Deep Learning Based Network Intrusion Detection

Mrs. M. Swapna[1], V.B.S Krishna[2], A.Deepthi[3], T.Chandan[4]
[1]*Assistant Professor, Matrusri Engineering College*
[2,3,4]*Matrusri Engineering College*

Abstract—**Network intrusion detection is a pivotal component of cybersecurity, focusing on identifying and mitigating unauthorized access, misuse, or malicious activities within a computer network. As cyber threats grow increasingly sophisticated, the demand for effective intrusion detection mechanisms is paramount to ensure the security and integrity of networked systems. Traditional methods typically utilize a combination of signature-based and anomaly-based approaches, each with its inherent strengths and limitations. However, achieving high detection accuracy while minimizing false positives and adapting to evolving threats remains a significant challenge.**

In this research project, we explore the efficacy of advanced machine learning techniques, including Support Vector Machines (SVM), Multilayer Perceptron (MLP), Random Forest, Logistic Regression, Decision Trees, and AdaBoost Classifiers, to enhance network intrusion detection. Through rigorous experimentation, our study reveals that SVM achieves a notable accuracy rate of 94%, effectively distinguishing normal network traffic from various types of attacks. Furthermore, the integration of Convolutional Neural Network (CNN) algorithms bolsters our detection capabilities, attaining an impressive accuracy rate of 95%. Additionally, the implementation of Ensemble Learning Methods, such as the Voting Classifier, further elevates accuracy to an outstanding 98%. This robust performance underscores the potential of ensemble learning in amalgamating predictions from diverse classifiers, significantly enhancing the overall efficacy of intrusion detection systems. Our findings demonstrate that leveraging deep learning and ensemble learning methods can substantially improve the precision and adaptability of network intrusion detection systems, addressing contemporary cybersecurity challenges.

Keywords: Network Intrusion Detection System, Deep Learning, CNN, SVM, Ensemble Learning, Cybersecurity

I. INTRODUCTION

In the realm of cybersecurity, network intrusion detection plays a critical role amidst a rapidly evolving threat landscape characterized by increasingly sophisticated cyber-attacks. Traditional signature-based systems, while effective against known threats, often struggle to adapt to the dynamic tactics employed by cybercriminals. This limitation underscores the importance of adopting advanced techniques such as machine learning and ensemble learning in intrusion detection.

These modern approaches empower intrusion detection systems to analyze vast amounts of network data comprehensively. By leveraging artificial intelligence and data-driven algorithms, these systems can detect subtle anomalies and identify emerging threats that might evade detection by conventional methods. This proactive approach not only enhances the capability to detect intrusions but also enables organizations to stay ahead of adversaries and strengthen their cybersecurity defenses against a wide array of attack vectors.

This project focuses on advancing intrusion detection systems through the application of advanced machine learning techniques. The primary objective is to significantly improve the accuracy and efficiency of identifying unauthorized access and malicious activities within computer networks. By harnessing the capabilities of machine learning and ensemble learning, the project aims to empower organizations to proactively safeguard sensitive data and uphold operational integrity in an ever-changing threat landscape.

II. RELATED WORK

Base Paper: Several studies have advanced the field of network intrusion detection systems (IDS) through innovative applications of machine learning (ML) and deep learning (DL) techniques. Chuanlong Yin et al. explored the use of recurrent neural networks (RNNs) in IDS, demonstrating their efficacy in analyzing sequential data to detect complex intrusions. Published in IEEE (2017), their research achieved high accuracy in both binary and multiclass classification tasks. However, challenges such as sensitivity to hyperparameters and prolonged training times on high GPU resources could hinder real-time deployment.

Another study focused on convolutional neural networks (CNNs) for IDS in large-scale networks, addressing issues like detection accuracy and computational demands. While promising, this model faces challenges such as reduced accuracy and high resource utilization, limiting its practical deployment.

A comprehensive survey reviewed ML and DL methods in IDS, highlighting their capability to extract intricate features automatically from raw data, thereby enhancing effectiveness. Nonetheless, reliance on outdated datasets and computational complexities remain significant hurdles.

Enhancing signature-based IDS efficiency, another study employed the Myers algorithm within the MapReduce framework for parallel pattern matching on multi-core CPUs. Results showed substantial speedups over serial implementations, yet identified potential for better performance in alternative frameworks and noted limitations in in-memory sharing. Comparing DL against traditional methods, one research focused on deep belief systems for big data analysis, demonstrating DL's superior detection accuracy and performance. Nevertheless, challenges persist with traditional methods in handling unknown protocols and manual preprocessing requirements.

