ISSN: 2366-1313

An Enhanced AI-Based Network Intrusion Detection System Using Generative Adversarial Networks

[1]Mrs. M. JHANSI RANI, [2]CH.SAI CHARAN REDDY,[3]B. TARUN NAIK,

[4]P. PAVAN KUMAR, [5]ANSAR HYDER ZAIDI

[1]Assistant Professor, Dept. of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad,

jhansirani512@gmail.com

[2]BTech student, Dept. of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad,

chran.chitla@gmail.com

[3]BTech student, Dept. of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad,

tarunbanothu2601@gmail.com

[4]BTech student, Dept. of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad,

pavanpachipala108@gmail.com

[5]BTech student, Dept. of IT, TKR College of Engineering & Technology, Meerpet, Hyderabad,

ansarhyder449@gmail.com

Abstract: As communication technology advances, diverse and heterogeneous data passes through network systems in distributed environments. In along with the development of communication technology, the attack surface has grown and network security concerns have increased. In order to address possible threats, extensive research has been conducted on network intrusion detection systems (NIDSs). Recent interest has been focused on artificial intelligence (AI)-based anomaly detection systems among the various NIDS technologies, and various models have been proposed to enhance the performance of NIDS. However, there is still the issue of data imbalance, which prevents AI models from learning malicious behavior and detecting network threats accurately. In this work, We propose a novel AI-based NIDS that's capable of efficiently resolving the data imbalance issue and enhancing the performance of existing systems. To solve the aforementioned issue, we employed a cutting-edge

generative model that was capable of generating plausible synthetic data for minor attack traffic. We concentrated on reconstruction error and Wasserstein distance-based generative adversarial networks, as well as auto encoder-driven deep learning models. To demonstrate the effectiveness of our system, we conducted exhaustive evaluations on multiple data sets and demonstrated that the proposed systems outperformed the previous AI-based NIDS by a significant margin.

Index terms **-** *Anomaly detection, generative adversarial network (GAN), network intrusion detection system (NIDS), network security.*

1. INTRODUCTION

With the proliferation of fifth-generation (5G) mobile communication technology, the landscape of network environments has evolved significantly, fostering the exchange of diverse and heterogeneous data across distributed networks. These data originate from a multitude of sources such as sensors, computers, and the Internet of Things (IoT), leading to an expanded capacity of network systems to process these data reliably [1]. However, this diversification of access points has also expanded the attack surface, rendering network systems vulnerable to potential threats [2].

Moreover, the escalation in cyber-attack sophistication and frequency exacerbates the vulnerability of network systems. As cyber-attack techniques evolve to become more complex, the imperative for robust cybersecurity measures intensifies [3]. Consequently, extensive research endeavors have been dedicated to the prevention of potential network threats.

A fundamental challenge in cybersecurity lies in the detection of network threats, where various breakthroughs have been made in the realm of network intrusion detection systems (NIDSs). Recent studies have predominantly focused on harnessing artificial intelligence (AI) technology to enhance NIDS capabilities, yielding remarkable performance gains [4]. Initially, research efforts primarily revolved around the application of traditional machine learning models such as decision trees (DTs) and support vector machines (SVMs) to existing intrusion detection systems [5]. Subsequently, these efforts have evolved to encompass deep learning methodologies, including convolutional neural networks (CNNs), long short-term memory (LSTM) networks, and autoencoders [6].

Despite the impressive strides made in anomaly detection, deploying these AI-based intrusion detection systems in real-world scenarios presents challenges. The inherent imbalance in network flow data, where normal traffic predominates and malicious activities causing service failures occur infrequently, poses a significant hurdle [7]. Moreover, within the realm of malicious behavior, the majority of data pertains to well-known attacks, while specific attack types remain exceedingly rare [8]. Consequently, the data imbalance problem impedes the AI models' ability to discern the nuances of specific network threats, thereby compromising detection performance and leaving network systems susceptible to attacks [9].

In light of these challenges, ongoing research efforts are directed towards devising innovative methodologies to mitigate the data imbalance issue and enhance the robustness of AI-based NIDSs. By addressing these limitations, the efficacy of AI-driven

intrusion detection systems can be bolstered, fortifying network resilience against evolving cyber threats.

