IJSES
Volume 9, Issue 11, pp. 39-46, 2025.
International Journal of Scientific Engineering and Science
ISSN (Online): 2456-7361

Intelligent Detection of ARP Spoofing Attacks Using Machine Learning

Buli Fakada Taresa[1*], Dr. Gaurav Gupta[2]
[1] Department of Computer Science & Engineering, Punjabi University, Patiala
[2]Assistant Professor, Computer Science & Engineering, Punjabi University Patiala
Email Address: gaurav.shakti@gmail.com

Abstract: *Address Resolution Protocol (ARP) spoofing is a common attack in networks that currently enables adversaries to interfere with and interrupt or disrupt line communications between networks. Traditional security mechanisms, such as statistical ARP entries and intrusion detection systems, are often too difficult to detect fraudulent attempts in real time. In this research, we propose to detect ARP spoofing attacks using machine learning-based methods. The proposed system distinguishes its intrinsic properties like packet frequency, MAC–IP map matching, and anomalous patterns in ARP traffic. Using these supervised learning models, we train classifiers to detect whether ARP packets are legitimate or spurious. Experimental evaluations show that our method achieves high accuracy, precision, and recall, significantly outperforming conventional detection techniques. This research helps strengthen the potential of machine learning based on ARP Spoofing threats for network security.*

Keywords: *ARP Spoofing, Network Security, Machine Learning, Intrusion Detection, Cyber security.*

I. INTRODUCTION

The acronym for Address Resolution Protocol is ARP. IP (Internet Protocol) addresses are mapped to hardware addresses using MAC address and its matching IP address are correlated using a table, commonly referred to as the ARP cache [1].

The mapping of IP addresses to corresponding MAC addresses is the responsibility of the Address Resolution mechanism (ARP), a layer-2 communication mechanism. The ARP process excludes any authentication or integrity checks and ignores whether the packet originates from a legitimate source. Any node can send an ARP reply without first receiving an ARP request because ARP is stateless. ARP-based spoofing Man in the middle is a type of cyber-attack that uses flaws in the Address Resolution Protocol (ARP) to link the attacker's Media Access Control (MAC) address to a valid host's IP address. This enables unauthorized users to discreetly access intercepted or altered material. Numerous other crucial attacks, including Denial-of-Service (DDoS) and session hijacking attacks, can be made easier by ARP spoofing attacks [2].

Address Resolution Protocol (ARP) packets are exchanged between devices in a local network. The system gathers all ARP requests and replies to monitor the communication between devices. Real-time ARP data is gathered during the thorough training phase of the suggested work Figure 1 to address the need for efficient ARP Spoofing detection. Using predetermined logic that critically compares, this entails extracting pertinent features and applying labeling. There are three stages to a complete man -in-the- middle assault that uses ARP spoofing. By inserting himself between two valid hubs of hosts, the attacker uses ARP spoofing to sever their link and replace it with: PC1 , PC2, PC3. The data frame between the two legitimate hosts is intercepted, filtered, or altered as needed. The collected data frame is transmitted to ensure on going connectivity. A simple ARP Spoofing attack model.

The technique of forging ARP packets to mimic another host on the network is known as ARP spoofing. The most prevalent form of ARP spoofing involves the attacker repeatedly sending the victim phony ARP replies. The operating system's ARP cache entry timeout period for the victim host is significantly longer than the interval between the spoof responses.[3]

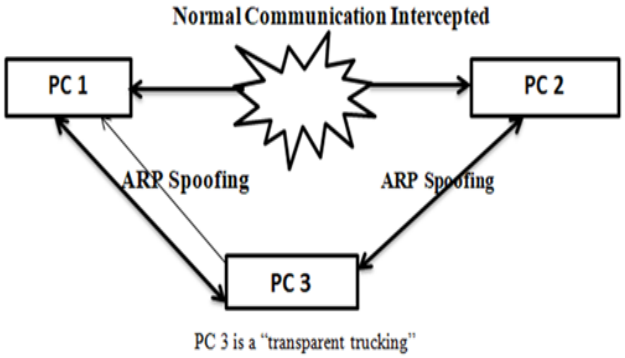

Figure 1: ARP Spoofing Attack

ARP spoofing attacks occur in regular networks when a fake IPMAC address mapping is added to the hosts' ARP cache. These ARP vulnerabilities are used by attackers to carry out the ARP Cache Poisoning attack, and end users are typically not aware that they are being spoofed [4]

By converting IP addresses to MAC (Media Access Control) addresses, the Address Resolution Protocol (ARP) is essential to IPv4-based local area networks (LANs). To find the MAC address that corresponds to a known IP address, a device in a LAN sends an ARP request to another device. Direct data-link layer communication is made possible by the legitimate owner of that IP responding with its MAC address.

The address resolution protocol (ARP) is one of the most important ways for computers to talk to each other on a local

IJSES
Volume 9, Issue 11, pp. 39-46, 2025.
International Journal of Scientific Engineering and Science
ISSN (Online): 2456-7361

area network. To talk to clients on a LAN, you need to know their MAC addresses. The ARP is a process that finds the MAC address of a specific IP address. For instance, if Client A wants to talk to Client B, Client A can send a broadcast ARP request. Client B will then respond to Client A with an ARP reply that includes its MAC address [6].

The Address Resolution Protocol (ARP) plays a critical role in IPv4-based local area networks (LANs) by mapping IP addresses to MAC (Media Access Control) addresses. When a device wants to communicate with another device in a LAN, it sends an ARP request to determine the MAC address corresponding to a known IP address. The legitimate owner of that IP responds with its MAC address, allowing direct communication at the data-link layer.

