Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

Predictive analysis network attack detection using AI model with advance technique

Sharnyaa. S[1], Girija. N[2], Jayshree Naykar. V[3]

[1]Associate professor
Dept.of Information Technology Panimalar Engineering College
Chennai, India
[2,3]Student of Information Technology Panimalar Engineering College Chennai, India

Abstract
In the evolving landscape of cybersecurity, the increasing frequency and sophistication of network attacks necessitate the adoption of advanced detection mechanisms. This paper presents an AI-driven approach for predictive network attack detection using machine learning models, including Random Forest, Decision Tree, and Extra Trees Classifier. The proposed system effectively identifies and classifies cyber threats such as Denial-of-Service (DoS) attacks, phishing attempts, and unauthorized intrusions. To enhance detection accuracy, advanced data preprocessing techniques—such as normalization, feature selection, and handling of imbalanced datasets—are incorporated. Random Forest and Decision Tree models are employed for their efficiency in managing hierarchical decision-making and complex data structures, while the Extra Trees Classifier offers robust similarity-based classification, enabling the detection of both known and emerging threats. The proposed methodology aims to deliver a scalable and highly accurate solution to mitigate network vulnerabilities and reinforce cybersecurity defenses. This work contributes to the advancement of predictive analytics in network security, enhancing digital resilience against evolving cyber threats.

Keywords – Cybersecurity, Machine Learning, Network Attack Detection, Predictive Analytics, Random Forest, Decision Tree, Extra Trees Classifier, Data Preprocessing

1. INTRODUCTION

Data science plays a transformative role in modern technology due to its ability to convert massive volumes of raw data into practical, insightful outcomes. Though its conceptual roots can be traced back to the mid-20th century, it has matured into a fully established discipline widely adopted across diverse domains. The explosive growth in data generation, combined with the urgent need for intelligent, data-driven decision-making, has significantly accelerated its relevance in recent years.This field encompasses a series of interconnected processes that begin with identifying the right problem to solve, followed by data collection, cleaning, transformation, model development, and finally, interpretation and presentation of results. Mastery of programming languages such as Python, R, Java, and SQL is essential for implementing algorithms and managing complex data workflows. In addition, data scientists employ machine learning methods—including classification, clustering, and regression—to derive meaningful insights from patterns in the data. Visualization tools and large-scale analytics platforms are also critical in conveying results clearly and making informed decisions. Artificial Intelligence (AI), a closely related discipline, focuses on creating intelligent systems that replicate cognitive abilities such as learning,

reasoning, and adaptation.

In fields like cybersecurity, AI enhances traditional detection systems by enabling them to continuously learn from incoming data and respond autonomously to new and evolving threats. These capabilities are increasingly vital in managing the scale and complexity of modern digital environments. Over the years, AI has evolved from early rule-based systems to advanced deep learning models that leverage neural networks and massive data sets. Machine learning, a key branch of AI, enables systems to learn from labeled examples and generalize those learnings to make future predictions. These technologies have become the backbone of applications like voice assistants, personalized recommendations, autonomous navigation, and real-time language translation. This capability is particularly useful in cybersecurity, where NLP can be used to scan phishing emails, analyze technical reports, or extract intelligence from unstructured text sources. Once limited to simple keyword detection, NLP has now advanced through deep learning and transformer-based architectures, resulting in more accurate and context-aware understanding of language.

Modern AI systems typically operate through a cyclical process involving learning from data, applying logic-based reasoning, and refining outputs through feedback mechanisms. These capabilities allow such systems to operate independently, adjust to new scenarios, and significantly reduce reliance on manual intervention in critical operations. In the context of cybersecurity, AI empowers systems to respond more rapidly to incidents, improve classification accuracy, and proactively identify suspicious behaviors in networks. In summary, the integration of data science and AI has revolutionized the approach to identifying and mitigating digital threats. Intelligent algorithms and scalable systems now enable organizations to transition from reactive to proactive security strategies. As digital infrastructures become more complex and interdependent, the continued convergence of these technologies will be essential to ensure robust protection and operational resilience.

