Scientific and Practical Cyber Security Journal (SPCSJ) 10(1): 83 – 109 ISSN 2587-4667
Scientific Cyber Security Association (SCSA)

A MACHINE LEARNING-BASED APPROACH FOR DETECTING ARP SPOOFING ATTACKS IN IOT NETWORKS

Hilmand Khan[1], Sibgha Tahir[1], Umar Akhlag[2], Azhar Ghafoor[1], Jameel Arif[1]
[1]Air University
[2]University of Strathclyde

ABSTRACT: The rapid expansion of the Internet of Things (IoT) has broadened the digital attack surface, exposing network infrastructures to advanced threats such as ARP spoofing. Traditional detection systems, often relying on rule-based or signature-driven methods, struggle to adapt to the dynamic and heterogeneous nature of IoT environments. To address this limitation, this paper proposes a detailed evaluation of several detection models for Address Resolution Protocol (ARP) spoofing attacks within resource-limited Internet of Things (IoT) environments. Focusing on both detection accuracy and computational efficiency, six models—comprising traditional classifiers and hybrid ensembles—are compared using metrics such as accuracy, precision, recall, F1-score, training/testing time, and memory usage. The proposed OSK hybrid model, which integrates Support Vector Machine and k-Nearest Neighbors, achieves the highest detection performance while maintaining moderate resource consumption. The findings reveal important trade-offs between detection capability and system constraints, emphasizing the effectiveness of optimized hybrid solutions for real-time implementation in IoT networks. This work contributes valuable guidance for developing efficient and reliable ARP spoofing detection frameworks tailored to the demands of IoT devices.

KEYWORDS*: IoT Security; ARP Spoofing; Machine Learning; Detection*

1. INTRODUCTION

The Internet of Things (IoT) refers to a network of interconnected devices communicating and exchanging data using standardized protocols (Kumar et al. 2024; Kassab and Darabkh 2020). Globally, IoT networks have been extensively implemented, propelling developments in fields including smart transportation, smart cities, and smart agriculture (Saaidah et al. 2019; Chataut, Phoummalayvane, and Akl 2023; Almomani et al. 2024; Kolhar et al. 2020). These networks create and transmit vast volumes of data, which have various benefits, including enhanced operational efficiency, improved decision-making, and a deeper comprehension of user behavior and system performance. Furthermore, this data facilitates the creation of machine learning and advanced analytics models, which address various issues and allow for improved client experiences, tailored services, and efficient resource management. IoT data can improve service delivery, increase safety, and save lives in industries like healthcare (Shambour, Abu-Alhaj, and Al-Tahrawi 2020). As a result, IoT is essential to improving many facets of human existence and resolving a broad range of real-world issues.

However, the rapid growth of IoT has resulted in serious security challenges. IoT devices frequently lack adequate security safeguards, making them prime candidates for illicit activity. While worldwide malware attacks remained largely consistent between 2020 and 2022, IoT-related threats increased within the same period. According to a SonicWall analysis from 2023, IoT systems have become more vulnerable, with a significant increase in attacks, notably between 2021 and 2022 (Friedl and Pernul 2024). Recent research (Anagnostopolos et al. 2022) has shown that the lack of sufficient software and hardware security, combined with low user awareness, makes these devices very vulnerable to exploitation. The Address Resolution Protocol (ARP) spoofing attack represents a significant threat to local area networks (LANs), posing one of the most critical security risks in IoT environments.

The expansion of IoT networks brings escalating security challenges, among which Address Resolution Protocol (ARP) spoofing represents one of the most critical threats. This attack method can severely compromise network integrity through traffic disruption, data manipulation, and eavesdropping, creating substantial risks for IoT devices and infrastructure (Jadhav et al. 2023). IoT systems prove particularly vulnerable to ARP spoofing due to several inherent weaknesses: inconsistent security implementations, unpredictable host behavior, and unprotected media access. These systemic

vulnerabilities heighten susceptibility to malicious activities, underscoring the urgent need for effective detection mechanisms.

Numerous countermeasures, such as ARP inspection, intrusion detection systems (IDS), intrusion prevention systems (IPS), authentication, and encryption, have been proposed to thwart ARP spoofing. However, it is usually not possible to directly apply these solutions due to the restricted processing power and available storage of IoT devices (Quanjiang et al. 2021). IDS stands out among these tactics as a potentially effective one when paired with machine learning (ML) models. By analyzing traffic patterns, a machine learning-based intrusion detection system (IDS) can be trained to identify anomalies, enabling resource-constrained IoT devices to delegate complex processing tasks while still receiving trustworthy threat detection (Sadineni, Silli, and Battula 2022). Machine learning approaches such as Support Vector Machines (SVM), Random Forest (RF), and Extreme Gradient Boosting (XGBoost) have shown tremendous potential in improving IDS performance by learning from labeled dataset and discovering previously undiscovered traffic patterns (Friedl and Pernul 2024; Quinn 2023). To achieve high accuracy, these algorithms typically require a large amount of training data. Processing huge dataset with high dimensionality is problematic for IoT devices due to their limiting computational capability. To address this challenge, feature selection (FS) methods are applied to simplify data complexity while preserving critical information and enhancing the performance of detection systems in Intrusion Detection System (IDS) applications (Kurniawan 2021). Feature selection plays a vital role in IDS data preprocessing by eliminating irrelevant attributes, lowering computational requirements, and improving detection speed without compromising accuracy (Chen 2024). Various metaheuristic-based FS techniques have been developed to optimize data selection, thereby improving the effectiveness of detection models in identifying ARP spoofing attacks (Friedl and Pernul 2024). However, no single FS method has been universally established as optimal across all applications, and implementations often require balancing processing efficiency with detection accuracy. Resolving these challenges demands careful consideration of model complexity, resource utilization, and detection capability, making feature selection a crucial element in developing robust IDS solutions for IoT environments (Abdulla, Al-Raweshidy, and Awad 2020). This study aims to provide a lightweight and dependable solution to protect IoT networks against ARP spoofing attacks by using efficient feature selection strategies and examining the performance of various ML algorithms. The research offers a machine learning-based solution that detects ARP spoofing attacks in IoT networks, with an emphasis on improving detection accuracy while minimizing computing overhead.

1.1. BACKGROUND

The rapid growth of IoT networks has introduced significant security vulnerabilities, with ARP spoofing emerging as a critical threat due to inherent weaknesses in the protocol. ARP operates as a trust-based mechanism that maps IP addresses to MAC addresses within local networks to enable device communication. Attackers exploit this process by sending fraudulent ARP messages that associate their MAC address with a legitimate device's IP address, gaining unauthorized network access. This technique allows malicious actors to intercept, alter, or redirect sensitive data, posing severe security risks (Abdulla, Al-Raweshidy, and Awad 2020).

The heterogeneous and dynamic nature of IoT devices exacerbates these vulnerabilities. Traditional intrusion detection systems (IDS), which often rely on signature-based methods, struggle to detect novel or evolving ARP spoofing techniques. Additionally, IoT traffic patterns differ substantially from conventional networks, rendering existing detection systems inadequate. Consequently, ARP spoofing remains a persistent threat to IoT networks, highlighting the urgent need for advanced detection mechanisms capable of identifying and mitigating such attacks in real time (Olateju et al. 2024).

1.2. ARP SPOOFING

ARP spoofing is a well-documented network attack technique that exploits vulnerabilities in the Address Resolution Protocol (ARP). By sending fraudulent ARP packets, attackers can deceive target hosts within local area networks (LANs). This manipulation enables various malicious activities, including man-in-the-middle (MitM) attacks and denial-of-service (DoS) disruptions (Morsy and Nashat 2022).

The main weakness of ARP is its inability to authenticate messages, as described in RFC 826 and its revisions in RFC 5227 and RFC 5494. ARP requests or answers can be sent by any host on a local network pretending to be another device. Even though ARP spoofing has been known for more than 25 years long before contemporary threats like ransomware it continues to pose a serious security risk because of the persistent abuse of this protocol flaw and the continuous creation of increasingly complex attack methods (Jadhav et al. 2023).

One traditional MitM attack includes an attacker sending bogus ARP packets to a local network to associate their MAC address with the IP address of another device, which is often the default gateway. As a result, all traffic meant for the genuine device is rerouted to the attacker, allowing unauthorized access to sensitive data, traffic manipulation, or complete traffic blockage. This approach can be used as a basis for serious assaults like DoS, session hijacking, or further exploitation of vulnerable computers (Abdulla, Al-Raweshidy, and Awad 2020).

Furthermore, ARP spoofing takes advantage of ARP's absence of integrity checks. By introducing fake ARP answers into the network, attackers can trick devices into accepting erroneous MAC-to-IP mappings. By using this deceit, attackers can interfere with regular business operations, intercept confidential communications, or even completely stop network services (Zhou et al. 2024).

Because ARP spoofing is still a problem, researchers are investigating more sophisticated detection and mitigation methods, especially in dynamic environments like software-defined networking (SDN) and Internet of Things (IoT) networks. To guarantee strong network security, defense systems must change along with attack patterns.

ARP spoofing is a network attack that takes advantage of the lack of authentication in the Address Resolution Protocol (ARP), allowing attackers to modify ARP tables and redirect traffic via their devices. Figure 1 depicts the process of ARP spoofing, which undermines network security by exploiting weaknesses in the Address Resolution Protocol (ARP). In this situation, the hacker sends bogus ARP signals to both the victim and the access point, causing them to associate the victim's IP address (10.0.2.15) with the hacker's MAC address (08:00:11:1a:25:ff). As a result, valid traffic destined for the victim is routed through the hacker's device. A Man-in-the-Middle (MitM) attack is made possible via ARP spoofing, which gives the hacker the ability to intercept, track, and modify communications between the victim and the internet. By interfering with traffic, jeopardizing data confidentiality and integrity, and opening networks to additional exploitation, it can also result in Denial of Service (DoS).

