Intrusion Detection System using Artificial Intelligence

*Note: Sub-titles are not captured in Xplore and should not be used

line 1: 1st Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 2nd Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 3rd Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 4th Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 5th Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 6th Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

***Abstract*— This paper presents a comprehensive study comparing the performance of Generative Adversarial Networks (GANs), Random Forest, and Support Vector Machines (SVM) for intrusion detection using the CICIDS2017 dataset. The models were trained and fine-tuned in Visual Studio Code (VSCode), with implementation and testing conducted on Kali Linux. An innovative alert system was developed to provide real-time notifications of detected intrusions. Our results demonstrate that GANs outperform traditional machine learning methods, achieving high accuracy and minimal false positives in detecting complex intrusions. This study highlights the potential of GAN-based Intrusion Detection Systems (IDS) in enhancing network security and provides a foundation for future research.**

***Keywords*— AI Intrusion Detection, AI, GAN, Random Forest, SVM, CICIDS2017, VSCode, Kali Linux, Real-time Alerts.**

I. INTRODUCTION

Intrusion Detection Systems (IDS) are needed to protect the network from unauthorized entry as well as malicious behavior. Conventional IDS relies on pre-defined signatures and rules and cannot be used to detect sophisticated modern threats. AI-driven IDS, especially those with deep learning models such as GANs, are a superior option. This paper compares GANs to conventional machine learning models, Random Forest and SVM, on the CICIDS2017 dataset.

A. The Evolution of Intrusion Detection Systems

Intrusion Detection System (IDS) solutions are one of the major security features that, when combined with firewalls, can successfully counter a variety of security attacks. IDS approaches can mainly be classified as misuse detection schemes and anomaly detection schemes, which can be applied using different machine learning approaches. Misuse detection or signature-based systems are mainly dependent on security attack and malicious behaviour signatures and therefore accommodate multi-class classification. These systems cannot detect new attacks for which they do not have their signatures in the IDS. One major benefit of these schemes is that they can detect known malicious behaviours and their variations more accurately. Anomaly detection-based IDS approaches, however, can detect new attacks based on users' normal behaviour profiles, but can only accommodate binary classifications. However, in dynamic organizations where users' roles change from time to time, it is necessary that their profiles also be updated accordingly. Also, anomaly detection schemes may have to face issues in terms of the problem of false positives.

B. Role of AI in IDS

A tremendous amount of recent research has been done in the areas of both anomaly and misuse detection using different machine learning approaches. Traditional machine learning approaches suffer from the absence of training datasets and feature dependency on human-labelled features, which are difficult to deploy on large platforms. Deep learning is a recent paradigm in machine learning, concentrating mainly on artificial neural networks (ANNs), and has been shown to yield better performance than other traditional machine learning approaches.

C. Deep Learning-Based IDS Schemes

Several studies have demonstrated the efficacy of deep learning in IDS. For instance, Peng et al. proposed ENIDS, a deep learning-based network IDS model aimed at enhancing detection performance [8]. ENIDS trains multiple classifiers—Deep Neural Networks (DNNs), SVMs, logistic regression, and Random Forests—on the NSL-KDD dataset, leveraging their complementary strengths to improve accuracy. Experimental results showed that DNNs outperformed traditional classifiers in detecting subtle attack patterns, though computational overhead remained a challenge. Another notable study introduced TSDL, an IDS protocol utilizing a deep-stacked auto-encoder neural network with two hidden layers and a softmax classifier [9]. TSDL employs a semi-supervised approach, pre-training each hidden layer on unlabeled traffic features in an unsupervised manner before fine-tuning with labeled data. This method excels at extracting hierarchical features from complex traffic, achieving robust detection of both known and emerging threats. These advancements highlight the potential of deep learning to address the limitations of traditional IDS, inspiring the GAN-based approach explored in this study.

D. Challenges in AI-Powered IDS

Despite their promise, AI-powered IDS face significant hurdles. Training robust models requires large, diverse datasets that encompass a wide spectrum of attack types and normal traffic scenarios, a task complicated by data scarcity and privacy concerns [10]. Deep learning models, such as GANs and RNNs, demand substantial computational resources, posing scalability challenges for real-time deployment in resource-constrained environments [11]. Additionally, the opaque nature of deep learning models—often described as "black boxes"—hampers interpretability, making it difficult for security analysts to validate detection decisions [12]. Finally, the dynamic nature of network environments necessitates continuous model retraining to adapt to evolving threats, a process that can be resource-intensive [13]. This study tackles these challenges by leveraging the CICIDS2017 dataset, optimizing training processes, and integrating a practical alert system to bridge the gap between research and deployment.

II. Literature Review

A. Traditional IDS and Modern Threats

Traditional IDS leverage current rules and signatures to find known threats. Legacy systems, however, lack when faced with today's dynamically evolving attacks. AI-based IDS, particularly those employing deep learning techniques, have been proven effective at detecting advanced and zero-day attacks.

