Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Improved Network Threat Detection Using Random Forest Algorithm with Django-Based Output Visualization

M.Kishore [1], Dr. P.J.Sathishkumar [2*], Dr. S.Balaji [3]

[1] *Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India. Email: mkavikishore@gmail.com*
[2] *Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India. Email: sathishjraman@gmail.com*
[3] *Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India. Email: balajiit@gmail.com*
**Corresponding Author: Dr. P.J.Sathishkumar*

ARTICLE INFO

Received: 12 Mar 2025

Revised: 04 May 2025

Accepted: 13 May 2025

ABSTRACT

The emergence of artificial intelligence (AI) methods has transformed network security by facilitating the use of predictive modelling for threat detection. This abstract to present an innovative methodology to improve network security through predictive modelling using emerging AI methods. Predictive modelling using AI entails processing a large volume of data of network traffic that uses AI algorithms to identify patterns within large data volumes to determine the possibility of threat(s) such as malware or intrusions that may result in security violation. The predictive modelling as proposed in this methodology allows for predictive models to determine the vulnerabilities within the network, and can further allow for the predictive modelling to give early detection of threats before they escalate into security violations. The methodology allows network defenders to exercise defendable action to improve the resilience of networks to absorb threats and allow for the constructive feedback to improve network defense. Through the use of AI and predictive modelling, this proposal ultimately seeks to add to developing practical and resilient security and network defense in an increasingly interconnected digital landscape. In this project, the three different algorithms will be described: Bernoulli Naive Bayes, Adaboost, and Random Forest algorithm. In my previous research, I used Bernoulli Naive Bayes and Adaboost algorithm, with the accuracy for Bernoulli Naive Bayes algorithm being 42% and Adaboost algorithm being 78%. In this project, I used the Random Forest algorithm, and the Random Forest Algorithm was executed on the project and reported 98%.

Keywords: Django, Random Forest Algorithm, Artificial Intelligence, Predictive models, Bernoulli Naive Bayes and Adaboost algorithm.

INTRODUCTION

With the fast-changing digital age, the threats to cybersecurity have increased in sophistication, coming against organizations that bank heavily on digital infrastructures. Cyberattacks like malware, phishing, and ransomware now become critical threats, causing massive financial losses and loss of reputation. With every advancement made, it becomes even more apparent how traditional security systems relying on rules are no longer effective. Their predefined designs are built on patterns, which impede their capability to discover novel and unfamiliar threat avenues capable of bypassing standard detection mechanisms. Considering such challenges, much more advanced approaches in the realm of threat detection need to be adopted, and artificial intelligence (AI) and machine learning (ML) present promising options. These technologies can estimate and detect obsolete and upcoming dangers by monitoring activities and anomalies within the network, thus allowing measures to be taken in advance. This research aims to develop a complex predictive model using the first two offered machine learning algorithms—Bernoulli Classifier and AdaBoost. Applying these methods, the research aims to improve the accuracy, adaptability, and efficiency of threat detection systems designed to respond to modern-day cyber security challenges.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Bernoulli Classifier:

A probabilistic model operating on Bernoulli distribution, and meant to be used to classify data as binary events. since much of the cybersecurity universe is binary or categorical, (i.e., whether a port is open/closed or an action is considered malicious/benign), this algorithm is best suited for distinguishing between safe behaviors and malicious behaviors in a network. Bernoulli Classifier excels at dealing with discrete and categorical data, which is prevalent in cybersecurity situations.

AdaBoost (Adaptive Boosting):

AdaBoost is an ensemble learning method that amalgamates weak classifiers into a single classifier that can be more accurate. In the cybersecurity arena, one of the contributions AdaBoost provides, is minimizing false negatives by emphasizing instances misclassified by previous model iterations. This adaptive learning has given AdaBoost the ability to best adapt to unique and novel attack vectors that are poorly modeled to traditional rule-based systems.

Random Forest Classifier:

The Random Forest Classifier (RFC) is an ensemble learning strategy used in machine learning for classification and regression problems. It is built on the foundation of decision trees. Many decision trees are trained and aggregated together to produce a more accurate and robust prediction. The algorithm is very popular because of its accuracy, ability to harness large datasets, and resistance to overfitting.

Random Forest Concept:

A Random Forest is a training set of decision trees that works together to produce a more accurate and generalizable prediction. The core premise is "the wisdom of the crowd." A random forest consists of many weak learners (decision trees) to produce a stronger learner. Each decision tree is trained using a random selection of the training data. The final prediction is achieved by aggregating the predictions of the individual trees.

