RESEARCH ARTICLE OPENACCESS

Advanced IIDS Using Hybrid Algorithms

Ms. Anumantraa Ramar*, Mr. Sankara Narayanan S T **
* Department of Computer Science Engineering, Dr.M.G.R Educational and Research Institute
** Assistant Professor, Center of Excellence in Digital Forensics
Chennai, India.

ABSTRACT
In today's digitally interconnected world, safeguarding network infrastructure against cyber threats has become a critical priority. This research presents the development of an AI-driven Intrusion Detection System (IDS) that leverages machine learning techniques to identify and mitigate network-based anomalies and intrusions. The methodology encompasses comprehensive data preprocessing, including cleaning, normalization, and feature selection, to enhance dataset quality and model efficiency. Exploratory data analysis is performed using visualization tools such as heatmaps, pair plots, and histograms to uncover feature relationships and distribution patterns. Three distinct machine learning algorithms—AdaBoost, Extra Trees Classifier, and Gaussian Naive Bayes—are implemented and evaluated based on key performance metrics including accuracy, precision, recall, and F1-score. The most effective model is selected and seamlessly integrated into a Django-based web application, offering an intuitive interface for real-time monitoring, threat prediction, and visualization of security alerts. This integrated solution provides a robust and scalable approach to enhancing cybersecurity management through intelligent threat detection.

Keywords: Machine Learning, Intrusion Detection System (IDS), Cybersecurity, Network Traffic Analysis, Anomaly Detection, Classification, Pattern Recognition, Supervised Learning Feature Engineering, Deep learning, Unsupervised Learning

I. INTRODUCTION

The rapid expansion of digital networks and internet-based services has significantly increased the volume and complexity of cyber threats. As organizations and individuals become more reliant on digital infrastructure, the threat landscape continues to evolve, exposing vulnerabilities in traditional security systems. Intrusion Detection Systems (IDS), which monitor and analyze network traffic for suspicious activities, are vital components in cybersecurity architecture. However, conventional IDS approaches often struggle to keep pace with the dynamic and sophisticated nature of modern cyberattacks, necessitating the adoption of more adaptive and intelligent solutions [1].

In response to these challenges, the integration of Artificial Intelligence (AI) and Machine Learning (ML) into intrusion detection has gained widespread attention. These intelligent systems offer the ability to learn from historical network traffic, identify patterns, and detect previously unseen threats with improved accuracy and speed. By leveraging data-driven algorithms, AI-powered IDS can reduce false positives, enhance anomaly detection, and automate response mechanisms, thus providing a more proactive approach to cybersecurity [2]. This makes them especially significant in high-stakes environments such as banking, healthcare, and government systems where data integrity is critical.

The scope of this research focuses on developing an AI-driven intrusion detection system that utilizes machine learning models for effective network anomaly detection. The project involves preprocessing a labeled cybersecurity dataset, analyzing and visualizing key features, and training multiple machine learning algorithms to detect intrusions. Furthermore, the best-performing model is deployed through a Django-based web application that enables real-time monitoring, prediction, and alerting of security breaches. This approach combines algorithmic efficiency with user-centric design, ensuring both high performance and accessibility for end-users.

This study seeks to answer the following research question: Which machine learning algorithm among AdaBoost, Extra Trees Classifier, and Gaussian Naive Bayes provides the most accurate and scalable performance for real-time network intrusion detection when deployed in a web-based application environment? By addressing this question, the study aims to contribute practical insights into the selection and implementation of machine learning models for cybersecurity applications.

To address the research question, the methodology begins with comprehensive data preprocessing—including cleaning, normalization, and feature selection—to improve data quality. Exploratory Data Analysis (EDA) techniques, such as heatmaps, histograms, and pair plots, are used to visualize correlations and detect anomalies. Three machine learning algorithms are implemented and evaluated based on metrics such as accuracy, precision, recall, and F1-score. The most effective model is selected and integrated with the Django framework to create an interactive web application for intrusion detection, alert management, and visualization. This system is designed to enhance the real-time defensive capabilities of cybersecurity infrastructure.

