IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Network Intrusion Detection And Prevention System Using ML Algorithm

[1]Bhushan Ajit Jadhav, [2]Prof: Amit Nichat

[1] B.Tech (Cloud Technologies and Information Security) Students School of Engineering, Ajeenkya DY Patil University Lohegaon Pune, Maharashtra, India, [2]

Abstract: This paper investigates the integration of Machine Learning (ML) for anomaly detection within an Intrusion Detection and Prevention System (IDPS). We propose a two-pronged approach utilizing Golang and Python. Golang handles real-time network traffic analysis and signature-based detection, while Python tackles data pre-processing, training ML models, and potentially generating synthetic attack data using a Generative Adversarial Network (GAN). This synthetic data, focusing on anomaly patterns, will be evaluated in a simulated environment to assess its effectiveness in detecting real-world attacks. If successful, the generated data, along with real-world data emphasizing attack severity, will be used to enhance the training dataset, targeting the most relevant threats. This research aims to achieve a balance between real-time performance, improved detection accuracy and reduced false positives while acknowledging the challenges and ethical considerations of implementing ML in IDPS. The project leverages Golang's strengths for a less complex and scalable application, while Python provides the environment for ML development. Trained models will be saved in a format like ONNX for seamless integration with Golang for signature and anomaly-based detection, enabling the system to make informed decisions and generate alarms. Ultimately, this approach aims to simplify the system while enhancing its ability to identify zero-day attacks.

Keywords - Intrusion Detection and Prevention System (IDPS), Machine Learning (ML), Generative Adversarial Network (GAN), Real-Time Network Traffic Analysis, Zero-Day Attacks

I. INTRODUCTION

The digital landscape is experiencing explosive growth, leading to a heightened need for robust and constantly evolving security measures. Safeguarding the integrity and confidentiality of data traversing interconnected networks is paramount in today's world. However, this expansion also fosters a persistent concern: cybersecurity threats. Malicious actors are continuously developing sophisticated techniques to exploit vulnerabilities in network infrastructure. These threats can have devastating consequences, causing financial losses, data breaches, and disruption of critical services. Traditional Intrusion Detection and Prevention Systems (IDPS) have served as the first line of defense against malicious activity on networks. These systems rely on pre-defined rules to identify known attack patterns [3]. However, the effectiveness of these rule-based systems is constantly challenged by the ever-evolving nature of cyber threats. Attackers are adept at developing novel attack vectors and obfuscating their methods, rendering signature-based detection less effective. This has led to a growing interest in leveraging Machine Learning (ML) algorithms to enhance the capabilities of IDPS. Recent advancements in ML, particularly deep learning and hybrid systems, hold significant promise for improving the accuracy and efficiency of intrusion detection [2][4]. These approaches harness the power of neural networks to automatically learn intricate patterns within network traffic data. This enables them to identify both known and novel attack methods with greater precision.

Network Attacks: A Classification

Network attacks can be broadly categorized into two main groups:

- **Passive Attacks:** These attacks aim to eavesdrop on or intercept sensitive information transmitted across a network without altering or disrupting the network traffic itself. Examples include packet sniffing and traffic analysis.
- **Active Attacks:** These attacks actively disrupt or manipulate network traffic with the intent to harm a system or steal data. Common active attacks include:
  - **Denial-of-Service (DoS) Attacks:** These attacks overwhelm a system or network with a flood of traffic, rendering it unavailable to legitimate users.
  - **Man-in-the-Middle (MitM) Attacks:** These attacks involve the attacker inserting themselves into the communication channel between two parties, allowing them to eavesdrop on or manipulate the communication.
  - **SQL Injection Attacks:** These attacks exploit vulnerabilities in web applications to inject malicious SQL code into a database server, potentially allowing unauthorized access to sensitive data.
  - **Phishing Attacks:** These attacks attempt to trick users into revealing sensitive information, such as usernames and passwords, by posing as legitimate entities.

The Need for Adaptive Security Solutions

The dynamic nature of cyber threats demands a more adaptable approach to network security. Traditional rule-based systems struggle to keep pace with the evolving tactics of attackers. Here, Machine Learning (ML) emerges as a powerful tool. It offers the adaptability and pattern recognition capabilities needed to effectively detect and thwart a wide range of network attacks, both known and unknown.

Understanding Intrusion Detection Systems (IDS)

Intrusion Detection and Prevention Systems (NIDS) play a critical role in network security, working alongside firewalls to fortify system defenses. NIDS can be categorized into two primary types based on their deployment:

- **Host-Based Intrusion Detection System (HIDS):** HIDS monitors and analyzes the internal activities of a single host, such as a computer or server, to detect malicious activity within the system. This is particularly valuable in identifying insider threats and unauthorized access attempts [6].
- **Network-Based Intrusion Detection System (NIDS):** NIDS focuses on real-time network traffic monitoring, identifying suspicious patterns that might indicate a network intrusion. It excels at detecting attacks targeting vulnerabilities in network services or protocols [6].

Traditional Detection Methods Employed by NIDS

There are three main detection methods employed by NIDS:

- **Signature-based Detection:** This method compares network traffic or system activity against a database of known attack signatures. If a match is found, the system triggers an alert. While effective for known threats, it struggles to detect novel attacks.
- **Anomaly-based Detection:** This method establishes a baseline of normal network behavior. Any significant deviations from this baseline are flagged as potential intrusions. While effective against unknown attacks, it can generate false positives by mistaking legitimate activity for anomalies.
- **Hybrid Detection:** Combining the strengths of both signature-based and anomaly-based detection, hybrid systems leverage signatures to detect known attacks and anomaly detection to identify novel threats.

The Role of Machine Learning (ML) in NIDS

Machine learning (ML) is a subset of artificial intelligence (AI) that empowers computer systems to learn from data and make decisions without explicit programming. In the context of NIDS, ML algorithms can significantly improve cyber threat detection and prevention:

- **Enhanced Detection Accuracy:** Through historical data analysis, ML algorithms can outperform traditional rule-based systems by performing deep packet analysis and comparing various data packet features. This leads to more accurate detection of malicious activity.
- **Adaptability to Evolving Threats:** ML algorithms possess the remarkable ability to adapt to new and evolving threats. They can continuously learn from new data and update their models to detect previously unknown attack patterns.
- **Reduced False Positives:** ML algorithms can significantly reduce false positives, which occur when legitimate network activity is mistakenly identified as an attack. By learning what constitutes normal network behavior, these algorithms can more accurately distinguish anomalies that signal a potential attack.
- **Real-time Threat Detection:** ML algorithms can analyze network traffic in real-time, enabling immediate detection of suspicious activity and faster response times to mitigate threats.
- **Scalability:** ML algorithms can handle large volumes of network traffic efficiently, making them ideal for high-speed networks where manual analysis becomes impractical.

This research builds upon existing studies in the field by exploring the application of various ML techniques, including Random Forest, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Generative Adversarial Networks (GANs)[12][11][6], in developing a novel NIDS. The primary focus lies on improving detection accuracy, minimizing false positives, and addressing the limitations of traditional IDPS. Leveraging expertise in both cybersecurity and machine learning, this research investigates the performance of the ML-based NIDS using real-world network datasets. Through experiments conducted in a controlled environment, the paper aims to contribute valuable insights to the growing body of knowledge surrounding ML-driven approaches for network security.

LITERATURE REVIEW

Cybersecurity research has made significant strides, thanks to the adoption of deep learning and hybrid systems, offering novel approaches for detecting and mitigating cyber threats, especially in light of evolving attack strategies. However, a major challenge in cybersecurity research is obtaining large and diverse datasets necessary for training and evaluating machine learning models. Privacy concerns and limited access to real-world datasets have prompted researchers to explore the use of generative models, such as Generative Adversarial Networks (GANs), for generating synthetic attack data [9][12]. In the field of network intrusion detection systems (NIDS), researchers have made notable advancements, particularly with the integration of machine learning (ML) and deep learning (DL) techniques. Various datasets and algorithms have been explored to enhance detection accuracy, reduce false positives, and address challenges faced by traditional NIDS. However, datasets used for training ML models for NIDS can exhibit class imbalances and overlap, potentially leading to overfitting [1][2][3].

Zoghi and Serpen conducted a detailed analysis of the UNSW-NB15 computer security dataset, emphasizing visualization techniques to understand its characteristics [1]. This dataset has been valuable for evaluating NIDS approaches, although it can lead to overfitting in ML training.

Niyaz et al. proposed a deep learning approach for NIDS, demonstrating its effectiveness in detecting intrusions [2]. Their work showcases the potential of DL in enhancing NIDS capabilities, including implementing sparse autoencoder and soft-max regression to improve effectiveness.

Ahmad et al. conducted a systematic study comparing machine learning and deep learning approaches in NIDS [3]. Their study provides insights into the performance of different techniques.

Ashiku and Dagli presented a deep learning-based NIDS at the Complex Adaptive Systems Conference, highlighting the role of big data, IoT, and AI in NIDS development [4]. Their work demonstrates the application of DL in real-world scenarios.

Haugerud et al. proposed a dynamic and scalable parallel NIDS using intelligent rule ordering and network function virtualization [5]. Their approach focuses on improving NIDS efficiency and scalability.

Elmubarak et al. implemented a hybrid NIDS system combining the anomaly Holt Winter algorithm and signature-based scheme, showcasing the effectiveness of hybrid approaches in NIDS [6].
Thirimanne et al. developed a real-time NIDS based on deep neural networks, emphasizing the importance of real-time detection in cybersecurity [7]. Researchers have explored the use of generative adversarial networks (GANs) in cybersecurity. Tasneem et al. discussed the challenges and opportunities of using GANs for cybersecurity [9], while Chenna highlighted the application of GANs for generating synthetic data and in cybersecurity [11]. Dunmore et al. conducted a comprehensive survey of GANs in cybersecurity intrusion detection, providing an overview of the advancements in this area [12]. In the realm of cybersecurity, the integration of machine learning (ML), deep learning (DL), and Generative Adversarial Networks (GANs) has shown promise. GANs can create synthetic cyberattack data that mimics real-world patterns, aiding intrusion detection and prevention. Additionally, combining GANs with models like Random Forest, K-Nearest Neighbors, and Support Vector Machines enhances attack detection. However, training GANs for cybersecurity remain complex due to convergence challenges. Analyzing abnormal data generated by GANs and exploring their synergy with large language models (LLMs) can further improve cybersecurity readiness [11][12]. Moreover, Dunmore et al. conducted a comprehensive survey of Generative Adversarial Networks (GANs) in cybersecurity intrusion detection, providing an in-depth overview of the advancements and applications of GANs in this field [13]. This survey serves as a valuable resource for understanding the current state and future directions of using GANs in cybersecurity.

Gap studies model

1. Limited availability of diverse and large-scale datasets hinders the development and evaluating of robust machine learning models for network intrusion detection systems.
2. Challenges in handling class imbalances and overlapping data in training datasets can lead to overfitting and reduced model generalization.
3. The complexity of training Generative Adversarial Networks (GANs) for generating synthetic attack data poses challenges in effectively simulating real-world cyber threats.
4. Further research is also using Deep learning and ML but it needs more computation power.
5. Real-time detection capabilities are crucial for network intrusion detection systems to effectively identify and respond to cyber threats as they occur.

RESEARCH METHODOLOGY

The research methodology for the Intrusion Detection and Prevention System (IDPS) leverages Machine Learning (ML) for anomaly detection, integrating the strengths of Golang and Python for real-time performance and efficient ML development.

Network Traffic Analysis and Signature-Based Detection (Golang):

1. Packet Capture and Decoding: Utilizing Golang to capture raw network packets and decode them efficiently for analysis.
2. Signature Matching: Implementing predefined signatures from Snort rules or similar sources for signature-based detection, identifying known attack patterns.
3. Matching Function: Developing a matching function in Golang to compare captured network traffic against signatures and identify potential threats.

Anomaly Detection with Machine Learning (Python):

1. Data Preprocessing and Feature Engineering: Preprocessing and feature engineering of data from network traffic analysis in Python, potentially including features extracted from captured packets.
2. Model Training: Training supervised learning algorithms like Random Forest, KNN, or SVM using labeled datasets containing normal and attack traffic data.

3. Model Conversion: Converting trained models into a format like ONNX for seamless integration with Golang, facilitating efficient deployment within the IDPS.

Integrated Detection and Response (Golang):

1. Prediction and Decision Making: Feeding captured network traffic into the deployed ML models (in ONNX format) for real-time prediction. Combining results from signature matching and ML models to make informed decisions about potential threats.
2. Action and Alert Generation: Taking actions such as blocking suspicious traffic or generating alerts based on the decisions made.
3. Data Collection and Feedback Loop: Feeding both signature-based and anomaly-detection data into the GAN model for continuous improvement.

Generative Adversarial Network (GAN) for Enhanced Anomaly Detection (Python):

1. Data Augmentation: Using a pre-trained GAN model to generate synthetic anomaly data that complements real-world data, addressing limitations in real-world datasets.
2. GAN Training and Evaluation: Continuously training and evaluate the GAN model in a simulated environment to ensure the generated synthetic data effectively represents real-world attack patterns.
3. Integration with Anomaly Detection Models: Incorporating successful synthetic data generated by the GAN into the training data for the ML anomaly detection models to enhance their ability to identify novel attacks.

Additional Considerations:

Feature Selection: Carefully selecting relevant features from network traffic data to improve ML model performance, using techniques like feature importance analysis. Hyperparameter Tuning: Optimizing hyperparameters of the ML algorithms for optimal performance using techniques like grid search or randomized search. Continuous Monitoring and Improvement: Regularly monitoring the IDPS effectiveness and retraining the ML models with new data to maintain high detection accuracy and adapt to evolving attack patterns.

Tools and Hardware Used:

Coding and Preprocessing: Visual Studio Code for Golang coding and preprocessing in Python. ML Model Training: Jupyter Notebook for training ML models. System Configuration: Own system with 8 GB RAM, integrated Radeon graphics, and a 6-core AMD Ryzen 5 5500U processor. The work flow diagram.

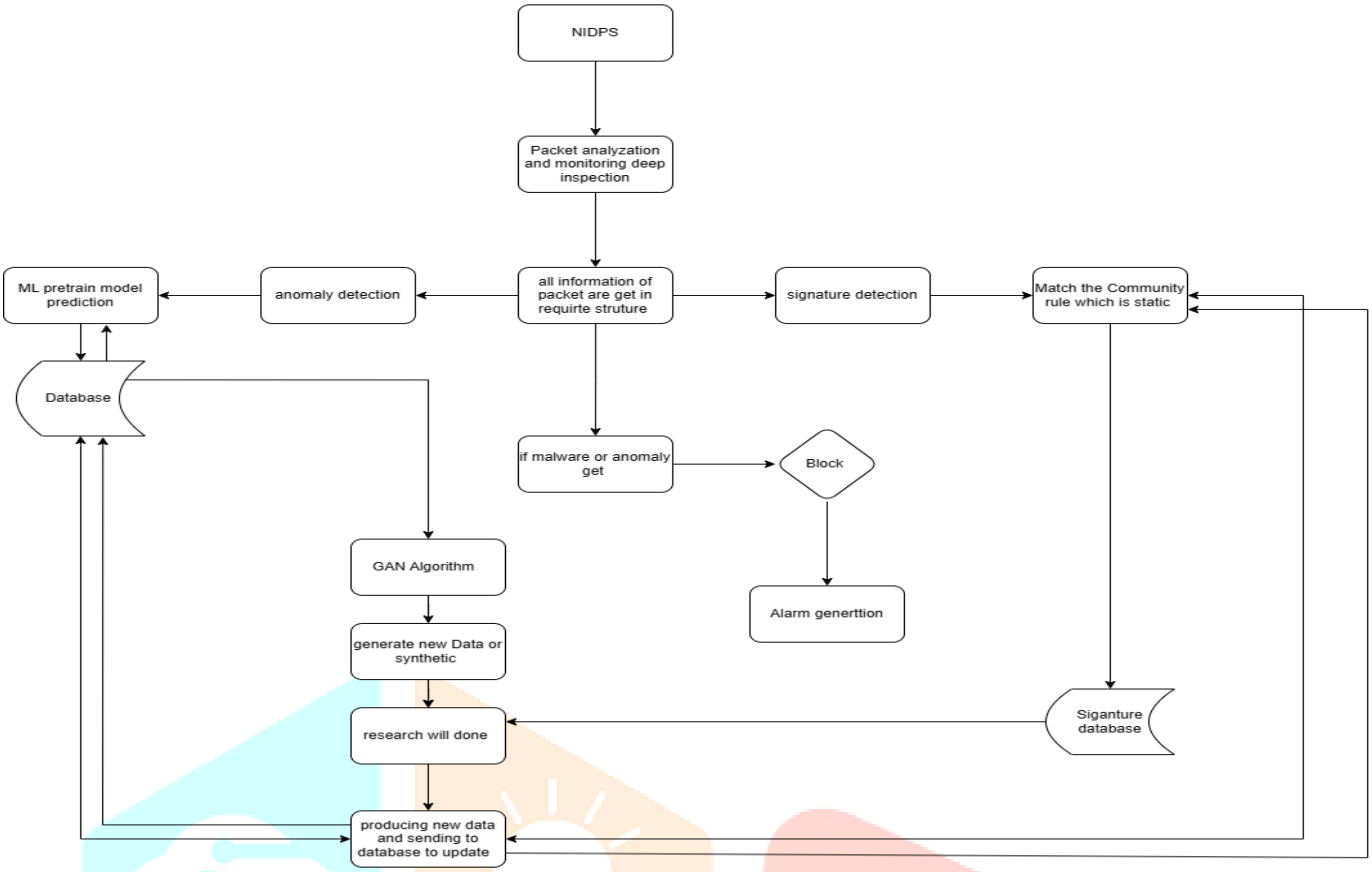
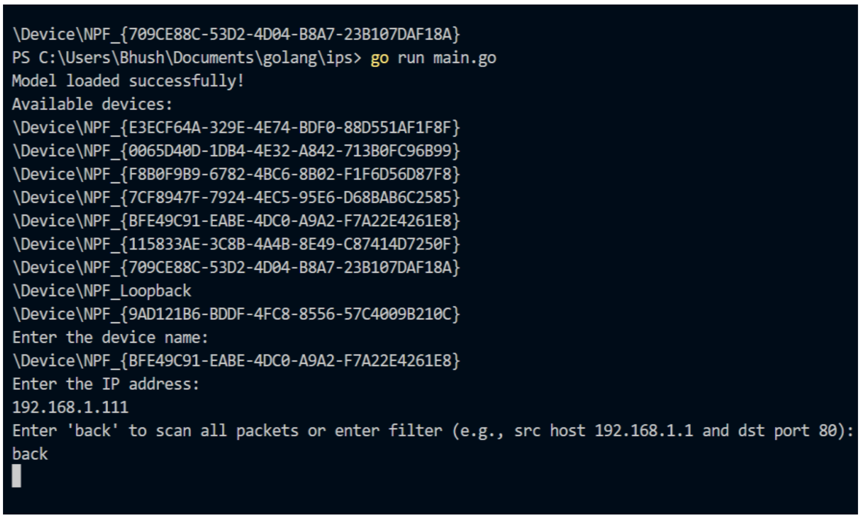

NIDPS
Packet analyzation and monitoring deep inspection
all information of packet are get in requirte struture
anomaly detection
ML pretrain model prediction
Database
signature detection
Match the Community rule which is static
if malware or anomaly get
Block
Alarm generttion
GAN Algorithm
generate new Data or synthetic
research will done
Siganture database
producing new data and sending to database to update

Fig (1)

RESULTS AND DISCUSSION

Results of the system

The project on the Network Intrusion Detection and Prevention System (NIDPS) in Golang has made significant advancements. Using an anomaly dataset similar to KDDCUP 99 from Kaggle, the Random Forest model achieved an impressive 99% accuracy, while KNN performed slightly lower at 98% accuracy, and for SVM I received an accuracy of 94 %. However, on the more challenging UNSW dataset, the Random Forest model achieved 86% accuracy, and KNN achieved 67% accuracy for SVM is not because, for the initial stage, I used two models. Additionally, a GAN was successfully trained to generate new data, which was saved in a .csv file. The models were converted to the ONNX format, enabling their use in any language. In particular, the Random Forest model was successfully loaded in Golang and is currently operational in the background. This project demonstrates the effectiveness of machine learning models in detecting and preventing network intrusions, with the potential for further improvements and real-world applications. At last, my system looks like fig (2)



```
\Device\NPF_{709CE88C-53D2-4D04-B8A7-23B107DAF18A}
PS C:\Users\Bhush\Documents\golang\ips> go run main.go
Model loaded successfully!
Available devices:
\Device\NPF_{E3ECF64A-329E-4E74-BDF0-88D551AF1F8F}
\Device\NPF_{0065D40D-1DB4-4E32-A842-713B0FC96B99}
\Device\NPF_{F8B0F9B9-6782-4BC6-8B02-F1F6D56D87F8}
\Device\NPF_{7CF8947F-7924-4EC5-95E6-D68BAB6C2585}
\Device\NPF_{BFE49C91-EABE-4DC0-A9A2-F7A22E4261E8}
\Device\NPF_{115833AE-3C8B-4A4B-8E49-C87414D7250F}
\Device\NPF_{709CE88C-53D2-4D04-B8A7-23B107DAF18A}
\Device\NPF_Loopback
\Device\NPF_{9AD121B6-BDDF-4FC8-8556-57C4009B210C}
Enter the device name:
\Device\NPF_{BFE49C91-EABE-4DC0-A9A2-F7A22E4261E8}
Enter the IP address:
192.168.1.111
Enter 'back' to scan all packets or enter filter (e.g., src host 192.168.1.1 and dst port 80):
back
```

Fig (2)

CONCLUSION

This research successfully built a foundation for a real-time, ML-based Network Intrusion Detection and Prevention System (NIDS) leveraging Golang's network handling and Python's machine-learning capabilities. While the core functionality is established, future work should focus on a user-friendly interface, scalability for growing data volumes, and continuous learning through automatic data updates and exploration of advanced anomaly detection techniques. By addressing these considerations, we can create a robust, adaptable, and user-friendly NIDS to safeguard our digital infrastructure.

REFERENCES

1. Zoghi, Z., & Serpen, G. (n.d.). UNSW-NB15 Computer Security Dataset: Analysis through Visualization. Electrical Engineering & Computer Science, University of Toledo, Ohio, USA.
2. Niyaz, Q., Sun, W., Javaid, A. Y., & Alam, M. A. (n.d.). A Deep Learning Approach for Network Intrusion Detection System. College Of Engineering, The University of Toledo, Toledo, OH-43606, USA. {quamar.niyaz, weiqing.sun, Ahmad.javaid, mansoor.alam2}@utoledo.edu
3. Musa, U. S., & Chhabra, M. (Year of Publication). Intrusion Detection System using Machine Learning Techniques: A Review. Department of Computer Science & Engineering, Sharda University, Greater Noida, Uttar Pradesh, India. Email: usmanmusa04@gmail.com, Megha.chhbr@gmail.com.
4. Ahmad Z, Shahid Khan A, Wai Shiang C, Abdullah J, Ahmad F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Trans Emerging Tel Tech. 2021;32:e4150. https://doi.org/10.1002/ett.415
5. Ashiku, L., & Dagli, C. (2021). Network Intrusion Detection System using Deep Learning. In Complex Adaptive Systems Conference Theme: Big Data, IoT, and AI for a Smarter Future. Malvern, Pennsylvania, June 16-18, 2021. Department of Engineering Management and Systems Engineering, Missouri University of Science and Technology, Rolla, MO 65401, USA.
6. Haugerud, H., Tran, H. N., Aitsaadi, N., & Yazidi, A. (n.d.). A Dynamic and Scalable Parallel Network Intrusion Detection System using Intelligent Rule Ordering and Network Function Virtualization.
7. Elmubarak, M., Karrar, A., & Hassan, N. (2019). Implementation of a Hybrid (NIDS) System using Anomaly Holt Winter Algorithm and Signature-based Scheme. *International Journal of Advanced Scientific Research & Engineering*, 5(6), doi:10.31695/IJASRE.2019.33278. E-ISSN: 2454-8006.

8. Thirimanne, S. P., Jayawardana, L., Yasakethu, L., & Hewage, C. (2022). Deep Neural Network Based Real-Time Intrusion Detection System. SN Computer Science, 3, 145. doi:10.1007/s42979-022-01031-1.
9. Cilloni, T., & Fleming, C. (2023). Privacy Threats in Stable Diffusion Models. University of Mississippi, Oxford, MS; Cisco, San Jose, CA. Email: tcilloni@go.olemiss.edu, chflemin@cisco.com. arXiv:2311.09355v1 [cs.CV].
10. Tasneem, S., Gupta, K. D., Roy, A., & Dasgupta, D. (2023). Generative Adversarial Networks (GAN) for Cyber Security: Challenges and Opportunities. Conference Paper. Retrieved from https://www.researchgate.net/publication/366962736.
11. Wang, S. (2017). Generative Adversarial Networks (GAN): A Gentle Introduction. Presentation. Department of Statistics and Data Science, University of Texas at Austin.
12. Chenna, S. (n.d.). Application of Generative Adversarial Networks (GANs) for Generating Synthetic Data and in Cybersecurity.
13. Dunmore, J. Jang-Jaccard, F. Sabrina and J. Kwak, "A Comprehensive Survey of Generative Adversarial Networks (GANs) in Cybersecurity Intrusion Detection," in IEEE Access, vol. 11, pp. 76071-76094, 2023, doi: 10.1109/ACCESS.2023.3296707.

.