Do's and Don'ts:

- It is an ensemble method - improves accuracy.
- It is an example of Bagging (Bootstrap Aggregation) - taking random samples of the data.
- It introduces random feature selection at each split, which increases diversity of the decision trees.
- It can be used for both classification and regression tasks.

The Random Forest classifier, among the many machine learning classifiers available, has drawn so much interest for its flexibility, strong accuracy, and ability with very large datasets and large number of features. Random Forests are considered ensemble methods as they limit the amount of variance of predicted outcomes by creating multiple decision trees and using the observations with those decision trees to combine outcomes to make better, more stable predictions. Also, it can be used for many classification problems. In network security, the use of these decision trees as new ideas may require fast and precise identification of malicious patterns from large data sets. This study outlined the possibility of using Random Forest's for better detection of threats in network environments. The study proposed a modelling approach with an understanding of traffic data being observed and the model attempting to grow and learn from known attacks or attack patterns. Other machine learning approaches are only conducted with historical traffic, but if they are successful, ultimately the model must be able to detect suspicious activity before considerable security damage occurs. The Random Forest model is best equipped to handle unbalanced data sets and minimize the effects of overfitting, both common issues in cybersecurity datasets Additionally, a web-based interface implemented in Django adds another layer of practicality to the model by allowing personnel to visualize detected threats in real-time. The fused application of both machine-learning and web technologies increases the detection performance, which returns great, usable value to potential users affording on-demand monitoring and assessment of potential threats.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

LITERATURE SURVEY

M. T. Abdelaziz et al., (2025). [1] published in the Journal of Network and Systems Management (2025), explores the application of machine learning techniques—specifically Random Forest (RF)—in improving Network Intrusion Detection Systems (NIDS). The authors propose an enhanced model that integrates permutation feature importance to improve the interpretability and performance of threat detection mechanisms.

J. Ramprasath et al. (2023) [2] proposed a method for detecting anomalous traffic in cloud services using the Random Forest algorithm. The study addresses growing concerns over security breaches in cloud environments due to unpredictable traffic patterns. By analyzing multiple datasets, the model demonstrated high accuracy and robustness in identifying anomalies. The Random Forest classifier outperformed traditional techniques in terms of precision and detection rate. This work contributes to improving real-time cloud monitoring and proactive threat management.

P. Bajpai et al. (2024) [3] explores cyber-attack detection within Internet of Things (IoT) environments using the Random Forest algorithm. It emphasizes the growing vulnerability of IoT devices to security threats due to their interconnectivity. The study demonstrates that Random Forest provides high accuracy and robustness in identifying malicious activities. The paper contributes to the development of intelligent and efficient intrusion detection systems for IoT networks.

Z. K. Maseer et al. (2021) [4] conducted a benchmarking study on various machine learning algorithms for anomaly-based intrusion detection using the CICIDS2017 dataset. The study compared classifiers including Decision Trees, Random Forest, SVM, and KNN in terms of accuracy, precision, recall, and F1-score. It highlighted that ensemble methods like Random Forest generally outperform individual classifiers. The paper also emphasized the challenges of imbalanced data and high false positives in intrusion detection systems.

Elmasri et al. (2020) [5] conducted an evaluation of the CICIDS2017 dataset to assess the effectiveness of various machine learning algorithms in detecting cyber threats. The study provides a qualitative comparison between classifiers like Decision Trees, Random Forest, and Naïve Bayes. Results highlight the strengths and weaknesses of each algorithm concerning accuracy and detection performance. The paper emphasizes the importance of dataset selection in IDS research. It concludes that no single model fits all scenarios, urging the need for hybrid and context-aware approaches.

K. Jiang et al. (2020) [6] proposed a novel intrusion detection framework that integrates hybrid sampling techniques with a Deep Hierarchical Network (DHN) to address data imbalance in network traffic. The model combines SMOTE and edited nearest neighbor (ENN) methods to enhance the quality of training samples. Their DHN architecture effectively captures deep abstract features from network data, improving classification performance. The approach was tested on benchmark datasets like NSL-KDD, showing superior accuracy and reduced false positive rates. This work highlights the importance of advanced preprocessing combined with deep learning for robust intrusion detection.

Bagui et al. (2023) [7] addressed the challenge of imbalanced datasets in network intrusion detection systems (NIDS), which often fail to identify rare but critical attack types. The study explored various resampling techniques to enhance the detection accuracy of underrepresented classes. By applying synthetic data generation and sampling strategies, the authors improved model performance for minority attacks. Their experiments demonstrated the effectiveness of balanced datasets in training machine learning classifiers. The paper highlights the importance of data preprocessing in developing reliable and robust intrusion detection systems.