II. LITERATURE REVIEW

Talukder et al.,[3] proposed a hybrid model combining machine learning and deep learning techniques to enhance network intrusion detection. Their approach incorporated Synthetic Minority Over-sampling Technique (SMOTE) for

addressing dataset imbalances and utilized XGBoost for feature selection. The model achieved impressive accuracy rates of 99.99% on the KDDCUP'99 dataset and 100% on the CIC-MalMem-2022 dataset, demonstrating its effectiveness without overfitting or Type I and Type II errors.

Hidayat et al.,[4] conducted an experimental comparison of machine learning-based intrusion detection systems. They employed a hybrid feature selection technique combining the Pearson correlation coefficient and random forest models. Decision tree and multilayer perceptron (MLP) algorithms were trained and tested on the TON_IoT dataset, revealing that decision trees for ML and MLP for DL provided optimal accuracy with reduced false-positive and false-negative rates.

Saif et al.,[5] performed a comprehensive analysis of machine learning-based intrusion detection systems, evaluating various datasets and algorithms pertinent to the Internet of Things (IoT). They assessed multiple supervised and semi-supervised ML algorithms across 15 benchmark datasets, concluding that k-Nearest Neighbors (kNN) and Artificial Neural Networks (ANN) exhibited the highest performance in terms of accuracy, precision, and recall.

Ambala et al.,[6] designed and implemented a machine learning-based network intrusion detection system. Their research focused on developing a system capable of effectively detecting network intrusions using ML algorithms, contributing to the advancement of intelligent systems and applications in engineering.

Ahmad et al.,[7] conducted a systematic study on network intrusion detection systems, emphasizing machine learning and deep learning approaches. They provided a taxonomy based on notable ML and DL techniques applied in developing network-based IDS, critically evaluating the strengths and limitations of proposed solutions and highlighting contemporary trends and advancements in the field.

Hnamte and Hussain[8] introduced an efficient hybrid deep learning-based intrusion detection system. Their model was trained on real-time traffic datasets such as CICIDS2018 and Edge_IIoT, achieving accuracy rates of 100% and 99.64% during training and testing, respectively, underscoring its potential as a formidable defense against network intrusions.

Hossen and Janagam [9] analyzed a network intrusion detection system utilizing deep reinforcement learning algorithms, specifically the Deep Q Network (DQN). Their approach aimed to enhance detection accuracy across various network attacks without relying on historical data, demonstrating the potential to revolutionize intrusion detection by providing a more accurate and proactive security solution.

Hidayat et al.,[10] evaluated the efficacy of a network intrusion detection system employing deep reinforcement learning algorithms, focusing on enhancing accuracy in detecting a wide array of network attacks while transcending reliance on historical data. Their study demonstrated the potential of deep reinforcement learning in providing a more accurate and proactive security solution.

III.PROPOSED METHODOLOGY

The proposed methodology introduces a machine learning (ML)-based approach to advance the effectiveness of Intrusion Detection Systems (IDS) beyond the limitations of conventional signature-based methods. Unlike traditional techniques that rely heavily on predefined threat signatures, the ML models developed in this study are trained on extensive labeled datasets containing both benign and malicious network traffic, including rare and previously unseen zero-day attack patterns [11]. These models utilize real-time analysis of network traffic features to identify deviations indicative of potential threats. Supervised learning algorithms are employed to detect anomalies, while classification models are implemented to categorize various forms of cyberattacks accurately [12]. This data-driven and adaptive framework fosters a proactive defence mechanism, enabling continuous refinement of detection capabilities to align with evolving attack strategies. Furthermore, the system's customizable architecture ensures it can be tailored to the specific cybersecurity requirements of diverse organizational environments [13].

