Dissertation on

"BAKSHI: An ML approach for detection of API attacks"

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering

UE20CS461A – Capstone Project Phase - 2

Submitted by:

Sameeraa Prinakaa	**PES2UG20CS303**
Sanjana S	**PES2UG20CS310**
Sneha Srinivasan	**PES2UG20CS342**
V Bavanika	**PES2UG20CS374**

Under the guidance of

Dr. Sarasvathi V
Professor
PES University

June - Nov 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

'BAKSHI: An ML approach for detection of API attacks'

is a bonafide work carried out by

Sameeraa Prinakaa	**PES2UG20CS303**
Sanjana S	**PES2UG20CS310**
Sneha Srinivasan	**PES2UG20CS342**
V Bavanika	**PES2UG20CS374**

In partial fulfilment for the completion of seventh semester Capstone Project Phase - 2 (UE20CS461A) in the Program of Study -Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period June 2023 – Nov. 2023. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 7th semester academic requirements in respect of project work.

Signature	Signature	Signature
Dr. Sarasvathi V	Dr. Sandesh B J	Dr. B K Keshavan
Professor	Chairperson	Dean of Faculty

External Viva

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

ANNEXURE-I

Bakshi: An ML approach for detection of API attacks

Sameeraa Prinakaa
Dept. of Computer Science
PES University
Bengaluru, India
sprinakaaa@gmail.com

Sanjana S
Dept. of Computer Science
PES University
Bengaluru, India
shetty.sanjana476@gmail.com

Sneha Srinivasan
Dept. of Computer Science
PES University
Bengaluru, India
snehapesu@gmail.com

V Bavanika
Dept. of Computer Science
PES University
Bengaluru, India
vbavanika028@gmail.com

Sarasvathi V
Dept. of Computer Science
PES University)
Bengaluru, India
sarsvathiv@pes.edu

***Abstract*—The rise of the internet led to widespread web Application Programming Interface (API) usage, fostering remote software communication. However, this popularity attracts attacks exploiting vulnerabilities, posing threats like DdoS, MITM, and injection attacks. Monitoring API calls is crucial to safeguard data security, service availability, and system integrity. This project focuses on real-time API attack detection, specifically identifying Economic Denial of Service (EDoS), Expired JWT Tokens, and Brute Force Broken Authentication attempts. It also offers a comprehensive approach to network, such as brute force on FTP, SSH,SQL Injection. Additionally, the system demonstrates robust capabilities in detecting various malware, such as Exploits, Worms, Shellcodes, and Backdoors, ensuring a holistic defence against malicious API calls. It adheres to stringent security and privacy requirements, achieving notable accuracies of 99.90% for network traffic and 98.81% for malware detection. The proposed system addresses the limitations of single-dataset reliance and presents advancements in attack specific datasets and a multi-tiered threat classification approach.**

***Index Terms*—API, Alerts, Real-time detection, Malware, Network Attacks, XGBoost, Random Forest, JWT, Spring Boot**

I. Introduction

The past two decades have witnessed a remarkable expansion of the internet, giving rise to the development of web-based software solutions. In response to the need for seamless communication between distant software entities, web APIs emerged as a vital component of modern technology. In the words of Tim Cook, ”The advancement of technology has been a double-edged sword. On one hand, it has made our lives easier and more convenient. On the other hand, it has made us more vulnerable to cyber-attacks. As technology continues to evolve, so too must our efforts to secure it.” This shift towards technology has led to an increased reliance on API’s, which in turn, has contributed to a surge in API- specific attacks, consequently intensifying the risk of security breaches.

In 2018 and 2022, Google Fi Service encountered significant breaches characterized by API vulnerabilities, resulting in unauthorized access and the compromise of sensitive data for millions of users. Facebook, in 2020, faced a breach of its Photo API, exposing the private information of 6.8 million users due to a Facebook login bug. Twitter confronted a Broken Object Level Authorization exploit in November 2022, leading to the inadvertent exposure of personal data for 5.4 million users. In 2019, Justdial experienced a major breach involving its main API and additional APIs, exposing private data for a staggering 100 million users through publicly accessible endpoints. Moreover, in 2021, a series of attacks beginning in April and escalating in June resulted in a massive data breach affecting an initial estimate of 500 million users, eventually reaching 700 million. These incidents highlight the urgent need for robust cybersecurity measures to safeguard user data and prevent unauthorized access.