Bagui and Li (2021) [8] address the problem of class imbalance in network intrusion detection datasets, which often leads to biased machine learning models. They explore various resampling techniques such as SMOTE, ADASYN, and random oversampling to improve detection accuracy. The study evaluates these methods on benchmark intrusion datasets like NSL-KDD and UNSW-NB15. Experimental results show that appropriate resampling significantly enhances the performance of classifiers. Their work emphasizes the importance of data balancing in building effective intrusion detection systems.

Research Article

Silva et al. (2021) [9] conducted a comparative study of under sampling techniques to improve the performance of Network Intrusion Detection Systems (NIDS). The study addressed the issue of class imbalance commonly found in intrusion detection datasets. Various under sampling methods were tested to evaluate their impact on detection accuracy and false positive rates. The authors found that specific under sampling strategies significantly enhanced classifier performance. Their findings support the importance of balancing data distribution in designing effective and reliable NIDS.

Megantara and Ahmad (2020) [10] focused on enhancing the performance of Intrusion Detection Systems (IDS) by employing feature importance ranking techniques. They evaluated various feature selection methods to determine the most influential attributes affecting IDS accuracy. The study utilized machine learning classifiers to assess system performance with and without feature optimization. Their findings revealed that reducing irrelevant features significantly improved detection accuracy and efficiency.

Reis, Maia, and Praça (2020) [11] investigates the feature selection and performance analysis on the CICIDS2017 dataset for intrusion detection systems. It explores various feature importance techniques to identify the most relevant attributes impacting model accuracy. The authors employ machine learning algorithms to assess detection performance using selected features. Their findings highlight how feature reduction enhances classification accuracy and reduces computational complexity. The study provides valuable insights into optimizing IDS models using efficient feature selection methods.

Disha and Waheed (2022) [12] conducted a performance analysis of various machine learning models for Intrusion Detection Systems (IDS). They introduced a Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique to enhance detection accuracy. The study evaluated multiple classifiers, including Random Forest, SVM, and KNN, on benchmark cybersecurity datasets. Results demonstrated that GIWRF significantly improved the classification performance and reduced computational overhead. The research highlights the importance of optimized feature selection in building efficient and accurate IDS models.

Latif et al. (2020) [13] proposed DRAIN, a Deep Random Neural Network model designed for intrusion detection in Industrial IoT (IoT) environments. The model addresses the increasing security threats in IoT by integrating deep learning with randomness to improve anomaly detection accuracy. It was trained and validated using real-world datasets, showing high detection rates and low false positives. The architecture outperformed traditional machine learning methods in identifying complex intrusion patterns. This study highlights the potential of advanced neural architectures in enhancing IoT cybersecurity.

Kumar et al. (2021) [14] proposed a fog-cloud integrated architecture utilizing ensemble learning to detect cyber-attacks in Internet of Medical Things (IoMT) networks. The framework improves threat detection accuracy and reduces latency by processing data closer to the source using fog computing.

Mahmood et al. (2021) [15] developed S-DPS, a Software-Defined Networking (SDN)-based system for defending smart grids against Distributed Denial of Service (DDoS) attacks. Their system enhances detection and mitigation efficiency by dynamically managing network traffic through programmable SDN controllers.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