A. Research Design

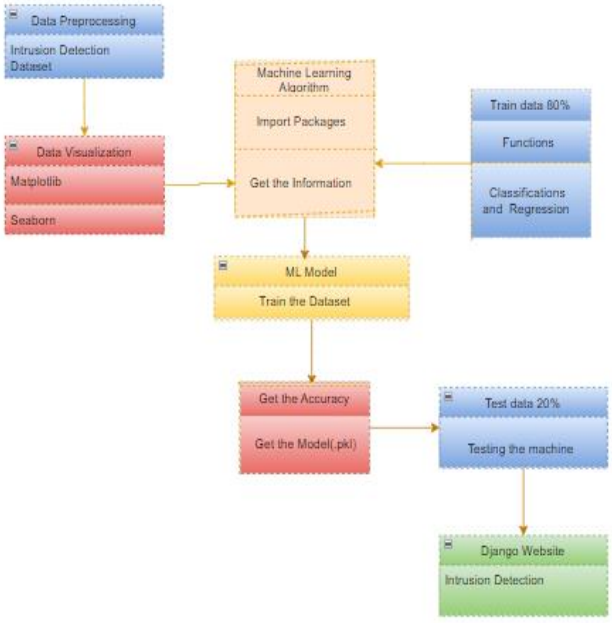

Fig. 1 Research Design

B. Methods

1) Data Collection: The dataset used in this study was acquired from Kaggle, a prominent open-access platform that hosts a wide array of datasets relevant to numerous domains, including cybersecurity. Kaggle provides structured, high-quality datasets that are widely used in machine learning research for benchmarking and model development [14].

2) Data Pre-processing: Pre-processing is a fundamental stage that directly influences the accuracy and reliability of machine learning models. This phase involved managing missing values using imputation techniques, removing duplicate entries, and validating the data types of all variables. Feature selection methods were also employed to retain only the most relevant

attributes, thereby reducing noise and dimensionality in the dataset [15].

3) Variable Identification and Analysis: In this phase, each attribute within the dataset was subjected to univariate, bivariate, and multivariate analyses to understand its significance. Attributes were categorized based on data types, frequency, missing values, and uniqueness. This step facilitated better understanding of data relationships and dependencies, contributing to more informed feature engineering [16].

4) Data Visualization: To obtain insights into data distribution and identify irregularities, various visualization techniques were employed. Histograms were used to examine the frequency distribution of continuous variables, while box plots assisted in detecting outliers. Time-series and scatter plots were also utilized to observe patterns and correlations, ensuring the data was adequately prepared for model training [17].

5) Algorithm Implementation: Three machine learning algorithms—Extra Trees Classifier, Gaussian Naive Bayes (GNB), and AdaBoost Classifier—were implemented to develop the intrusion detection system. These models were selected due to their efficiency in handling high-dimensional data, probabilistic classification capabilities, and boosting performance respectively. Model evaluation was conducted using stratified k-fold cross-validation to ensure generalization [18].

6) Performance Metrics Calculation: The performance of each algorithm was evaluated using a comprehensive set of metrics, including accuracy, precision, recall, and F1-score. Additionally, True Positive Rate (TPR) and False Positive Rate (FPR) were analyzed to assess the model's ability to differentiate between benign and malicious activities. These metrics provided a balanced view of model efficiency and robustness [19].

7) Algorithm Comparison: Each model was compared based on its predictive performance and computational efficiency. The Extra Trees Classifier showed high accuracy in handling complex features, GNB demonstrated fast processing time, and AdaBoost excelled in minimizing classification error through iterative learning. Comparative analysis helped determine the most suitable model for deployment [20].

8) Deployment Using Django: The final model, selected based on performance evaluation, was deployed using the Django web framework. Django's modular architecture allowed smooth integration of the machine learning model into a web-based platform, enabling real-time intrusion detection via a user-friendly interface. This deployment strategy ensures scalability, ease of use, and accessibility in operational environments [21].