Lastly, the BAT model integrated Bidirectional Long Short-term Memory (BLSTM) and attention mechanisms to improve IDS accuracy using the NSL-KDD dataset. While achieving superior performance, particularly in traffic classification, the model's BLSTM implementation led to longer classification times, posing a trade-off between accuracy and computational efficiency. Further refinements are suggested to optimize the use of structured network traffic data and address evolving cybersecurity threats. These studies collectively underscore the evolving landscape of IDS with advanced ML and DL techniques, while also highlighting ongoing challenges in scalability, real-time processing, and adaptability to new cyber threats.

III. PROPOSED METHOD

The proposed work aims to significantly strengthen cybersecurity defenses against the escalating threat of network intrusions through the application of advanced machine learning techniques. Leveraging the NSL-KDD dataset, the project will focus on accurately classifying network packets into four distinct categories: R2L (Remote-to-Local), U2R (User-to-Root), DoS (Denial-of-Service), Probe, alongside normal network traffic.To achieve this, the project will harness the capabilities of several powerful algorithms, including Support Vector Machines (SVM), Convolutional Neural Networks (CNN), AdaBoost, Random Forest, and Gradient Boosting. These algorithms will be integrated with ensemble learning techniques to enhance the accuracy, robustness, and real-time detection capabilities of the Intrusion Detection System (IDS).Moreover, the project will emphasize advanced feature engineering methodologies to extract relevant features from raw network data, optimizing model performance. Model training will be complemented by rigorous optimization processes to fine-tune hyperparameters and ensure optimal performance under varying network conditions.

A pivotal aspect of this endeavor involves the development of a user-friendly website interface. This interface will serve as a practical tool for cybersecurity experts and network administrators, enabling them to monitor network activity in real-time, swiftly evaluate potential security threats, and initiate proactive response measures. The integration of a responsive and intuitive interface aims to empower users with actionable insights into network security, facilitating informed decision-making and enhancing overall cybersecurity posture.By combining cutting-edge machine learning techniques with effective visualization and user interface design, the proposed work seeks to establish a proactive defense mechanism against a wide range of network intrusions, thereby safeguarding sensitive data and preserving operational continuity in today's dynamic cybersecurity landscape.

IV. IMPLEMENTATION

The implementation phase of this project encompasses a systematic approach to develop and deploy an advanced network intrusion detection system. Commencing with data preprocessing, the raw dataset undergoes meticulous cleaning and categorical variable encoding to ensure data integrity and uniformity. Subsequent feature engineering techniques are applied to select pertinent features and unveil intricate relationships latent within the dataset. Following preprocessing, an array of classification algorithms, including Support Vector Machines (SVM), Multilayer Perceptron (MLP), AdaBoost, RandomForest, and Gradient Boosting, are trained on the prepared data. Rigorous model evaluation, employing established metrics such as F1-score, accuracy, and confusion matrices, ensues to gauge classification efficacy and robustness. Moreover, a deep learning framework is employed to construct a sequential neural network model, crafted using frameworks like Keras, to discern intricate patterns in network traffic data. Upon successful training and evaluation, the models are stored in a deployable format for subsequent integration into an intuitive web-based interface

V.SYSTEM ARCHITECTURE

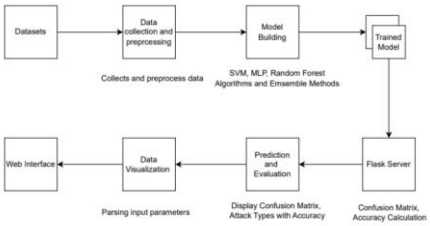

Datasets
Data collection and preprcessing
Model Building
Trained Model
Collects and preprocess data
SVM, MLP, Random Forest Algorithms and Emsemble Methods
Web Interface
Data Visualization
Prediction and Evaluation
Flask Server
Parsing input parameters
Display Confusion Matrix, Attack Types with Accuracy
Confusion Matrix, Accuracy Calculation

Fig-1.0

VI. ALGORITHMS

(i) Support Vector Machine (SVM)

SVM is a supervised machine learning algorithm that finds the optimal hyperplane to categorize new examples based on labelled training data. Introduced in the 1960s and refined in the 1990s, SVMs have gained popularity for their ability to achieve outstanding results. The optimal decision boundary maximizes the distance from the nearest data points of all classes, known as support vectors.

- Kernel SVM: For non-linearly separable data, Kernel SVM maps the data to higher dimensions to find a linear separation. Common kernels include:
- Linear Kernel: Suitable for linearly separable data.
- RBF Kernel: Captures complex relationships in both linearly and non-linearly separable data.
- Sigmoid Kernel: Models non-linear decision boundaries using the sigmoid function.

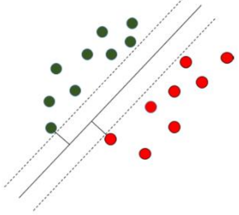


Fig-1.1 Decision Boundary with support vectors

(ii) Multilayer Perceptron (MLP)

An MLP is a neural network with input, output, and one or more hidden layers of neurons. It falls under feedforward algorithms, combining inputs with weights and using activation functions like ReLU or sigmoid.

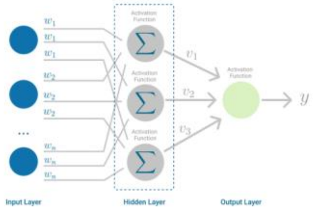


Fig-1.3

(iii) Backpropagation: This mechanism allows the MLP to adjust weights iteratively to minimize the cost function using Gradient Descent. It involves computing the gradient of the error and propagating it back through the network to update the weights.

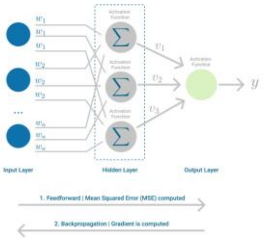


Fig-1.4 Feedforward and Backpropagation in Multilayer Perceptron

(iv) Decision Trees (DT)

DTs are composed of nodes and branches, where intermediate nodes represent features, and leaf nodes represent class labels. They decompose complex decision-making into simpler decisions by recursively splitting covariate space into subspaces. DTs are popular in areas like character identification, medical diagnosis, and voice recognition.

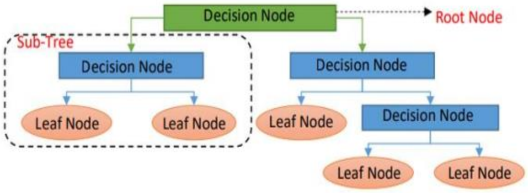


Fig-1.5 Decision Tree Algorithm

(v) Random Forest (RF)
RF is an ensemble learning algorithm composed of multiple decision trees. It combines the results from numerous decision trees to reach a single result. RF manages high-dimensional data without feature selection and has good generalization functionality. It trains quickly and is robust against overfitting.

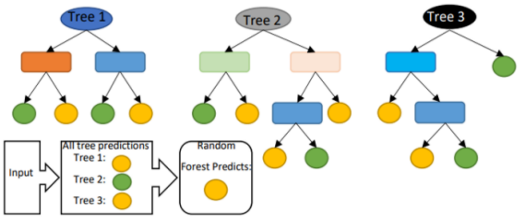


Fig- 1.6 Random Forest

(vi) AdaBoost Classifier
AdaBoost combines multiple weak learners, often decision trees, to create a strong classifier. It adjusts the weights of misclassified data points in each iteration, allowing subsequent weak learners to focus on difficult instances. This process continues until a predefined number of learners are generated or perfect classification is achieved.

(vii) Voting Classifier
A Voting Classifier aggregates predictions from multiple models and predicts the output class based on the highest probability (majority vote). It supports:

- Hard Voting: The class with the highest majority vote is chosen.
- Soft Voting: The output class is predicted based on the average probability.

(viii) Logistic Regression
Logistic regression is a supervised algorithm for classification tasks that predict the probability of an instance belonging to a given class. It uses the sigmoid function to map predicted values to probabilities, producing an S-shaped curve.

(ix) Gradient Boosting
Gradient Boosting combines weak learners into strong learners by minimizing the loss function using gradient descent. Each new model is trained to reduce the residual errors of the previous model. This iterative process continues until the loss function converges.

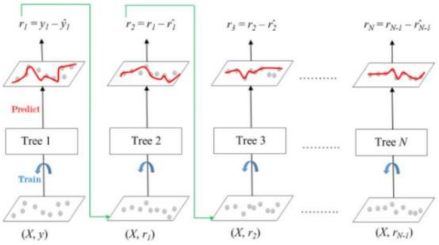


Fig 1.7 Gradient Boosted Trees For Regression

(x) Convolutional Neural Network (CNN)
CNN enhances network intrusion detection by discerning intricate patterns indicative of malicious activity. Key components include:

- Layers: Sequentially added for feature extraction and classification, using ReLU activation functions.
- Dropout Regularization: Prevents overfitting by randomly deactivating neurons during training.
- Training: Utilizes stochastic gradient descent (SGD) optimizer and binary cross-entropy loss function. The model is trained over 100 epochs, processing data in batches of 32 instances to optimize parameters and minimize loss.

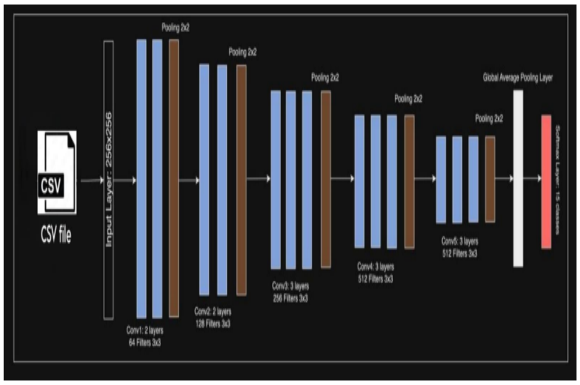


Fig 1.8 Convolution Neural Network Layers

By employing these algorithms, the intrusion detection system aims to effectively detect and classify network intrusions with high accuracy and robustness.

VII.PROCEDURE

(i). Data Preprocessing

Data preprocessing is essential for preparing datasets for model training. It involves several key steps:

- Loading Datasets: Load the training and testing datasets into memory from CSV files using appropriate libraries.
- Preprocessing: Clean the data, handle missing values, and encode categorical variables using techniques like one-hot encoding.
- Target Mapping: Map categorical target labels to numeric values for classification tasks.
- Feature Scaling: Standardize features using techniques like StandardScaler to ensure all features have the same scale, improving model convergence.

(ii) Feature Engineering

Feature engineering involves creating new features or transforming existing ones to enhance the predictive power of machine learning models. Key techniques include:

- Recursive Feature Elimination (RFE): Select the top features using RFE with various classification algorithms to identify the most relevant features.
- Polynomial Features: Generate polynomial features to capture non-linear relationships between variables and improve model performance.

(iii) Model Training and Evaluation

Model training involves fitting machine learning algorithms to the pre-processed data and evaluating their performance. Key steps include:

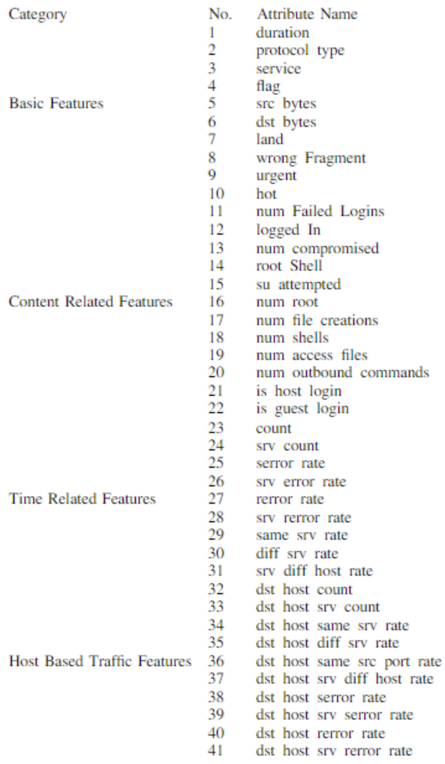
- Classification Algorithms: Train multiple classification algorithms such as SVM, MLP, AdaBoost, Random Forest, etc., on the pre-processed data.
- Model Evaluation: Evaluate model performance using metrics such as F1-score, accuracy, precision, recall, and confusion matrices to assess classification accuracy and model robustness.

(iv) Deep Learning Model

In addition to traditional machine learning algorithms, deep learning models such as neural networks can be employed for more complex tasks. Key steps include:

- Sequential Model Creation: Design a sequential neural network model using frameworks like Keras, specifying the architecture with input and hidden layers.
- Training: Train the neural network model on the pre-processed data and apply ensemble learning techniques such as Voting Classifier to combine predictions from multiple classifiers.

TABLE I
LIST OF NSL-KDD DATASET RECORDS ATTRIBUTES [11]



Category	No.	Attribute Name
	1	duration
	2	protocol type
	3	service
	4	flag
Basic Features	5	src bytes
	6	dst bytes
	7	land
	8	wrong Fragment
	9	urgent
	10	hot
	11	num Failed Logins
	12	logged In
	13	num compromised
	14	root Shell
	15	su attempted
Content Related Features	16	num root
	17	num file creations
	18	num shells
	19	num access files
	20	num outbound commands
	21	is host login
	22	is guest login
	23	count
	24	srv count
	25	serror rate
	26	srv error rate
Time Related Features	27	rerror rate
	28	srv rerror rate
	29	same srv rate
	30	diff srv rate
	31	srv diff host rate
	32	dst host count
	33	dst host srv count
	34	dst host same srv rate
	35	dst host diff srv rate
Host Based Traffic Features	36	dst host same src port rate
	37	dst host srv diff host rate
	38	dst host serror rate
	39	dst host srv serror rate
	40	dst host rerror rate
	41	dst host srv rerror rate

(v) Web Interface Development

- Develop a Web Interface: Use Flask frameworks to allow users to input network parameters.
- Integrate the Trained Model: Incorporate the trained model into the web interface to make predictions based on user input.
- Display Results: Show the predicted results to the user in a user-friendly format.

(vi) Testing and Evaluation

- Evaluate Performance: Assess the performance of the ensemble model and individual classifiers using the testing dataset.
- Calculate Evaluation Metrics: Determine evaluation metrics such as accuracy, precision, recall, and F1-score to measure the model's effectiveness.

VIII. RESULTS

6.1 Accuracy Comparision

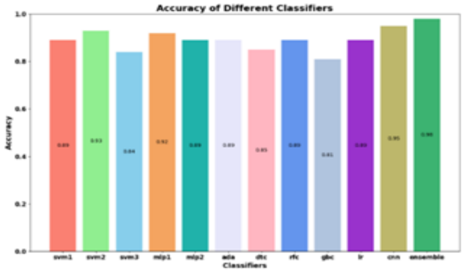
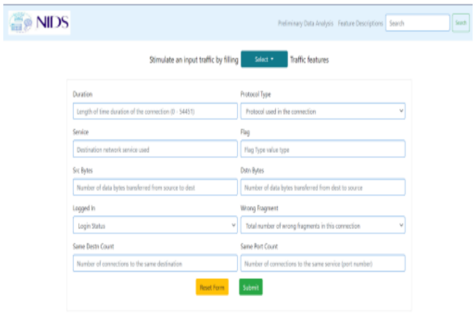


Fig 6.1 Accuracy Bar Graph

Fig .1.9 Accuracy Bar Graph

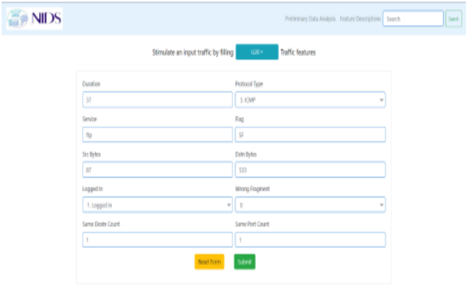
Fig 1.1.0 Data Analysis Page



Stimulate an input traffic by filling Select Traffic features

Duration: Length of time duration of the connection (0 - 54451)
Protocol Type: Protocol used in the connection
Service: Destination network service used
Flag: Flag Type value type
Src Bytes: Number of data bytes transferred from source to dest
Dstn Bytes: Number of data bytes transferred from dest to source
Logged In: Login Status
Wrong Fragment: Total number of wrong fragments in this connection
Same Destn Count: Number of connections to the same destination
Same Port Count: Number of connections to the same service (port number)

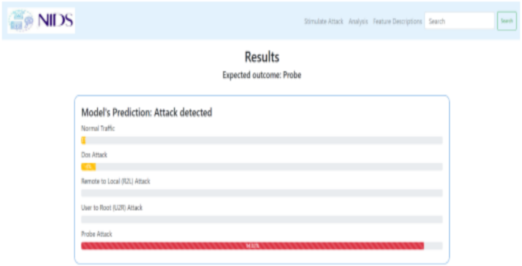
Fig 1.1.1 Home Page



Stimulate an input traffic by filling U2R Traffic features

Duration: 37
Protocol Type: 3. ICMP
Service: ftp
Flag: SF
Src Bytes: 07
Dstn Bytes: 533
Logged In: 1. Logged in
Wrong Fragment: 0
Same Destn Count: 1
Same Port Count: 1

Fig 1.1.2 Home Page with Input Parameters



Results

Expected outcome: Probe

Model's Prediction: Attack detected

Normal Traffic

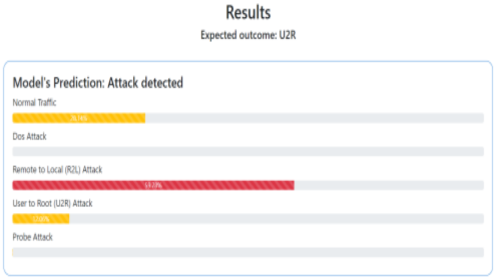
Dos Attack

Remote to Local (R2L) Attack

User to Root (U2R) Attack

Probe Attack

Fig 1.1.3 Prediction-1



Results

Expected outcome: U2R

Model's Prediction: Attack detected

Normal Traffic 28.14%

Dos Attack

Remote to Local (R2L) Attack 59.28%

User to Root (U2R) Attack 12.00%

Probe Attack

Fig 1.1.4 Prediction-2

IX. CONCLUSION

In the face of escalating cyber threats and sophisticated attack vectors, the development of an advanced network intrusion detection system stands as a beacon of resilience and security for modern-day digital ecosystems. Through the amalgamation of cutting-edge machine learning and deep learning methodologies, the project has realized a formidable defense mechanism against a myriad of network intrusions, spanning from Denial-of-Service (DoS) attacks to sophisticated User-to-Root (U2R) exploits. By harnessing the power of ensemble learning techniques such as AdaBoost, Random Forest, and Voting classifiers, coupled with base classifiers like Support Vector Machines (SVMs) and Multilayer Perceptron (MLP), the system has demonstrated remarkable prowess in accurately identifying and categorizing malicious network activities. The journey of this project has been marked by meticulous experimentation, rigorous testing, and continuous refinement, resulting in the emergence of a robust and adaptive network intrusion detection framework. The deployment of polynomial feature engineering, recursive feature elimination, and extensive algorithmic evaluation has culminated in a solution that not only meets but exceeds the stringent requirements of modern cybersecurity standards. Furthermore, the integration of a user-friendly web interface enhances accessibility and usability, empowering cybersecurity practitioners to leverage the system's capabilities with ease and efficiency

X. ACKNOWLEDGEMENT

It is our privilege and pleasure to express our profound sense of respect, gratitude, and indebtedness to our guide, Mrs. M. Swapna Reddy, Assistant Professor, Department of Computer Science and Engineering, Matrusri Engineering College, for her inspiration, guidance, cogent discussion, constructive criticisms and encouragement throughout this dissertation work. We express our sincere thanks to Project Coordinator Dr. P. Vijayapal Reddy, Professor & Head of the Department of Computer Science and Engineering, and Mrs. B. J. Praveena, Assistant Professor, Department of Computer Science and Engineering, Matrusri Engineering College, for their valuable suggestions and constant helping in completing the work. We express our sincere gratitude to Dr. P. Vijayapal Reddy, Professor and head of the Department of Computer Science and Engineering, Matrusri Engineering College, for his precious suggestions, motivation, and cooperation. We express our sincere thanks to Dr. D. Hanumantha Rao, Principal, Matrusri Engineering College, Saidabad, Hyderabad, for his encouragement and constant help. We extend our sincere thanks to all the teaching and non-teaching staff of the Computer Science and Engineering Department for their support and encouragement. Last but not least, we wish to acknowledge our friends and family members for giving us moral strength and helping us to complete this dissertation.

XI. FUTURE SCOPE

The future scope of this project encompasses several promising avenues for enhancement and extension. Exploring the integration of anomaly detection techniques alongside traditional signature-based methods, such as Isolation Forests and One-Class SVMs, could offer a more comprehensive approach to intrusion detection, particularly in identifying novel attack patterns and mitigating zero-day threats. Furthermore, refining the ensemble learning framework by incorporating advanced techniques like stacking and meta-learning holds potential for boosting classification performance. This involves stacking multiple heterogeneous classifiers and employing metalearners to combine their predictions, thus mitigating individual model limitations and enhancing overall detection accuracy. Additionally, extending the system to support real-time monitoring and automated response capabilities remains crucial, alongside exploring scalability for handling large-scale network environments and diverse datasets. These efforts aim to continuously improve the system's effectiveness in safeguarding against evolving cyber threats and ensuring the resilience of network infrastructures