EXISTING SYSTEM

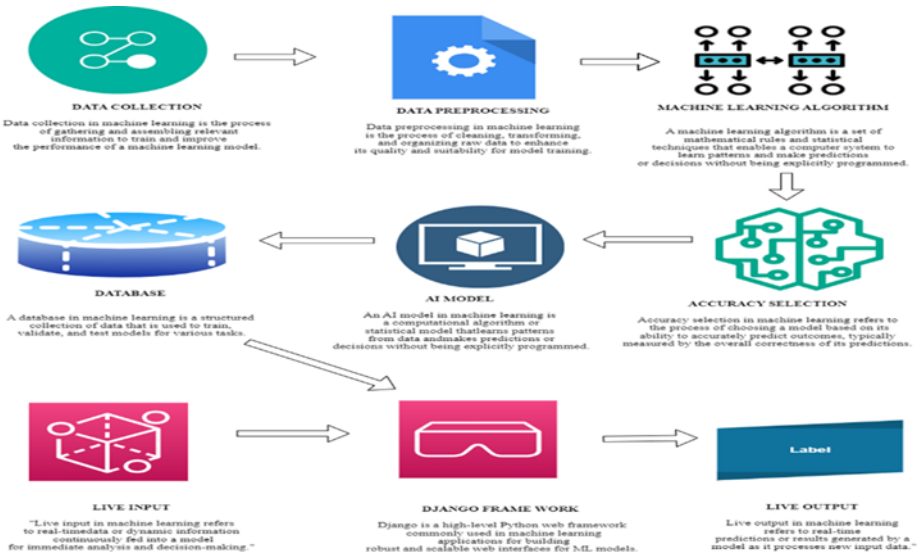

Fig. 1. Existing System Architecture

More than 150 cellular networks worldwide have rolled out LTE-M (LTE-Machine Type Communication) and/or NB-IoT (Narrow Band Internet of Things) technologies to support massive IoT services such as smart metering and environmental monitoring. Such cellular IoT services share the existing cellular network architecture with non-IoT (e.g., smartphone) ones. When they are newly integrated into the cellular network, new security vulnerabilities may happen from imprudent integration. In this work, we explore the security vulnerabilities of the cellular IoT from both system-integrated and service-integrated aspects. We discover several vulnerabilities spanning cellular standard design defects, network operation slips, and IoT device implementation flaws. Threateningly, they allow an adversary to remotely identify IP addresses and phone numbers assigned to cellular IoT devices, interrupt their power saving services, and launch various attacks, including data/text spamming, battery draining, device hibernation against them. We validate these vulnerabilities over five majors cellular IoT carriers in the U.S. and Taiwan using their certified cellular IoT devices. The attack evaluation result shows that the adversary can raise an IoT data bill by up to $226 with less than 120 MB spam traffic, increase an IoT text bill at a rate of $5 per second, and prevent an IoT device from entering/leaving power saving mode; moreover, cellular IoT devices may suffer from denial of IoT services. We finally propose, prototype, and evaluate recommended solutions.

PROPOSED SYSTEM

By exploiting artificial intelligence (AI) methods, the proposed predictive modeling system for network threat detection fortifies cybersecurity infrastructure. The proposed model attempts to detect and eliminate attacks proactively to protect critical networks, combining sophisticated machine learning algorithms and data analysis of complete network data. Continuous monitoring and analysis of network traffic patterns utilizing anomaly detection algorithms can reveal unusual behavior that may be indicative of cyber threats such as unauthorized access, malware infection, and data exfiltration. Furthermore, the system may be able to adapt and improve threat detection capabilities through the use of deep learning models trained on large collections of prior cyber events data. This enables the system to stay ahead of emerging threats and changing attack patterns. Additionally, incorporating AI-driven predictive modeling improves response times, enabling security teams to quickly implement countermeasures and minimize possible harm. The suggested system acts as a crucial defense mechanism against the constantly changing landscape of cyber threats by proactively protecting network infrastructures, guaranteeing a strong cybersecurity posture and continuous operations for businesses in a variety of industries. The Random Forest Classifier (RFC) is an ensemble learning method used in machine learning for classification and regression tasks. It is an extension of Decision Trees, where multiple decision trees are trained

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

and combined to produce a more accurate and robust prediction. This algorithm is widely used due to its high accuracy, ability to handle large datasets, and resistance to overfitting.

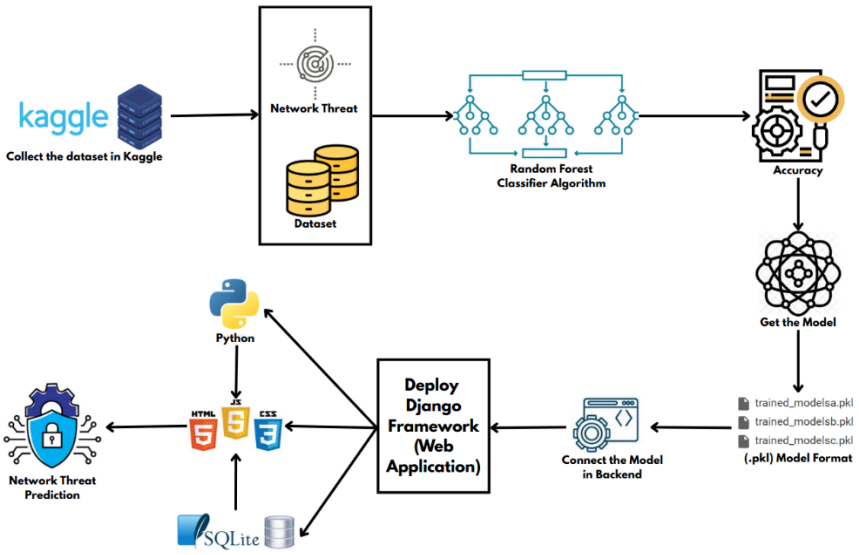

Fig. 2. Proposed System Architecture

RANDOM FOREST MODULES

Bootstrapping:

Random Forest starts by creating multiple subsets of the original dataset through a process called bootstrapping. This involves randomly sampling the data with replacement, creating new datasets of the same size as the original.