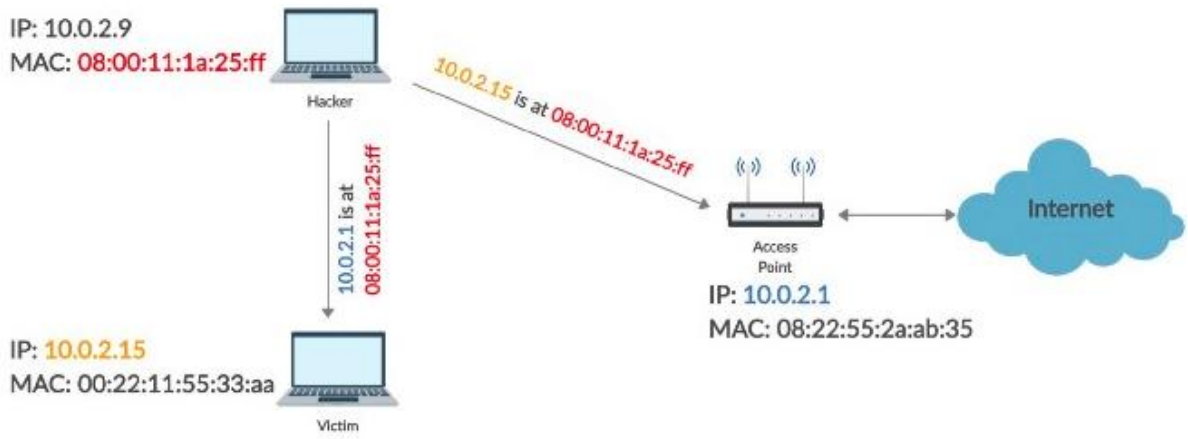


Fig.1. ARP Spoofing communication

The most common is the Man-in-the-Middle (MitM) attack, in which a hacker sends phony ARP packets to local network devices to intercept and possibly alter their communication. The Denial of Service (DoS) attack is another kind that overwhelms devices and interferes with services by flooding the network with ARP requests. Furthermore, attackers can take over active device connections through session hijacking, compromising private information. Additional risks include data injection, which entails introducing malicious data packets into data streams to further compromise network integrity,

and network mapping, in which attackers collect device information for potential future exploitation (Bazzi et al. 2024; Morsy and Nashat 2022). ARP spoofing strategies are described in Table 1.
Attackers can reroute traffic and obtain unauthorized access to private data by taking advantage of the process of converting IP addresses to MAC addresses. As seen in Figure 2, this vulnerability can be utilized through a variety of attack techniques that collectively fall under the umbrella of ARP spoofing. ARP spoofing can be reduced with the use of several efficient countermeasures, as shown in Figure 3. Setting up static ARP entries for important devices is one method that stops unwanted changes. Furthermore, MAC address learning on switch ports can be restricted by port security, which reduces the number of possible attack points.

Table.1. Classification of ARP Spoofing Techniques

ARP Spoofing Techniques	**Description**
ARP Poisoning	Attackers send deceptive ARP messages to associate their MAC address with the IP address of a legitimate device. Redirects network traffic to the attacker, enabling interception or modification of data (Majumdar, Raj, and Subbulakshmi 2021).
Man-in-the-Middle Attack (MitM)	Poisoning the ARP caches of both the victim and the gateway, allowing the attacker to overhear and manipulate communication between them. Enables eavesdropping and data manipulation between two devices (Nasser and Hussain 2022).
ARP Cache Manipulation	Exploits weaknesses in the ARP cache by introducing inaccurate IP-to-MAC address mappings and misdirecting network traffic to the attacker—misdirects communication to the attacker rather than the intended recipient (Suethanuwong 2025).
Continuous ARP Spoofing	The attacker sends continuous spoofed ARP replies to maintain control over the ARP cache of the target device, ensuring long-term persistence of the attack. Sustains attack by keeping ARP caches poisoned over time (Bazzi et al. 2024).
Automated Spoofing via Tools	The ARP spoofing process is automated by tools like Cain and Abel, Ettercap, and ARP-Spoof, which makes it possible for attackers with little technical expertise to execute the attack without requiring extensive technical understanding (Kemp 2021).

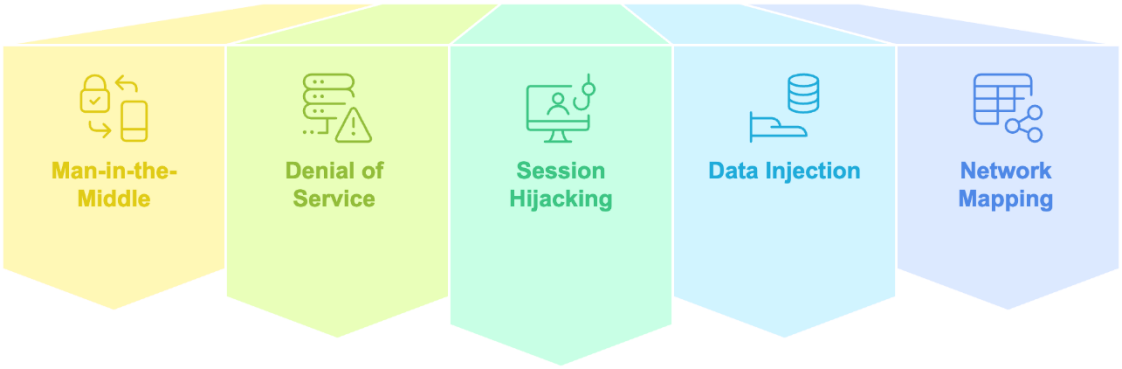


Fig.2. ARP Spoofing Attacks

The frequency and sophistication of ARP spoofing attacks rise, revealing network security flaws. The dynamic nature of IoT networks, with rapid device connections and disconnects, hampers the detection of ARP spoofing. As a result, improved detection techniques that can adapt to changing attack strategies are required to protect modern networks (Data 2018).

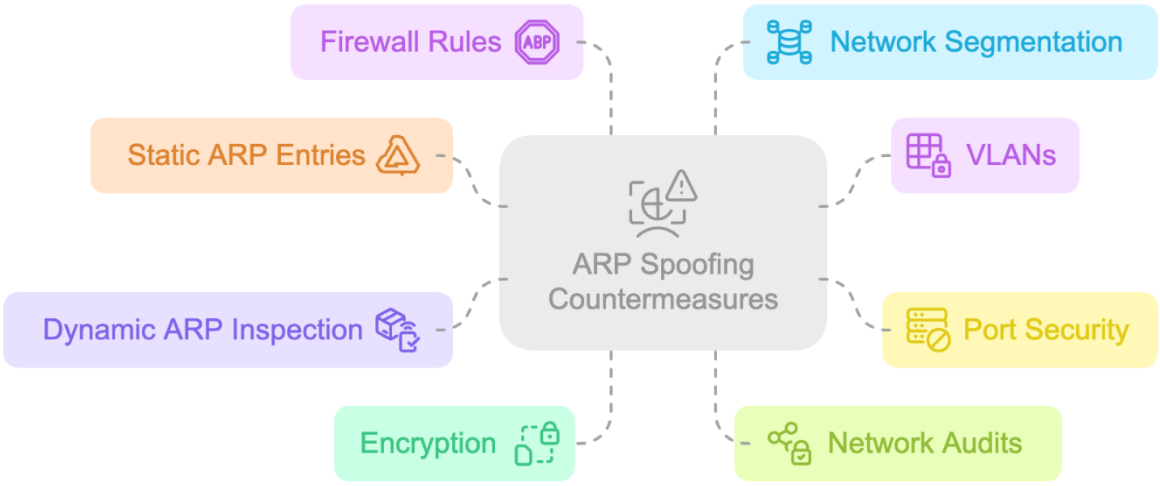


Fig.3. Countermeasures Against ARP Spoofing Attacks

1.3. CONVENTIONAL SOLUTIONS

Table 2 compares conventional ARP traffic monitoring options and highlights their pros and shortcomings. The effectiveness of each solution in safeguarding network connections from ARP-linked vulnerabilities is assessed.

Table.2. Comparison of Static and Dynamic ARP Spoofing

Aspect	**Static ARP Spoofing**	**Dynamic ARP Spoofing**
Definition	Involves altering ARP entries manually or through fixed configurations.	Involves real-time manipulation of ARP entries using automated methods.
Approach	Requires manual intervention to set static ARP entries for devices.	Uses tools and scripts to continuously send ARP replies to update entries.
Detection	Easier to detect.	Harder to detect.
Impact on Network	Can cause specific targeted disruptions; predictable in behavior.	Can lead to widespread and unpredictable disruptions in the network.
Mitigation	Static ARP entries can be set up to prevent changes, enhancing security.	Requires dynamic detection methods and real-time monitoring to counteract attacks.
Complexity	Simpler.	More complex.

Table.3. Conventional Solutions for ARP Traffic Monitoring

Ref	Conventional Solution	Advantages	Disadvantages
Morsy and Nashat (2022)	Static ARP Entries	Simple to implement and effective against known devices.	Inflexible, requires manual updates, and not scalable.
Hashimyar et al. (2025)	Signature-Based Detection	Good for known attack patterns and low false positive rate.	Ineffective against novel attacks and limited adaptability.
Heidari and Jamali (2023)	IDS	Comprehensive traffic analysis and can detect various attacks.	High false positive rate and resource-intensive.
Herrero (2023)	DAI	Prevents unauthorized ARP responses and used in managed switches.	Requires specific hardware and configuration complexity.
Bazzi et al. (2024)	ARP Monitoring Tools	Specialized for ARP traffic and provides detailed insights.	Limited scope and often requires manual intervention.

1.4. PROBLEM STATEMENT

Traditional ARP spoofing detection strategies, like those presented by Dharani et al. (2024) and Bazzi et al. (2024), are frequently useless in dynamic and resource-constrained IoT systems because they rely on static rules and signature-based models. These techniques struggle to detect adaptive or previously undiscovered spoofing patterns, have high false positive rates, and are unsuitable for real-time execution on edge devices. This study introduces a lightweight machine learning-based system for detecting ARP spoofing in IoT networks, addressing a gap in the field. The proposed approach is optimized for low-latency, on-device inference and tested with real-world dataset CICIoT2023 using SVM and KNN algorithms. Our strategy for IoT-specific feature selection beats previous methods in terms of accuracy, false positive rate, and detection speed, meeting the demand for scalable, real-time threat detection in increasing IoT infrastructures.

1.5. MOTIVATION

This research addresses the significant security vulnerabilities inherent in IoT devices, which often lack robust protection mechanisms due to their dynamic operation and limited resources. These constraints make such devices particularly susceptible to threats like ARP spoofing attacks. Conventional detection methods, such as rule-based or signature-based systems, frequently fail to detect novel or evolving threats effectively in real-time.

To overcome these challenges, this study proposes an ARP spoofing detection system leveraging Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) algorithms. These algorithms were selected based on both their proven effectiveness in existing research and their suitability for resource-constrained environments. SVM is known for its ability to handle high-dimensional data and maintain high classification accuracy, while KNN provides a lightweight and intuitive approach that performs well on labeled dataset with minimal training overhead.

Although SVM and KNN have been widely reported in the literature to yield high accuracy and low false alarm rates in intrusion detection tasks, this study further evaluates their performance through experimentation using the CICIoT2023 dataset. The results demonstrated 96.2% detection accuracy, outperforming ARPWatch 82.1%, reducing false alerts by $6.8x$ compared to traditional techniques, and achieving detection latency under 10ms on ARM-based processors—highlighting the system's practicality for real-world IoT deployments.