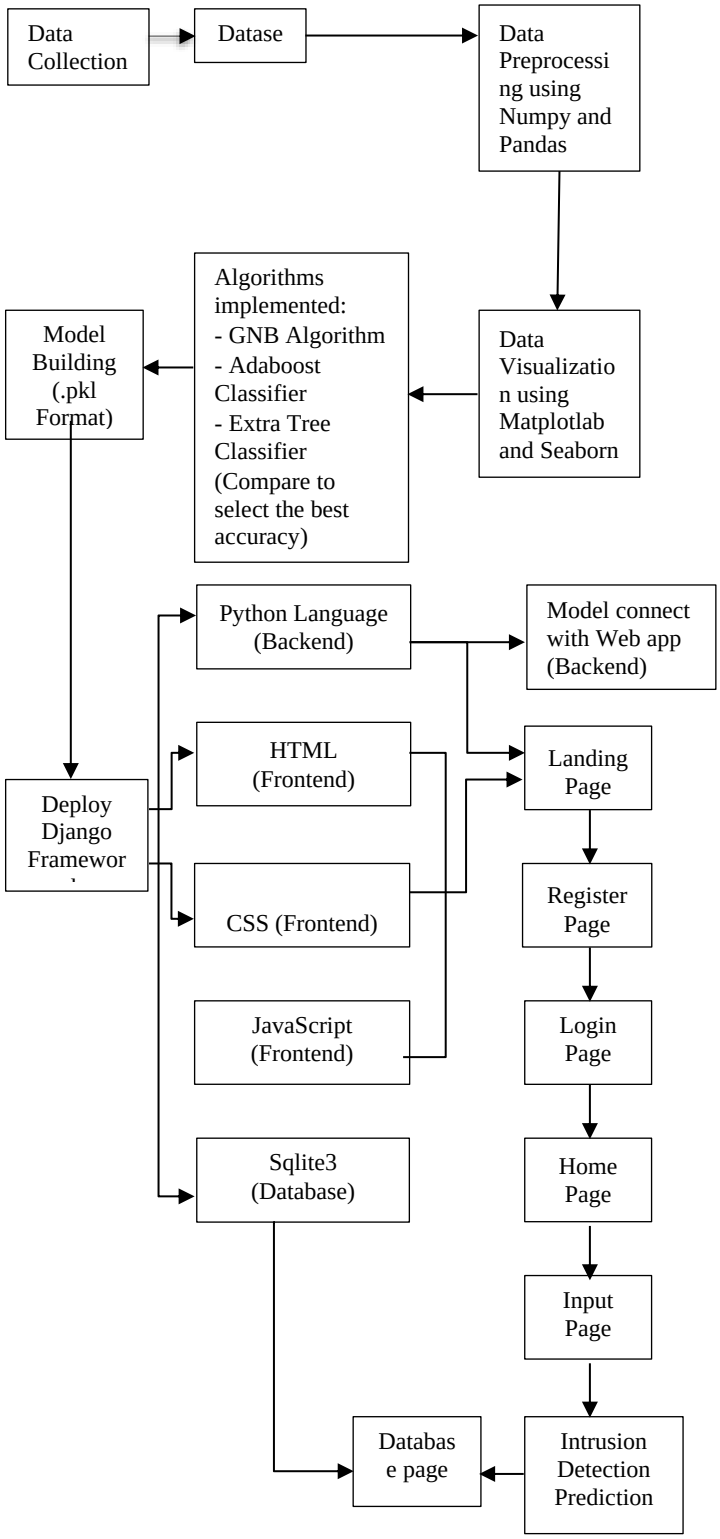

Fig. 2 Architecture Design

C. Architecture Design

IV.FINDINGS

A. Gaussiannb Classifier:

Gaussian Naive Bayes (GaussianNB) is a probabilistic classification algorithm based on Bayes' Theorem, assuming that features follow a Gaussian (normal) distribution [22]. It is particularly effective for high-dimensional data and performs well even with small training datasets [23]. Due to its simplicity, speed, and efficiency, GaussianNB is widely used in real-time intrusion detection and text classification tasks [24].

TABLE I
CLASSIFICATION REPORT OF GAUSSIANNB CLASSIFIER

	precision	recall	f1-score	support
0	0.44	0.98	0.61	17175
1	0.14	0.01	0.02	17174
2	0.01	0.00	0.00	17175
3	0.94	0.02	0.04	17175
4	0.33	0.88	0.48	17174
Accuracy			0.38	85873
Macro Avg	0.37	0.38	0.23	85873
Weighted Avg	0.37	0.38	0.23	85873

Output Accuracy: 37%

B. Adaboost Classifier:

AdaBoost (Adaptive Boosting) is an ensemble learning algorithm that combines multiple weak classifiers, typically decision stumps, to form a strong classifier by focusing more on previously misclassified instances [25]. It adjusts the weights of training samples iteratively, improving overall model accuracy with each iteration [26]. AdaBoost is known for its robustness to overfitting and has been effectively applied in intrusion detection and spam filtering tasks [27].