Building Decision Trees:

For each subset, a decision tree is constructed. Decision trees are built by selecting the best feature from a random subset of features at each node, considering various criteria such as Gini impurity for classification or mean squared error for regression.

Voting (Classification) or Averaging (Regression):

Once all the trees are built, they "vote" for the class (in classification) or provide a prediction (in regression) for a new data point. For classification, the class that receives the most votes becomes the predicted class. For regression, the predictions are averaged.

Reducing Overfitting:

One of the key advantages of Random Forest is that it reduces overfitting. Each tree in the forest is trained on a different subset of the data, and by averaging or voting, the model becomes more robust and less prone to the noise present in individual trees.

ALGORITHM

Gini Index & Entropy (Impurity Measures)

Random Forest uses decision trees, which employ impurity measures like Gini Index or Entropy to decide splits.

- **Gini Index (Gini Impurity):** Measures how often a randomly chosen element would be incorrectly classified.

Gini=1−∑pi2Gini = 1 - \sum p_i^2Gini=1−∑pi2

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Where pip_ipi is the probability of a class.

- **Entropy:** Measures the uncertainty in data.

Entropy=−∑pilog⁡2piEntropy = -\sum p_i \log_2 p_iEntropy=−∑pilog2pi

The lower the Gini or Entropy, the purer the node.

A Random Forest is a collection of decision trees that work together to make predictions. In this article, we'll explain how the Random Forest algorithm works and how to use it. Understanding Intuition for Random Forest Algorithm. Random Forest algorithm is a powerful tree learning technique in Machine Learning to make predictions and then we do vote of all the tress to make prediction. They are widely used for classification and regression task.

It is a type of classifier that uses many decision trees to make predictions. It takes different random parts of the dataset to train each tree and then it combines the results by averaging them. This approach helps improve the accuracy of predictions. Random Forest is based on ensemble learning. Imagine asking a group of friends for advice on where to go for vacation. Each friend gives their recommendation based on their unique perspective and preferences (decision trees trained on different subsets of data). You then make your final decision by considering the majority opinion or averaging their suggestions (ensemble prediction).

As explained in image: Process starts with a dataset with rows and their corresponding class labels (columns). Then - Multiple Decision Trees are created from the training data. Each tree is trained on a random subset of the data (with replacement) and a random subset of features. This process is known as bagging or bootstrap aggregating. Each Decision Tree in the ensemble learns to make predictions independently. When presented with a new, unseen instance, each Decision Tree in the ensemble makes a prediction. The final prediction is made by combining the predictions of all the Decision Trees. This is typically done through a majority vote (for classification) or averaging (for regression).

Key Features of Random Forest Handles Missing Data: Automatically handles missing values during training, eliminating the need for manual imputation. Algorithm ranks features based on their importance in making predictions offering valuable insights for feature selection and interpretability. Scales Well with Large and Complex Data without significant performance degradation. Algorithm is versatile and can be applied to both classification tasks (e.g., predicting categories) and regression tasks (e.g., predicting continuous values).

Assumptions of Random Forest: Each tree makes its own decisions: Every tree in the forest makes its own predictions without relying on others.

Random parts of the data are used: Each tree is built using random samples and features to reduce mistakes. Enough data is needed: Sufficient data ensures the trees are different and learn unique patterns and variety. Different predictions improve accuracy: Combining the predictions from different trees leads to a more accurate final result.