In light of this, this project embarks on the mission to develop an IDS that can effectively detect as well as classify these API based attacks along with network and malware based attacks. Our vision is to construct an IDS that can analyze and classify multiple types of traffic such as network, malware and API calls to fortify the security infrastructure in the evolving landscape of technology.This paper introduces a security framework designed to detect API, network, and malware traffic. A SpringBoot API has been created over which three attacks are simulated and detected in real time. Moreover, the framework utilizes a signature-based intrusion detection system (IDS) tailored for the identification of both malware and network threats.

The rest of this paper is organized as follows: Section II discusses the existing research efforts towards detecting API, network and malware attacks, while section III presents the proposed security framework. Section IV discusses the experiments results followed by concluding remarks and references at the end.

II. LITERATURE SURVEY

The widespread adoption of web APIs has heightened security risks, resulting in an upsurge of API-specific, malware, and network-related attacks. This literature survey addresses the critical need for accurate detection methods, diverse datasets, and robust machine-learning models in the context of API, malware, and network traffic security. Identifying deficiencies in API extraction process descriptions and references within existing literature. Hence, a concise overview of all referenced papers is provided below.

The authors in [1] address surging API traffic challenges in large enterprises, proposing an innovative API-TAD system. Utilizing a Support Vector Machine (SVM) for classification, the system employs a Gaussian distribution for dataset creation, encompassing crucial steps like feature extraction, selection, model training, and anomaly detection. Validation compares with traditional methods, and a synthetic dataset enhances implementation. A notable contribution is the creation of a proprietary dataset, featuring client IP addresses, HTTP request types, response codes, consecutive response numbers, request bandwidth, and connection tokens. The approach balances precision and minimizes false positives, seamlessly integrating with microservice frameworks. This pragmatic solution offers insights for future research, though its focus on a specific enterprise environment may limit direct applicability to diverse settings, emphasizing DDoS issues.

The authors in [2] address the challenge posed by the substantial API traffic generated in large enterprises, supporting their business activities. This surge in API calls makes analysis and prediction complex. To tackle this, the paper proposes an ML-based technique utilizing a linear kernel to detect and classify API traffic anomalies. SVM is employed for classifying abnormal API traffic and detecting potential attacks, utilizing a Gaussian distribution for dataset creation. The approach undergoes evaluation, comparing it with traditional signature-based methods, and outlines a comprehensive anomaly detection framework involving feature extraction, selection, model training, and detection using ”Snort” and ”Suricata.” A proprietary synthetic dataset, inspired by real-world APIs, is a notable feature, encompassing client IP addresses, HTTP request types, response codes, consecutive response numbers, request bandwidth, and connection tokens. Striking a balance between precision and false positives, the approach integrates seamlessly with existing microservice frameworks as a plug-and-play solution. While providing a practical API security solution for large enterprises, the paper focuses on a specific enterprise environment. The included dataset serves as a valuable resource for further research.

The author in [3], Subiksha Srinivasa Gopalan et al., examined imbalanced data in intrusion detection systems, focusing on CSE-CIC IDS datasets. The study categorizes ML techniques, emphasizing balanced datasets' role in reducing bias. It scrutinizes prior work on CSE-CIC IDS datasets, urging further research on supervised ML to counter bias. The paper highlights the absence of an evaluation framework, advocates for diverse datasets, and calls for techniques addressing imbalance. Using a rigorous methodology, it defines criteria around intrusion detection, AI, and key terms. Covering 2018-2020, it details ML techniques, identifies imbalance issues, and presents a classification system. However, it lacks technique performance evaluation and efficacy comparisons. The study unveils a benign to malicious instance imbalance ratio, stressing dataset improvement compared to the 2017 version against set standards.