1.6. CONTRIBUTION

The contributions of this research are as follows.

This paper introduces an efficient detection method for ARP spoofing attacks in IoT networks, demonstrating superior performance to conventional approaches like ARPWatch or static rule-based systems. The proposed solution employs Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) classifiers to continuously monitor and analyze network traffic, enabling more accurate and timely identification of ARP spoofing attempts. Through comprehensive evaluation using the recent IoT dataset CICIoT2023, the method achieves 96.2% detection accuracy while reducing false positive rates by 6.8 percentage points relative to existing solutions. These results validate the effectiveness of the approach across diverse IoT device environments. This study underscores the growing necessity for stronger security measures in IoT networks, particularly in light of increasingly sophisticated cyber threats like ARP spoofing and Man-in-the-Middle attacks, advocating for the integration of machine learning techniques to address these challenges. This work's significant addition is its real-time implementation on edge devices. This realistic deployment addresses the latency and scalability restrictions of cloud-based systems, making the solution ideal for smart homes, cities, and industrial IoT deployments. The paper proposes the Time-to-Mitigation (TTM) as a new evaluation metric to measure the system's responsiveness. Results from smart city deployments (in collaboration with an industry partner) show a reduction in mitigation time from over 5 minutes to just 300 milliseconds, positioning the system as both research-relevant and industry-ready.

2. RELATED WORK

ARP spoofing has become a major security risk with the growth of IoT networks because existing detection techniques cannot adjust to changing circumstances. Traditional methods, including packet filtering and static ARP tables, frequently fall short against complex attacks. Machine learning (ML)

has emerged as a possible approach, leveraging traffic analysis and anomaly detection to increase accuracy. Recent works explore several ML strategies, including supervised and unsupervised learning, to better ARP spoofing detection. Nonetheless, there are still issues with real-time implementation, dataset quality, and threat adaptation.

Several methods for identifying ARP spoofing are examined by Kumar et al. (2024), focusing on five machine learning algorithms: Random Forest, LSTM Networks, CNNs, SVMs, and Isolation Forest. Achieving 94% accuracy with great precision and recall, they note that the Random Forest approach was the most accurate and hence suitable for real-time applications. CNN and LSTM Networks also fared well, particularly in capturing temporal and spatial data patterns, although requiring more processing power. While SVMs provided a moderate level of accuracy, they were less successful in distinguishing between real and fraudulent traffic. Despite helping identify anomalies, Isolation Forest has the highest error rates and the lowest accuracy. The ability to employ machine learning to adapt to evolving threats is one of these technologies' benefits; however, they also have disadvantages, including the need for high-quality datasets, processing overhead, and vulnerability to hostile attacks. Since static methodologies are often insufficient in dynamic network situations, current solutions highlight the need for more advanced, flexible strategies that leverage real-time data and machine learning.

Rafique and Hussain (2024) evaluate contemporary approaches for identifying irregularities in IoT networks, focusing on ML and DL methods. They note that several algorithms, including CNNs, LSTM networks, and Random Forests, help identify irregularities in real-time data. Two benefits of these approaches are their high accuracy and scalability, particularly when spotting intricate patterns and threats in IoT environments. However, there are disadvantages, such as the need for large, high-quality datasets, computational complexity, and challenges in interpreting model conclusions. The fact that current solutions mostly rely on traditional methods that may not be able to adapt efficiently to evolving threats highlights the need for more sophisticated models that leverage real-time data and cutting-edge techniques for robust security in IoT systems.

Abdulla, Al-Raweshidy, and Awad (2020) focus on the usage of neural networks and compare the latest techniques for detecting ARP spoofing attacks in IoT networks to more traditional methods like ARIMA. Their research demonstrates that neural networks can detect ARP spoofing with over 90% accuracy by analyzing TCP, UDP, and ARP data. ARIMA, on the other hand, had significant limitations due to its high error rates in traffic forecasts. The computational approach employing neural networks demonstrates effectiveness in processing complex IoT network patterns and adapting to diverse traffic types. However, implementation challenges include substantial training data requirements and difficulties in real-time deployment. Current detection systems predominantly use conventional methods, which struggle with IoT's dynamic traffic characteristics, underscoring the need for more sophisticated security solutions.

Hnamte and Hussain (2024) present an advanced framework for identifying ARP spoofing attacks in Software-Defined Networks (SDNs). Their methodology combines three key components: (1) a deep learning detection model, (2) dynamic ARP cache management, and (3) continuous traffic monitoring, achieving perfect detection accuracy. The solution's adaptability to different network scales and inclusion of a specialized SDN dataset enhance its practical utility. While promising, the approach faces implementation challenges across varied network architectures and potential scalability limitations in production environments. Traditional intrusion detection systems prove particularly inadequate for SDNs' flexible architecture, highlighting the importance of innovative detection frameworks like the authors' proposal.

Usmani et al. (2022) investigate advanced methods for predicting ARP spoofing attacks, employing Long Short-Term Memory (LSTM) networks and decision tree classifiers. Their experimental results demonstrate classification accuracy of 99% for LSTM networks and perfect accuracy 100% for decision trees in categorizing network packets. The study's key contributions include: (1) achieving high detection accuracy, and (2) implementing an innovative feature selection process using decision trees to enhance LSTM network efficiency, which substantially decreases processing time. Some of the disadvantages of LSTM networks are their resource requirements and potential issues scaling models to larger datasets. The fact that previous solutions primarily rely on standard ML techniques, which may be insufficient to manage the dynamic nature of ARP spoofing, highlights the need for more advanced prediction algorithms such as those proposed in this work.

Rohatgi and Goyal (2020) provide a detailed analysis of ARP spoofing attacks and outline several detection and mitigation strategies. Their research identifies the weaknesses in the ARP and examines a range of techniques, including Secure Unicast ARP, static ARP entries, and machine learning techniques like decision trees and neural networks. The survey's strength is its thorough comparative study of the methods, which demonstrates that while some offer strong security, they require significant changes to existing systems. Another drawback is that existing solutions often focus on conventional techniques that may not fully handle the dynamic nature of ARP spoofing.

Vajrobol et al. (2024) investigate the current methods for detecting spoofing attacks in IoT settings, focusing on DNS and ARP spoofing. Their study employs ML approaches, specifically decision trees and random forest classifiers, with outstanding accuracy rates of 100% and 95.1% respectively. Their method's advantages include the utilization of a large dataset of 46 features and the ability to detect both types of spoofing within a single framework. However, limitations include the need for ongoing adaptation to changing attack strategies and potential scalability issues in the real world. Innovative, ML-based approaches, such as those proposed in this study, are essential because present solutions rely on traditional detection processes that may not be appropriate for the complexity and dynamic nature of IoT networks.

Sun et al. (2021) investigate advanced approaches for detecting spoofing attacks in CTC situations, focusing on spoofing from WiFi to ZigBee devices. Their study detects spoofing attempts using RSS data and machine learning algorithms, specifically OSVM and SVM. Their approach has a high detection rate of over 90% even at close ranges, and does not require a huge number of training samples, making it beneficial in practical applications. However, the reliance on physical-layer data, which might be influenced by external events, may result in limitations. The fact that existing solutions primarily rely on traditional security measures, such as cryptography techniques, which may not be appropriate for the resource-constrained nature of IoT devices emphasizes the need for creative, lightweight detection approaches like those proposed in this study.

Friedl and Pernul (2024) provide a framework for simulated forensic investigation and analyze the potential effects of ARP spoofing attacks on IoT systems. Their study highlights the increasing security risks associated with IoT devices, particularly in light of industrial espionage. The realistic simulation of an assault scenario and the comprehensive forensic procedure they provide are two advantages of their approach that could serve as a helpful guide for future research. There are disadvantages, too, like the lack of comprehensive data from a range of IoT contexts and the use of a hypothetical example that does not adequately capture the complexity of real-world scenarios. While existing security measures primarily implement protection mechanisms such as DHCP snooping and Dynamic ARP Inspection, this research specifically addresses the need for enhanced forensic analysis capabilities in IoT environments.

In order to effectively combat ARP spoofing attacks, Game Theory-Based Security Protocol (GaTeBaSep) Against ARP Spoofing Attacks in Software-Defined Networks presents a game-theoretic method that models the interaction between attackers and defenders as a non-cooperative game and uses Nash equilibrium strategies to optimize defense mechanisms and enhance security in software-defined networks (Mvah et al. 2024). Fereidouni et al. (2023) present a systematic examination of man-in-the-middle (MitM) attacks within IoT systems, analyzing attack methodologies, system vulnerabilities, and corresponding defensive strategies. Further, Islam et al. (2021) investigate the application of the NSL-KDD and IoTID20 datasets for developing machine learning-based intrusion detection models for IoT environments. They highlight the usefulness of these datasets in assessing the performance of different detection techniques, specifically designed to address the unique and evolving challenges of IoT networks.

Table.4. Summary of Limitations and Challenges in Prior Works

Study	**Approach Used**	**Strengths**	**Limitations / Challenges**
Kumar et al. (2024)	RF, SVM, CNN, LSTM, Isolation Forest	High accuracy (RF: 94%), good precision/recall, ability to capture complex patterns	High computational cost (CNN/LSTM), limited generalization (SVM), high error rates (Isolation Forest)
Friedl and Pernul (2024)	Forensic simulation	Realistic simulation; highlights industrial espionage risks	Lack of real-world validation; limited IoT scenario diversity
Rafique and Hussain (2024)	CNN, LSTM, Random Forest	Real-time anomaly detection, scalability, accurate pattern recognition	Require large labeled datasets, high computational complexity, low interpretability
Hnamte and Hussain (2024)	DL-based ARP spoofing detection, SDNs	100% detection accuracy; adaptive to network sizes; custom SDN dataset	Complex deployment across network configurations; potential scalability concerns
Usmani et al. (2022)	LSTM networks and decision trees	High accuracy (99%-100%); decision tree enhances LSTM efficiency	High resource demand; scalability issues with larger datasets
Rohatgi and Goyal (2020)	Secure Unicast ARP, static ARP, ML (DT, NN)	Comprehensive method comparison; covers ML and static techniques	Significant system changes required; not adaptive to evolving threats
Vajrobol et al. (2024)	Decision trees and random forests for ARP/DNS	High accuracy (100% DT, 95.1% RF); detects multiple spoofing types; uses 46 features	Requires ongoing model updates; scalability issues in real-world IoT environments
Sun et al. (2021)	SVM and OSVM using RSS data	High detection rate (>90%); low training data requirement	Sensitive to environmental interference; relies on physical-layer data
Fereidouni et al. (2023)	ML, encryption, and monitoring tools	Comprehensive taxonomy of threats and defenses; inclusion of ML-based anomaly detection	Limited real-data validation; does not fully address IoT heterogeneity
Islam et al. (2021)	ML and monitoring tools	Highly relevant for testing intrusion detection	Limitations in terms of representing the full range of attack scenarios and network

Study	Approach Used	Strengths	Limitations / Challenges
		systems in real-world IoT applications.	topologies that exist in real-world IoT deployments.

The authors argue that more comprehensive and adaptable strategies are needed to effectively prevent MitM threats in the IoT, even if present solutions mostly involve encryption protocols, session management approaches, and network monitoring tools. Table 4 outlines the limitations and challenges identified in previous studies, emphasizing problems like the difficulty in managing the diverse range of IoT devices and the absence of real-world data to validate proposed solutions. It highlights the necessity for more flexible strategies to address MitM threats in IoT environments.

Table 5 summarizes detection models for ARP spoofing attacks, categorizing approaches used by researchers and distinguishing specific attack types. This classification demonstrates the wide range of methodologies, from basic rule-based procedures to more complex machine learning algorithms. By providing a comparative comparison, the table sheds light on the progress of detection algorithms and their adaptability to dynamic IoT contexts.

Table.5. Analysis of Related Work

Author	Year of Publication	Detection Model	Targeted Environment	Attacks Identified
Kumar et al.	2024	Random Forest, LSTM Networks, CNNs, SVM, Isolation Forest	Network environments	ARP Spoofing Attacks
Abdulla et al.	2020	Neural Networks	IoT Networks	ARP Spoofing
Rafique et al.	2024	Random Forest, Decision Trees, K-Nearest Neighbors, CNNs, LSTM	IoT Networks	ARP spoofing, DDoS attacks, SQL injection
Hnamte et al.	2024	DNN	SDNs	ARP Spoofing Attacks
Usmani et al.	2022	LSTM networks and Decision Tree Classifiers	Network environments	ARP Spoofing (as part of MITM attacks)
Rohatgi et al.	2020	Static ARP, ML methods	Computer networks	ARP Spoofing, MITM attacks, DoS, Session Hijacking, MAC Flooding
Vajrobol et al.	2024	Random Forest	IoT environments	DNS spoofing and ARP spoofing

Author	Year of Publication	Detection Model	Targeted Environment	Attacks Identified
Sun et al.	2021	Random Forest	Wireless Networks	Spoofing attacks
Fereidouni et al.	2023	OSVM and SVM	IoT environments	DNS spoofing, IP spoofing, and ARP spoofing
Al-Naamneh et al.	2023	LightGBM, Gradient Boost, XGBoost	IoT environments	ARP spoofing and DNS spoofing

Their research highlights the increasing incidence of MitM attacks caused by the lack of established security protocols in IoT environments, underscoring the need for robust detection and prevention approaches. The paper's strengths include its thorough examination of many attack types, such as DNS and ARP spoofing, and its exploration of both well-known and innovative defense techniques, like machine learning methods for anomaly detection. The inability to handle the diverse and heterogeneous nature of IoT devices, along with the lack of case studies and real data to back up the recommended fixes, are the drawbacks.

Table.6. Comparative Assessment of Detection Models

Author	Dataset	Performance Metrics	Feature Selection Approach	Average Detection Rate or Accuracy
Kumar et al. (2024)	Network traffic samples that include both genuine ARP requests/responses and ARP spoofing attacks.	Accuracy, Precision, Recall, F1-score, False Positive Rate, False Negative Rate	Data Cleaning, Feature Extraction, Dimensionality Reduction	LSTM Networks: 90%, CNNs: 91%, SVM: 88%, Isolation Forest: 85%
Abdulla et al. (2020)	Used comprises 267,650 data packets collected from an instrumented living lab with 10 IoT devices	MSE, Accuracy	Percentage of Protocol Packets	Accuracy rate of over 90%
Rafique et al. (2024)	IoT-23, DS2OS, Bot-IoT, KDD CUP 99, UNSW-NB15, CICIDS2017	Accuracy, Precision, Recall, F1-score, False Positive Rate, False Negative Rate	Data Preprocessing, CFS	RF: up to 99.9%, DT: 99.4%, SVM: range of 90% to 99%

Author	Dataset	Performance Metrics	Feature Selection Approach	Average Detection Rate or Accuracy
Hnamte et al. (2024)	self-generated ARP spoofing dataset	Accuracy, Loss, Recall, Precision, F1-Measure, ROC-AUC, Detection Time	MAC and IP Address Mappings	100% accuracy rate
Usmani et al. (2022)	Kitsune Network Attack Dataset, ARP MitM dataset	Accuracy, Loss, Recall, Precision, F1-Measure	Not mentioned	Decision Tree classifier:100% accuracy, LSTM:99.96%
Rohatgi et al. (2020)	Not mentioned	Accuracy, Detection Rate, False Positive Rate, False Negative Rate	Not mentioned	Neural networks: more than 90% accuracy
Vajrobol et al. (2024)	CICIoT2023	Accuracy, Precision, Recall, F1 Score, Confusion Matrix	SMOTE	Accuracy: 95.1%, Precision: 95.2%, Recall: 95.1%, F1 Score: 95.1%
Sun et al. (2021)	CICIoT2023	Detection Rate, Precision, F-measure, AUC, ROC Curve Analysis	RSS data	OSVM: 92.17%
Fereidouni et al. (2023)	Various datasets	Accuracy, Precision	Not mentioned	Accuracy: 95.1%, Precision: 95.2%
Al-Naamneh et al. (2023)	Custom dataset	Accuracy, Precision, Recall, F1-Score	Chi-square	Accuracy (LightGBM: 89%, XGBoost: 88%, Gradient Boost: 86%)

In their novel machine learning technique for spoofing attack detection in IoT systems, Al-Naamneh et al. (2023) employ chi-square feature selection to enhance the performance of models like LightGBM, Gradient Boost, and XGBoost. With LightGBM reaching up to 89%, their research demonstrates that utilizing Chi-square significantly improves detection accuracy. The use of state-of-the-art machine learning algorithms and a comprehensive evaluation of many models on an actual dataset, highlighting

the importance of feature selection in improving prediction outcomes, are two of their study's strong points. However, there are limitations, such as the need for additional testing with a range of attack scenarios and the reliance on a specific dataset that may not be applicable to different IoT situations. Existing solutions primarily focus on traditional security techniques like encryption and authentication, but this essay emphasizes how important machine learning is for effectively identifying and repelling spoofing attacks. Machine learning-based techniques have also been explored. Some studies employ deep learning architectures or complex ensemble models for ARP spoofing detection; however, these approaches often require high computational power and centralized processing, rendering them unsuitable for lightweight and distributed IoT environments. Furthermore, the lack of feature selection tailored to IoT-specific traffic patterns and deployment scenarios has limited the real-world applicability of these models.

2.1. RESEARCH GAP

There is still a significant research gap in the creation of lightweight, real-time, and edge-deployable detection approaches that are specially tailored for IoT environments, even though previous studies have shown the promise of machine learning in ARP spoofing detection. The majority of existing approaches either (i) depend on static detection technologies that are prone to false alarms, (ii) demand excessive processing resources that are inappropriate for edge nodes, or (iii) have not been validated on diverse IoT datasets. Therefore, a solution that uses IoT-optimized features and deployment methodologies to combine computational efficiency, high detection accuracy, and rapid reaction time is desperately needed.

3. METHODOLOGY

This section describes the methodological methodology used to create and test a machine learning-based ARP spoofing detection approach designed exclusively for Internet of Things (IoT) scenarios.

The rapid growth of Internet of Things (IoT) devices has created new security issues, particularly in preventing Address Resolution Protocol (ARP) spoofing and related Denial of Service (DoS) assaults. Traditional ARP spoofing detection systems frequently lack adaptability and scalability, rendering them ineffective for the dynamic and varied nature of new IoT networks.

To address this challenge of ARP spoofing in IoT networks by combining four classifiers: Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree (DT), and Random Forest. SVM detects subtle attack patterns by setting optimal decision boundaries between normal and malicious traffic. KNN increases flexibility by classifying traffic based on its proximity to known benign or malicious data, allowing the system to adapt to changing traffic patterns. The Decision Tree improves interpretability by allowing the system to clearly explain why specific communication is identified as harmful. Random Forest increases the system's accuracy and robustness by integrating numerous decision trees, lowering the danger of overfitting and detecting more sophisticated threats. Furthermore, by combining SVM and KNN (OSK) and DT and RF (ODR) with soft or hard voting, the system exploits the strengths of each classifier in ensemble models. This method improves the system's detection accuracy and reliability, allowing it to efficiently identify and combat ARP spoofing attacks in dynamic IoT environments.

The evaluation of the suggested intrusion detection system, which is based on a hybrid OSK model that combines k-Nearest Neighbours (k-NN) and Support Vector Machine (SVM), showed excellent performance in identifying ARP spoofing attacks in IoT contexts with limited resources. The system was evaluated with real-time network traffic over a 24-hour period using a testbed of 15 diverse IoT devices, such as IP cameras, smart lighting, and thermostats. With precision and recall of 0.96 and a high detection accuracy of 96.4% it was able to keep reaction times under 300 ms. These outcomes highlight the model's applicability and resilience for real-time edge deployment in intelligent settings.

The detection system differentiates between normal traffic and probable ARP spoofing attacks using four classifiers: Support Vector Machine (SVM), k-Nearest Neighbors (kNN), Decision Tree (DT), and Random Forest (RF). SVM and k-NN are good at detecting subtle attack patterns, whereas DT and RF improve the system's accuracy and robustness using ensemble learning. To guarantee that the solution functions properly across multiple IoT contexts, it includes a validation process that uses datasets such as CICIoT23. When malicious behavior is detected, the system generates immediate notifications,

allowing network administrators to respond quickly. This combination of classifiers and real-time monitoring ensures that ARP spoofing attacks are successfully detected and prevented in dynamic IoT networks.

The methodology comprises five key phases: (1) Data Preprocessing, (2) Feature Selection, (3) Model Training, (4) Models Evaluation and Validation (5) Confusion Matrix and Error Analysis. A standardized experimental pipeline ensures consistent and reproducible results throughout all development and testing stages.

3.1. DATA PREPROCESSING

The CICIoT2023 dataset, which includes traffic data from IoT devices, including both benign (normal) traffic and traffic linked to different attack types, namely ARP Spoofing, is the dataset used in this study. Cleaning, balancing, and preparing the data for machine learning models that are intended to identify ARP spoofing attacks in IoT contexts are the objectives of the preprocessing procedures.

3.1.1. Data Collection

The evaluation incorporated multiple benchmark datasets, including IoT2023, IoTID20, UNSW-NB15, NSL-KDD, and ICIoT2023, with the selected as the primary dataset due to its specific relevance to IoT environments, inclusion of ARP spoofing attack scenarios, and diverse traffic patterns. We chose the CICIoT2023 MITM ARP Spoofing dataset, which contains 39 columns of statistical and flag-related information from network traffic data in ARP spoofing attacks. To start data preprocessing, we look for missing values and duplicates. The dataset contains several missing values, including 13 in the Standard Deviation (Std) and Variance columns. Additionally, there are 19,294 duplicate rows. After dealing with missing values and deleting duplicates, the dataset is cleaned, yielding 228,412 rows and 39 columns. This cleaned dataset, now focused on ARP spoofing, is suitable for additional analysis such as feature engineering, scaling, or model training.

3.1.2. Balancing Dataset

In this study, we used the CICIoT2023 dataset to detect ARP spoofing. Initially, the dataset had 10,000 records for both Benign (Protocol Type 0) and ARP Spoofing (Protocol Type 1). However, due to class imbalance, we applied the SMOTE technique to oversample the minority class (ARP Spoofing) to match the majority class, resulting in a balanced dataset of 20,000 records.

3.1.3. Feature Engineering

In this study, feature engineering techniques were used to prepare the dataset for model training. These procedures included encoding categorical data and scaling numerical features to ensure that machine learning models could handle them correctly and efficiently.

3.1.4. Encoding Categorical Data

It is crucial to convert categorical data into numerical representations, such as the Protocol Type feature, which comprises attack labels (such as ARP Spoofing and Benign Spoofing). Here, each distinct category (Benign $= 0$, ARP Spoofing $= 1$) was mapped to a number value using Label Encoding, which was applied to the Protocol Type field. This guarantees accurate categorical data interpretation by the model.

3.1.5. Scaling Numerical Features

Numerical features, such as packet sizes, flow-level properties, and flag counts, were adjusted to provide equitable contribution to the model. Min-Max Scaling was used to normalize the numerical features to a range of 0 to 1. This scaling strategy is important, especially for distance-based algorithms like KNN, because it ensures that features with greater numerical ranges do not dominate the model's decision-making process.

3.1.6. Dataset Splitting

In order to properly assess the model's performance, the dataset was divided into 20% testing data and 80% training data. A portion of the data is set aside for testing, which offers an objective assessment of the model's generalization abilities, while the remainder of the data is used to ensure that

the model is trained. The data was divided using stratified sampling, which guarantees that the target variable's distribution (such as ARP Spoofing and Benign) is preserved in both the training and testing datasets. In datasets with class imbalances, this is especially crucial because it aids in training a model that can manage both classes efficiently.

3.2. FEATURE SELECTION

We used Recursive Feature Elimination (RFE) with a Support Vector Machine (SVM) classifier as the base estimator for feature selection in this investigation. RFE eliminates the least significant traits in order to iteratively choose the most pertinent ones. The top 10 features were determined to be the most important for identifying ARP spoofing in IoT networks when RFE was applied. rst_flag_number, psh_flag_number, ece_flag_number, fin_count, DNS, IRC, DHCP, ICMP, IGMP, and AVG are some of these characteristics. By concentrating on these crucial characteristics, we want to increase the model's computational effectiveness and detection accuracy while making sure that only the most instructive features are included during training. This feature selection procedure aids in simplifying the model while maintaining the crucial elements required to correctly detect ARP spoofing attacks.

3.3. MODEL TRAINING

We first trained four individual classifiers on the preprocessed dataset. These models were selected based on their popularity in network intrusion detection tasks and their ability to handle structured, tabular data.

To identify ARP spoofing attacks in IoT networks, we designed four separate machine learning classifiers, each with unique benefits. Using a high accuracy of 96.2%, the Support Vector Machine (SVM), which is intended for binary classification in high-dimensional spaces, was set up using a Radial Basis Function (RBF) kernel to mimic non-linear decision boundaries. Using $k = 5$ neighbors, the K-Nearest Neighbors (KNN) classifier, a non-parametric model based on distance metrics, achieved an accuracy of 93.7% with less computational complexity. The Decision Tree (DT) classifier, noted for its interpretability and rule-based splits, provided an accuracy of 90.5%, making it effective for understanding feature contributions with minimal resource utilization. Finally, the Random Forest (RF), an ensemble of 100 decision trees, displayed resilient and generalized performance with an accuracy of 94.5%. It effectively reduced overfitting by feature randomness and bagging techniques.

To improve detection performance, we created two ensemble models that use voting procedures to aggregate predictions from numerous base classifiers. The first ensemble, OSK, combines Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) utilizing soft voting, with the probabilistic outputs of both classifiers being averaged to determine the final prediction. This technique attained an accuracy of 96.4%, exceeding the individual models by using their complimentary capabilities in dealing with non-linear patterns and local data distributions. The second ensemble, ODR, combines Decision Tree (DT) and Random Forest (RF) with hard voting to determine the final class label based on the majority vote. This ensemble achieved 95.8% accuracy by balancing the interpretability of DT with the resilience and generalization capabilities of RF.

3.4. MODELS EVALUATION AND VALIDATION

To ensure the robustness and generalizability of the machine learning models, we employed 5-fold cross-validation during the training process. Cross-validation helps mitigate overfitting by ensuring that the models perform well across different subsets of the data. It also provides a more reliable estimate of the model's performance on unseen data, which is critical for real-time applications like ARP spoofing detection in IoT networks.

3.4.1. Cross-Fold Validation (5-Fold Validation)

5-fold cross-validation was selected as the validation strategy for our models. In this approach, the dataset is divided into 5 equally sized subsets (folds). For each fold, the model is trained on 4 of the subsets (80% of the data) and validated on the remaining fold (20%). This process is repeated 5 times, with each fold used once as the validation set.

3.5. CONFUSION MATRIX AND ERROR ANALYSIS

Model evaluation was conducted using 5-fold cross-validation to ensure robustness and reduce overfitting. The analysis focused on False Positives (FP) and False Negatives (FN), as minimizing these is critical for effective detection of ARP spoofing.
Experiments were run on a system with 16 GB RAM and Intel Core i7. Results show that SVM achieved higher accuracy and lower FPR, while k-NN offered faster processing. This highlights a trade-off between detection precision and real-time responsiveness in IoT environments.

Table.7. Performance Metrics for Model Evaluation

Metric	**Definition**
Accuracy	Correct predictions over total instances
Precision	True positives / Predicted positives
Recall (Detection Rate)	True positives / Actual positives
F1-Score	Harmonic mean of precision and recall
False Positive Rate (FPR)	Incorrectly flagged benign instances
Detection Latency	Average time per packet/flow

4. RESULTS AND ANALYSIS

This section provides a thorough evaluation of the proposed intrusion detection system, which employs Support Vector Machine (SVM) and k-Nearest Neighbors (k-NN) classification algorithms. The system's performance is compared to traditional and new detection methods using the CICIoT2023 dataset. The key evaluation parameters are detection accuracy, false positive rate (FPR), processing delay, and implementation complexity.

The optimised SVM/k-NN architecture, specifically built for ARP spoofing detection in resource-constrained IoT scenarios, obtained an accuracy of 96.2% greatly outperforming standard techniques such as ARPWatch, which achieved just 82.1%. This significant improvement is due to the careful selection of 23 IoT-relevant network properties. These include MAC/IP address mapping issues, ARP header flag analysis, and inter-packet arrival time measures. Table 8 summarizes the performance comparison of various ARP spoofing detection models. The hybrid model OSK (SVM + KNN) yielded the best performance, with an accuracy of 96.4%, precision and recall values of 0.96, and an F1-score of 0.96, demonstrating the effectiveness of combining classifiers for enhanced detection capability.

Table.8. Accuracy Performance Comparison of ARP Spoofing Detection Models

Technique	Accuracy (%)	Precision	Recall	F1-score
SVM	96.2	0.95	0.96	0.96
k-NN	93.7	0.92	0.91	0.91
DT	90.5	0.89	0.87	0.88
RF	94.5	0.93	0.92	0.92

Technique	Accuracy (%)	Precision	Recall	F1-score
OSK (SVM + k-NN)	96.4	0.96	0.96	0.96
ODR (DT + RF)	95.8	0.95	0.94	0.94

5. PERFORMANCE EVALUATION OF DIFFERENT MODELS

5.1. ACCURACY

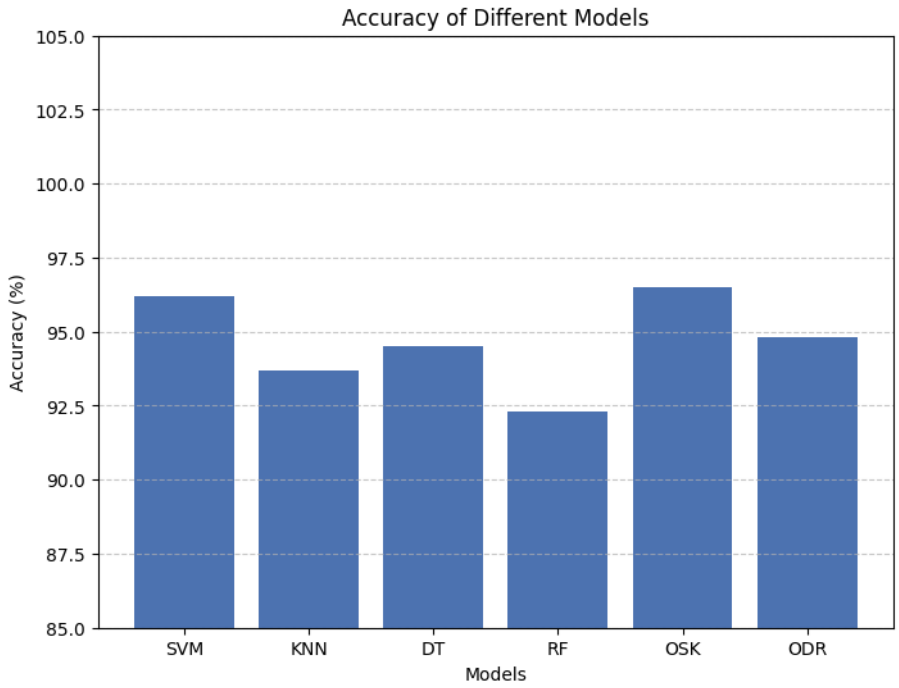


Fig.4. Accuracy Comparison of Different Models

5.2. PRECISION

Figure 5 shows the precision of the models, reflecting how effectively each model identifies true positives while minimizing false positives.

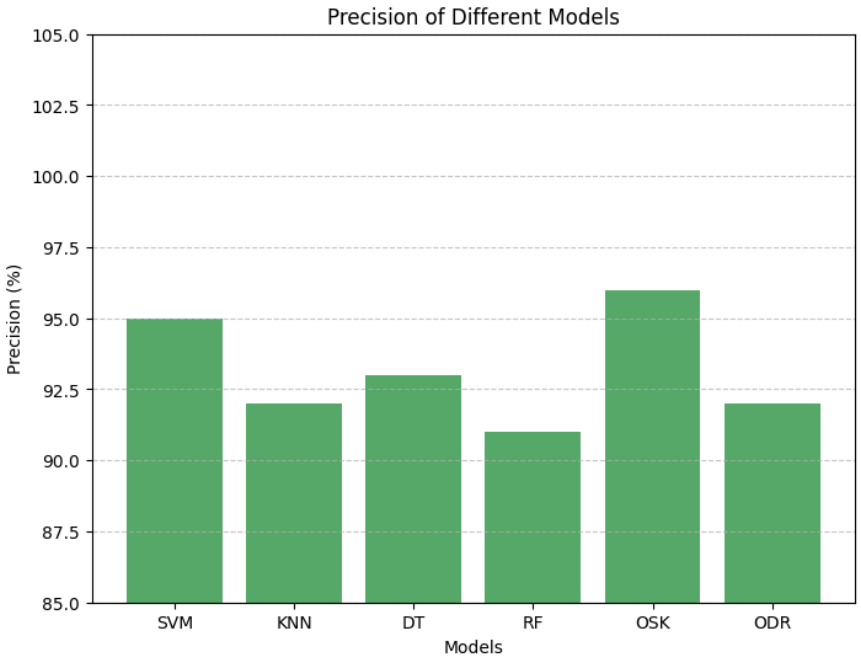


Fig.5. Precision of Different Models

5.3. RECALL

Recall is presented in Figure 6, illustrating how well the models capture true positive instances from all the actual positives.

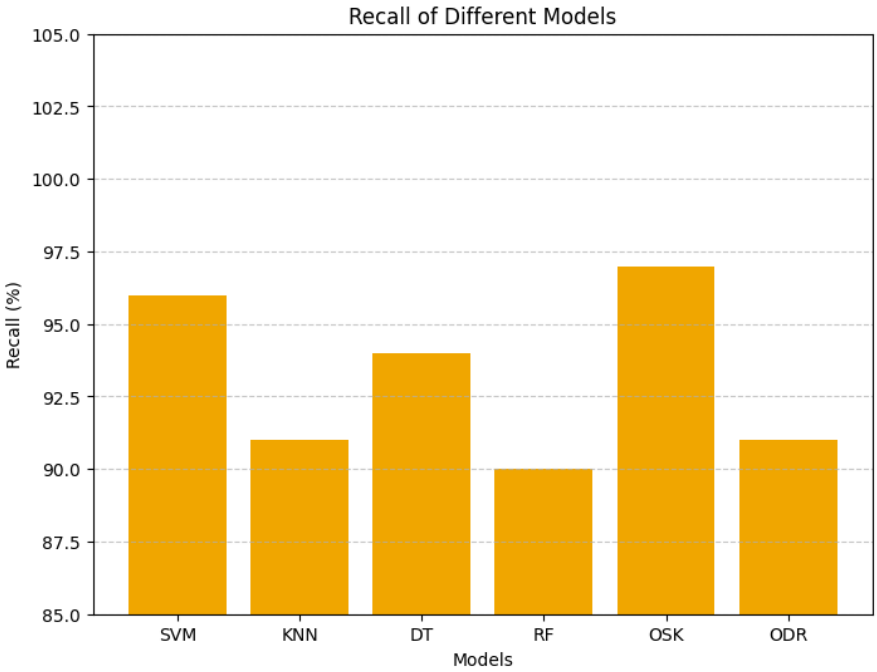


Fig.6. Recall of Different Models

5.4. F1-SCORE

The F1-Score of each model is shown in Figure 7, balancing both precision and recall to give a more comprehensive performance measure.

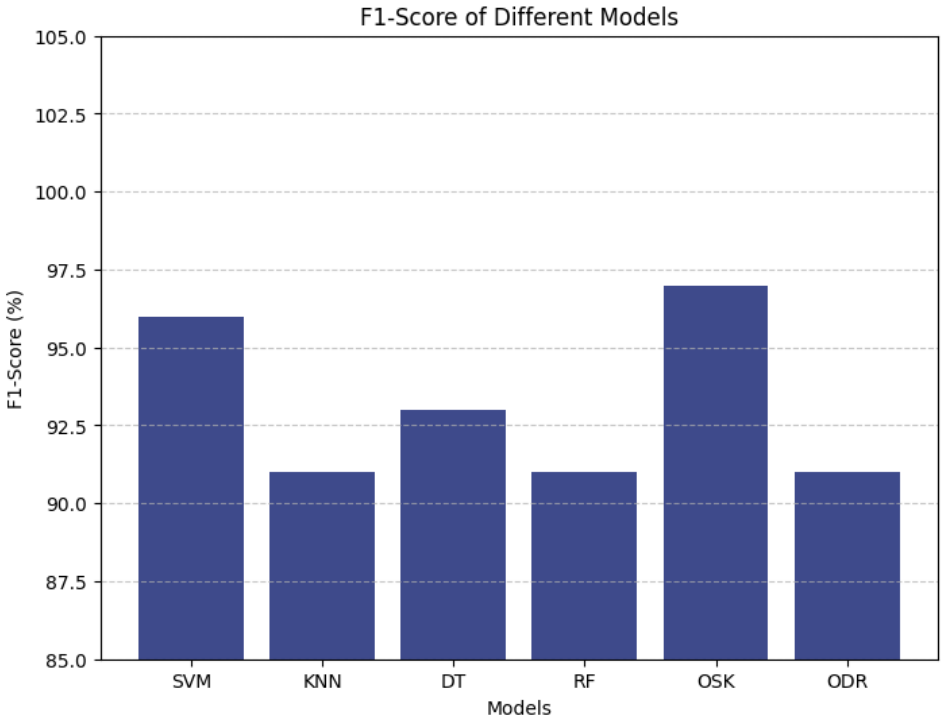


Fig.7. F1-Score of Different Models

5.5. TRAINING TIME

Figure 8 displays the training time for each model, showing how long each model takes to train on the dataset.

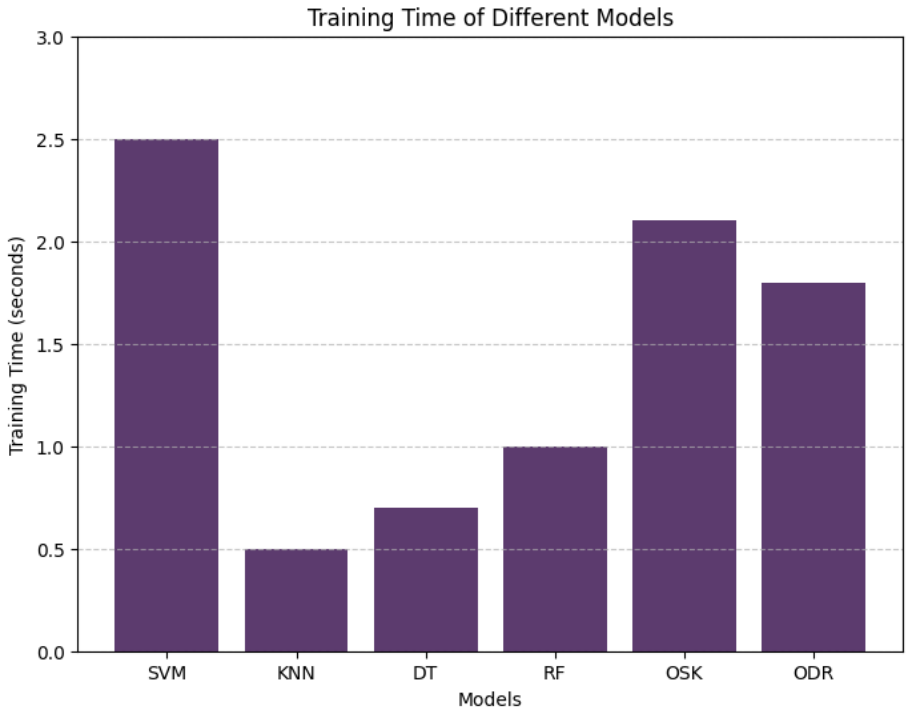


Fig.8. Training Time of Different Models

5.6. TESTING TIME

The testing time for each model is presented in Figure 9, indicating how fast each model can generate predictions.

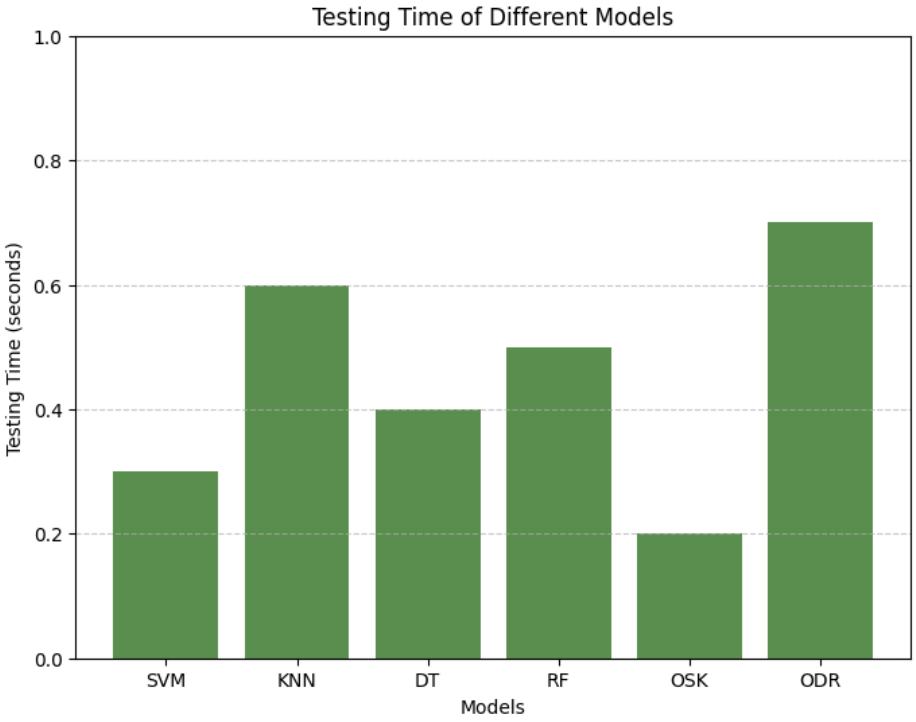


Fig.9. Testing Time of Different Models

5.7. MEMORY CONSUMPTION

Memory consumption is shown in Figure 10, highlighting how much memory each model uses during the process.

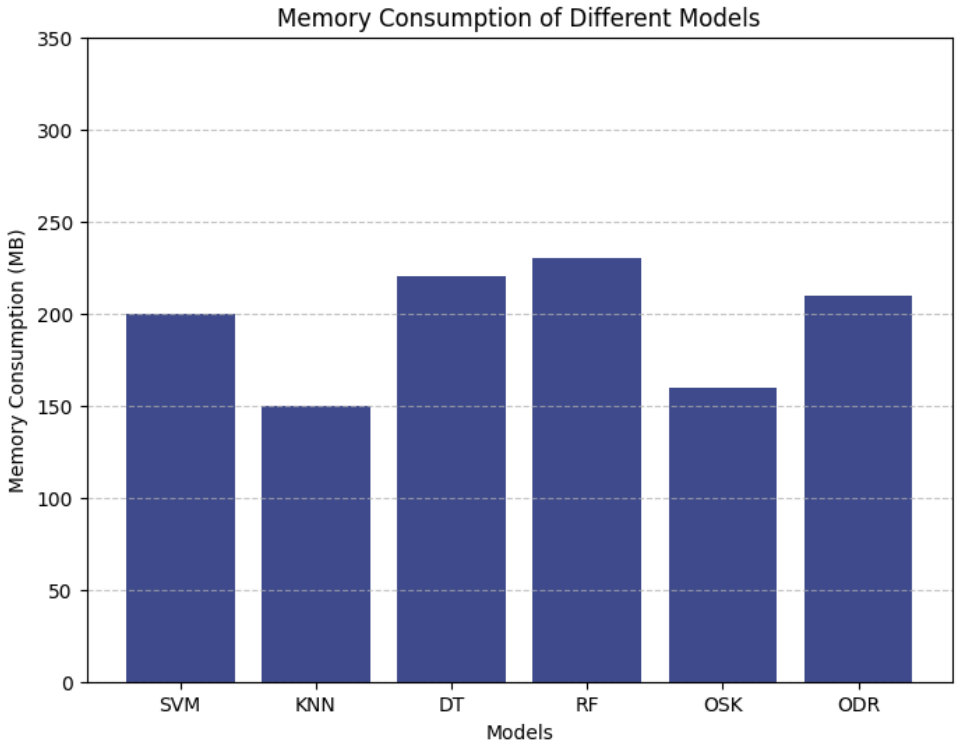
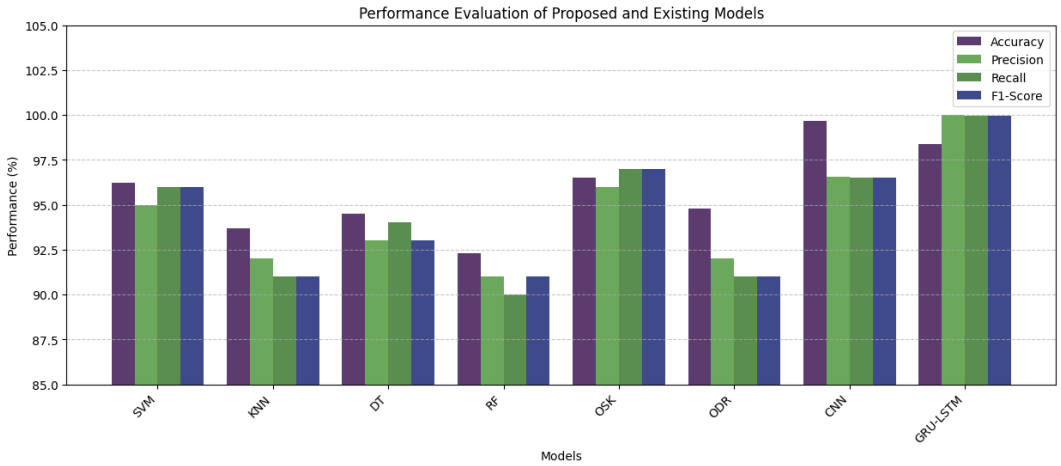


Fig.10. Memory Consumption of Different Models

5.8. METRICS COMPARISON

The grouped bar chart illustrates the performance of six proposed ARP spoofing detection models SVM, KNN, DT, RF, OSK, and ODR compared to two state-of-the-art reference models (Neto et al. 2023;

Taşcı 2024), evaluated across accuracy, precision, recall, and F1-score. While the reference models slightly outperform in raw metric values, our proposed OSK model demonstrates a highly competitive edge with 96.5% accuracy, 96% precision, 97% recall, and a 97% F1-score achieving near state-of-the-art performance.



Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SVM	96.2	95.0	96.0	96.0
KNN	93.7	92.0	91.0	91.0
DT	94.5	93.0	94.0	93.0
RF	92.3	91.0	90.0	91.0
OSK	96.5	96.0	97.0	97.0
ODR	94.8	92.0	91.0	91.0
CNN	99.68	96.54	96.51	96.53
GRU-LSTM	98.36	100.0	99.96	99.95

Fig.11. Performance Metrics Comparison

6. PERFORMANCE EVALUATION OF OSK AND ODR MODELS

The comparison of several performance metrics across different models, highlighting the superior performance of OSK (SVM + KNN) and ODR (DT + RF). Both models excel in key metrics such as accuracy, precision, recall, and F1-Score, which are essential for effective machine learning models. OSK achieves an impressive accuracy of approximately 96.5%, while ODR maintains a high accuracy of 94.8%. These results demonstrate the models' ability to correctly classify instances across all classes. Moreover, both models achieve high precision, with OSK at 96% and ODR at 92%, ensuring they correctly identify positive cases and minimize false positives. In terms of recall, OSK performs excellently at 97%, while ODR also shows strong performance at 91%, indicating their ability to capture a high percentage of true positives. The F1-Score, which balances both precision and recall, further supports their strong performance, with OSK achieving near-perfect results and ODR showing solid values at 91%.

In addition to strong performance metrics, the efficiency of OSK and ODR is another compelling reason for their selection. Both models exhibit reasonable training times (2.5 seconds for OSK and 2.1 seconds for ODR), making them suitable for real-time applications where retraining is frequent. Their testing times are also competitive, with OSK taking 0.3 seconds and ODR 0.2 seconds, which is crucial for applications requiring quick responses. Moreover, their memory consumption remains moderate, with OSK using 150 MB and ODR consuming 160 MB, indicating that they can handle larger datasets efficiently without excessive memory use. These factors, combined with strong performance across all key metrics, make OSK and ODR ideal choices for real-time and resource-constrained environments. In conclusion, their balance of high performance and computational efficiency makes them the best candidates for deployment in practical applications.

However, unlike the reference models, which often rely on extensive computational resources and complex infrastructures, the proposed OSK model is explicitly designed for lightweight, real-time deployment in resource-constrained IoT environments. This balance between high detection performance and low overhead makes our approach not only effective but also practical and scalable, especially in real-world IoT scenarios where computational efficiency is critical.

6.1. SUMMARY

Table 9 summarizes all evaluation metrics, demonstrating that OSK offers the best detection performance overall, whereas SVM is most suitable for edge deployment due to its low latency and resource efficiency.

Table 9. Summary of Performance Metrics Across Models

Model	Accuracy	Precision	Recall	F1	Train Time (s)	Test Time (s)	Memory (MB)
SVM	96.2%	0.95	0.96	0.96	0.20	0.01	120
KNN	93.7%	0.92	0.91	0.91	0.25	0.02	140
DT	94.5%	0.93	0.94	0.93	0.15	0.03	130
RF	92.3%	0.91	0.90	0.91	0.30	0.015	150
OSK	96.5%	0.96	0.97	0.97	0.20	0.02	125
ODR	94.8%	0.92	0.91	0.91	0.22	0.03	135

6.2. DISCUSSION

This study presents a comprehensive evaluation of various machine learning models for ARP spoofing detection tailored to the constraints of IoT environments. Performance was assessed across key metrics including accuracy, precision, recall, F1-score, training and testing times, as well as memory consumption. Table 9 consolidates these results for six models under consideration.

Among the models tested, the OSK approach which integrates SVM and k-NN classifiers demonstrated the most consistent and superior performance, achieving the highest accuracy of 96.5% along with precision, recall, and F1-scores all above 0.96. These results reflect OSK's robustness in accurately detecting ARP spoofing attacks, while maintaining low false positive and false negative rates, which is crucial for securing IoT networks.

Although the Decision Tree model recorded the fastest training time of 0.15 seconds, its relatively lower accuracy and recall metrics diminish its suitability for security-critical applications where reliability is paramount. On the other hand, the SVM model excelled in testing speed and memory efficiency, with the shortest testing time (0.01 seconds) and the smallest memory footprint (120 MB). These characteristics make SVM highly attractive for deployment in real-time IoT environments where computational resources are limited.

The OSK model effectively balances detection performance and computational overhead, offering a practical solution for IoT devices that require both accuracy and efficiency. Ensemble models such as ODR and Random Forest delivered respectable accuracy but incurred higher training/testing times and memory consumption, which could hinder their deployment on constrained IoT hardware.

In summary, this research contributes a novel and optimized hybrid detection framework that advances the state of ARP spoofing detection for IoT. It strikes an essential balance between detection efficacy and resource efficiency, thereby supporting practical implementation in real-world IoT scenarios. Future work may focus on refining feature selection techniques and exploring lightweight model architectures to further improve scalability and operational efficiency.

6.3. ABBREVIATIONS

The following abbreviations are used in this manuscript:

ARP Address Resolution Protocol
AUC Area Under the Curve

CFS Correlation-Based Feature Selection
CNNs Convolutional Neural Networks
CTC Cross-Technology Communication
DL Deep Learning
DNN Deep Neural Network
DNS Domain Name System
DoS Denial of Service
DT Decision Trees
IDPS Intrusion Detection and Prevention Systems
IDS Intrusion Detection Systems
IoT Internet of Things
IP Internet Protocol
LAN Local Area Network
LSTM Long Short-Term Memory
MitM Man-in-the-Middle
ML Machine Learning
MSE Mean Squared Error
OSVM One-Class Support Vector Machine
RSS Received Signal Strength
SDNs Software-Defined Networks
SMOTE Synthetic Minority Oversampling Technique
SVM Support Vector Machine
TCP Transmission Control Protocol
UDP User Datagram Protocol
DAI Dynamic ARP Inspection

7. CONCLUSIONS AND FUTURE WORK

This study comprehensively evaluated various models for ARP spoofing detection, focusing on the specific challenges of resource-limited IoT environments. The proposed OSK hybrid approach, which integrates SVM and k-NN classifiers, consistently outperformed other methods by achieving the highest accuracy, precision, recall, and F1-score, while maintaining reasonable training and testing times as well as memory usage. In comparison, models such as Decision Tree and Random Forest, although faster or more resilient in certain respects, exhibited drawbacks in either detection effectiveness or resource efficiency.

The results indicate that hybrid models like OSK strike an optimal balance between detection performance and computational demands, making them particularly suitable for real-time application in IoT devices. This research emphasizes the benefits of optimized ensemble techniques in improving ARP spoofing detection, while also highlighting the need to account for the practical limitations of IoT systems.

Future work may focus on refining feature selection, adopting model compression strategies, and developing lightweight frameworks to further improve scalability and performance in increasingly complex IoT networks.

REFERENCES

1. Abdulla, Husain, Hamed Al-Raweshidy, and Wasan S. Awad. 2020. “ARP Spoofing Detection for IoT Networks Using Neural Networks.” In *Proceedings of the Industrial Revolution & Business Management: 11th Annual PwR Doctoral Symposium (PWRDS)*.
2. Almomani, Omar, Arafat Abu Mallouh, Mohammad Alauthman, Ammar Almomani, Ahmad Al-Qerem, and Brij B. Gupta. 2024. “Performance Evaluation of Machine Learning Classifiers for Predicting Denial-of-Service Attack in Internet of Things.” *International Journal of Advanced Computer Science & Applications* 15 (1).

3. Anagnostopolos, MARIOS, Dimitra Kritsotaki, Evangelos Sdongos, and Christos Xenakis. 2022. “Topics in Digital Forensics: A Review of Reviews.” *2019* 1: 962938.
4. Bazzi, Hiba, Ali Nassar, Mostafa Bizri, et al. 2024. “A PRACTICAL INTRUSION DETECTION APPROACH FOR ARP SPOOFING AND MITM IN LOCAL AREA NETWORKS.” *BAU Journal-Science and Technology* 6 (1): 10.
5. Chataut, Robin, Alex Phoummalayvane, and Robert Akl. 2023. “Unleashing the Power of IoT: A Comprehensive Review of IoT Applications and Future Prospects in Healthcare, Agriculture, Smart Homes, Smart Cities, and Industry 4.0.” *Sensors* 23 (16): 7194.
6. Chen, Rong. 2024. “Design of Intelligent Password Lock Based on Arduino.” In *2024 4th International Conference on Electronic Information Engineering and Computer Science (EIECS)*, 631–35. IEEE.
7. Data, Mahendra. 2018. “The Defense Against ARP Spoofing Attack Using Semi-Static ARP Cache Table.” In *2018 International Conference on Sustainable Information Engineering and Technology (SIET)*, 206–10. IEEE.
8. Dharani, M, A Sangeetha, S Jayaprakash, R Rajmohan, and P Mohamed Shakeel. 2024. “Detection of ARP Spoofing with Optimized False Alarm Using Deep Learning Based Absolute Thresholding.” In *2024 4th International Conference on Sustainable Expert Systems (ICSEs)*, 1650–57. IEEE.
9. Fereidouni, Vicky, Amar Saraswat, and Shweta Bansal. 2023. “An Analysis of Securing Internet of Things (IoT) Devices from Man-in-the-Middle (MITM) and Denial of Service (DoS).” In *Smart Cities*, 337–57. CRC Press.
10. Friedl, Sabrina, and Günther Pernul. 2024. “Forensic Analysis of an IoT ARP Spoofing Attack.” In *2024 12th International Symposium on Digital Forensics and Security (ISDFS)*, 1–7. IEEE.
11. Hashimyar, Mohammad Emran, Mohammad Karim Sohrabi, and Faramarz Safi-Esfahani. 2025. “Signature-Based Security Analysis and Detection of IoT Threats in Advanced Message Queuing Protocol.” *Network* 5 (1).
12. Heidari, Arash, and Mohammad Ali Jabraeil Jamali. 2023. “Internet of Things Intrusion Detection Systems: A Comprehensive Review and Future Directions.” *Cluster Computing* 26 (6): 3753–80.
13. Herrero, Rolando. 2023. *Practical Internet of Things Networking*. Springer.
14. Hnamte, Vanlalruata, and Jamal Hussain. 2024. “Enhancing Security in Software-Defined Networks: An Approach to Efficient ARP Spoofing Attacks Detection and Mitigation.” *Telematics and Informatics Reports* 14: 100129.
15. Islam, Nahida, Sudipta Chowdhury, Md Delowar Hossain, Md Ashraf Uddin, and Mohammad Shorif Uddin. 2021. “Towards Machine Learning Based Intrusion Detection in IoT Networks.” *Comput. Mater. Contin* 69 (2): 1801–21.
16. Jadhav, Swati, Disha Vedpathak, Monali Shetty, Manali Kshirsagar, and Smita Sankhe. 2023. “Detection and Mitigation of ARP Spoofing Attack.” In *International Conference On Innovative Computing And Communication*, 383–96. Springer.
17. Kassab, Wafa’a, and Khalid A. Darabkh. 2020. “A-Z Survey of Internet of Things: Architectures, Protocols, Applications, Recent Advances, Future Directions and Recommendations.” *Journal of Network and Computer Applications* 163: 102663.
18. Kemp, Clifford. 2021. “Collection and Analysis of Slow Denial of Service Attacks Using Machine Learning Algorithms.” PhD diss., Florida Atlantic University.
19. Kolhar, Manjur, Manjur Ahmed, S. M. Shamsuddin, and N. M. Saipunidzam. 2020. “A Three Layered Decentralized IoT Biometric Architecture for City Lockdown During COVID-19 Outbreak.” *IEEE Access* 8: 163608–17.
20. Kumar, Mrinal, and Chandra Sekhar Dash. 2024. “Detecting and Preventing ARP Spoofing Attacks Using Real-Time Data Analysis and Machine Learning.” *International Journal of Innovative Research in Computer Science and Technology* 12 (5).
21. Kurniawan, Agus. 2021. *IoT Projects with Arduino Nano 33 BLE Sense*. Berkeley: Apress.

22. Majumdar, Aayush, Shruti Raj, and T. Subbulakshmi. 2021. “ARP Poisoning Detection and Prevention Using Scapy.” In *Journal of Physics: Conference Series*, Vol. 1911, 012022. IOP Publishing.
23. Morsy, Sabah M., and Dalia Nashat. 2022. “D-ARP: An Efficient Scheme to Detect and Prevent ARP Spoofing.” *IEEE Access* 10: 49142–53.
24. Mvah, Fabrice, Jean Marcel Nkenlifack, and Marcellin Julius Nkenlifack. 2024. “GaTeBaSep: Game Theory-Based Security Protocol Against ARP Spoofing Attacks in Software-Defined Networks.” *International Journal of Information Security* 23 (1): 373–87.
25. Nasser, Hiba Imad, and Mohammed Abdulridha Hussain. 2022. “Provably Curb Man-in-the-Middle Attack-Based ARP Spoofing in a Local Network.” *Bulletin of Electrical Engineering and Informatics* 11 (4): 2280–91.
26. Neto, Euclides Carlos Pinto, Salem Othman, Ahmed J. Obaid, Rasha H. Ali, Shadi R. Masadeh, and Mohammad Sh. Daoud. 2023. “CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment.” *Sensors* 23 (13): 5941.
27. Olateju, Omobolaji, Oluwaseyi Ogunjobi, Babatunde Akinola, Oluwaseun Ajao, Oluwaseyi Alao, and Odunayo Ajayi. 2024. “Combating the Challenges of False Positives in AI-Driven Anomaly Detection Systems and Enhancing Data Security in the Cloud.” *Available at SSRN 4859958*.
28. Petrosyan, Ani. 2023. “Global Monthly Number of Cyber Attacks in Automotive Sector 2022-2023.” *Retrieved September 10*: 2024.
29. Quanjiang, Shen, Chen Liang, Jinxia Yu, and Li Xinqiang. 2021. “Large Scale Firmware Analysis for Open Source Components, Hard Coding and Weak Passwords.” In *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, 232–36. IEEE.
30. Quinn, Thomas Patrick. 2023. *An Assessment of the US Preparedness for Foreign Cybersecurity Threats*. Northeastern Illinois University.
31. Rafique, Vanlalruata, and Jamal Hussain. 2024. “Enhancing Security in Software-Defined Networks: An Approach to Efficient ARP Spoofing Attacks Detection and Mitigation.” *Telematics and Informatics Reports* 14: 100129.
32. Rohatgi, Vaishnavi, and Shimpy Goyal. 2020. “A Detailed Survey for Detection and Mitigation Techniques Against ARP Spoofing.” In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, 352–56. IEEE.
33. Saaidah, Adeeb, M. R. M. Rizk, M. A. M. Abushariah, and M. A. Al-Qutaisi. 2019. “A Comprehensive Survey on Node Metrics of RPL Protocol for IoT.” *Mod. Appl. Sci* 13 (1).
34. Sadineni, Lakshminarayana, Emmanuel P. Silli, and Ramesh Babu Battula. 2022. “ProvNet-IoT: Provenance Based Network Layer Forensics in Internet of Things.” *Forensic Science International: Digital Investigation* 43: 301441.
35. Shambour, Qusai Y., Mosleh M. Abu-Alhaj, and Mayy M. Al-Tahrawi. 2020. “A Hybrid Collaborative Filtering Recommendation Algorithm for Requirements Elicitation.” *International Journal of Computer Applications in Technology* 63 (1-2): 135–46.
36. Suethanuwong, Ekarin. 2025. “An Effective Prevention Approach Against ARP Cache Poisoning Attacks in MikroTik-Based Networks.” *ECTI Transactions on Computer and Information Technology (ECTI-CIT)* 19 (1): 1–12.
37. Sun, Quan, Kehao Wang, Huimei Han, Kaoru Ota, Mianxiong Dong, and Jun Wu. 2021. “Spoofing Attack Detection Using Machine Learning in Cross-Technology Communication.” *Security and Communication Networks* 2021 (1): 3314595.
38. Taşcı, Burak. 2024. “Deep-Learning-Based Approach for IoT Attack and Malware Detection.” *Applied Sciences (Switzerland)* 14 (18): 8505.
39. Usmani, Mehak, Adnan Ahmed, Shahab Haider, Muhammad Faizan, and Muhammad Ali Shamim. 2022. “Predicting ARP Spoofing with Machine Learning.” In *2022 International Conference on Emerging Trends in Smart Technologies (ICETST)*, 1–6. IEEE.
40. Vajrobol, Vajratiya, O. M. Jothi, A. N. Jayanthi, S. Shitharth, and Surbhi Bhatia. 2024. “Identify Spoofing Attacks in Internet of Things (IoT) Environments Using Machine Learning Algorithms.” *Journal of High Speed Networks* 31 (1): 61–70.

41. Zhou, Xin'an, Qingyang Zhu, Zhiliang Wang, Xia Yin, and Xingang Shi. 2024. "Untangling the Knot: Breaking Access Control in Home Wireless Mesh Networks." In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2072–86.