Fig. 3. Network Threat Home Page

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

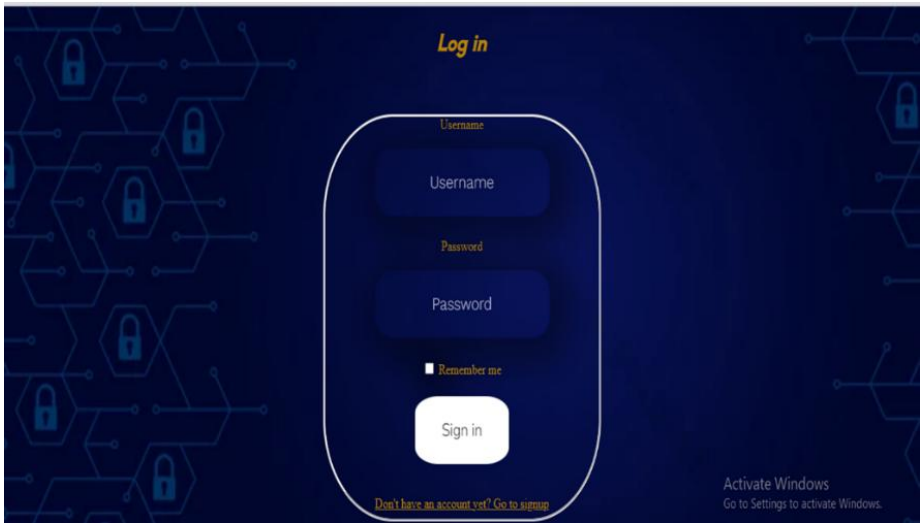

Fig. 4. Network Threat login Page

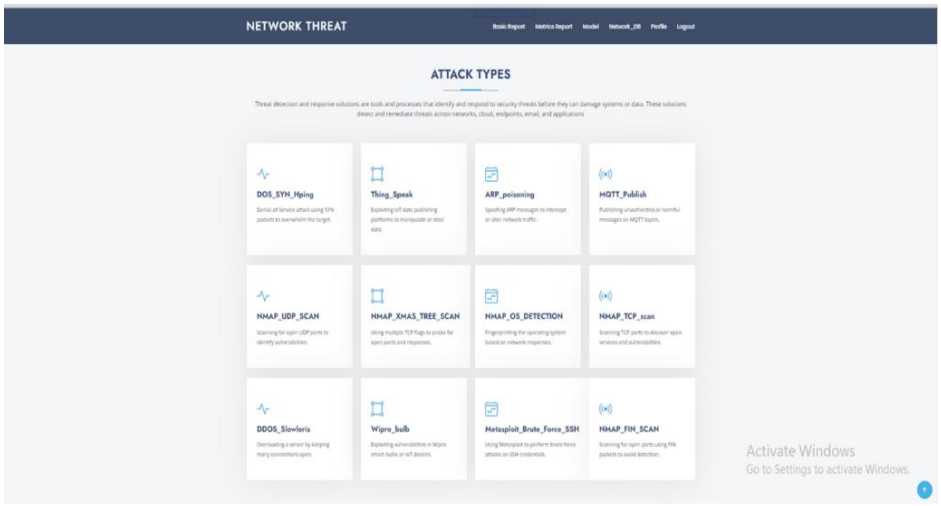

Fig. 5. Network Threat Attack types and Menu Page

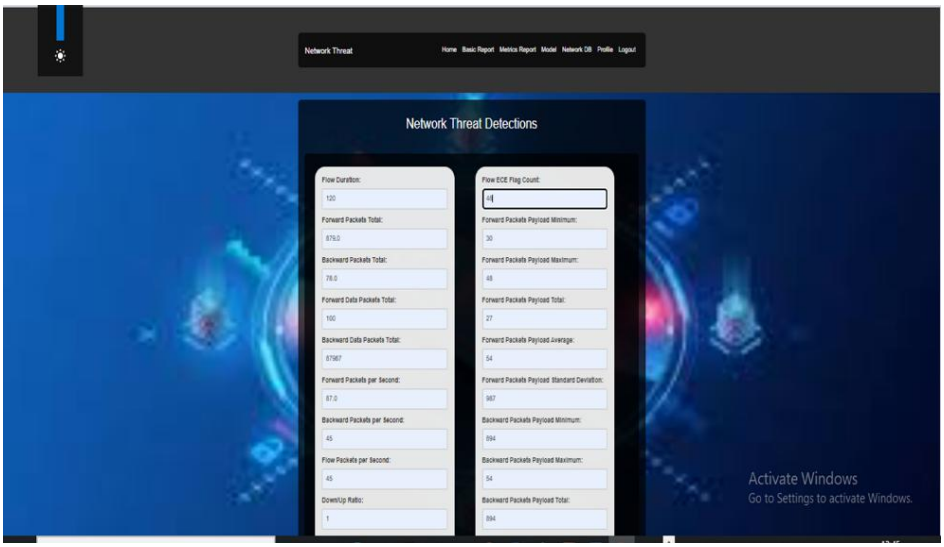

Fig. 6. Network Threat Detection View

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

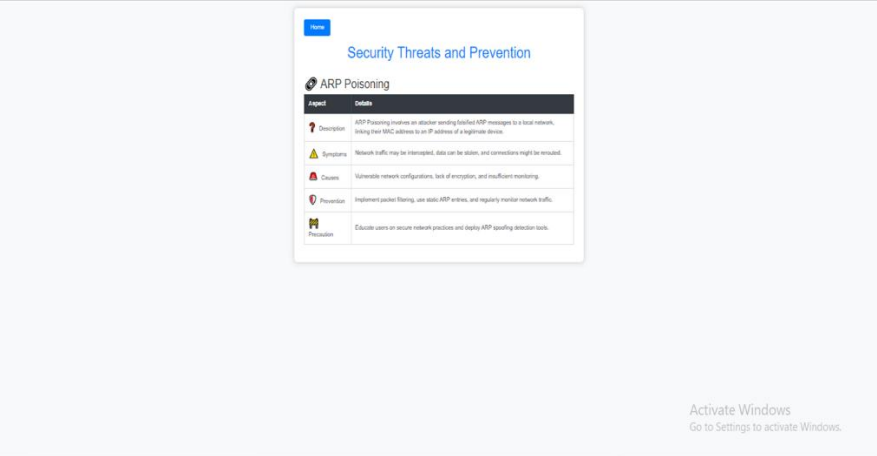

Fig. 7. Network Security Threats and Prevention Image

RESULT AND DISCUSSION

The system that was proposed used a Random Forest algorithm to improve network threat detection, coupled with a Django-based visualization interface. Two algorithms, Bernoulli Naive Bayes and Adaboost, were tested in previous research stages and achieved accuracies of 42% and 78% respectively. Although these models showed some promise, their performance was not adequate for high-reliability threat detection in dynamic networks. On the other hand, the application of the Random Forest algorithm provided a notable enhancement, with an accuracy rate of 98%. This increase is due to the ensemble attribute of the Random Forest, which promotes a decrease in variance and classification strength through the aggregation of several decision trees. It was also very useful in the detection of complex patterns in network traffic that went unspotted by the lesser models. The system analyzes high levels of network traffic, detects abnormal patterns, and issues early alerts before threats have a chance to escalate. The front end based on Django enables users to see detected anomalies in real-time, enabling instant response measures. The outcome shows the promise of marrying sophisticated AI models with intuitive interfaces to enable proactive network defense measures. Additionally, the very high accuracy produced by the Random Forest algorithm highlights its dependability for application in practical situations. The model's flexibility, with repeated learning from new data, provides a scalable and robust solution to threat detection. These results indicate that utilizing ensemble machine learning techniques can substantially enhance cybersecurity systems in ever more complex and connected networks.

CONCLUSION AND FUTURE WORK