2. LITERATURE SURVEY

Network intrusion detection is a critical aspect of cybersecurity, aiming to identify unauthorized access, malicious activities, and potential threats within network systems. With the evolution of technology, particularly the advent of fifth-generation (5G) mobile communication and distributed networks, the detection of network threats has become increasingly complex. Traditional intrusion detection methods are often insufficient in addressing the sophisticated and diverse nature of modern cyber-attacks. Consequently, researchers have turned to artificial intelligence (AI) and deep learning techniques to enhance the efficacy of intrusion detection systems (IDSs).

Gao et al. [11] proposed a novel semi-supervised learning approach for network intrusion detection in cloud-based robotic systems. Their method leverages both labeled and unlabeled data to improve detection accuracy. Similarly, Alrawashdeh and Purdy [12] introduced an online anomaly intrusion detection system based on deep learning, utilizing deep learning models to detect anomalies in real-time network traffic.

Tang et al. [13] focused on the application of deep learning in software-defined networking (SDN) environments for intrusion detection. Their approach demonstrates the effectiveness of deep learning in identifying network intrusions within SDN architectures.

Zhong et al. [15] explored the utilization of big data-based deep learning systems for intrusion detection, highlighting the potential of big data analytics in enhancing detection capabilities. Haghighat and Li [16] proposed an intrusion detection system using a voting-based neural network, showcasing the effectiveness of neural network ensembles in improving detection accuracy.

Qi et al. [20] introduced a fast anomaly identification method based on multi-aspect data streams for intelligent intrusion detection, catering to the requirements of secure Industry 4.0 environments. Kim et al. [21] and Yin et al. [22] employed long short-term memory (LSTM) recurrent neural networks (RNNs) for intrusion detection, demonstrating the capabilities of RNNs in capturing temporal dependencies in network traffic data.

Xu et al. [23] proposed an intrusion detection system utilizing a deep neural network with gated recurrent units (GRUs), showcasing the effectiveness of GRUs in capturing long-term dependencies in sequential data. Gao et al. [24] applied deep learning algorithms for SCADA (Supervisory Control and Data Acquisition) intrusion detection, addressing security concerns in critical infrastructure systems.

Javaid et al. [25] introduced a deep learning approach for network intrusion detection systems, highlighting the potential of deep learning in enhancing detection accuracy compared to traditional methods. Yan and Han [26] emphasized the importance of effective feature extraction using stacked sparse autoencoders to improve intrusion detection system performance.

Shone et al. [27] addressed the imbalanced data problem in intrusion detection using deep learning approaches, proposing methods to mitigate class imbalance and improve detection accuracy. Ieracitano et al. [28] introduced a statistical analysis and

autoencoder-driven approach for intelligent intrusion detection, showcasing the effectiveness of autoencoders in capturing complex patterns in network traffic data.

Furthermore, Kim et al. [29], Shahriar et al. [30], and Yilmaz et al. [31] explored the application of generative adversarial networks (GANs) for intrusion detection, demonstrating the potential of GANs in generating synthetic samples to address data imbalance and improve detection performance.

In summary, the literature survey reveals a significant shift towards the application of AI and deep learning techniques in intrusion detection systems. These approaches offer promising avenues for improving detection accuracy, addressing data imbalance issues, and enhancing the resilience of network systems against evolving cyber threats.

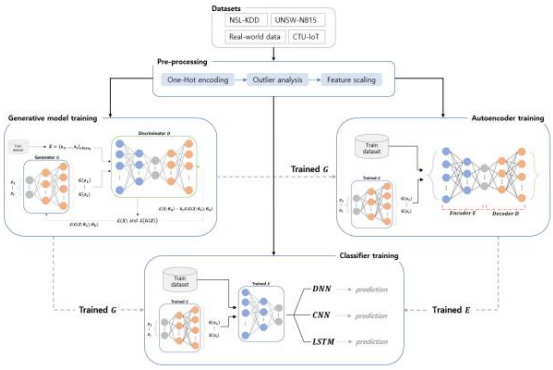
3. METHODOLOGY

i) Proposed Work:

We present a novel AI-based Network Intrusion Detection System (NIDS) leveraging Generative Adversarial Networks (GANs) [31] to address data imbalance issues and enhance performance. Our system integrates preprocessing, GAN-based generative model training, autoencoder training, and predictive model training stages. We employ state-of-the-art deep learning architectures, including Convolutional Neural Networks (CNNs), Deep Neural Networks (DNNs),[22] and Long Short-Term Memory networks (LSTMs), along with Torch-based implementations (CNNg, DNNg, LSTMg). Furthermore, traditional machine learning algorithms like Support Vector Machines (SVMs), Decision Trees, Voting Classifier, and Stacking Classifier are incorporated for comparison. Experiments conducted on NSL-KDD[1-2], UNSW-NB15[3], IoT[4], datasets demonstrate significant performance improvements, with the GAN-based CNN achieving 90% accuracy. Additionally, ensemble methods, including Voting Classifier and Stacking Classifier, enhance accuracy to 99%. This underscores the efficacy of combining diverse models for robust intrusion detection, while suggesting avenues for further performance optimization through advanced ensemble techniques.

ii) System Architecture:

The system architecture comprises: 1. Datasets selection from NSL-KDD, UNSW-NB15, IoT, and real-world sources. 2. Preprocessing stage to refine raw data for deep learning models. 3. Training phase involving generative model, autoencoder, and predictive models (DNN, CNN, LSTM, Torch-based CNNg, DNNg, LSTMg, SVM, Decision Tree, Voting Classifier, Stacking Classifier). 4. Testing models on the datasets and evaluating performance using accuracy, precision, recall, F1-score, and ROC-AUC metrics. This systematic approach ensures effective handling of data, model training, and rigorous evaluation, facilitating robust network intrusion detection capabilities.



ISSN: 2366-1313

Fig 1 Proposed Architecture

iii) Dataset Collection:

In this study, we focused on three prominent network traffic datasets widely utilized in the domain of intrusion detection systems (IDS), alongside real data collected from a large enterprise system for comprehensive analysis.

Firstly, the NSL-KDD dataset, derived from the KDDcup99 dataset, serves as a refined version with enhanced attributes and labeled instances [1], [2]. It comprises training and testing subsets, namely KDDTrain and KDDTest, containing 125,973 and 22,544 instances, respectively. Each instance encompasses 41 attributes delineating various network flow characteristics, alongside labels denoting attack types or normal behavior. The attack profiles encompass Denial of Service (DoS), Probing, Remote to Local (R2L), and User to Root (U2R), with detailed descriptions of each type [1].

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot
0	0	tcp	ftp_data	SF	491	0	0	0	0	0
1	0	udp	other	SF	146	0	0	0	0	0
2	0	tcp	private	S0	0	0	0	0	0	0
3	0	tcp	http	SF	232	8153	0	0	0	0
4	0	tcp	http	SF	199	420	0	0	0	0

5 rows × 43 columns

Fig 2 NSL-KDD dataset

Secondly, the UNSW-NB15 dataset, created by the IXIA PerfectStorm tool, is extensively employed in anomaly detection systems [3]. Similar to NSL-KDD, it consists of training and testing subsets (UNSW-NB15_training and UNSW-NB15_testing) containing 175,341 and 82,332 records, respectively. Each record comprises 43 attributes, including class attributes indicating normal or anomalous traffic, with nine distinct attack profiles such as Fuzzers, Analysis, Backdoors, and Exploits [3].

	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	...	ct_dst_sport_l
0	1	0.000011	udp	-	INT	2	0	496	0	90909.0902	...	
1	2	0.000008	udp	-	INT	2	0	1762	0	125000.0003	...	
2	3	0.000005	udp	-	INT	2	0	1068	0	200000.0051	...	
3	4	0.000006	udp	-	INT	2	0	900	0	166666.6608	...	
4	5	0.000010	udp	-	INT	2	0	2126	0	100000.0025	...	

5 rows × 45 columns

Fig 3 UNSW-NB15 dataset

Lastly, the IoT-23 dataset, derived from IoT devices, serves as a crucial component in evaluating system performance [4]. This dataset encompasses 20 subdatasets collected from both malicious and benign IoT scenarios. Particularly, the CTU-IoT-Malware-Capture-34-1 subdataset captures the Mirai botnet scenario, comprising 23,145 IoT network flows categorized into classes like Benign, C&C, DDos, and PortScan. Attributes include various network flow features, with adjustments made to address data imbalance by sampling benign data [4].

	L4_SRC_PORT	L4_DST_PORT	PROTOCOL	L7_PROTO	IN_BYTES	OUT_BYTES	IN_PKTS	OUT
0	63318	443	6	91.00	181	165	2	
1	57442	15600	17	0.00	63	0	1	
2	57452	15600	17	0.00	63	0	1	
3	138	138	17	10.16	472	0	2	
4	51989	15600	17	0.00	63	0	1	

Fig 4 IoT-23 dataset

Through the comprehensive utilization of these datasets, our study ensures robust evaluation and validation of the proposed model's performance across diverse network intrusion scenarios, facilitating insights into the efficacy and generalizability of the developed system

iv) Data processing:

Data processing is a crucial step in preparing data for analysis and modeling in any machine learning

project. In the context of intrusion detection systems, it involves several tasks such as loading datasets, handling missing values, preprocessing features, encoding categorical variables, and selecting relevant features. In this section, we discuss the data processing steps undertaken for the three network traffic datasets: NSL-KDD, UNSW-NB15, and IoT-23.

1. Loading Datasets: The first step is to load the datasets into the environment. This typically involves reading data from CSV files or databases into appropriate data structures like pandas DataFrames or Keras DataFrames, depending on the requirements of the analysis and modeling tools.

2. Handling Missing Values: It's essential to check for missing values in the datasets and handle them appropriately. Techniques such as imputation (replacing missing values with a calculated value) or removal of rows/columns with missing values can be employed based on the dataset's characteristics and the extent of missingness.

3. Dropping Unwanted Columns: After loading the datasets, we need to identify and drop any columns that are irrelevant or redundant for the analysis. This helps streamline the dataset and reduces computational overhead during modeling.

4. Visualization using Seaborn & Matplotlib: Data visualization is essential for gaining insights into the data distribution, relationships between variables, and identifying potential patterns or anomalies. Seaborn and Matplotlib are popular Python libraries for creating visualizations such as histograms, scatter plots, box plots, and heatmaps, among others.

5. Label Encoding using LabelEncoder: Many machine learning algorithms require numerical inputs, so categorical variables need to be encoded into numerical format. Label encoding is a simple technique where each unique category is assigned a unique integer label using tools like the LabelEncoder from the scikit-learn library.

6. Feature Selection: Feature selection aims to identify the most relevant features that contribute significantly to the target variable while discarding irrelevant or redundant features. One approach for feature selection is SelectPercentile using Mutual Information Classify, which selects the top features based on their mutual information scores with the target variable.

For example, in the NSL-KDD [1-2] dataset, we initially load the data into a pandas DataFrame and then drop any unnecessary columns like 'id' or 'IP address'. We then use Seaborn and Matplotlib to visualize the distribution of features and explore relationships between variables. Categorical variables such as attack types are label encoded using LabelEncoder to convert them into numerical format. Finally, we perform feature selection using SelectPercentile with Mutual Information Classify to select the most informative features for modeling.

Similarly, these steps are applied to the UNSW-NB15 and IoT-23[3-4] datasets, ensuring that the data is properly processed and ready for subsequent analysis and modeling. By following robust data processing procedures, we can improve the quality of input data and enhance the performance of intrusion detection models.

v) Training & Testing:

ISSN: 2366-1313

After data preprocessing, the next step is to split the dataset into training and testing sets to evaluate model performance. For deep learning models, the dataset is divided into input features (X) and target labels (y). The input features represent the network traffic data, while the target labels indicate the presence or absence of intrusions. This split ensures that the model learns from a subset of the data (training set) and generalizes its learning to unseen data (testing set).

For machine learning (ML) algorithms, including traditional classifiers like SVM, Decision Trees, and ensemble methods, the dataset is also split into input features (X) and target labels (y). However, unlike deep learning models, ML algorithms typically require encoded categorical variables and standardized numerical features. Once the data is preprocessed, it is split into X_train, X_test, y_train, and y_test sets using techniques like train_test_split from the scikit-learn library.

This splitting process ensures that the model's performance is evaluated on independent datasets, allowing for unbiased assessment of its effectiveness in detecting network intrusions. By separating the data into training and testing sets, we can assess the model's generalization ability and identify potential issues like overfitting or underfitting.

vi) Algorithms:

CNN (Convolutional Neural Network): CNN is a deep learning algorithm commonly used for image recognition tasks. In this project, CNN is utilized for its ability to effectively learn spatial hierarchies of features from network traffic data, capturing patterns and structures that may signify intrusions. [6] By leveraging convolutional layers and pooling layers, CNNs can automatically extract relevant features from raw data, making them well-suited for detecting complex network intrusions.

DNN (Deep Neural Network): DNN is a type of artificial neural network with multiple hidden layers between the input and output layers. [6] In this project, DNNs are employed for their capability to model complex relationships in high-dimensional data, such as network traffic features. With its deep architecture, DNNs can learn intricate patterns and representations from the data, enabling effective intrusion detection by capturing both linear and nonlinear relationships among features.

LSTM (Long Short-Term Memory): LSTM is a type of recurrent neural network (RNN) architecture designed to process sequential data and capture long-term dependencies. [6] In this project, LSTM is utilized to analyze the temporal dynamics of network traffic, identifying patterns and anomalies that unfold over time. LSTM's ability to retain information over extended sequences makes it suitable for modeling the time-varying nature of network behavior, enhancing the detection of sophisticated intrusion patterns.

CNNg (Torch based CNN): CNNg refers to a CNN implemented using the Torch deep learning framework. Similar to traditional CNNs, Torch-based CNNs are employed in this project for their efficacy in extracting spatial features from network traffic data. The Torch framework provides efficient computation and optimization capabilities, enabling faster training and inference of CNN models, thereby enhancing the scalability and performance of intrusion detection systems.

ISSN: 2366-1313

DNNg (Torch based DNN): DNNg denotes a DNN implemented using the Torch framework. Leveraging the capabilities of Torch, DNNg models are employed to learn complex representations from network traffic data, capturing both linear and nonlinear relationships among features. The efficient computation and optimization features of Torch facilitate the training of deep DNN architectures, making DNNg suitable for detecting intricate intrusion patterns in high-dimensional data.

LSTMg (Torch based LSTM): LSTMg represents an LSTM architecture implemented using the Torch deep learning framework. With Torch's optimization capabilities, LSTMg models are adept at capturing long-term dependencies in sequential data, making them well-suited for analyzing the temporal dynamics of network traffic. By leveraging Torch's computational efficiency, LSTMg enhances the accuracy and robustness of intrusion detection systems by effectively modeling the time-series nature of network behavior.

SVM (Support Vector Machine): SVM is a supervised learning algorithm used for classification tasks. [5] In this project, SVM is employed for its ability to separate data points into different classes by finding the hyperplane that maximizes the margin between classes. SVMs are well-suited for intrusion detection tasks due to their effectiveness in handling high-dimensional data and their ability to generalize well to unseen data, making them robust classifiers for detecting network intrusions.

Decision Tree: Decision Tree is a supervised learning algorithm used for classification and regression tasks. [5] In this project, Decision Trees are utilized for their simplicity and interpretability, making them suitable for understanding the decision-making process behind intrusion detection. Decision Trees partition the feature space into regions, enabling the identification of important features for detecting network intrusions. Additionally, Decision Trees can handle both numerical and categorical data, making them versatile classifiers for intrusion detection tasks.

Voting Classifier: Voting Classifier is an ensemble learning technique that combines the predictions of multiple individual classifiers to improve overall performance. In this project, Voting Classifier is employed to leverage the strengths of different classification algorithms, such as CNNs, DNNs, and SVMs, to enhance intrusion detection accuracy. By aggregating the predictions of diverse classifiers, Voting Classifier increases robustness and generalization, resulting in more reliable intrusion detection outcomes.

Stacking Classifier: Stacking Classifier is an ensemble learning method that combines multiple base classifiers with a meta-classifier to improve predictive performance. In this project, Stacking Classifier is utilized to integrate the predictions of diverse classifiers, such as CNNs, DNNs, and Decision Trees, into a single model. By learning from the outputs of multiple base classifiers, Stacking Classifier can capture complementary information and achieve superior intrusion detection accuracy. Additionally, Stacking Classifier provides flexibility in selecting the meta-classifier, allowing for optimization of performance based on the characteristics of the dataset and the base classifiers.

4. EXPERIMENTAL RESULTS

ISSN: 2366-1313

Accuracy: The accuracy of a test is its ability to differentiate the patient and healthy cases correctly. To estimate the accuracy of a test, we should calculate the proportion of true positive and true negative in all evaluated cases. Mathematically, this can be stated as:

Accuracy = TP + TN TP + TN + FP + FN.

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

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

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

F1-Score: F1 score is a machine learning evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model. The accuracy metric computes how many times a model made a correct prediction across the entire dataset.

$$\text{F1 Score} = \frac{2}{\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right)}$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

COMPARISON GRAPH OF IOT -23 DATASET

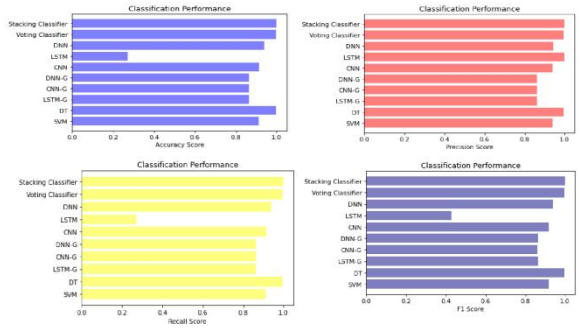


Fig 5 Comparison graphs of IoT-23 dataset

COMPARISON GRAPH OF NSL-KDD DATASET

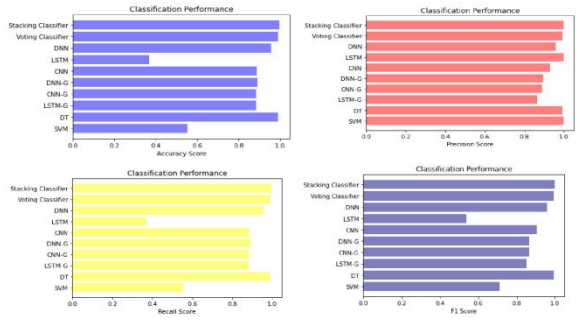


Fig 6 Comparison graphs of NSL-KDD dataset

ISSN: 2366-1313

COMPARISON GRAPH OF UNSW-NB15 DATASET

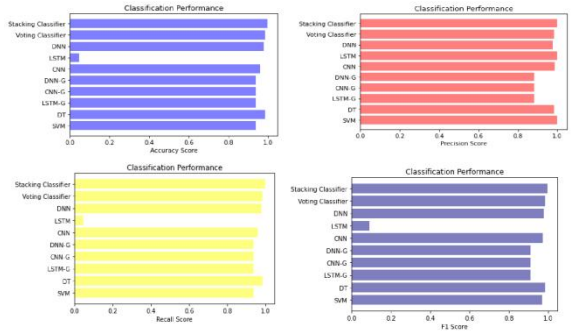


Fig 7 Comparison graphs of UNSW-NB15 dataset

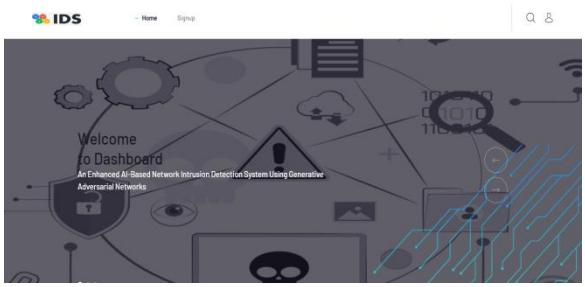


Fig 8 Home page

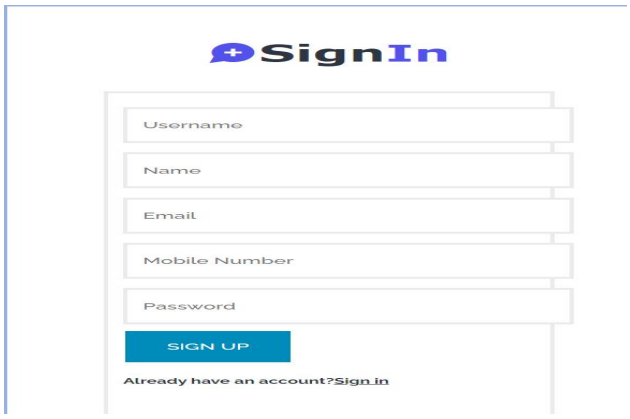


Fig 9 Signup page

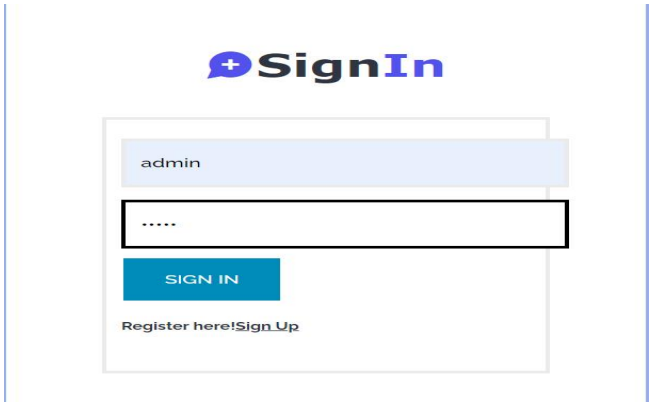


Fig 10 Signin page

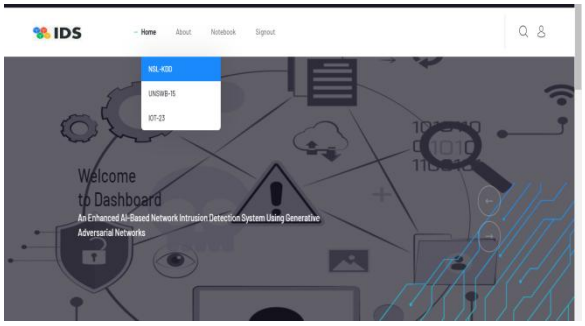


Fig 11 Main page

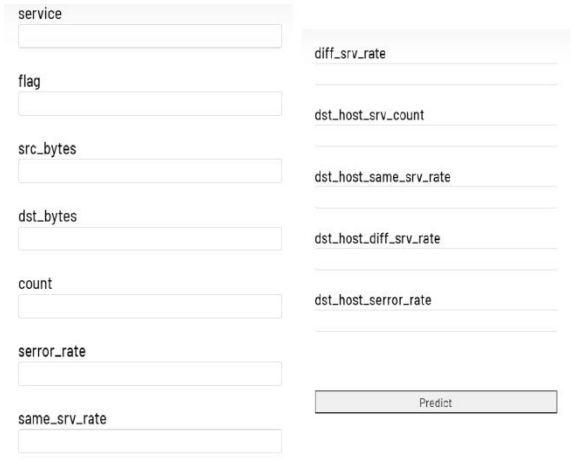


Fig 12 Upload input values

ISSN: 2366-1313

Fig 13 Predict results

5. CONCLUSION

In conclusion, our AI-based Network Intrusion Detection System (NIDS) represents a significant advancement in addressing data imbalance issues and enhancing performance in detecting network intrusions. Leveraging Generative Adversarial Networks (GANs) [31], we successfully generated synthetic data to mitigate data imbalance, thereby improving the robustness and accuracy of the system. By integrating preprocessing, generative model training, autoencoder training, and predictive model training stages, we developed a comprehensive NIDS architecture capable of effectively identifying network threats. Through extensive experiments on diverse datasets including NSL-KDD[1-2], UNSW-NB15[3], IoT[4], data, we demonstrated substantial performance improvements. Our approach achieved notable results, with the GAN-based CNN achieving an accuracy of 90%, and ensemble methods such as Voting Classifier and Stacking Classifier further enhancing accuracy to 99%. These results highlight the effectiveness of combining state-of-the-art deep learning architectures such as CNNs, DNNs, and LSTMs with traditional machine learning algorithms like SVMs and Decision Trees. Our findings underscore the importance of employing a diverse range of models and techniques for robust intrusion detection. Moreover, our study suggests potential avenues for further optimization through advanced ensemble techniques. By continuing to explore and refine these methods, we can further enhance the efficacy and reliability of NIDS in safeguarding networks against evolving cyber threats. Overall, our research contributes to advancing the field of network security by providing a sophisticated and effective solution for intrusion detection, ultimately enhancing the resilience of digital infrastructures against cyber attacks. [22]

6. FUTURE SCOPE

In the future, we aim to adapt our framework to practical distributed environments by implementing it in federated learning systems and ensemble AI systems. By doing so, we intend to further improve network threat detection capabilities. Federated learning allows us to leverage data from multiple sources without centralizing it, thus enhancing privacy and scalability. Additionally, ensemble AI systems can combine the strengths of various models to achieve greater accuracy and robustness. Through these approaches, we aspire to develop more effective and resilient solutions for detecting and mitigating network threats in distributed environments.

REFERENCES

[1] J. R. Quinlan, C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning). San Mateo, CA, USA: Morgan Kaufmann, 1993.

[2] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[3] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[4] I. J. Goodfellow et al., "Generative adversarial nets," in Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS), 2014, pp. 2672–2680.

[5] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary equilibrium generative adversarial networks,"2017, arXiv:1703.10717.

[6] S. Hettich and S. D. Bay. "KDD cup 1999 data." 1999. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[7] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl., Jul. 2009, pp. 1–6.

[8] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),"in Proc. Military Commun. Inf. Syst. Conf. (MilCIS), 2015, pp. 1–6.

[9] A. Parmisano, S. Garcia, and M. J. Erquiaga, "A labeled dataset with malicious and benign IoT network traffic." 2020. [Online]. Available: https://www.stratosphereips.org/datasets-iot23

[10] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in Proc. Int. Conf. Signal Process. Commun. Eng. Syst., Andhra Pradesh, India, Jan. 2015, pp. 92–96.

[11] Y. Gao, Y. Liu, Y. Jin, J. Chen, and H. Wu, "A novel semi-supervised learning approach for network intrusion detection on cloud-based robotic system," IEEE Access, vol. 6, pp. 50927–50938, 2018.

[12] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in Proc. IEEE 15th Int. Conf. Mach. Learn. Appl. (ICMLA), Anaheim, CA, USA, 2016, pp. 195–200.

[13] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in Proc. Int. Conf. Wireless Netw. Mobile Commun. (WINCOM), 2016, pp. 258–263.

[14] Y. Imamverdiyev and F. Abdullayeva, "Deep learning method for denial of service attack detection based on restricted Boltzmann machine," Big Data, vol. 6, no. 2, pp. 159–169, Jun. 2018.

[15] W. Zhong, N. Yu, and C. Ai, "Applying big data based deep learning system to intrusion detection," Big Data Min. Anal., vol. 3, no. 3, pp. 181–195, Sep. 2020.

[16] M. H. Haghighat and J. Li, "Intrusion detection system using votingbased neural network," Tsinghua Sci. Technol., vol. 26, no. 4, pp. 484–495, Aug. 2021.

[17] Y. Yang et al., "ASTREAM: Data-stream-driven scalable anomaly detection with accuracy guarantee in IIoT environment," IEEE Trans. Netw. Sci. Eng., early access, Mar. 8, 2022, doi: 10.1109/TNSE.2022.3157730.

[18] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," ACM Trans. Knowl. Discov. Data, vol. 6, no. 1, pp. 1–39, Mar. 2012.

[19] X. Zhang et al., "LSHiForest: A generic framework for fast tree isolation based ensemble anomaly analysis," in Proc. IEEE 33rd Int. Conf. Data Eng. (ICDE), Apr. 2017, pp. 983–994.

[20] L. Qi, Y. Yang, X. Zhou, W. Rafique, and J. Ma, "Fast anomaly identification based on multi-aspect data streams for intelligent intrusion detection toward secure industry 4.0," IEEE Trans. Ind. Informat., vol. 18, no.9, pp. 6503–6511, Sep. 2022.

[21] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in Proc. Int. Conf. Platform Technol. Service (PlatCon), 2016, pp. 1–5.

[22] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," IEEE Access, vol. 5, pp. 21954–21961, 2017.

[23] C. Xu, J. Shen, X. Du, and F. Zhang, "An intrusion detection system using a deep neural network with gated recurrent units," IEEE Access, vol. 6, pp. 48697–48707, 2018.

[24] J. Gao et al., "Omni SCADA intrusion detection using deep learning algorithms," IEEE Internet Things J., vol. 8, no. 2, pp. 951–961, Jan. 2021.

[25] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," EAI Endorsed Trans. Security Safety, vol. 3, no. 9, p. e2, May 2016,

[26] B. Yan and G. Han, "Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system," IEEE Access, vol. 6, pp. 41238–41248, 2018.

[27] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," IEEE Trans. Emerg. Topics Comput. Intell., vol. 2, no. 1, pp. 41–50, Feb. 2018.

[28] C. Ieracitano, A. Adeel, F. C. Morabito, and A. Hussain, "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," Neurocomputing, vol. 387, pp. 51–62, Apr. 2020.

[29] J. Y. Kim, S. J. Bu, and S. B. Cho, "Malware detection using deep transferred generative adversarial networks," in Proc. Int. Conf. Neural Inf. Process., 2017, pp. 556–564.

[30] M. H. Shahriar, N. I. Haque, M. A. Rahman, and M. Alonso, "G-IDS: Generative adversarial networks assisted intrusion detection system," in Proc. IEEE 44th Annu. Comput., Softw., Appl. Conf. (COMPSAC), Jul. 2020, pp. 376–385.