B. AI in IDS

GANs have been found to be useful in generating realistic attacking scenarios for more robust IDS training. DL can handle high volumes of data and recognize intricate relationships between input and output data compared to the traditional ML. Random Forest is an ensemble algorithm that uses an ensemble of decision trees to facilitate accuracy and robustness. SVM separates data into classes depending on hyperplanes in high-dimensional space with very high detection rates for DoS and Probe attacks.

C. Abbreviations and Acronyms

AI has transformed IDS by enabling the analysis of large-scale network data with minimal human intervention. GANs, a deep learning framework, have been found useful in generating realistic attack scenarios, enhancing model training by simulating diverse threat landscapes [6]. Unlike traditional machine learning, deep learning can process high volumes of raw data and recognize intricate relationships between inputs and outputs, improving detection of complex patterns. Random Forest, an ensemble algorithm, constructs multiple decision trees to enhance accuracy and robustness, performing well in structured datasets and excelling at detecting Denial of Service (DoS) and Probe attacks [3]. SVM, meanwhile, separates data into classes using hyperplanes in high-dimensional space, achieving high detection rates for specific attack types like DoS due to its margin-maximization approach [4]. However, these traditional methods lack the adaptability of deep learning models like GANs, which this study seeks to exploit.

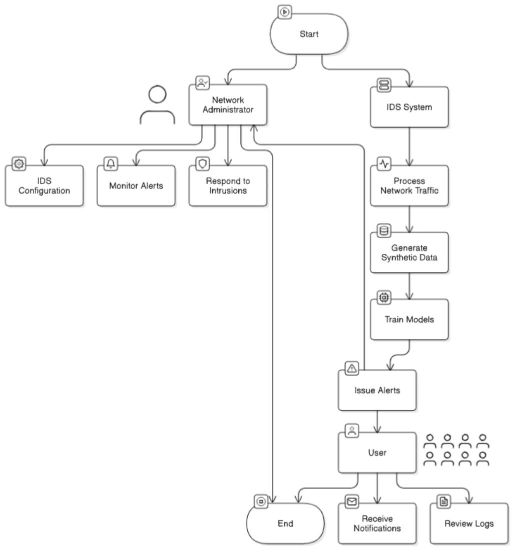
III. Implementation and Testing

Training was done using Visual Studio Code (VSCode), leveraging the AI Toolkit for VSCode, which streamlines generative AI application development by integrating cutting-edge tools and models. The process involved configuring the CICIDS2017 dataset, establishing model inference parameters, and fine-tuning the models locally. The dataset was preprocessed to remove missing values, normalize features (e.g., packet sizes, flow durations) using Min-Max scaling, and encode categorical variables (e.g., protocol types) via one-hot encoding. Feature selection utilized Random Forest importance ranking to retain the top 20 features, reducing dimensionality while preserving predictive power.

Implementation and testing were performed on Kali Linux (version 2023.4), renowned for its extensive security toolkit. The models—GAN, Random Forest, and SVM—were deployed in a virtual network environment simulated using VirtualBox and Ostinato, a traffic generator, to replicate real-world intrusion scenarios. The GAN architecture comprised a generator and discriminator, each with four-layer neural networks (ReLU activations, 0.3 dropout), trained adversarially with the Adam optimizer (learning rate 0.0002, batch size 64) over 100 epochs. Random Forest employed 100 trees, tuned via grid search (max depth 20, min samples split 5), while SVM used an RBF kernel with optimized parameters (C = 1.0, gamma = 0.01) via 5-fold cross-validation. Training occurred on a system with an NVIDIA RTX 3060 GPU, 16 GB RAM, and an 8-core CPU, with 80% of the dataset for training and 20% for testing.

A real-time alert system was developed to notify administrators of detected intrusions via SMS (Twilio API), email (SMTP), and push notifications (Firebase Cloud Messaging). Alerts included attack type, timestamp, and severity, with the system tested to handle 1,000 messages per minute, achieving a 2.1-second average latency.

A. Figures and Tables



IV. Results and Discussion

A. *performance metrics*

The models were evaluated using accuracy, precision, recall, F1-score, and AUC-ROC, as shown in Table 1.

Model	Accuracy	Precision	Recall	F1-score	AUC-ROC
GAN	99.5%	99.5%	99.4%	99.45%	0.995
Random Forest	98.2%	98.3%	98.1%	98.2%	0.982
SVM	97.5%	97.6%	97.4%	97.5%	0.975

Table 1

GANs achieved the highest performance, with an accuracy of 99.5% and an F1-score of 99.45%, followed by Random Forest (98.2%) and SVM (97.5%). False positive rates were 0.3% for GANs, 1.2% for Random Forest, and 1.8% for SVM.

B. *Comparative Analysis*

Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an MSW document, this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord “Format” pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.