This study aimed at improving network threat detection through the application of sophisticated machine learning methods, with specific focus on the Random Forest algorithm. The method combines predictive modeling features to examine intricate patterns in network traffic and detect potential threats with high accuracy. In contrast to conventional security systems that tend to respond to threats after their occurrence, this model prioritizes proactive defense through early threat detection. The use of the Random Forest classifier proved to be a much better performance than that of existing models, such as Bernoulli Naive Bayes and Adaboost, which have achieved accuracy values of 42% and 78% respectively. The Random Forest model, on the other hand, posted an impressive 98% accuracy, which verifies its effectiveness in real-time threat detection applications. In addition, the use of Django as a visualization framework brought significant value through its ability to present a simple, easy-to-use interface for security analysts to understand threat data. Not only did this simplify the monitoring process but also facilitated improved decision-making via real-time feedback and visual indication. The findings indicate that integrating effective classification algorithms with easy-to-use front-end technologies can make network defense mechanisms much faster and more accurate.

Research Article

Although the present deployment has shown encouraging results, there are various avenues to pursue in the future. Firstly, the training dataset can be augmented with newer and diverse types of cyberattacks to further equip the model with generalizability across disparate threat streams. Secondly, the model can be extended with advanced techniques like Long Short-Term Memory (LSTM) or Convolutional Neural Networks (CNNs) for deep learning to incorporate temporal or spatial patterns in sequential traffic data. Additionally, integrating this threat detection system with cloud-based systems could make it possible for large-scale deployment and real-time protection of distributed networks. Improvements to the Django interface can also include introducing alert systems, live updates, and API integration with other security solutions. In the end, the ultimate goal is to create an autonomous, adaptive threat detection system that can evolve along with new threats in an ever-changing digital environment.

REFERENCES

[1] M. T. Abdelaziz, A. Radwan, H. Mamdouh, and A. A. El-Sayed, "Enhancing Network Threat Detection with Random Forest-Based NIDS and Permutation Feature Importance," Journal of Network and Systems Management, vol. 33, no. 2, pp. 1–23, 2025, doi: 10.1007/s10922-024-09874-0.

[2] J. Ramprasath, S. Ramakrishnan, V. Tharani, R. Sushmitha, and D. Arunima, "Cloud Service Anomaly Traffic Detection Using Random Forest," in Advances in Data and Information Sciences, Lecture Notes in Networks and Systems, vol. 522, Springer, Singapore, 2023, pp. 263–272, doi: 10.1007/978-981-19-5292-0_25.