2. LITERATURE SURVEY

Recent advancements in cybersecurity research have focused on enhancing malware detection and classification methods to address the limitations of traditional systems, especially in the face of rapidly evolving attack techniques. A variety of deep learning and machine learning approaches have been proposed to tackle these challenges and improve detection accuracy in real-world scenarios.

[1] Daniel Gibert (2016) addressed the increasing difficulty of detecting polymorphic and metamorphic malware, which frequently alter their structure to bypass signature-based security solutions. In his study titled "Convolutional Neural Networks for Malware Classification", he introduced two scalable CNN-based models. The first technique involved transforming malware binaries into grayscale images for feature extraction, while the second focused on classifying malware using x86 instruction sequences. Utilizing the Microsoft BIG 2015 dataset, these methods achieved high classification accuracies of 93.86% and 98.56%, respectively, showcasing the significant potential of deep learning in malware analysis.

[2] In a detailed survey, Adel Abusitta, Miles Q. Li, and Benjamin C. M. Fung (2019) examined the evolution of malware classification strategies and composition analysis techniques. Their study, "Malware Classification and Composition Analysis: A Survey of Recent Developments", explored how modern malware has grown increasingly complex, requiring more advanced detection mechanisms. The authors categorized existing methods based on the algorithms and features used and highlighted common strengths and limitations. Their work emphasized the value of understanding malware structure and behavior in order to improve detection systems and outlined future directions for enhancing classification techniques.

[3] Sumit S. Lad and Amol C. Adamuthe (2020) contributed to malware detection research by developing an improved CNN model that utilizes grayscale image representations of malware for classification. Working with the Maling dataset, which contains over 9,000 samples across 25 malware families, they applied data augmentation and preprocessing to enhance model performance. Furthermore, they introduced a hybrid architecture that integrates CNN feature extraction with a Support Vector Machine (SVM) classifier. This combination achieved a classification accuracy of 99.59% while significantly reducing computation time compared to standalone CNN models, demonstrating its efficiency and effectiveness for large- scale malware detection tasks.

[4] Fangtian Zhong and Xiuzhen Cheng (2023) explored the generation of adversarial malware designed to evade black-box detection models. In their work, "MalFox: Camouflaged Adversarial Malware Example Generation Based on Conv- GANs Against Black-Box Detectors", they developed a framework using convolutional generative adversarial networks (Conv-GANs) that employed three perturbation methods— Obfusmal, Stealmal, and Hollowmal—to modify malware features. Evaluations conducted on a comprehensive dataset revealed that the MalFox framework achieved 99.01% classification accuracy while simultaneously reducing the detection rate by an average of 45.1%. The system also increased evasion success by up to 56%, underscoring the need for more resilient detection mechanisms in adversarial contexts.

[5] Jiaxing Chen, Shiwen Sun, and Chengyi Xia (2010) investigated malware spread within wireless networks using a hypergraph-based propagation model. Their study, "Modeling and Analyzing Malware Propagation Over Wireless Networks Based on Hypergraphs", illustrated how malware can exploit the broadcast nature of wireless communications to infect multiple devices simultaneously. The model revealed that even in the absence of internet connectivity, malware could propagate rapidly, especially in dense and heterogeneous networks. Through mathematical modeling and simulations, the study demonstrated the heightened vulnerability of wireless environments and the importance of tailored prevention strategies.

[6] Adding to this body of work, Alazab et al. (2020) examined the application of machine learning models—specifically Random Forest, Naive Bayes, and Decision Tree—in detecting various forms of cyberattacks. Their research utilized the CICIDS2017 dataset to evaluate these models against threats such as brute-force attacks, denial-of-service attempts, and infiltration behavior. The authors highlighted that effective feature selection and data preprocessing significantly enhanced detection rates and reduced false positives, reinforcing the value of ensemble learning models for building intelligent intrusion detection systems.

3. EXISTING SYSTEM

In the modern digital ecosystem, cyber-physical systems (CPS) have emerged as critical infrastructure components across various domains such as healthcare, smart transportation, and industrial automation. These systems combine embedded computing with real-world physical processes to deliver automation, efficiency, and real-time responsiveness. However, their heavy reliance on interconnected networks and intelligent devices exposes them to a broader range of sophisticated cyber threats. Conventional cybersecurity frameworks, which often rely on predefined rules or static signatures, fall short in defending against these evolving and dynamic attack vectors, particularly when decisions need to be made in real time. To address these growing security concerns, researchers have increasingly turned toward deep learning (DL) as a promising tool for cyber threat detection. Deep learning models, known for their ability to extract layered, non-linear relationships from complex datasets, are well-suited for

uncovering previously unseen or stealthy attacks. Unlike traditional machine learning methods that often require manual feature engineering, DL algorithms can learn features autonomously from raw input, making them advantageous. These capabilities have led to a growing number of DL-based techniques being developed.

One of the major challenges lies in their computational intensity, which can hinder real-time implementation in resource-constrained environments such as embedded systems within CPS. Additionally, the availability of clean, labeled, and comprehensive datasets suitable for training high-performance DL models remains limited. Another concern is the lack of interpretability; DL models often operate as opaque systems with little transparency, making it difficult for human experts to understand the rationale behind specific predictions. This lack of explainability hinders trust and acceptance, especially in safety-critical applications where accountability is essential. Even minimal perturbations in input data can deceive a trained model into producing inaccurate classifications, thereby creating serious vulnerabilities. These adversarial weaknesses can be exploited by attackers to craft inputs that bypass security defenses undetected. While existing literature has proposed a variety of DL-based intrusion detection mechanisms, many still fall short in terms of scalability, robustness, and explainability. These challenges highlight the need for next-generation solutions that are not only accurate but also resilient, transparent, and capable of adapting to new forms of attack— goals that form the foundation of the system proposed in this study.

4. PROPOSED ARCHITECTURE

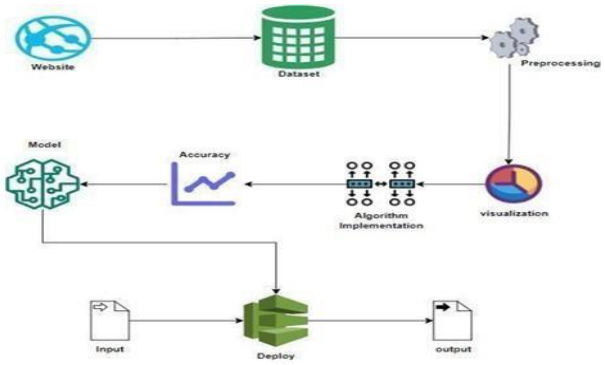

Figure 1: architecture diagram

The proposed system, titled Predictive Analysis Network Attack Detection Using AI Models with Advanced Techniques, is designed to tackle the growing complexity and sophistication of cyberattacks by employing intelligent and automated threat detection methods. Traditional intrusion detection systems (IDS) often depend on static rules and signature-based detection, making them ineffective against advanced or previously unseen attack patterns. To overcome these shortcomings, the proposed system integrates powerful machine learning algorithms using an ensemble learning strategy that enhances scalability, adaptability, and overall detection performance. This system focuses on identifying and classifying various types of network-based threats, including Denial-of-Service (DoS) attacks, phishing attempts, brute-force intrusions, and other forms of malicious activity. It utilizes three robust classifiers—Random Forest, Decision Tree, and Extra Trees Classifier—which are known for their efficiency in processing high-dimensional data, identifying complex feature interactions, and detecting subtle anomalies. By combining these models, the ensemble technique ensures improved accuracy and reduces the occurrence of false positives, which are critical challenges in network security.

a. Data Acquisition:

The process starts with the collection of network traffic datasets containing labeled examples of both normal and malicious activities. Well-known datasets such as CICIDS2017 or UNSW- NB15 are commonly used to train and validate the system, offering diverse attack scenarios and real-world traffic behaviour

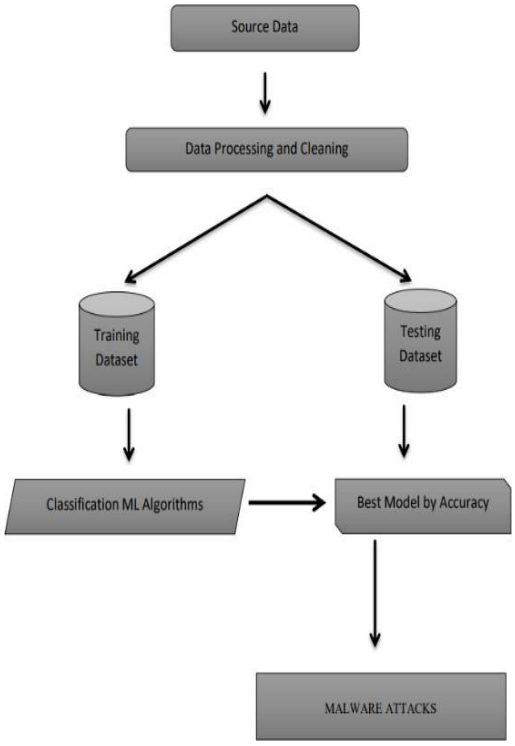

Figure 2: diagram of proposed model

b. Data Preprocessing and Cleaning:

Raw network data often includes inconsistencies and noise. Preprocessing ensures that the data is structured and relevant for model training. Data Cleaning: Removing duplicate entries, null values, and irrelevant features. Scaling numerical data to a standard range to optimize model convergence. Feature Selection: Identifying and retaining the most informative features using statistical and correlation-based techniques, thereby improving learning efficiency and reducing model complexity.

c. Dataset Splitting:

The training set is used to teach the model how to recognize malicious behaviour, while the testing set evaluates its ability to

generalize to new, unseen data. This ensures that the model can operate reliably in real-world conditions.

d. Model Training:

Each model learns to distinguish between benign and malicious traffic by identifying underlying patterns, relationships and statistical features associated with known attack types.

e. Best Model Selection

Once training is complete, each model is evaluated using metrics such as accuracy, precision, recall, and F1-score. The model with the highest performance—particularly in terms of recall and F1- score—is selected for deployment, as it provides the best balance between detecting true threats and minimizing false positives.

f. Real-Time Detection and Monitoring

The best-performing model is integrated into a real-time monitoring pipeline. In this environment, it continuously analyses live network traffic to detect anomalies and suspicious patterns. When a potential threat is identified, the system classifies it and triggers the corresponding response mechanisms.

g. Intelligent Alerting System

Upon detection of an attack, the system generates an automatic alert containing critical details such as the type of attack, its severity, time of occurrence, and affected endpoints. These alerts are sent to the security operations center (SOC) or administrators for immediate action, ensuring a rapid response to prevent further damage.

h. Adaptive Learning and Continuous Improvement

To remain effective against evolving threats, the system includes an adaptive learning component. This allows it to retrain periodically using newly acquired or labeled data, enabling the detection of zero-day threats and sophisticated malware variants that may not have been previously encountered.

System Architecture and Operational Flow:

the outlines the sequential workflow from data ingestion and preprocessing to model training, evaluation, and final deployment. The selected model is deployed in a real-time environment to classify incoming traffic and respond to potential threats based on learned patterns and behaviors. The diagram depicts key stages including data preprocessing, dataset partitioning, algorithm training and evaluation, and real-time malware detection using the most accurate model. In summary, the proposed system presents an intelligent, data-driven solution for network attack detection.

By leveraging ensemble machine learning models, advanced preprocessing techniques, and adaptive learning, the system achieves high detection accuracy while minimizing false positives. Its integration of real-time traffic analysis, automatic alerting, and continuous model improvement makes it suitable for dynamic, large-scale network infrastructures. This proactive approach significantly strengthens an organization's cyber resilience and ensures effective mitigation of both known and emerging cyber threats.

5. MODULE DESCRIPTION

a. Data Preprocessing

It focuses on improving data quality by handling missing values, duplicates, inconsistent formats, and outliers. Validation techniques are also used to estimate error rates and enhance model reliability, especially with limited or biased datasets. The process includes detecting and fixing missing entries, incorrect data types, and redundant records. Tools like Python's Pandas library tasks type conversion and data cleaning. Missing data is addressed using strategies like mean or median replacement, preserving dataset integrity. Feature engineering transforms raw inputs into useful representations. Techniques like normalization, encoding of categorical data, and feature selection (e.g., correlation analysis or recursive elimination) reduce complexity and improve learning efficiency. These steps help algorithms identify relevant patterns more accurately.

Preprocessing also handles imbalanced datasets through methods like SMOTE and corrects outliers using statistical measures (e.g., Z-score, IQR). Finally, splitting the dataset into training, validation, and testing sets—along with k-fold cross-validation— ensures robust evaluation. Together, these practices refine the dataset into a high-quality, structured input, optimizing model performance and generalization.

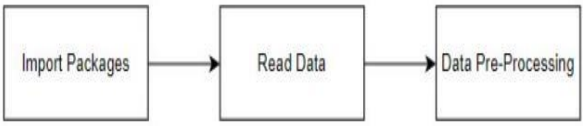

Figure 3: module diagram of data preprocessing

b. Data Visualization

Data visualization acts as a crucial link between raw datasets and actionable insights, making it an essential component of applied statistics and machine learning. While statistical methods emphasize numerical analysis, visualization provides a more intuitive and qualitative view, enabling analysts to spot trends, anomalies, and inconsistencies with greater ease. Common visualization techniques—such as line charts for time-series data, bar graphs for category comparisons, histograms for distribution patterns, and box plots for outlier detection—simplify complex datasets and improve communication of findings to stakeholders.

Beyond interpretation, visualization also contributes to feature engineering by revealing correlations and variable importance, which aids in selecting the most impactful attributes for model training. More advanced tools, including scatter plots, heatmaps, and correlation matrices, offer clarity when working with high- dimensional data. Interactive visual dashboards further enhance exploratory data analysis by allowing real-time filtering and manipulation. As machine learning models become more sophisticated, visualization techniques like SHAP (Shapley Additive explanations) values and partial dependence plots support model transparency and interpretability, making them essential in understanding algorithmic behavior.

The data processing workflow typically involves three main steps: importing necessary libraries, loading the dataset, and preparing the data for analysis. In the first step, tools such as Pandas, NumPy, and Matplotlib are imported to manage data structures and generate visual representations. The second step includes acquiring datasets from various sources such as CSV files, databases, or APIs. Finally, during pre-processing, the data is cleaned and transformed—missing values are handled, attributes are

normalized, and categorical variables are encoded— ensuring the dataset is ready for further statistical analysis or machine learning modeling.

By combining visual exploration with structured processing, this approach ensures high-quality inputs, supports informed decision-making, and enables deeper insights into both data and model performance.

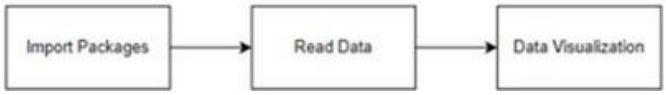

Figure 4: module diagram of data visualization

c. Decision Tree Classifier

A Decision Tree Classifier is a widely used algorithm in supervised learning that operates by dividing the input data into subsets based on feature values, forming a branching structure that resembles a tree. Each decision node in the tree is constructed by evaluating which feature provides the most informative split, typically measured using criteria like Gini Index or Information Gain derived from Entropy. The process continues recursively, creating branches until certain conditions—such as reaching a defined tree depth or a minimum number of samples—are met. At the end of each branch, or leaf node, the algorithm assigns a predicted class label based on the dominant class of the samples that reached that point. The hierarchical structure visually maps out how decisions are made, making it easy for users and stakeholders to follow the logic behind predictions. These models also work well with both categorical and numerical data, and they naturally rank the importance of features, which helps in understanding which variables contribute most to the final prediction.

Nonetheless, decision trees have certain drawbacks. They can overfit the training data if allowed to grow without constraints, leading to poor generalization on unseen data. Furthermore, when the data is imbalanced, decision trees may lean toward the majority class, reducing accuracy for minority classes unless techniques like cost-sensitive learning, pruning, or ensemble methods are applied. Even with these limitations, decision trees remain a popular choice in domains requiring transparent decision rules and fast inference. They are commonly applied in financial scoring systems, medical diagnostics, customer behavior analysis, and fraud detection, among others. Their balance of flexibility, transparency, and effectiveness makes them a reliable tool for both simple and moderately complex classification tasks.

Extra Tree Classifier

Unlike traditional decision trees or even Random Forests, this model introduces a higher degree of randomness by selecting split thresholds completely at random instead of using impurity- based measures like Gini or Entropy to determine the optimal split. This added randomness acts as a form of regularization, which helps the model reduce the risk of overfitting, especially when applied to large or complex datasets. The training process involves constructing a large ensemble of trees, each trained on random samples of the data. However, what sets Extra Trees apart is how splits are made: for each node, both the feature and the threshold for splitting are chosen randomly, rather than through an exhaustive search for the best split.

This makes them particularly efficient for high-dimensional datasets where feature space is large. They are also more robust to noise and perform well with limited hyperparameter tuning. However, due to their reliance on randomized splits, Extra Trees models are generally less interpretable than traditional decision trees, making it harder to understand the reasoning behind individual predictions. Despite the

trade-offs, Extra Trees classifiers are widely adopted in real-world applications that demand high accuracy and efficiency. Their ability to handle large-scale data while mitigating overfitting makes them suitable for tasks such as fraud detection, gene expression analysis, and document categorization. By leveraging randomness and ensemble learning, Extra Trees provide a fast and effective solution for classification and regression tasks in modern machine learning workflows.

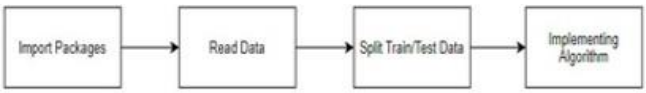

Figure 5: module diagram of Extra Tree Classifier

d. Random Forest Algorithm

The Random Forest algorithm is a widely recognized ensemble technique designed to enhance the reliability and accuracy of decision tree models. It achieves this by constructing a large number of decision trees and combining their outputs to form a more stable and generalized prediction. This ensemble strategy reduces the impact of individual tree errors, leading to improved performance across diverse datasets. Additionally, at each node split, the model considers only a random subset of features rather than the entire feature set, which helps prevent overfitting and reduces correlation between trees.

It performs well even when dealing with datasets that contain a large number of features or missing values. Another benefit is its ability to calculate feature importance scores, which rank features based on their contribution to predictive performance. This adds interpretability to the model, helping practitioners identify which variables are most influential in decision-making. Such capabilities are especially valuable in domains like healthcare diagnostics, credit risk evaluation, image recognition, and fraud detection. While Random Forest is relatively easy to implement, it can be computationally intensive depending on the number of trees and depth of each tree. Fine-tuning hyperparameters such as the number of estimators, maximum depth, and minimum samples per split is often required to balance performance with efficiency. Nonetheless, its ability to reduce variance, handle complex datasets, and deliver strong predictive accuracy makes Random Forest one of the most trusted and widely used algorithms in modern machine learning applications.

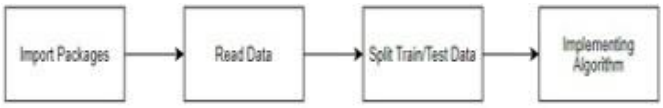

Figure 6: module diagram of Random Forest Classifier

6. RESULT

The login interface features a modern UI with cybersecurity- themed visuals, enhancing both usability and security

Figure 7: login page

Figure 8 illustrates the Cyber Network Detection webpage, which highlights the importance of securing networks using advanced technology and expert knowledge. The interface includes navigation options and a visual representation of cybersecurity mechanism.

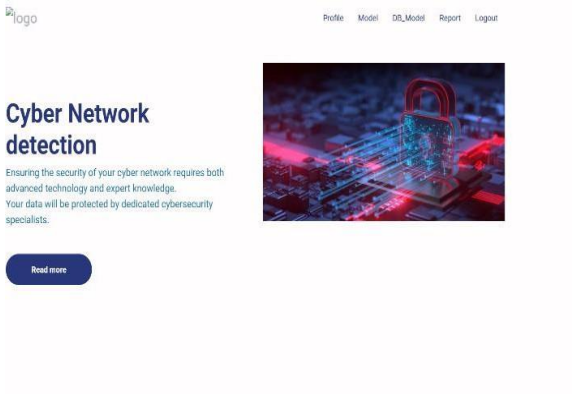

Figure 8: Home Page

Figure 8 presents a dataset extracted from a database, displaying various network flow parameters such as packet length statistics and flow metrics. This structured data is essential for analyzing network traffic behavior and detecting anomalies in cybersecurity applications.

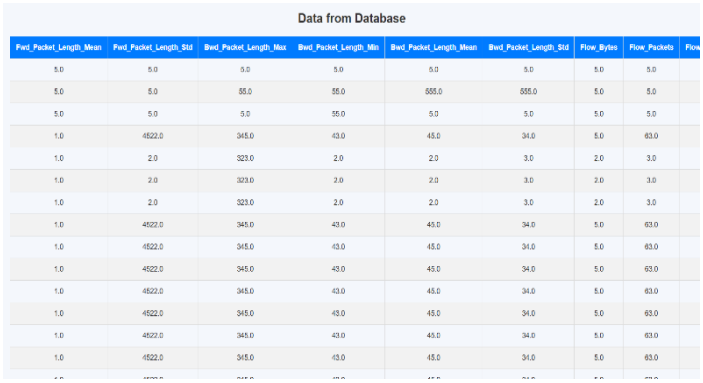

Figure 9: Data From Database

Figure 9 presents a dataset extracted from a database, displaying various network flow parameters such as packet length statistics and flow metrics. This structured data is essential for analyzing network traffic behavior and detecting anomalies in cybersecurity applications.

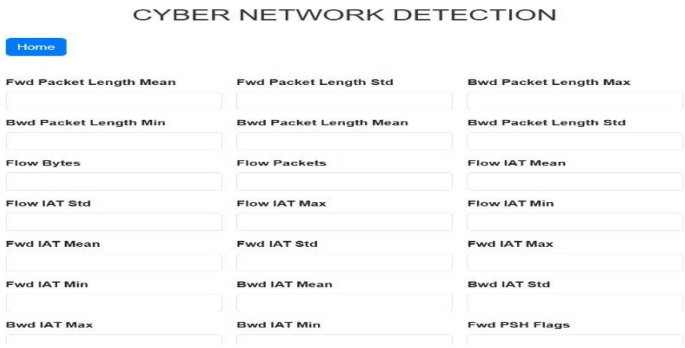

Figure 10: Input Data

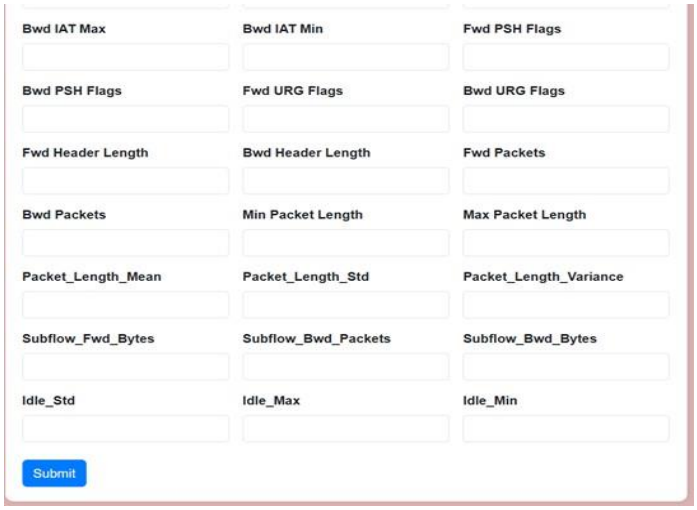

Figure 11: After Filling the input proceed to submit

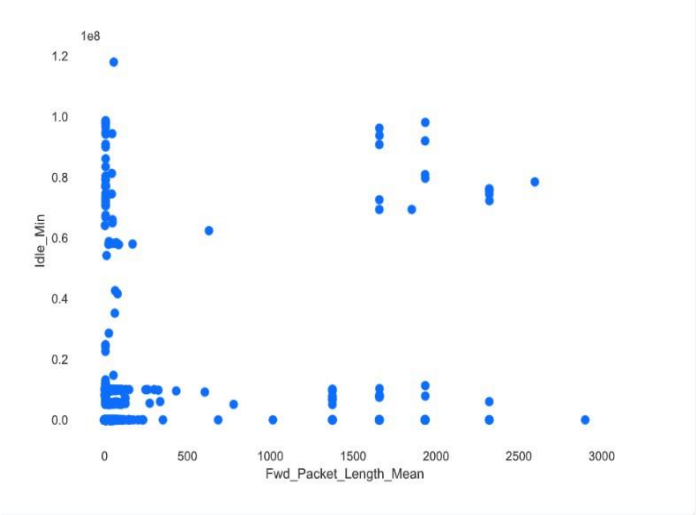

Figure 12: Final output

Figure 12 illustrates the scatter plot of Idle_Min versus Fwd_Packet_Length_Mean. The data exhibits a high concentration of points at lower packet lengths, with several outliers showing significantly high idle times. This distribution suggests potential clustering and anomalies, indicating the need for further statistical analysis.

7. CONCLUSION

The integration of machine learning into malware detection systems, particularly through the analysis of memory dumps, represents a promising advancement in the field of cybersecurity. Memory dump analysis provides a rich source of information for identifying malicious behaviours that may not be evident through traditional detection methods. By leveraging machine learning algorithms, it becomes possible to automatically uncover hidden patterns and subtle anomalies that signal the presence of malware. These intelligent systems enable proactive threat identification by detecting irregularities in volatile memory, offering a powerful response to the growing complexity of modern cyber threats. As adversaries continue to develop more advanced attack techniques, the adaptability and learning capabilities of machine learning models make them an essential component of next-generation defence mechanisms. Moreover, the ability of these models to generalize across varying types of malware and evolving attack signatures enhances the overall resilience of digital systems. As the field of machine learning progresses, its continued integration into cybersecurity architectures will play a vital role in protecting sensitive data, minimizing attack surfaces, and ensuring robust, real-time threat mitigation.

8. FUTURE SCOPE

The proposed system lays a strong foundation for intelligent malware detection using machine learning and memory dump analysis; however, there is substantial potential for future advancements. One key direction involves expanding the model to support real-time detection in live environments, enabling immediate threat response without relying solely on post-incident memory snapshots. Integrating the system with live monitoring tools could provide continuous surveillance of system memory, improving detection speed and reducing incident response time. Future work can also explore the use of deep

learning models, such as LSTM or transformer-based architectures, which are capable of understanding temporal and sequential patterns within memory activity. These models may enhance the system's ability to detect complex, stealthy malware that evades traditional detection techniques. Additionally, incorporating adversarial training can improve model robustness against evasion tactics employed by attackers to manipulate detection outputs. Another promising area lies in the creation of a comprehensive threat intelligence platform that combines static analysis, behavioral profiling, and memory-based insights into a unified detection system. Support for cross-platform analysis, including mobile and cloud-based systems, can further extend the applicability of the model in diverse computing environments.

Finally, introducing explainable AI (XAI) techniques will improve transparency and user trust by providing clear justifications for each detection decision. This is especially valuable in sectors where accountability is critical, such as healthcare, defense, and finance. With continuous refinement and integration of advanced AI techniques, the system holds great promise for evolving into a scalable, adaptive, and industry-grade cybersecurity solution capable of combating the next generation of malware threats.

REFERENCE

1. D.Gibert(2016): Proposed a CNN-based malware classification method using grayscale images and x86 instructions, achieving 93.86% and 98.56% accuracy with the BIG 2015 dataset.
2. A.Abusitta, M.Q.Li, and B.C.M. Fung (2019): Presented a survey of malware classification techniques, feature extraction methods, and evasion strategies, highlighting current challenges and future directions.
3. S.S.Lad and A.C. Adamuthe (2020): Introduced a hybrid CNN-SVM model for malware detection using grayscale images, achieving 99.59% accuracy and outperforming traditional CNN models.
4. F.Zhong and X.Cheng (2023): Introduced MalFox, an adversarial malware generation approach using convolutional GANs, designed to evade black-box detection systems, achieving a high detection evasion accuracy of 99.01%.
5. J. Chen,S. Sun, and C.Xia (2010): Proposed a hypergraph-based model to analyze malware spread in wireless networks, revealing high vulnerability even without internet access.