XII. REFERENCES

[1] B B Zarpelo, R S Miani, C T Kawakani, and S C de Alvarenga, A survey of intrusion detection in Internet of Things, J Netw Comput Appl vol 84 pp 25 37 Apr 2017

[2] B Mukherjee, L T Heberlein, and K N Levitt, Network intrusion detection, IEEE Netw vol 8 no 3 pp 26 41 May 1994

[3] S Kishorwagh, V K Pachghare, and S R Kolhe Survey on intrusion detection system using machine learning techniques, Int J Control Automat vol 78 no 16 pp 30 37 Sep 2013

[4] N Sultana, N Chilamkurti, W Peng, and R Alhadad Survey on SDN based network intrusion detection system using machine learning approaches, Peer to Peer Netw Appl vol 12 no 2 pp 493 501 Mar 2019

[5] M Panda, A Abraham, S Das, and M R Patra, Network intrusion detection system A machine learning approach, Intell Decis Technol vol 5 no 4 pp 347 356 2011

[6] W Li, P Yi, Y Wu, L Pan, and J Li, A new intrusion detection system based on KNN classification algorithm in wireless sensor network, J Electr Comput Eng vol 2014 pp 1 8 Jun 2014

[7] Ammar, M., et al. "Intrusion detection using machine learning algorithms: A comprehensive review." IEEE Access 8 (2020): 47550-47580.

[8] Chitrakar, Suraj, et al. "Deep learning based network intrusion detection system: A review." Journal of Information Security and Applications 60 (2021): 102693

[9] S. J. Jian, Z. G. Lu, D. Du, B. Jiang and B. X. Liu, "Overview of network intrusion detection technology", *J. Cyber Secur.*, vol. 5, no. 4, pp. 96-122, 2020.

[10] K. Wu, Z. Chen and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks", *IEEE Access*, vol. 6, pp. 50850-50859, 2018.

[11] T. Acharya, I. Khatri, A. Annamalai and M. F. Chouikha, "Efficacy of heterogeneous ensemble assisted machine learning model for binary and multi-class network intrusion detection", *Proc. IEEE Int. Conf. Autom. Control Intell. Syst. (I2CACIS)*, pp. 408-413, Jun. 2021.

[12] M. Injadat, F. Salo, A. B. Nassif, A. Essex and A. Shami, "Bayesian optimization with machine learning algorithms towards anomaly detection", *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, pp. 1-6, Dec. 2018.

[13] Y. J. Chew, S. Y. Ooi, K.-S. Wong, Y. H. Pang and N. Lee, "Adoption of IP truncation in a privacy-based decision tree pruning design: A case study in network intrusion detection system", *Electronics*, vol. 11, no. 5, pp. 805, Mar. 2022.

[14] L. L. Ray, "Training and testing anomaly-based neural network intrusion detection systems", *Int. J. Inf. Secur. Sci.*, vol. 2, no. 2, pp. 57-63, 2013.

[15] A. Ponmalar and V. Dhanakoti, "An intrusion detection approach using ensemble support vector machine based chaos game optimization algorithm in big data platform", *Appl. Soft Comput.*, vol. 116, Feb. 2022.

[16] M. Mehmood, T. Javed, J. Nebhen, S. Abbas, R. Abid, G. R. Bojja, et al., "A hybrid approach for network intrusion detection", *Comput. Materials Continua*, vol. 70, no. 1, pp. 91-107, 2022.

[17] M. U. Ilyas and S. A. Alharbi, "Machine learning approaches to network intrusion detection for contemporary internet traffic", *Computing*, vol. 104, no. 5, pp. 1061-1076, May 2022.

[18] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa and C. F. M. Foozy, "Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset", *IEEE Access*, vol. 9, pp. 22351-22370, 2021.

[19] Z. Li, A. L. G. Rios, G. Xu and L. Trajkovic, "Machine learning techniques for classifying network anomalies and intrusions", *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, pp. 1-5, May 2019.

[20] T. Saba, A. Rehman, T. Sadad, H. Kolivand and S. A. Bahaj, "Anomaly-based intrusion detection system for IoT networks through deep learning model", *Comput. Electr. Eng.*, vol. 99, Apr. 2022.