TABLE I
RULE BASED VERSUS SVM BASED APPROACH

Method	Unseen detected	FP(%)	TP(%)	F-score
SVM	83.5	7.3	92.7	0.964
Snort	20.45	22.5	76.5	0.751
Suricata	22.45	21.5	77.5	0.761

The authors in [4] introduce a novel approach to malware detection, utilizing a graph convolutional network (GCN), contrasting its performance with traditional machine learning methods. Their GCN-based approach captures both structural and content features of malware through a graph-based representation, demonstrating superior accuracy in detection compared to traditional methods. The implementation involves creating a GCN-based classifier adaptable to diverse malware characteristics, reducing feature extraction challenges. The workflow includes extracting API call sequences, generating Directed Cyclic Graphs, and utilizing Markov chain concepts for feature map extraction. The proposed method, evaluated on a dataset of 13,624 samples, exhibits advantages such as outperforming False Positive Rate (FPR) and accuracy in traditional techniques and resilience to evasion attacks. However, limitations include the lack of real-world scenarios, resource-intensive computing requirements, and a need for expertise in graph analysis and machine learning. Despite limitations, the GCN-based approach, detailed in Table II, achieves a notable accuracy rate of 98.32%, making it a promising solution for intelligent malware detection in the face of feature extraction challenges.

TABLE II
GCN VERSUS EXISTING METHODS

Detection Approach	OS Type	TPR	FPR	Accuracy
Smita et al.	Windows	0.9930	0.0460	0.9542
Tang et al.	Windows	0.9900	0.0105	None
Lie et al.	Android	None	None	0.9647
Raf et al.	Windows	None	None	0.9400
Alzaylaee et al.	Android	0.9956	0.0330	0.9780
GCN	Windows	0.9951	0.0037	0.9832

The authors in [5] present CruParamer, a Windows malware detection system leveraging deep neural networks. Using parameter-augmented API sequences, it evaluates sensitivity

through rule-based and clustering-based classification. These sensitivities inform API runtime parameters. After labeling, native and sensitive embeddings connect security semantics with labeled APIs. Processed by a DNN, the concatenated sequences achieve malware detection through binary classification. This paper underscores runtime parameter significance, introducing a parameter-assisted API labeling method that outperforms raw API sequence feeding. Employing Text-CNN and BiLSTM models on datasets containing 94 and 98 APIs, the system achieves 98.52% accuracy, 98.63% precision, and a 98.52% F1 score in two minutes. Outperforming Naive Bayes, KNN, Logistic regression, and Decision tree, it boasts high recall, low misclassification, and a large AUC. API labeling enhances F1 score by 1.56%, and embedding increases it by 1.52%. Limitations include challenges in detecting new malware, imperfect classification for unknown parameters, exclusive binary classification, and longer processing time compared to raw API call methods.

TABLE III
TEST RESULTS INFERRED

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	71.69	84.80	52.85	65.12
DT	76.32	90.28	58.99	71.36
Test-CNN	95.54	97.65	93.33	95.44
Bi-LSTM	84.16	93.36	73.55	82.28
Cru-BiLSTM	92.62	96.51	88.43	92.29
Cru-TextCNN	98.52	98.63	98.41	98.52

III. DATASET

A. CIC-IDS2018

This project utilizes the CIC-IDS2018 dataset for effective network traffic classification. This publicly available dataset offers a comprehensive view of contemporary network traffic, featuring seven distinct attack types simulated in a controlled environment. Notably, our focus narrows down to web attacks and brute force, constituting approximately 16% of the overall network traffic. Acknowledged as a benchmark, CIC-IDS2018 stands out for its expansive and diverse dataset, encompassing various attacks over multiple days. However, challenges like potential overfitting, data cleaning, preprocessing, and class imbalance necessitate careful consideration.

B. ADFA Dataset (NGIDS - DS)

This project utilizes the ADFA NGIDS-DS dataset for effective malware traffic classification. This publicly accessible dataset, derived from a military network, captures realistic scenarios with diverse attacks such as reverse shell, privilege escalation, and backdoor exploitation. Notably, our focus narrows down to Exploits, Backdoors, Shellcode and Worms. The features of this dataset include date, time, process ID, and system call details, it serves as a valuable resource for system call-based HIDS research on Linux operating systems. The dataset's advantages include its reflection of real-world military traffic, extensive diversity, and size, collected over several days. However, challenges involve potential dataset imbalance, it predominantly centers on Linux systems.

IV. PROPOSED METHODOLOGY

A. Background Techniques

This project aims to accurately detect three major types of traffic: API, Network and Malware. In the proposed framework, different classifiers are used viz; XGBoost for network traffic classification and Random Forest for malware traffic classification. XGBoost combines the predictions of multiple weak learners to create a robust model. Random forest casts a majority voting on the decision of multiple trees to vote for the most popular class for a certain input. Both the models perform multi-class classification on the given traffic. We have created a SpringBoot API over which three attacks are simulated and detected in real-time.

B. Design of the Proposed Framework

This study focuses on detecting API, Network, and Malware traffic using an IDS to enhance overall system security. The proposed framework, as shown in Fig. 1, integrates signature-based detection for network and malware traffic with real-time API traffic detection. The user engages with the web interface, submitting instances through a JSON-formatted payload directed to the input parser for queuing. After preprocessing and feature selection, the data is transformed and only selected features proceed for classification. Based on the instance's nature, it goes to either the malware or network model for classification, predicting benign or malicious status and specific attack types. The alert system promptly notifies users of the outcome, ensuring timely awareness of potential security threats. The user gets to interact with the application with a basic login using a username and password. Followed by which users can view and update the product info via API "GET" and "POST" methods.

Signature Detection of network and malware

Signature Detection of network and malware has four modules:

- Data Cleaning
- Data Pre-Processing
- Model Training for Multi-Class Classification
- Storing the Trained Model
- Integration with Interface

1) Data Cleaning: The Signature-based Intrusion Detection System (IDS) undergoes initial training using the CICIDS2018 dataset, encompassing 77 distinct features such as protocols, ports, and labels denoting traffic nature (Benign or Malicious) including attacks such as Brute Force (SSH, XSS, Web, FTP) and SQL Injection. Concurrently, the ADFA Dataset (NGIDS - DS) is employed for training the malware model, featuring over a million records with attributes like Date, Time, Processid, System call eventid, path, and labels indicating traffic type (Benign or Malicious), encompassing categories like Benign, Exploits, Backdoors, Shellcode, and Worms.

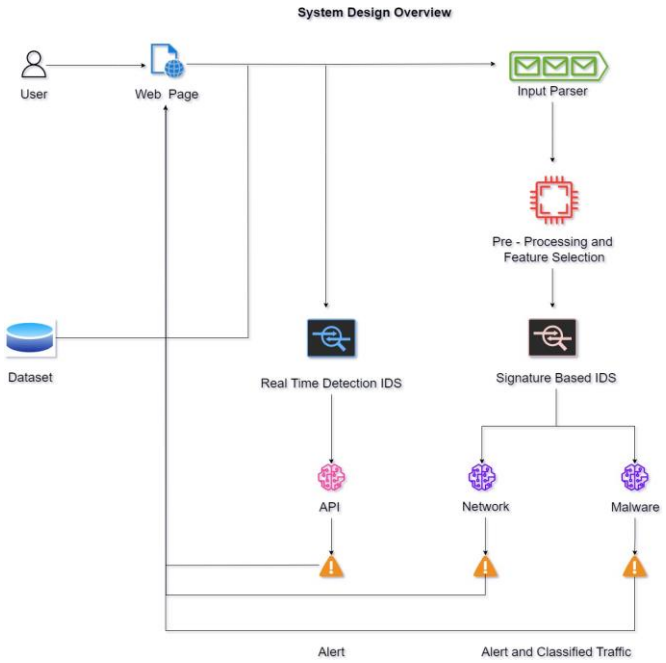

Fig. 1. Working flow of the Proposed Framework

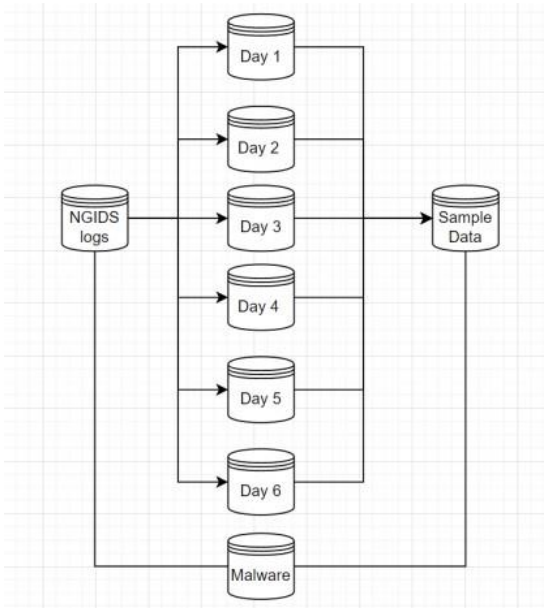

Fig. 2. Systematic Random Sampling of Malware

These datasets are then cleaned by removing Null, negative, zero variance, and unnecessary values, ensuring data integrity and appropriate data types.

2) Data Pre-Processing: To deal with the data imbalace of the CICIDS2018 dataset, targeted random under sampling and Synthetic Minority Over-sampling Technique (SMOTE) were used. Simultaneously, one-hot encoding was applied for effective classification. Moreover, Recursive Feature Elimination using Random Forest was used to select the relevant features of the dataset. Conversely, for malware instances, normalization was achieved through L2 norm, and systematic random sampling was used on both benign and malware instances, which were then combined together as depicted in Fig. 2. Moreover,hash encoding and one-hot encoding techniques were used for successful classification.

3) Model Training for Multi-Class Classification The datasets were then split in the ratio of 80:20 for training and testing respectively. Finally, to perform multi-class classification, the XGBoost model was used for network instances and the Random Forest Model was used for malware instances.

4) Integration with Interface: Our project introduces a robust cybersecurity framework comprising a Java server with enhanced intrusion detection capabilities, specifically designed for API attacks, coupled with a Python server proficient in classifying network and malware threats. The effective integration of these servers via API endpoints facilitates comprehensive threat analysis and timely alerts, enhancing overall system security.

Real-Time Detection of API attacks

This system is designed to detect API abuse within a Spring Boot-based application integrated with a React front-end utilizing JWT authentication. It is also engineered to identify and alert upon encountering three distinct attacks, each delineated by specific thresholds.

1) **Expired Token Alert:** An alert is triggered when a login attempt using an expired token occurs, preemptively thwarting potential misuse by attackers attempting to exploit stolen or outdated tokens.
2) **Economical DoS Detection:** This system mitigates the risk of economic Distributed Denial-of-Service (DDoS) attacks. Specifically, it monitors reset password requests, safeguarding against excessive requests that could target third-party APIs responsible for OTP generation, potentially incurring financial losses. An alert is generated upon surpassing predefined thresholds for such requests.
3) **Brute Force Broken User Authentication:** It proactively identifies and alerts upon detecting login attempts utilizing the same username but with brute-force attacks emanating from different sources. Notably, the attack instances are coded in Node.js for evaluation purposes.

V. RESULTS AND ANALYSIS

A. Evaluation Indicator

Accuracy, precision, detection rate, F1-Score, FPR and FNR have been chosen as the performance metrics for the intrusion detection model suggested in this research.

- **Accuracy:** The ratio of correctly predicted instances to the total instances.

- **Precision:** The ratio of correctly predicted positive observations to the total predicted positives.
- **Recall:** The ratio of correctly predicted positive observations to all observations in the actual class.
- **F1 Score:** The weighted average of precision and recall, balancing false positives and false negatives.
- **False Positive Rate (FPR):** The ratio of false positives to the total actual negatives.
- **False Negative Rate (FNR):** The ratio of false negatives to the total actual positives.

B. Evaluation performance of the proposed model

Network Traffic

TABLE IV
OVERALL PRECISION, RECALL, F1 SCORE AND ACCURACY USING XGBOOST

Model	Precision	Recall	F1 Score	Accuracy
XGBoost	0.7164	0.8945	0.7614	0.9990

TABLE V
PRECISION, RECALL, F1 SCORE AND ACCURACY OF EACH CLASS USING XGBOOST

Class	Precision	Recall	F1 Score	Accuracy
Benign	0.999976	0.999053	0.999514	0.999053
Brute Force - Web	0.199575	0.846847	0.323024	0.846847
Brute Force - XSS	0.916667	0.956522	0.936170	0.956522
FTP-BruteForce	0.800000	0.800000	0.800000	0.800000
SQL Injection	0.382353	0.764706	0.509804	0.764706
SSH-Bruteforce	0.999894	0.999894	0.999894	0.999894

Malware Traffic

TABLE VI
ACCURACY, FPR, FNR FOR EACH CLASS USING RANDOM FOREST

Class	Accuracy	FPR	FNR
Benign	0.98931	0.0103	0.0144
Exploits	0.99198	0.0111	0.0053
Backdoors	0.96069	0.0328	0.0013
Shellcode	0.95836	0.0374	0.0012
Worms	0.94752	0.0310	0.0003

The general findings indicate that the classifier successfully identified and categorized malicious activity with elevated levels of accuracy and precision. The XGBoost model to classify network instances achieves a commendable overall accuracy of 99.90 % and the Random Forest model achieves a high accuracy of 98.81 %.

VI. CONCLUSION AND FUTURE WORK

This project introduces a comprehensive approach to enhance threat detection and alert capabilities, leveraging a diverse array of models, datasets, and techniques. The emphasis on layered and event-driven architectures underscores adaptability and scalability, tailoring a robust system to meet our specific requirements. Achieved significant accuracies, notably 99.90% for network traffic detection using XGBoost on the CICIDS2018 dataset and 98.81% for malware detection with Random Forest on the ADFA-NGIDS dataset, with emphasis on the real-time effectiveness of our system in identifying Spring Boot API attacks. Future work can include real-time traffic capture for instant inspection, enabling early anomaly detection and immediate response. Implementation of dashboards for real-time attack tracking and data organization can be done to enhance understanding and responsiveness. Additionally, detection mechanisms aligned with OWASP Top 10 vulnerabilities, such as CSRF, Broken Object and Function Level Authentication, etc. and zero-day attacks can be incorporated.

REFERENCES

[1] M. Sowmya, A. J. Rai, V. Spoorthi, M. Irfan, P. B. Honnavalli and S. Nagasundari, ”API Traffic Anomaly Detection in Microservice Architecture,” 2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing Workshops (CCGridW), Bangalore, India, 2023, pp. 206-213, doi: 10.1109/CCGridW59191.2023.00044.

[2] G. Baye, F. Hussain, A. Oracevic, R. Hussain and S. M. Ahsan Kazmi, ”API Security in Large Enterprises: Leveraging Machine Learning for Anomaly Detection,” 2021 International Symposium on Networks, Computers and Communications (ISNCC), Dubai, United Arab Emirates, 2021, pp. 1-6, doi: 10.1109/ISNCC52172.2021.9615638.

[3] Gopalan, Subiksha & Ravikumar, Dharshini & Linekar, Dino & Raza, Ali & Hasib, Maheen. (2021). Balancing Approaches towards ML for IDS: A Survey for the CSE-CIC IDS Dataset. 1-6. 10.1109/ICCSPA49915.2021.9385742

[4] Li, S., Zhou, Q., Zhou, R. et al. Intelligent malware detection based on graph convolutional network. J Supercomput 78, 4182–4198 (2022). https://doi.org/10.1007/s11227-021-04020-y

[5] X. Chen et al., ”CruParamer: Learning on Parameter-Augmented API Sequences for Malware Detection,” in IEEE Transactions on Information Forensics and Security, vol. 17, pp. 788-803, 2022, doi: 10.1109/TIFS.2022.3152360

[6] P. Wei, Y. Li, Z. Zhang, T. Hu, Z. Li and D. Liu, ”An Optimization Method for Intrusion Detection Classification Model Based on Deep Belief Network,” in IEEE Access, vol. 7, pp. 87593-87605, 2019, doi: 10.1109/ACCESS.2019.2925828

[7] D. Li, D. Kotani and Y. Okabe, ”Improving Attack Detection Performance in NIDS Using GAN,” 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 2020, pp. 817-825, doi: 10.1109/COMPSAC48688.2020.0-162.