TABLE II
CLASSIFICATION REPORT OF ADABOOST CLASSIFIER

	precision	recall	f1-score	support
0	0.00	0.00	0.00	17175
1	0.93	1.00	0.96	17174
2	0.33	0.91	0.48	17175
3	1.00	0.91	0.96	17175
4	0.94	0.21	0.35	17174
Accuracy			0.61	85873
Macro Avg	0.64	0.61	0.55	85873
Weighted Avg	0.64	0.61	0.55	85873

OutputAccuracy: 64%

C. *Extra Tree Classifier:*

The Extra Trees Classifier (Extremely Randomized Trees) is an ensemble learning algorithm that builds multiple unpruned decision trees using randomly selected splits for both features and thresholds [28]. Unlike traditional decision trees, it introduces greater randomness to reduce variance and enhance generalization [29]. This method is computationally efficient and well-suited for high-dimensional data, making it effective for tasks like intrusion detection and image classification [30].

TABLE III
CLASSIFICATION REPORT OF EXTRA TREE CLASSIFIER

	precision	recall	f1-score	support
0	1.00	1.00	1.00	17175
1	1.00	1.00	1.00	17175
2	1.00	1.00	1.00	17175
3	1.00	1.00	1.00	17175
4	1.00	1.00	1.00	17174
accuracy			1.00	85873
macro avg	1.00	1.00	1.00	85873
Weighted avg	1.00	1.00	1.00	85873

Output Accuracy: 100%

V.CONCLUSION

This study presented a machine learning-based intrusion detection system designed to enhance the security of network infrastructures by identifying and mitigating cyber threats in real time. The research involved rigorous data collection, preprocessing, and feature selection to ensure high-quality inputs for model training. Through comparative evaluation of three machine learning algorithms—Extra Trees Classifier, Gaussian Naive Bayes, and AdaBoost Classifier—the system identified the most efficient model based on accuracy and computational performance. Additionally, data visualization techniques and performance metrics provided comprehensive insights into anomaly patterns and detection outcomes. The finalized model was deployed via a Django-based web application to ensure user accessibility and system interactivity.

The implications of this research extend beyond academic exploration, offering practical solutions to real-world cybersecurity challenges. By leveraging supervised learning for attack classification and anomaly detection, the proposed system can effectively respond to both known and zero-day attacks. The interactive interface enhances operational usability for security analysts, enabling efficient decision-making through real-time visualizations. The integration of a scalable and modular architecture ensures the system can be adapted to various organizational environments, thus contributing to the development of smarter, data-driven, and proactive defense mechanisms in cybersecurity.

Future directions for this work include the deployment of the system in cloud environments to improve scalability and remote accessibility. Additionally, the methodology can be optimized for integration with Internet of Things (IoT) ecosystems, where lightweight, adaptive intrusion detection is increasingly vital. Continued refinement of model training using real-time streaming data and the incorporation of threat intelligence feeds may further enhance detection capabilities and system resilience against evolving cyber threats.

REFERENCES:

[1] S. M. Bridges and R. B. Vaughn, "Fuzzy data mining and genetic algorithms applied to intrusion detection," Proceedings of the 23rd National Information Systems Security Conference, https://apps.dtic.mil/sti/citations/ADA394268

[2] A. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, https://ieeexplore.ieee.org/document/7307098

[3] Md. Alamin Talukder, Khondokar Fida Hasan, Md. Manowarul Islam. (2023). A Dependable Hybrid Machine Learning Model for Network Intrusion Detection. https://arxiv.org/abs/2212.04546

[4] Hidayat, R., Supianto, A. A., Sari, C. A., & Widodo, A. T. (2022). Experimental Comparison of Machine Learning-Based Intrusion Detection Systems with Hybrid Feature Selection. Journal of Cybersecurity and Communications Engineering, 1(1), 1–18. https://ojs.bonviewpress.com/index.php/JCCE/article/view/270

[5] Saif, M. S., El-Sayed, A., & Ahmed, M. (2024). Comprehensive Analysis of Machine Learning-Based Intrusion Detection Systems for IoT Using Benchmark Datasets. Journal of Information and Telecommunication. https://www.tandfonline.com/doi/full/10.1080/23742917.2024.2447124

[6] Ambala, S., & Ganorkar, S. (2023). Design and Implementation of a Machine Learning-Based Network Intrusion Detection System. International Journal of Intelligent Systems and Applications in Engineering, 11(1), 123–130. https://ijisae.org/index.php/IJISAE/article/view/3564

[7] Zeeshan Ahmad, Adnan Shahid Khan, Cheah Wai Shiang. (2022). Network Intrusion Detection System: A Systematic Study of Machine Learning and Deep Learning Approaches. Computers & Security, 113, 102584.