[3] P. Bajpai, "Cyber Attack Detection in an Internet of Things Employing Random Forest," Journal of Recent Innovations in Computer Science and Technology, vol. 1, no. 1, pp. 1–7, Oct. 2024, doi: 10.70454/JRICST.2024.10101.

[4] 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, doi: 10.1109/ACCESS.2021.3056614.

[5] T. Elmasri, N. Samir, M. Mashaly, and Y. Atef, "Evaluation of CICIDS2017 with Qualitative Comparison of Machine Learning Algorithm," in 2020 IEEE Cloud Summit, 2020, pp. 46–51, doi: 10.1109/IEEECloud Summit48914.2020.00013.

[6] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network Intrusion Detection Combined Hybrid Sampling with Deep Hierarchical Network," IEEE Access, vol. 8, pp. 32464–32476, 2020, doi: 10.1109/ACCESS.2020.2973730.

[7] S. Bagui, D. Mink, S. Bagui, S. Subramaniam, and D. Wallace, "Resampling Imbalanced Network Intrusion Datasets to Identify Rare Attacks," Future Internet, vol. 15, no. 4, p. 130, 2023, doi: 10.3390/fi15040130.

[8] S. Bagui and K. Li, "Resampling Imbalanced Data for Network Intrusion Detection Datasets," Journal of Big Data, vol. 8, no. 1, p. 13, 2021, doi: 10.1186/s40537-020-00390-x.

[9] B. Silva, R. Silveira, M. Silva Neto, P. Cortez, and D. Gomes, "A Comparative Analysis of Undersampling Techniques for Network Intrusion Detection Systems Design," Journal of Communication and Information Systems, vol. 36, no. 1, pp. 31–43, 2021, doi: 10.14209/jcis.2021.3.

[10] A. Megantara and T. Ahmad, "Feature Importance Ranking for Increasing the Performance of Intrusion Detection System," in 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE), 2020, pp. 1–6, doi: 10.1109/IC2IE50715.2020.9274570.

[11] Reis, E. Maia, and I. Praça, "Selection and Performance Analysis of CICIDS2017 Features Importance," in Advances in Intelligent Systems and Computing, vol. 1194, Springer, Cham, 2020, pp. 56–71, doi: 10.1007/978-3-030-45371-8_4.

[12] R. A. Disha and S. Waheed, "Performance Analysis of Machine Learning Models for Intrusion Detection System Using Gini Impurity-Based Weighted Random Forest (GIWRF) Feature Selection Technique," Cybersecurity, vol. 5, no. 1, p. 3, 2022, doi: 10.1186/s42400-021-00103-8.

[13] S. Latif, Z. Idrees, Z. Zou, and J. Ahmad, "DRAIN: A Deep Random Neural Network Model for Intrusion Detection in Industrial IoT," in 2020 International Conference on UK-China Emerging Technologies (UCET), 2020, pp. 1–4, doi: 10.1109/UCET51115.2020.9205486.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[14] P. Kumar, G. P. Gupta, and R. Tripathi, "An Ensemble Learning and Fog-Cloud Architecture-Driven Cyber-Attack Detection Framework for IoMT Networks," Computer Communications, vol. 166, pp. 110–124, 2021, doi: 10.1016/j.comcom.2020.12.017.

[15] H. Mahmood, D. Mahmood, Q. Shaheen, R. Akhtar, and W. Changda, "S-DPS: An SDN-Based DDoS Protection System for Smart Grids," Security and Communication Networks, vol. 2021, Article ID 6629098, 2021, doi: 10.1155/2021/6629098.

[16] Sihan Wang et al. "Dissecting Operational Cellular IoT Service Security: Attacks and Defenses." IEEE/ACM Transactions on Networking, vol. 32, no. 2, April 2024.

[17] Animesh Srivastava, Hemendra Shanker Sharma, Rishikesh Rawat, Navin Garg, "Detection of Cyber Attack in IoT Based Model Using ANN Model with Genetic Algorithm," 2024 International Conference on Computing, Power, and Communication Technologies (IC2PCT), 2024.

[18] Harsh Kumar, Patrick Nnaji, Sanjeev Kumar, "Smart Meter Performance Under Wired and Wireless Cyber Security Attack," 2024 IEEE World AI IoT Congress (AIIoT), 2024.

[19] José Cecílio, André Souto. "Security Issues in Industrial Internet-of-Things: Threats, Attacks and Solutions." 2024 IEEE International Workshop on Metrology for Industry 4.0 & IoT, 2024.

[20] Vikas Malhotra, Shaminder Kaur, Monika Parmar, "Algorithm for Generation of Over Clock Attack for Security Assessment of Cyber-physical Systems," 2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT), 2024.