: Springer US, 2010, pp. 869–875.