However, the ARP protocol is inherently vulnerable because it lacks authentication mechanisms. This flaw allows malicious actors to forge ARP replies and associate their own MAC addresses with legitimate IP addresses in the network. An attacker conducts an ARP spoofing attack by sending bogus ARP messages that link their MAC address to the IP address of someone else in the neighborhood. host, such as the DNS server or default gateway. As a result, traffic intended for the legitimate device is sent to the attacker instead, enabling interception, modification, or redirection of data packets

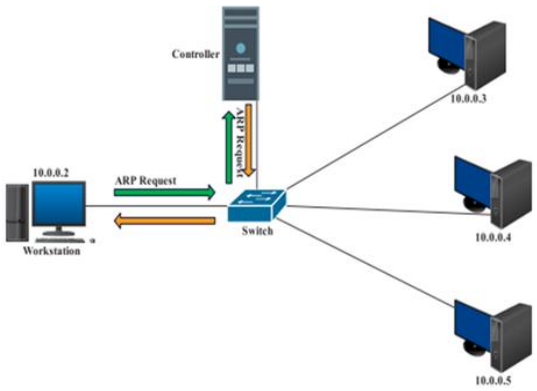

Figure 2: ARP Packet Transfer[7]

1.1. Problem statement of the study

ARP spoofing remains a prevalent and damaging form of network attack. Despite being a well-known vulnerability, its detection and prevention continue to challenge cyber security

An attacker who successfully executes an ARP spoofing attack can intercept sensitive information, launch Denial-of-Service (DoS) attacks, or redirect network traffic for malicious purposes (Srinivas et al., 2019). The primary issue lies in the fact that ARP, by design, does not authenticate incoming requests or responses, making it an easy target for malicious entities.

While some signature-based intrusion detection systems (IDS) attempt to identify known attack patterns, they are often ineffective against new, unseen variations of ARP spoofing. These systems also suffer from high false positive and false negative rates. Thus, there is a pressing need for an adaptive, data-driven approach capable of learning from network traffic patterns and generalizing to detect both known and previously unseen ARP spoofing attacks.

1.2. Motivation of the study

Recent advancements in machine learning (ML) have opened up new opportunities for intelligent network intrusion detection. Unlike traditional rule-based systems, ML models can learn complex patterns and behaviors from data, allowing them to identify anomalies that may indicate security breaches. ML algorithms such as Support Vector Machines (SVM), Random Forest, K-Nearest Neighbors (KNN), and Neural Networks have demonstrated success in various domains of cyber security, including malware detection, phishing detection, and anomaly-based intrusion detection (Revathi & Malathi, 2013).

The motivation behind this research is to harness the potential of machine learning to develop an effective and efficient detection system for ARP spoofing attacks. By analyzing network traffic features—such as packet arrival times, ARP request-response consistency, MAC-IP mappings, and sender-receiver behavior—ML models can be trained to differentiate between legitimate and malicious ARP activities. Such a solution not only enhances detection accuracy but also helps in reducing false alarms, making it more practical for real-time deployment in enterprise environments.

1.3. Contributions

This Research tries to contribute to the fields of Network security and Machine Learning Based model in this aspect. Integrated ARP Spoofing Detection: We propose an approach that combines network traffic data with process data to improve the detection of cyber-attacks in Network security environments. This hybrid approach enables a more comprehensive detection strategy by capturing both network security ARP spoofing attacks and operational deviations. ML Models for ARP Spoofing attack: A range of machine learning models, including Random Forests, Support Vector Machines (SVM), and Naïve Bayes, are applied. This research explores how these models perform binary classification when trained on a combined dataset compared to when they are trained on either network or process data alone.

1.4. Objectives of the study

This research aims to develop a machine learning-based system capable of detecting ARP spoofing attacks with high accuracy and low latency. The specific objectives include:

- To analyze ARP traffic and extract relevant features indicative of spoofing behavior.
- To evaluate and compare the performance of various machine learning classifiers in detecting ARP spoofing attacks.
- To implement an anomaly-based detection mechanism capable of identifying both known and novel attack patterns.
- To minimize false positives and false negatives, thereby increasing the reliability of the detection system.
- To propose a lightweight and scalable architecture suitable for real-time network environments.

1.5. Significance of the Study

The significance of this study lies in its potential to address a critical vulnerability in modern network infrastructure. As networks continue to grow in complexity and scale, traditional security mechanisms are no longer sufficient to protect against sophisticated threats. By incorporating machine learning into the detection process, this research contributes to the development of autonomous and intelligent security systems that can proactively identify and mitigate ARP spoofing attacks. Moreover, the proposed system can serve as a foundation for future research in machine learning-based intrusion detection, extending beyond ARP spoofing to encompass other layers and types of attacks. It also contributes to the growing field of cyber threat intelligence, where data-driven approaches are used to detect, analyze, and respond to cyber security incidents.

1.6. Challenges in ARP Spoofing Detection

- Data Collection: Obtaining a high-quality dataset that accurately reflects both normal and spoofed ARP traffic is non-trivial. Many public datasets are outdated or lack detailed ARP-specific information.
- Feature Engineering: Identifying the right set of features that can distinguish between legitimate and malicious ARP packets is crucial for model performance.
- Real-Time Detection: Ensuring that the model operates efficiently in real-time without introducing latency into the network is a key requirement.

1.7. Overview of Machine Learning Techniques

One subfield of artificial intelligence is machine learning, which allows systems to learn from data and get better at what they do over time without needing to be specifically coded. In the context of ARP spoofing detection, various Machine Learning techniques are used to identify abnormal patterns in network traffic. By identifying the ideal hyper plane that divides data points, the supervised learning method known as Support Vector Machine (SVM) excels at binary classification problems. An ensemble technique called Random Forest (RF) creates numerous decision trees to increase accuracy and minimize over fitting**.** K-Nearest Neighbors (KNN) classifies data based on proximity to labeled examples, making it simple and effective for small datasets. Naive Bayes (NB) uses probability and Bayes' theorem to classify data efficiently, especially in large-scale datasets. (XGBoost) is an unsupervised anomaly detection algorithm that isolates rare behaviors, ideal for spotting spoofing attempts without labeled data. These techniques offer varied strengths in detection accuracy, scalability, and computational efficiency, making them valuable tools in securing networks against ARP spoofing attacks.

1.8. Machine Learning-Based Detection

Machine learning-based detection leverages intelligent algorithms to identify ARP spoofing attacks by analyzing patterns in network traffic. Unlike traditional rule-based systems, ML models can learn from historical data and detect both known and unknown attack behaviors. By extracting key features such as MAC-IP inconsistencies, unusual ARP reply frequencies, and timing anomalies, ML algorithms like SVM, Random Forest, KNN, and XGBoost can accurately differentiate between legitimate and malicious activities. This approach enhances detection accuracy, reduces false positives, and supports real-time monitoring, making it highly effective for securing modern network environments. Numerous developments in the field of artificial intelligence, such as chess engines, self-driving cars, and numerous other tools that improve people's daily lives; have been fueled by improvements in computing power and storage capacity over the past few decades. Networks are another area where machine learning (ML) has found use.

II. Literature Review

The evolution of cyber threats has prompted significant advancements in intrusion detection systems (IDS), particularly with the integration of machine learning (ML) techniques. Address Resolution Protocol (ARP) spoofing is one of the ongoing and harmful attack vectors in local area networks. This literature review explores the existing work on ARP spoofing detection, highlighting traditional detection methods, the application of ML techniques, and their respective strengths and weaknesses. The cyber-security research community has become more aware of ARP spoofing attacks due to their frequency in both public and enterprise networks. A thorough review of the state of the art is provided in this section, covering hybrid approaches, machine learning-based anomaly detection, conventional ARP spoofing detection techniques, real-time system difficulties, and recent developments.

a. ARP Spoofing: Nature and Threats

The stateless ARP protocol maps IP addresses to MAC addresses in a network. Its lack of authentication makes it susceptible to spoofing attacks, where malicious entities forge ARP replies to mislead a host about the MAC-IP mapping. This enables attackers to perform man-in-the-middle, session hijacking, and denial-of-service attacks (Gupta & Shukla, 2020).

Despite being widely known, ARP spoofing attacks are difficult to detect using conventional security mechanisms due to their low visibility and minimal resource usage.

b. Traditional Detection Approaches

Conventional methods for identifying ARP spoofing attacks often depend on preset attack signatures, protocol-based monitoring, or fixed rules. Methods such as static ARP tables involve manually assigning MAC-IP bindings to prevent spoofing, but these lack scalability in dynamic networks. Intrusion Detection Systems (IDS) based on signatures compare network traffic to known attack patterns, but they frequently fall short when faced with novel or altered spoofing tactics. Unnecessary ARP monitoring identifies unusual ARP transmissions but is susceptible to false positives because it has trouble differentiating between legitimate and harmful actions. Although these traditional approaches provide some degree of defense, they are often constrained by their rigidity, lack of flexibility, and inability to identify novel or covert ARP spoofing assaults.

c. Related Works

Several studies have used machine learning for ARP spoofing detection within network intrusion research. Models like SVM, random forests, and neural networks analyze packet features to spot anomalies Using ML, attacks may be identified with greater precision than using rule-based approaches. It lessens false positives and is able to respond to emerging dangers. As a result, ML is a good technique for identifying ARP spoofing.

Launched a fresh dataset dubbed NSL-KDD, which is a subset of the complete KDD dataset and avoids the aforementioned problems [8].

Introduced an active method for identifying ARP spoofing. To check for discrepancies, we send TCP SYN and ARP request packets into the network. Compared to passive approaches, our technology is more dependable, intelligent, scalable, and faster at detecting attacks [3].

Examines ARP traffic sequence data on LANs by calculating the number and degree of packets. Additionally, an unsupervised auto encoder neural network is trained using feature vectors that are transformed from underlying suspicious actions, which are detected using a dynamic threshold [9].

[10] Demonstrate that a bank of simple classifiers, each with a front end designed to identify various spoofing assaults, may be combined at the score. Utilizing non-linear fusion to achieve a level of performance that surpasses more complex ensemble solutions that are dependent on intricate neural network designs.

[11] Recommend an online system for identifying attacks and classifying network traffic that combines stream machine learning, deep learning, and ensemble learning approaches. By applying multi-stage data analysis, the system can continuously monitor network traffic, detect malicious flows in real time, and accurately classify them based on the specific type of attack. This hybrid approach enhances detection accuracy and supports timely responses to evolving threats in dynamic network environments.

Shows that, in contrast to state-of-the-art techniques, their suggested method can accurately detect Low Earth Orbit (LEO) spoofing attacks delivered from a variety of altitudes. Given the ever-changing nature of satellite-based threats, their strategy demonstrates a high degree of adaptability. They have also made their gathered dataset publicly available as open source, which promotes more study and advancement in the area of satellite security[12].

Machine learning methods can automatically discover the essential differences between normal data and abnormal data with high accuracy. Furthermore, machine learning approaches are very generalizable, allowing them to identify previously unknown threats as well. Deep learning is a branch of machine learning whose performance is remarkable and has become a research hotspot [13].

[14] Concentrate on machine learning methods for detecting protocol tunneling assaults. The system can efficiently identify variations that indicate possible ARP-based attacks by learning common traffic patterns. This technique lessens the need for static rules while improving spoofing detection accuracy. It helps create network security solutions that are more intelligent and flexible. Proposed a machine learning-based approach for detecting Man in the middle attacks in large-scale software-defined networks (SDNs).

[15] Study is to demonstrate how algorithmic performance provides a range of solutions that satisfy different quality requirements, albeit at the price of speed and accuracy. Numerous algorithms were evaluated to determine the optimal trade-off between processing time and result accuracy.

In authors [16] Proposed the K Nearest Neighbor (KNN) model for the detection of Man in the middle attacks. Based on the experimental result, the rate of detection of man-in-the-middle attacks by the KNN model is 0.98.

In authors [17] determined a D-ARP-based detection scheme for man in the middle attacks via ARP spoofing, achieving zero false positives and false negatives.

[18] Proposed the utilizing kaggle website data to establish four machine learning approaches detecting two common threats that attack connected devices in a network. To verify their capacity to defend devices against such attacks, the analysis revealed over 99% accuracy in man in the middle identification and over 97% accuracy in DoS identification by all the methods.

In authors [19] focused on cyber-attack detection and network traffic anomalies through the development of an LSTM-RNN Intrusion Detection System. The suggested ID model's performance was evaluated using a performance evaluation approach that included measures of accuracy, detection rate, and false alarms.

[20] Proposed framework on Raspberry Pis and carried out experiments. We assess our framework in different locations within our experiment test bed and demonstrate that it can identify MC-Man in the middle attacks with an average correctness of 98%.

[21] Proposed a regression-based approach for secure routing PC-1, PC-2 and PC-3 in Internet of Things networks. Among LR, MLR, and GPR models, GPR showed the highest attack detection accuracy and lowest misclassification rates.

[22] proposed a reinforcement learning-based security model to defend against man in the middle attacks in fog computing. Their approach integrates SDN, MPTCP, and MTD to improve network security and resource efficiency.

[2] Developed a real-time anomaly detection system for Man in the middle attacks via ARP spoofing, using various machine learning models. Their CNN model achieved a 99% F1- score in training and 99.26% in real-time detection.

[23] Present a detection technique that employs explainable deep learning to identify ARP spoofing in Internet of Things networks. The system extracted features from network packets and achieved an F1 score of 0.999 and an accuracy of 99.98%.

[24]demonstrated the use of machine learning algorithms to detect end-point MIME attacks using Address Resolution Protocol (ARP) information. Linear-based classifiers were tested by the authors, and their detection rates were 99.72% when using machine learning and signal processing techniques.

[25] Designed to address imbalanced data through rescaled class weights and selecting the most informative features.

[26] presented an LSTM, and Decision Tree classifiers are used in classification. On performing different experiments, we observed that both methods can predict ARP spoofing at 99.9% and 100% accuracy, respectively.

[27] Demonstrate a machine learning method for identifying denial of-service assaults on Internet of Things platforms. The method's effectiveness in identifying denial-of-service attacks is demonstrated. The authors come to the conclusion that Internet of Things systems can be made more secure by using their method.

[28] Concentration specifically on detecting Distributed Denial of service attacks using machine learning an algorithm does not cover Man in the middle attacks or its detection methods and focuses on the classification of Distributed Denial of service traffic and mitigation strategies.

[29] Focus on Perceptron, Random Forest. This work does not address man-in the-middle attacks. The article is dedicated to Distributed Denial of service detection methods and their evaluation.

[30] engage in the detection of human attacks on MLP and CNNMSTM. We investigated various methods of scaling up the functions and got the best results of 99.74% in the CNN-MLP model.

[31] The main focus is on the evaluation of several machines.

[32] Describes an Intrusion Detection System (IDS) based on machine learning (ML) that can identify Man-in-the-Middle attacks in a smart grid (SG) advanced metering infrastructure (AMI).

[33] Develop a Man in the middle -Intrusion Detection System (MITM-IDS) for wireless sensor networks (WSN) to enhance security by detecting, isolating, and reconfiguring attacked nodes.

III. Methodology and Implementation

The proposed ARP spoofing detection model includes several key stages: dataset preparation, data preprocessing, feature selection, classification, and performance evaluation. The system's implementation utilized the Python programming language within the Jupyter Notebook framework. The necessary python classes for the selected machine learning algorithms (Naïve Bayes, Random forest and Naïve Bayes and Support Vector Machine) were imported and subsequent model modeling. Following experimentation, the evaluation outcomes were studied from which their performance rankings were carried out as outlined in the following subsections.

Machine Learning Algorithms

In this study, four prominent machine learning algorithms were selected for the task of detecting ARP spoofing attacks: Random Forest, Naive Bayes, Support Vector Machine (SVM), and XGBoost. These models represent a diverse set of classification techniques, each with distinct strengths and learning mechanisms.

A. Support Vector Machine

The Support Vector Machine (SVM), first put forth by Coretes and Vapnik in 1995, is useful for high-dimensional, nonlinear, and limited-sample pattern recognition. It finds a hyperplane of separation between various groups by utilizing the Vapnik-Chervonenkis (VC) dimension and structural risk reduction principles. This approach has made it a popular choice in fields such as image recognition, bioinformatics, and text classification. By effectively managing the trade-off between model complexity and generalization, Support Vector Machine continues to be a powerful tool in machine learning applications [35]. Support Vector Machine (SVM) is a supervised learning algorithm that constructs a hyperplane or set of hyperplanes in a high-dimensional space to separate different classes. SVMs are effective in cases where the number of features is greater than the number of samples and are particularly suited for binary classification problems

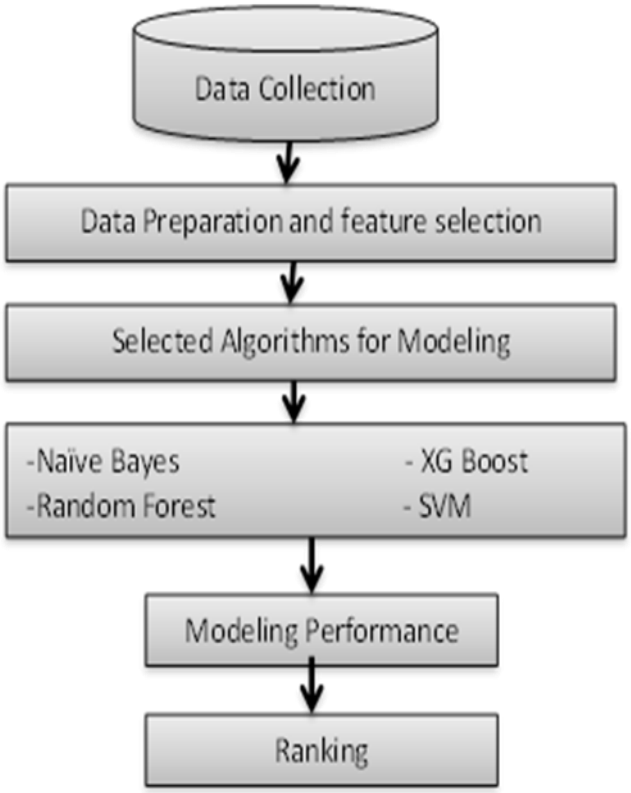

Figure 3: Overview of the Model

B. Random Forest:

Breiman first presented Random Forest in 2001. It is a potent supervised learning algorithm that predicts sample types based on classification outcomes and data properties. Using decision trees and a random subspace division, it uses the bagging technique to create a variety of training samples and a voting system to classify input samples [34]. The ensemble learning approach known as Random Forest generates many decision trees during training and then outputs the class that is the mode of the classes of each tree. It is known for its great precision, resilience to noise, and resistance to overfitting, especially in high-dimensional data. Another popular supervised machine learning method that is skilled at solving regression and classification issues is Random Forest (RF). It's an aggregation of DTs that functions by majority, using the combined decisions of several decision trees created from

different subsets of the dataset. 'voting' for categorization jobs and averaging for regression estimates. Notably, Random Forest is adaptable across a wide range of fields because it can effectively handle both continuous and categorical data variables.

C. XGBoost (Extreme Gradient Boosting)

XGBoost, which was created by Chen and Guestrin as an extension of gradient descent decision trees, uses regularization and parallel regression trees to avoid overfitting. By arranging parallelized searches for ideal split points and employing a first-order Taylor expansion for residual value estimation, it improves computational speed. The evaluation measures divide the predictions into four possible results depending on how accurately the model predicts the actual labels. Mathematical equations are used to determine these metrics, which enable a quantitative analysis of the model's effectiveness. Extreme Gradient Boosting (XGBoost) is a potent and adaptable gradient boosting framework that employs tree-based models. It concentrates on fixing the mistakes made by prior models and creates models one at a time. XGBoost is optimized for speed and performance, making it a strong candidate for complex classification tasks.

D. Naive Bayes algorithm

The Naive Bayes algorithm is one approach in machine learning-based models and serves as a versatile tool in numerous classification scenarios. It is utilized in many classification tasks such as document categorization, spam detection, and predictive modeling; its name “Bayes” comes from the foundational principles laid by Thomas Bayes. The algorithm's core premise of feature independence is encapsulated by the "Naive" setting, which indicates that modifications to one feature have no impact on other features in the model. With the assumption of feature independence, Bayes' theorem forms the basis of the probabilistic classifier known as Naive Bayes. Despite its simplicity, it performs competitively in many domains and is particularly efficient for large datasets with categorical features.

IV. Result and Evaluation Metrics

The experimental Model performance is evaluated using a number of metrics; we utilized a wide range of Machine Learning algorithms, such as Random Forest (RF), Support Vector Machine (SVM), XGBoost and Naive Bayes. The results yielded valuable insights into the performance of each algorithm under varying conditions. The model's performance is evaluated using a number of metrics, such as F1-Score, Accuracy, Precision, and Recall. These metrics are used to evaluate the Machine learning based detection of ARP Spoofing attack. By analyzing the accuracy and efficiency of these models, we aimed to identify the most effective approach for handling imbalanced datasets in real-world applications. [34]. They are four machine learning models were evaluated against the most commonly used metrics for classification problems, these are; model accuracy, precision, recall and f1-score. The summary of the results is presented in the table. We conducted experiments comparing the performance of four models—SVM (support vector machine), Naive Bayes, Random Forest, XGBoost use the dataset contains 69,248 records. The target variable for classification is the Label column, which includes categories such as normal and ARP Spoofing attack Model encoding the target labels prepare the features of the ML based Train and evaluate the four machine learning models. The study employs a range of Machine Learning based Algorithms, including Random Forest, SVM and XGBoost to train and evaluate the algorithms.

TABLE 1: Normal and ARP Spoofing Predicted

Label	Normal Predicted	ARP Spoofing Predicted
Normal	34,000 (TN)	441 (FP)
ARP Spoofing	300 (FN)	34,507 (TP)

True Negatives (TN = 34,000) the model correctly predicted 34,000 normal packets as a dataset.
False Positives (FP = 441) the model incorrectly predicted 441 normal packets as ARP spoofing.
False Negatives (FN =300) the model incorrectly predicted 300 ARP spoofing packets as normal.
True Positives (TP = 34,507) the model correctly predicted 34,507 ARP spoofing packets.

TABLE 2: Result and Evaluation Metrics

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.9839	0.9801	0.9844	0.9822
Naive Bayes	0.9284	0.9257	0.9362	0.9312
SVM	0.9676	0.9588	0.9716	0.9651
XGBoost	0.9827	0.9844	0.9864	0.9854

RF is demonstrated balanced performance across all evaluation metrics, with an accuracy of 0.9839, an F1 score of 0.9822, and precision and recall of 0.9801and 0.9844, respectively. This indicates that the Random Forest Machine learning model can achieve high precision while maintaining a high recall rate, effectively distinguishing between normal and ARP Spoofing traffic. However, as a linear model, its ability to handle complex nonlinear features is limited, which may restrict its performance in certain data patterns.

SVM excelled in recall, achieving an almost perfect score of 0.96, meaning it detected nearly all attack behaviors. With an accuracy of 0.9676, Support Vector machine outperformed, though its precision of 0.9588 could point to a higher false positive rate in some cases. Nonetheless, SVM, with its nonlinear modeling capabilities, effectively handled the complexity of network data. MLP2 outperformed all other models, with an accuracy of 0.9676 and F1 score, precision, and recall all reaching 0.9717. This exceptional performance indicates that excels at classifying large and complex datasets, accurately identifying both normal and ARP Spoofing traffic attack.

a. Performance Metrics of ML Based

Accuracy

Accuracy measures the overall correctness of the model. It tells you the percentage of total predictions that were correct.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Where:

International Journal of Scientific Engineering and Science
Volume 9, Issue 11, pp. 39-46, 2025. ISSN (Online): 2456-7361

- TP = True Positives (correctly predicted ARP spoofing) these samples are correctly identified as positive, representing genuine positives.
- TN = True Negatives (correctly predicted normal traffic) these are samples that are accurately classified as negative, signifying genuine negatives.
- FP = False Positives (normal predicted as ARP spoofing) this category is comprises samples that are incorrectly predicted as positive when, in reality, they are negative.
- FN = False Negatives (ARP spoofing predicted as normal) is the false negative consists of positive samples that are erroneously classified as negative.

Precision

Precision focuses on the quality of positive predictions. It tells you how many predicted attacks were actually real attacks. High precision means fewer false alarms. This is important when the cost of a false positive is high (e.g., stopping normal users due to false ARP alerts).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Recall

Recall measures the model's ability to detect actual attacks. It tells you how many real ARP spoofing attempts were correctly identified. High recall means the system catches most attacks very important in cyber security, where missing an attack can be dangerous.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

F1-Score

The F1-score is the harmonic mean of precision and recall. It gives a single metric that balances both false positives and false negatives. Useful when you need a balance between precision and recall, especially in imbalanced datasets like ARP spoofing detection.

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

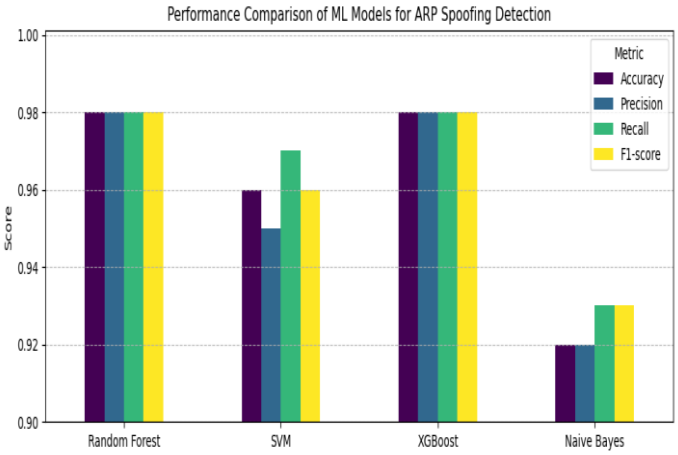

Figure 4: Performance Compression of ML based Model ARP Spoofing Detection

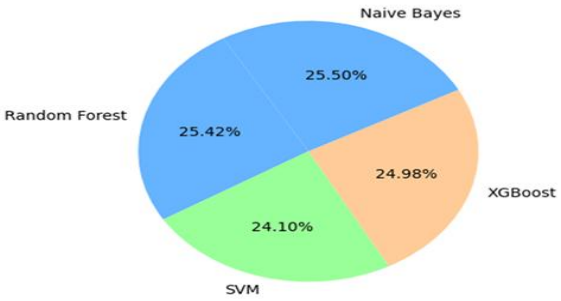

Figure 5: F1-Score Distribution

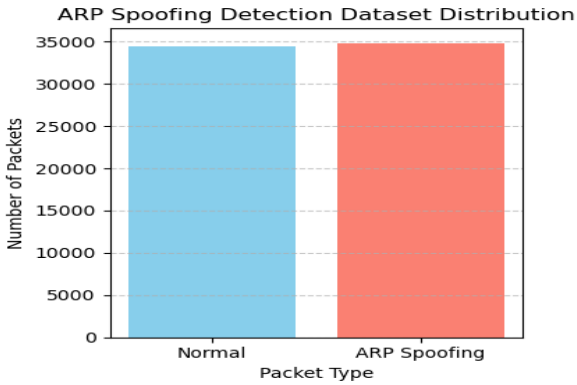

Figure 6: ARP Spoofing Detection Dataset Distribution

Feature Engineering in ARP Spoofing Detection

Feature engineering plays a crucial role in enhancing the accuracy of machine learning models used for ARP spoofing detection. It involves selecting and transforming raw network data into meaningful features that help distinguish between normal and malicious behavior. Key features include MAC-IP binding consistency, ARP reply frequency, packet timing, and request-response patterns. For instance, frequent or unsolicited ARP replies, inconsistent MAC-IP mappings, and irregular traffic timing can indicate spoofing activity. Well-crafted features not only improve detection performance but also reduce false positives, making the system more reliable and effective for real-time network security. Effective feature engineering can significantly reduce false positives and improve detection performance. Publicly available datasets such as UNSW-NB15, CICIDS2017, and custom ARP traffic captures are frequently used for training and evaluation[8]. However, the scarcity of ARP-specific labeled datasets remains a challenge, often requiring synthetic data generation or real-world traffic monitoring for model development

V. Conclusion and Future Work

This study investigates the use of machine learning to detect ARP spoofing attacks, which often evade traditional defenses due to the ARP protocol's lack of authentication. Techniques such as Support Vector Machines (SVM), Random Forest, Naïve Bayes and XGBoost were applied to identify anomalies in ARP traffic specifically MAC-IP mismatches, abnormal packet frequency, and timing irregularities. These methods improved detection accuracy and reduced false positives. Key challenges include limited labeled datasets, the need for real-

time detection, and model interpretability. Future work should focus on building lightweight, real-time systems for diverse environments like IOT and cloud networks, enhancing dataset diversity, and improving ML transparency. Integrating detection with Software-Defined Networking (SDN) could enable faster, centralized responses. This research supports the development of adaptive, intelligent, and scalable network security solutions to counter both current and evolving ARP spoofing threats.

REFERENCE

[1] P. Ramesh, D. L. Bhaskari, and C. S. -, "A Comprehensive Analysis of Spoofing," *Int. J. Adv. Comput. Sci. Appl.*, vol. 1, no. 6, pp. 157–162, 2010, doi: 10.14569/ijacsa.2010.010623.

[2] S. Majumder, M. K. Deb Barma, and A. Saha, *ARP spoofing detection using machine learning classifiers: an experimental study*, vol. 67, no. 1. Springer London, 2024. doi: 10.1007/s10115-024-02219-y.

[3] V. Ramachandran and S. Nandi, "Detecting ARP spoofing: An active technique," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3803 LNCS, pp. 239–250, 2005, doi: 10.1007/11593980_18.

[4] P. Shanmugaraj, K. Karthick, G. M. Muralikumar, R. Meena, and S. P. P. Sarathi, "A Survey on DoS/DDoS and ARP Spoofing Attack Solutions in Software-Defined Networks," no. Icsice 24, pp. 1396–1412, 2025, doi: 10.2991/978-94-6463-718-2_116.

[5] Amrit Kaur, "Detection of Phishing Websites Using SVM Technique," *Imp. J. Interdiscip. Res.*, vol. 2, no. 8, pp. 1273–1276, 2016.

[6] Y. Jeong, H. Kim, and H. J. Jo, "ASD: ARP Spoofing Detector Using OpenWrt," *Secur. Commun. Networks*, vol. 2022, 2022, doi: 10.1155/2022/2196998.

[7] V. Hnamte and J. Hussain, "Enhancing security in Software-Defined Networks: An approach to efficient ARP spoofing attacks detection and mitigation," *Telemat. Informatics Reports*, vol. 14, no. February, 2024, doi: 10.1016/j.teler.2024.100129.

[8] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *IEEE Symp. Comput. Intell. Secur. Def. Appl. CISDA 2009*, no. July, 2009, doi: 10.1109/CISDA.2009.5356528.

[9] Y. Sun, H. Ochiai, and H. Esaki, "Suspicious ARP Activity Detection and Clustering Based on Autoencoder Neural Networks," *Proc. - IEEE Consum. Commun. Netw. Conf. CCNC*, pp. 743–744, 2022, doi: 10.1109/CCNC49033.2022.9700697.

[10] H. Tak, J. Patino, A. Nautsch, N. Evans, and M. Todisco, "Spoofing attack detection using the non-linear fusion of sub-band classifiers," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2020–Octob, pp. 1106–1110, 2020, doi: 10.21437/Interspeech.2020-1844.

[11] D. Abreu and A. Abelem, "OMINACS: Online ML-Based IoT Network Attack Detection and Classification System," *2022 IEEE Latin-American Conf. Commun. LATINCOM 2022*, 2022, doi: 10.1109/LATINCOM56090.2022.10000544.

[12] J. Wigchert, S. Sciancalepore, and G. Oligeri, "Detection of Aerial Spoofing Attacks to LEO Satellite Systems via Deep Learning," *Comput. Networks*, vol. 269, pp. 1–11, 2025, doi: 10.1016/j.comnet.2025.111408.

[13] Z. Liu, J. Hu, Y. Liu, K. Roy, X. Yuan, and J. Xu, "Anomaly-Based Intrusion on IoT Networks Using AIGAN-a Generative Adversarial Network," *IEEE Access*, vol. 11, no. August, pp. 91116–91132, 2023, doi: 10.1109/ACCESS.2023.3307463.

[14] F. Sobrero, B. Clavarezza, D. Ucci, and F. Bisio, "Towards a Near-Real-Time Protocol Tunneling Detector Based on Machine Learning Techniques †," *J. Cybersecurity Priv.*, vol. 3, no. 4, pp. 794–807, 2023, doi: 10.3390/jcp3040035.

[15] A. Alsaaidah, O. Almomani, A. A. Abu-Shareha, M. M. Abualhaj, and A. Achuthan, "ARP Spoofing Attack Detection Model in IoT Networks Using Machine Learning: Complexity vs. Accuracy," *J. Appl. Data Sci.*, vol. 5, no. 4, pp. 1850–1860, 2024, doi: 10.47738/jads.v5i4.374.

[16] B. A. Mantoo and P. Kaur, "A Machine Learning Model for Detection of Man in The Middle Attack Over Unsecured Devices," *AIP Conf. Proc.*, vol. 2555, no. February, pp. 0–10, 2022, doi: 10.1063/5.0109151.

[17] S. M. Morsy and D. Nashat, "D-ARP: An Efficient Scheme to Detect and Prevent ARP Spoofing," *IEEE Access*, vol. 10, no. May, pp. 49142–49153, 2022, doi: 10.1109/ACCESS.2022.3172329.

[18] S. A. M. Al-Juboori, F. Hazzaa, Z. S. Jabbar, S. Salih, and H. M. Gheni, "Man-in-the-middle and denial of service attacks detection using machine learning algorithms," *Bull. Electr. Eng. Informatics*, vol. 12, no. 1, pp. 418–426, 2023, doi: 10.11591/eei.v12i1.4555.

[19] M. Ibrahim and R. Elhafiz, "Modeling an intrusion detection using recurrent neural networks," *J. Eng. Res.*, vol. 11, no. 1, p. 100013, 2023, doi: 10.1016/j.jer.2023.100013.

[20] M. Thankappan, H. Rifà-Pous, and C. Garrigues, "A distributed and cooperative signature-based intrusion detection system framework for multi-channel man-in-the-middle attacks against protected Wi-Fi networks," *Int. J. Inf. Secur.*, vol. 23, no. 6, pp. 3527–3546, 2024, doi: 10.1007/s10207-024-00899-9.

[21] N. Sivasankari and S. Kamalakkannan, "Detection and prevention of man-in-the-middle attack in iot network using regression modeling," *Adv. Eng. Softw.*, vol. 169, no. February, p. 103126, 2022, doi: 10.1016/j.advengsoft.2022.103126.

[22] H. Elmansy, K. Metwally, and K. Badran, "Reinforcement learning-based security schema mitigating man-in-the-middle attacks in fog computing," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 5, pp. 5908–5921, 2023, doi: 10.11591/ijece.v13i5.pp5908-5921.

[23] M. M. Alani, A. I. Awad, and E. Barka, "ARP-PROBE: An ARP spoofing detector for Internet of Things networks using explainable deep learning," *Internet of Things (Netherlands)*, vol. 23, no. February, 2023, doi: 10.1016/j.iot.2023.100861.

[24] J. J. Kponyo, J. O. Agyemang, and G. S. Klogo, "Detecting End-Point (EP) Man-In-The-Middle (MITM) Attack based on ARP Analysis: A Machine Learning Approach," *Int. J. Commun. Networks Inf. Secur.*, vol. 12, no. 3, pp. 384–388, 2020, doi: 10.17762/ijcnis.v12i3.4735.

[25] 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, pp. 1–22, 2022, doi: 10.1186/s42400-021-00103-8.

[26] M. Usmani, M. Anwar, K. Farooq, G. Ahmed, and S. Siddiqui, "Predicting ARP spoofing with Machine Learning," *2022 Int. Conf. Emerg. Trends Smart Technol. ICETST 2022*, no. September 2022, 2022, doi: 10.1109/ICETST55735.2022.9922925.

[27] B. R. KIKISSAGBE, M. Adda, P. Célicourt, I. T. HAMAN, and A. Najjar, "Machine Learning for DoS Attack Detection in IoT Systems," *Procedia Comput. Sci.*, vol. 241, no. 2019, pp. 195–202, 2024, doi: 10.1016/j.procs.2024.08.027.

[28] D. K. Suvra, "An Efficient Real Time DDoS Detection Model Using Machine Learning Algorithms," 2025, [Online]. Available: http://arxiv.org/abs/2501.14311

[29] W. Tay, S. Chong, and L. Chong, "DDoS Attack Detection with Machine Learning," vol. 3, no. 3, 2024.

[30] H. H. Satyanegara and K. Ramli, "Implementation of CNN-MLP and CNN-LSTM for MitM Attack Detection System," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 3, pp. 387–396, 2022, doi: 10.29207/resti.v6i3.4035.

[31] M. A. Rajput, Muhammad Umar, A. Ahmed, Ali Raza Bhangwar, Khadija Suhail Memon, and Misbah, "Evaluation of Machine Learning based Network Attack Detection," *Sukkur IBA J. Emerg. Technol.*, vol. 5, no. 2, pp. 57–66, 2023, doi: 10.30537/sjet.v5i2.1186.

[32] D. Jim Solomon Raja, R. Sriranjani, P. Arulmozhi, and N. Hemavathi, "Unified Random Forest and Hybrid Bat Optimization Based Man-in-the-Middle Attack Detection in Advanced Metering Infrastructure," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–12, 2024, doi: 10.1109/TIM.2024.3420375.

[33] H. Mohapatra, "Handling of Man-In-The-Middle Attack in WSN Through Intrusion Detection System," *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 5, pp. 1503–1510, 2020, doi: 10.30534/ijeter/2020/05852020.

[34] Z. M. Khan, "Network Intrusion Detection Utilizing Autoencoder Neural Networks," *Commun. Appl. Nonlinear Anal.*, vol. 31, no. 3S, pp. 336–354, 2024, doi: 10.52783/cana.v31.777.