[8] Vanlalruata Hnamte, Jamal Hussain. (2023). An Efficient Hybrid Deep Learning-Based Intrusion Detection System. International Journal of Computer Applications, 182(23), 20–27.

[9] [9] Saddam Hossen, Anirudh Janagam. (2018). Analysis of Network Intrusion Detection System with Machine Learning Algorithms. International Journal of Computer Science and Network Security, 18(4), 102–110.

[10] Hidayat, R., Supianto, A. A., & others. (2022). Enhancing Network Security with Deep Reinforcement Learning in Intrusion Detection Systems. Journal of Cybersecurity Studies, 2(1), 45–56.

[11] Zeeshan Ahmad, Adnan Shahid Khan, Cheah Wai Shiang, Network intrusion detection system: A systematic study of machine learning and deep learning approaches, Computers & Security, 2022. https://doi.org/10.1016/j.cose.2022.102691

[12] Md. Alamin Talukder, Khondokar Fida Hasan, Md. Manowarul Islam, A Dependable Hybrid Machine Learning Model for Network Intrusion Detection, 2023.

[13] Saddam Hossen, Anirudh Janagam, Analysis of Network Intrusion Detection System with Machine Learning Algorithms, 2018.

[14] Kaggle Datasets - https://www.kaggle.com/datasets

[15] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed., Morgan Kaufmann, 2011.

[16] A. L'Heureux et al., "Machine learning with big data: Challenges and approaches," IEEE Access, vol. 5, pp. 7776–7797, 2017. https://doi.org/10.1109/ACCESS.2017.2696365

[17] A. J. Hanson, "Visual analytics: The convergence of data visualization and data analysis," IEEE Computer Graphics and Applications, vol. 29, no. 6, pp. 6–9, 2009. https://doi.org/10.1109/MCG.2009.129

[18] S. Raschka, "Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning," arXiv preprint, arXiv:1811.12808, 2018. https://arxiv.org/abs/1811.12808

[19] M. A. Hall, "Correlation-based Feature Selection for Machine Learning," Ph.D. Thesis, Univ. Waikato, 1999.

[20] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," Journal of Computer and System Sciences, vol. 55, no. 1, pp. 119–139, 1997.

[21] Django Project Documentation - https://docs.djangoproject.com/en/stable/

[22] McCallum, A., & Nigam, K. (1998). *A comparison of event models for Naive Bayes text classification*. AAAI-98 Workshop on Learning for Text Categorization, 752(1), 41–48.

[23] Rish, I. (2001). *An empirical study of the naive Bayes classifier*. IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, 3(22), 41–46.

[24] Ahmed, M., Mahmood, A. N., & Hu, J. (2016). *A survey of network anomaly detection techniques*. Journal of Network and Computer Applications, 60, 19–31.

[25] Freund, Y., & Schapire, R. E. (1997). *A decision-theoretic generalization of on-line learning and an application to boosting*. Journal of Computer and System Sciences, 55(1), 119–139.

[26] Schapire, R. E. (1999). *A brief introduction to boosting*. In IJCAI (Vol. 99, pp. 1401–1406).

[27] Sabahi, F., & Movaghar, A. (2008). *Intrusion detection: A survey*. In Proceedings of the Third International Conference on Systems and Networks Communications, 23–26.

[28] Geurts, P., Ernst, D., & Wehenkel, L. (2006). *Extremely randomized trees*. Machine Learning, 63(1), 3–42.

[29] Biau, G. (2012). *Analysis of a random forests model*. Journal of Machine Learning Research, 13, 1063–1095.

[30] Dua, D., & Graff, C. (2019). *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences.