A Neutrosophic Model for Ranking Technical Sol utions for Three Types of ARP Attacks in SDN Architecture

Ehab R. Mohamed*, Heba M. Mansour, Osama M. El-Komy
Faculty of Computer and Informatics, Zagazig University, Zagazig, Egypt
Email: Ehab.rushdy@gmail.com ; Eng.hebamohsen.222@gmail.com ;osamaelkomy@yahoo.com

Abstract
Software-defined networks (SDN) have developed an understanding of the technological world in recent decades, which has led scholars to become interested in its problems. One of the primary issues facing SDN networks is security. We discovered that ARP assaults constitute a significant threat to SDN, so we provided in this survey the most recent solutions put forth to counteract these attacks, and rank the technical solutions based on the neutrosophic set. The neutrosophic set is used to deal with uncertain data in the evaluation process. The neutrosophic set is integrated with the TOPSIS method to obtain the rank of the proposed solutions. The TOPSIS method is used to give a rank of alternatives with specified criteria. This will make it easier for future researchers to identify and combat the three different forms of ARP attacks—ARP flooding assault, ARP spoofing attack, and ARP broadcasting attack. Prior to that, we went into more detail on SDN networks, including their design and, in particular, the shortcomings of the ARP protocol. Since SDN focuses on separating the controller plane from the data plane and centralizing the controller, it differs significantly from traditional networks and has stirred considerable controversy in the networking industry. Due to the fact that SDN is software-based, it offers greater flexibility, scalability, programmatic management, and control. The decoupling of the control and forwarding planes also enables SDN to connect to applications via application programming interfaces (APIs), supporting application security and performance and resulting in a scalable and dynamic network architecture. Contrarily, because traditional networks are hardware-based, they must use fixed functions and specialized hardware and devices to control the network. As a result, scaling traditional networks requires purchasing new hardware, which is a common issue. The SDN network architecture and its properties, as well as the most significant issues—particularly the three different types of ARP attacks that affect SDN—are covered in some sections of this research. These sections also discuss the best current remedies for these issues and outline the ongoing work that will eventually result in an ideal SDN network architecture free of significant security issues.

Keywords: ARP protocol; Neutrosophic theory; ARP attacks; Application programming interfaces; Neutrosophic Set; Uncertainty.

1. Introduction

Networks have emerged as one of the key distinguishing characteristics of the era of rapid development and expansion in the use of the Internet. Researchers' understanding of networks has also deepened, and most recently, SDN networks have accompanied this tremendous development. However, the more development and changes, the greater the complexity and issues, especially the problem of SDN security, in which researchers have tried their best but have been unsuccessful. Three different types of ARP assaults in SDN networks are the topic of this survey, and we provide these issues together with the most effective fixes. Additionally, we'll give a robust poll to compare and analyze the most recent and effective ideas offered by various scholars to find solutions to these issues based on several performance indicators, such as the time it takes to identify and mitigate an attack, as well as other standards based on whether or not to fight against the three different forms of attacks. This paper proposed the neutrosophic set methodology to rank set of tecgncial solutions for three types of ARP attacks.

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

It is challenging to provide a succinct description of a phenomenon in cognitive research due to the complexity of the data that must be analyzed. The data might be used to support or refute a claim. Additionally, ambiguous data may occur. Zadeh offered fuzzy sets as a solution, and the idea of membership function as a means of representing information ambiguity. Atanassov later proposed fuzzy sets with intuition, which provide a more comprehensive description of data thanks to its non-membership function. However, they are helpless in the face of the uncertainty. Smarandache then generalized the sets that had been offered before with neutrosophic sets (NSs), a potent tool for dealing with this issue. The degrees of truth, indeterminacy, and falsehood in each of its three components are established individually. Because they help decision-makers deal with uncertainty and inconsistency, NSs are an excellent reflection of the actual world.

However, this technology may be difficult to use because of the philosophical nature of NSs. Wang et al. created an implementation of NS (called SVNS) to go around this problem. When it comes to resolving decisions, SVNSs are more universal and applicable. This paper used the neutrosophic set with the TOPSIS method.

The term "multi-criteria decision making" (MCDM) refers to a method for deciding amongst alternatives that meet several, and often competing requirements. The TOPSIS approach is now considered to be among the best MCDM techniques available. The distances from the positive ideal solution (PIS) and the negative ideal solution (NIS) may be used to rank the options. If an option is closer to the PIS, it will be rated higher, and if it is closer to the NIS, it will be ranked less. The TOPSIS technique is quick, adaptable, and simple to use.

It has been used in a wide range of situations and contexts, including those involving fuzzy data and intuitionistic data. In addition, some scholars have used TOPSIS to address dilemmas of decision-making in a neutrosophic setting. For the purpose of resolving multiple criterion group decision-making issues using SVNS, Biswas et al. created an expanded TOPSIS technique. To handle linguistic data with a single value, you learned TOPSIS.

In this paper, we show in detail SDN features and challenges. This section, ARP protocol, ARP vulnerabilities, and ARP handing example in SDN in Section 2, are all covered in detail in Section 3. The related work that concluded only three surveys discussed solutions against ARP in SDN architecture; we documented in Section 4 the latest solutions of various researchers and compared these different solutions; in Section 5, we discussed, analyzed, and compared the performance metrics of some solutions supported with figures and tables; section 5 presented the steps of the proposed method; section 6 presented the solutions and ranked the set of solutions; and finally, in Section 8, we provided a brief summary of this survey and listed any potential future research projects.

1.1. SDN background and challenges

We mentioned at this section the main features of SDN that may make the SDN network immune from some kinds of attacks, or may put the network in a vulnerable position against attacks, we have presented SDN architecture at Fig. 1, which shows that SDN consists of three basic layers, which are the control layer, the infrastructure (device) layer and the application layer, at the application layer actions and request resources are transmitted to the controller through the APIs to implement various functions related to security, analytics, management, load balancing and quality of service management, after the instructions from the application layer reach the controller, the controller collects the information and relevant statistics from the different devices and analyzes and calculates this information and sends it back to the application layer, the infrastructure layer can make forwarding and data processing tasks, SDN network devices are all placed at the infrastructure layer, as author in [1] mentioned, the SDN network devices make a simple decision of what to do with incoming traffic (frames or packets) according to instructions programmed by their SDN controller, the controller can communicate with the application layer through north bound API, and can communicate with the device layer through south bound. The main advantages of SDN are that decoupling control from the forwarding process gives a global view of the whole network, which provides more control of the network behavior via a programmable controller. The concept of virtualization in SDN helps reduce OPEX and CAPEX, which makes SDN support flexibility and scalability. And the main disadvantage of SDN is SDN security issues, especially ARP attacks, which may affect both the hardware level and the software level, as in [2]. In the hardware level, the attack targets the controller and the forwarding devices, and in the software level, the attack is related to the security of the OpenFlow protocol that affects the performance of the network. Security is considered one of the biggest challenges facing SDN. Despite the many benefits of an SDN network, it suffers from some vulnerabilities, and its main weakness is due to the centralization of the controller, which is the focus of attention of researchers in this field because it is the heart of the network and

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

securing it is a necessary task to obtain a network free of defects to a large extent. We focused on the latest solutions for three main types of ARP attacks, like ARP flooding attacks, ARP spoofing attacks, and ARP broadcasting attacks, that may lead to the most important issues, especially those related to the controller, such as DDOS attacks and other serious attacks like MITM, etc. These attacks may exploit one or more of the three layers [3].

Data plane attacks: The network traffic can be disturbed by access points or switches, which results in malicious users launching denial of service (DoS) attacks that can result in network failure or disruption. Or the attacker compromises the network element to redirect the flow of traffic, followed by exploring further vulnerabilities through methods such as eavesdropping.

Control plane attacks: an intruder compromises the controller by constructing false data on the network and then simultaneously initiates other attacks on the network.

***Application-plane attacks*:** the attacker could gain access (sometimes with high privilege) to any SDN application and be successful in accomplishing illegal actions over the network. Exploitation of application vulnerabilities can cause adverse effects such as malfunction of the network, disruption of service, or eavesdropping on data.

1.2. SDN architecture

According to Fig. 1, the infrastructure layer, the control layer, and the application layer are the three fundamental layers of the SDN architecture. The centralization of the control process by severing the controller plane from the data plane is the fundamental idea behind SDN. SDN architecture consists of three layers [2]: the infrastructure layer, which consists of physical and virtual devices such as switches, routers, and other devices that become the network structure; the control layer, which is the brain of SDN and manages the policies and controls the flow of traffic across the whole network; and the application layer, which contains the network applications or functions like load balancing, firewalls, and intrusion detection systems. The application layer communicates with the control layer using northbound APIs, and the control layer communicates with the data plane using southbound APIs.

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

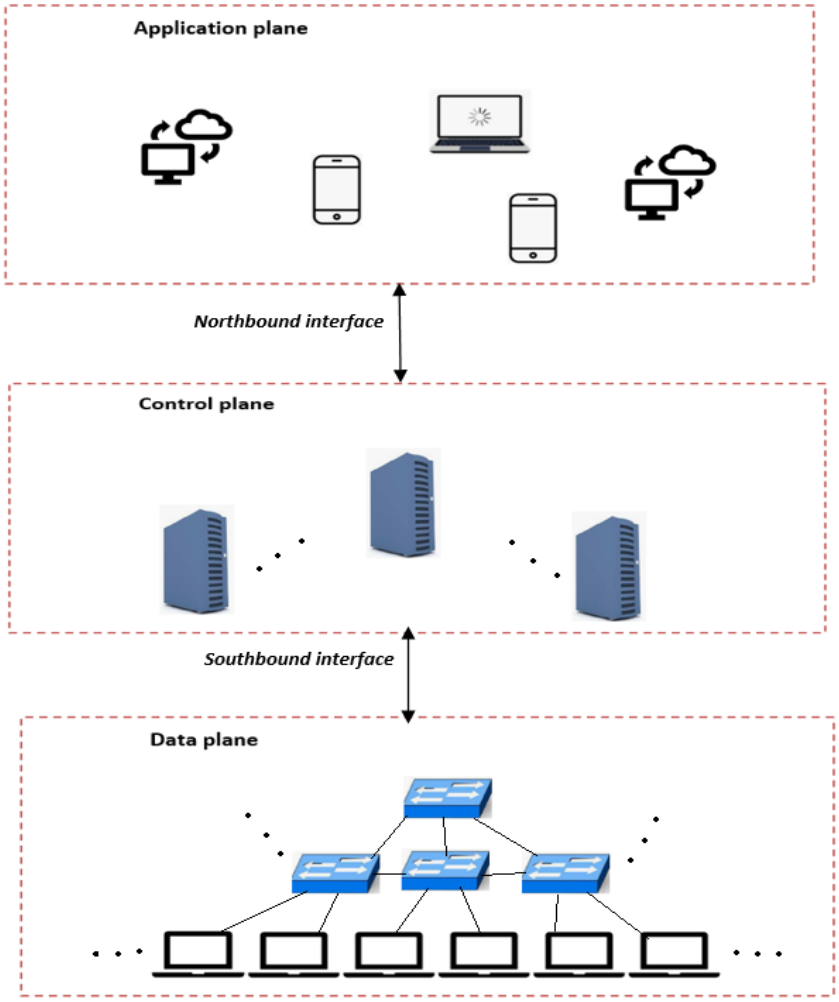

Figure 1: SDN architecture.

Four main features of SDN add to its benefits; these strengths make SDN a great option for large networking and new networking functions or innovations. Good network design and SDN together can give network engineers almost unlimited creative freedom in developing new networking functionality. The strengths can be listed briefly as below:

***Programmability*:** SDN enables the control of network behavior by isolated software from the networking hardware that provides the physical connection. As a result, network operators can directly program network features, change the behavior of their networks to fulfill the required services, and configure network resources quickly and easily through automated SDN functions.

agility: abstracting control from forwarding lets administrators dynamically adjust network configuration to meet changing requirements. Services and applications running on SDN technology are abstracted from the underlying technologies and hardware that provide physical connectivity for network control.

***Centralization*:** A centralized controller in software-based SDN maintains a global view of the network, which appears to applications and policy engines as a single, logical switch, which enables intelligent control and management of network resources.

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

Openness: The open APIs support a wide range of applications, including cloud orchestration, OSS/BSS, SaaS, and business-critical networked apps. In addition, intelligent software can control hardware from multiple vendors with open programmatic interfaces like OpenFlow. We used these factors as a criterion in the neutrosophic TOPSIS method.

The main advantage of SDN technology is the ability for network operators to write programs that utilize SDN APIs and give applications control over network behavior. SDN allows users to develop network-aware applications, intelligently monitor network conditions, and automatically adapt the network configuration as needed. So moving to software-defined networking is an essential step for keeping up with emerging trends and technologies, such as the internet of things, cloud computing, and other technologies that need this architecture of the network.

2 ARP Protocol Background and Vulenerabilities

SDN hosts use a variety of protocols to finish their communication process, ARP being one of them. Its primary use in a local area network is to convert a dynamic IP address to a physical machine address (Mac address). An end node's Mac address can be determined from its IP address using the ARP protocol. An ARP request is broadcast to do this. All devices receive these requests from the controller, and the one in question replies with its Mac address in an ARP reply. Every host has an ARP cache, which is a collection of entries created either statically or dynamically when an IP address is associated with a Mac address. Tab. 1 shows the packet format of ARP. This packet contains Sender Hardware Address (SHA) and Target Hardware Address (THA), which are the Mac addresses of the sender and the receiver, and also includes Sender Protocol Address (SPA) and Target Protocol Address (TPA), which are the addresses of the sender and the target. As in Fig. 2, host A must broadcast an ARP request with 10.0.0.2 (TPA), and host B must send a unicast packet to host A with its mac address 00:00:00:00:00:02 (THA) to begin their communication.

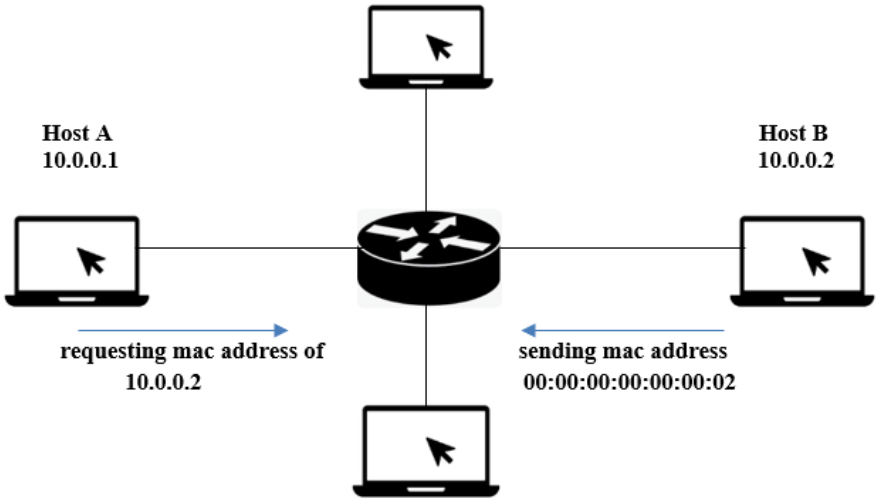

Figure 2: How ARP protocol work.

Table 1: ARP packet format.

Hardware Type (HTYPE) 16-bit	Protocol Type (PTYPE) 16-bit

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

Hardware Length (HLEN)	Protocol Length (PLEN)	Operational request (1), reply (2)
Sender Hardware Address (SHA)		
Sender Protocol Address (SPA)		
Target Hardware Address (THA)		
Target Protocol Address (TPA)		

2.1. ARP handling in SDN

We explained ARP handling in SDN, as in Fig. 3. If we suppose that H1 needs to know the mac address of H3, it sends an ARP request with sender address information (SPA = 10.0.0.1, SHA = 00:00:00:00:00:01, TPA = 10.0.0.3, THA = 00:00:00:00:00:00) with a dummy value of the unknown target mac address. After receiving the packet, S1 forwards it to the controller, which instructs S1 to flood the packet. When the packet reaches H3, it sends an ARP reply to S2, which then sends the packet to the controller. The controller installs a flow entry on S2 and hence sends the packet, which has address information for H3, to H1.

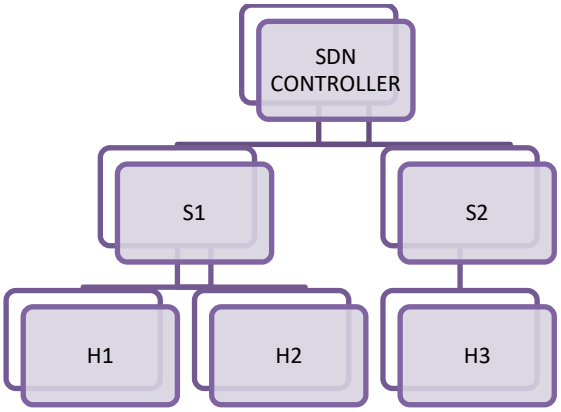

Figure 3: SDN handling ARP

2.2. ARP vulenerabilities

The ARP protocol has many vulnerabilities, especially security issues, but this protocol also has some vulnerabilities, and in this work, we'll take a close look to pinpoint these difficulties and try to find innovative solutions to them. ARP flooding has become an imperious problem in SDN networks, which makes huge traffic from one port by sending a large number of request packets, causing traffic overhead and overwhelming the controller's performance, which may lead to the breakdown of the whole network [3]. ARP spoofing occurs because the ARP protocol lacks authentication and privacy, and SDN has been the target of major assaults. The attacker takes advantage of the lack of authentication and spoofs ARP requests or replies to contaminate the controller's or other hosts' ARP caches, which opens the door for other attacks like MITM and DOS. This attack tries to transmit an ARP request with an unknown destination, an ARP packet with forged address information, fake ether header information, a Mac address that doesn't match an IP address in the host table, or an ARP packet with a mismatched Mac address and IP address. This overwhelms the controller and taints the ARP cache, rendering the entire network unusable. The network is now vulnerable to DDOS or MITM attacks [4]. ARP broadcasting becomes an issue in SDN when the attacker sends ARP replies to all Mac addresses [4], and the IP of

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

the sender is not equal to the destination IP. This type of attack is dangerous for the whole network and may cause DDOS. Various current solutions for ARP attacks include ARP authentication, operating system patching, dynamic ARP inspection (DAI), ARP mitigation tools, and static ARP mappings.

3 Related Work

There are many solutions for detecting and mitigating ARP attacks in SDN; thus, we briefly described and mentioned these solutions to help other researchers in this field find all modern solutions in one powerful survey. There are only three literature surveys:

In [4], Shah Z et al. proposed a detailed survey on various solutions to mitigate ARP Cache Poisoning attacks in SDN. In this survey, various solutions are classified into three categories: Flow graph-based solutions; Traffic patterns-based solutions; IP-MAC Address bindings-based solutions All these solutions are critically evaluated in terms of their working principles, advantages, and shortcomings. Another important feature of this survey is to compare various solutions with respect to different performance metrics, e.g., attack detection time, ARP response time, calculation of delay at the controller, etc.

In [5], Rohatgi V etal, first gives an overview of ARP and its working. Then discusses ARP Spoofing and its various attacks. Finally, provides the comparison of different detection and mitigation techniques along with their pros and cons to control ARP attacks.

In survey [6], Meghana J etal, present adeep investigation of existing ARP cache poisoning or ARP spoofing solutions. We first introduce Address resolution protocol and its operation. We then dwell on ARP spoofing, with an emphasis on its impact on network security. Then we provide an overview on existing solutions to detect and mitigate ARP spoofing in both traditional networking and software-defined networking (SDN) platforms. Finally, we compare the techniques discussed and conclude that SDN based solutions are more effective in detecting and eliminating any kind of ARP spoofing attack.

4 The Latest Solutions Proposed Against ARP Attacks In SDN

As in Tab. 2, we listed the latest solutions against three types of ARP attacks that threaten SDN architecture and indicated the paper name and the year of publication for each solution to facilitate work for future researchers in this field. The application years range from 2022 to 2010. In this section, we briefly mentioned these solutions and made a comparison among them based on some criteria, such as the three types of ARP attacks on SDN and ARP req/rep analyzing, and we discussed some metrics of these solutions like response time to detect and mitigate the attack, CPU utilization, bandwidth, and accuracy.

Table 2: the latest solutions against ARP attack in SDN architecture.

Paper name	Authors	Puplished-year
Implementing an intrusion detection and prevention system using Software-Defined Networking: Defending against ARP spoofing attacks and Blacklisted MAC Addresses	Girdler T etal, [7]	2021
Ascertain the efficient machine learning approach to detect different ARP attacks	Ahuja N etal, [8]	2022
Scalable and secure SDN based ethernet architecture by suppressing broadcast traffic	Munther M etal, [9]	2022
EFFICIENT MECHANISM FOR SECURING SOFTWARE DEFINED NETWORK AGAINST ARP SPOOFING ATTACK	KHALID H etal, [10]	2019
Research on An Approach of ARP Flooding Suppression in Multi-Controller SDN Networks	Du J etal, [11]	2021
Detection and Mitigation of ARP Poisoning Attack in Software Defined Network	Saritakumar etal,[12]	2021
Counteracting UDP Flooding Attacks in SDN	Wei H etal, [13]	2020

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

An Extendable Software Architecture for Mitigating ARP Spoofing-Based Attacks in SDN Data Plane Layer	Buzura S etal, [14]	2022
The detection and prevention for ARP Spoofing based on Snort	Hou X etal, [15]	2010
Distributed Responder ARP: Using SDN to Re-Engineer ARP from within the Network	Matties M etal, [16]	2017
A Technique for a Software-Defined and Network-based ARP Spoof Detection and Mitigation	Balagopal D etal,[17]	2018
OrchSec: An Orchestrator-Based Architecture For Enhancing Network-Security Using Network Monitoring And SDN Control Functions	Zaalouk A etal,[18]	2014
FICUR: Employing SDN Programmability to Secure ARP	Nehra A etal,[19]	2017
A novel mechanism to handle address spoofing attacks in SDN based IoT	Aldabbas H etal,[20]	2021
Denial of ARP spoofing in SDN and NFV enabled cloud-fog-edge platforms	Rangisetti A etal,[21]	2021
E2BaSeP: Efficient Bayes Based Security Protocol Against ARP Spoofing Attacks in SDN Architectures	Tchendji V etal,[22]	2020
RTNSS: a routing trace-based network security system for preventing ARP spoofing attacks	Moon D etal, [23]	2014

The following was contributed by Girdler T et al. in [7] SDN-based Intrusion Detection and Prevention System (IDPS) to protect against Blacklisted Media Access Control (MAC) Addresses and Address Resolution Protocol (ARP) Spoofing: (1) POX is the SDN controller used, and it implements the OpenFlow protocol for communication with an OpenvSwitch (OvS) switch. This IDPS protects against ARP spoofing attacks and Blacklisted MAC Addresses. (2) Techniques for carrying out the attacks and modifying the IDPS were developed. These are connected to an expert library that also verifies user input. The POX controller is improved by creating a module that logs attempted assaults and counteracts them by configuring flow rules on the OvS switch. (3) Various scenarios were used to comprehensively evaluate the IDPS against its design objectives. ARP Request Attack, ARP Reply Attack, ARP Reply Destination Attack, and Blacklisted MAC Address were the four types of experiments that were used in this. As a POX extension, the IDPS watches for these Packet-in events. Following the event's acceptance, one of four distinct processes—ARP Request Spoofing, ARP Reply Spoofing, ARP Reply Destination Spoofing, and Blacklisted MAC Addresses—handles its processing.

According to Ahuja N et al. in [8], "This paper presents the classification of benign network traffic from ARP poisoning and ARP Flooding attacks using machine learning (ML) techniques." The SDN controller creates a Python application using Mininet that gathers and logs the characteristics needed to identify the assault into a file known as a traffic dataset. The ML model is trained on this dataset in order to identify attacks. With an accuracy rating of 99.73%, the hybrid Convolution Neural Network-Long Short Term Memory (CNNLSTM) model outperforms all other machine learning (ML) models. and put forth an improved dataset with a strong characteristic set for ARP assault identification. There are two parts to the suggested solution. The dataset is created by the first unit, which extracts the important features and writes them into a comma-separated values (CSV) file. The other unit sets up the machine learning (ML) model and trains it using the suggested dataset to classify the traffic. The following is a brief summary of the research contribution: A mininet emulator is used to produce an SDN traffic dataset for an ARP-based MITM attack. To classify the traffic into one of the three categories (Benign, ARP Poison assault, or ARP Flooding attack), several ML algorithms are applied. The proposed dataset is used to train the ML algorithm, which is then implemented as an application at the controller to classify the traffic, and the analysis of the important aspects in the dataset using the ML model is used to detect attacks on hosts was put up by Munther M et al. in [9]. The three phases of the multistage security algorithm each integrate unique analysis to determine the status or behavior of the packet and respond appropriately based on that

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

condition. According to the performance assessment, the proposed solution's true positive ratio for attack detection is 57.14% for the first stage, 66.66% for the second stage, and in some situations, 100% for the final step. In this research, a floodless and secure technique is proposed that can both suppress broadcast traffic and defend the SDN network from ARP-based assaults. Three security phases of the security algorithm look at how the incoming ARP packets behave. The source MAC address is examined in the second stage, while the source IP address is examined in the first stage. The third stage involves examining the destination IP address.

KHALID H et al.'s suggestion in [10], This article examines ARP spoofing attacks and provides a thorough analysis of the current solutions, whether in conventional or SDN systems. In order to avoid ARP spoofing, a simple, quick, reliable, and effective approach that does not require any additional hardware or software has been proposed. The SDN controller has been enhanced in this work by a module that examines each ARP packet in the network to look for potential faked packets and stop them. Du J et al. presented a two-stage "storing-blocking" suppression technique in [11] for multi-controller SDN networks to reduce ARP flooding. It uses the strong consistency technique to synchronize ARP suppression entries between controllers and cache the ARP response messages as ARP suppression entries. The mechanism stops the flooding of ARP request messages after an ARP request message matches the ARP suppression entries, which can accomplish the suppression of ARP flooding purpose. We develop a two-stage "storing blocking" ARP suppression algorithm and suggest an ARP flooding suppression approach in multi-controller SDN networks to address synchronization and ARP flooding issues. After obtaining the initial ARP message, the algorithm creates an entry in the ARP request suppression table and synchronizes with other controller nodes using a strong consistency algorithm. At the same time, the controller constantly monitors the ARP request messages. If an entry from the table matches an entry from the controller, it signals that the ARP request message is invalid and that ARP flooding should be prevented.

To improve and resolve the issue of ARP poisoning assaults, Saritakumar N et al. presented a mechanism in [12]. The unauthenticated and stateless features of ARP are the two most noticeable restrictions on ARP assaults. The ARP packet, IP-MAC address bindings, and count value are the foundations for the detection. The controller begins dynamically allocating MAC addresses for each IP address in the network once the topology has been established. The controller stores the newly formed MAC address in the MAC_to_PORT database. During an attack, the attacker spoofs arriving packets and copies its own MAC address. However, the controller is not aware of whether the received packets' MAC addresses are genuine or duplicates. ARP poisoning attacks are created using dsniff. The attacker sends ARP reply packets repeatedly to establish a connection between the target and client, giving the controller the impression that the connection is valid. The attacker will use an ARP packet spoof to transmit 1000 packets per second, and since the switches are constantly receiving many duplicate packets, a threshold is set to control the attack. Thus, the threshold value is set at 100 to detect packets in the millisecond range. If the packet count process flow is raised, the controller discards the packets, and The module would send a warning of an ARP poisoning attack if the MAC address in the packet and the record in the list were different. Additionally, this module has the ability to instruct switches to send all incoming packets to the controller when this MAC address is detected. The action is taken by the controller, who halts the attacker's performance. Based on the ARP packet count value and the IP-MAC address matching the preset IP-MAC address, the ARP poisoning attack is recognized.

Wei H et al. offered two simple countermeasures in [13]. In order to achieve a high level of security, the first technique occasionally forgoes address-resolution protocol (ARP) requests, yet it can engage the analysis mechanism while still keeping track of the number of packets currently being sent. While it is occasionally necessary to sacrifice packets in the second method's attack before beginning to defend, the network state detection can stop normal packets from being sacrificed. Attacks from the afflicted port are directly prevented when a network attack is banned, leaving other ports unaffected.

In [14], Buzura S et al. apply an expandable and adaptable software architecture to the data plane layer of the SDN, which is only implemented in the data plane. The goal of the solution is to accelerate response times and increase attack mitigation efficiency. As a result, quicker response times increase network dependability by instantly stopping attackers. The contributions made in the current study are outlined below since attacks can be created using a range of tools and in networks with various traffic patterns. The contributions made to this study are listed below:

Putting in place a system that recognizes and counters ARP spoofing-based attacks in an SDN's data plane layer The general design of the SDN solution can be customized to support various operating environments (such as OpenVSwitch-based, npcap-based, etc.) and integrate several technologies at the software layer level. rigorous

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

implementation in the SDN data plane congested areas. In addition, the study shows how the solution affects network throughput, whether or not a MitM attack is occurring. An examination of the attacker's activity and the design of a distributed ARP spoofing-based attack, due to the fact that a defense against this distributed attack is offered, also introduces a novelty in attack management compared to similar studies. A novel benchmark suggestion that adds to the body of knowledge by offering fresh testing settings for comparative studies in the future.

Hou X et al. [15] implemented an ARP detection module to the Snort preprocessing plug-ins. Results indicate that doing so can give the Snort sniffer immunity and improve the timeliness and accuracy of the attacker's location. Static binding the host's IP and MAC addresses in the gateway will prevent the gateway from refreshing its cache list even if it receives forwarded ARP reply packets, which is a fairly simple defense against cheating gateways. We can alter the original ARP spoofing plug-ins to trick the host of the internal network and add an ARP inspection module to defend against and identify the spoofing.

The design of distributed responder ARP (DR-ARP), a software-defined networking (SDN)-enabled improvement of the conventional address resolution protocol (ARP), is presented by Matties M et al. in [16]. By exercising far more control over both what actually provides the ARP service as well as how ARP traffic flows over the network, it is designed to increase network security and performance. illustrate how to re-engineer ARP, one of the most fundamental end-host protocols, without changing the end-host's software. The DR-ARP responder service, which keeps an ARP service table that contains both the traditional ARP table and meta-data on all ARP transactions, receives all ARP request traffic from the SDN network and directs it there. The SDN network then forwards the properly structured ARP answers from the responder to the appropriate asking hosts. The service may be delivered by a single responder or by a group of cooperative responders, depending on the load and environment.

A limited ARP Cache Recovery module that recognizes the poisoned ARP packets and takes the appropriate steps to block the attacking device was presented by Balagopal D et al. in [17]. In addition to sending out an alarm and stopping the attacker, it restores the broken connection between the victim and other devices in order to preserve network functionality. The application must periodically examine the ARP traffic passing through the network. Traffic is redirected to the Controller if an abnormally high number of ARP packets are observed. The Controller makes an effort to find spoofing in the packets. If spoofing is found, a switch entry is added to prevent the malicious source's ARP and IP packets for a set amount of time. Capturing the IP and MAC of the attacker, the victim, and the spoofing host is required for spoofing identification. An Orchestrator-SDN control-based architecture that makes use of and works to construct security applications is proposed by Zaalouk A et al. in [18]. attacks that are impossible to identify without access to every packet in the network. Sampling alone won't be able to stop these attacks; hence, packets associated with this traffic flow should be sent to the orchestrator for additional examination (e.g., ARP Cache Poisoning, ARP Spoofing). attacks that don't always need to have access to every packet. Sampling can be used to find large-flow events and then tell the Orchestrator to take further measures (such as DoS, DDoS, or DNS Amplification attacks) in order to reduce overhead. In order to coordinate the functions of the network monitor and the SDN Controllers, the Orchestrator communicates with the network monitor for updates on significant events and network status. Based on the knowledge obtained from the network monitor, the Orchestrator directs the controller to watch specific traffic patterns and redirect traffic belonging to these patterns back to the Orchestrator for further inspection. As a result, the Orchestrator is regarded as the architecture's central nervous system.

An innovative technique for the verification and detection of ARP-based assaults was proposed by Nehra A et al. in [19] called Traffic Pattern Based Solution to ARP-Related Threats (FICUR). The suggested approach extends the SDN controller with a module that collects the necessary network parameters. To confirm and find the attacks, this module additionally analyzes these factors. Three methods were created by Ficur to identify ARP Poisoning and ARP flooding. The first algorithm aids in the development of a data structure required for both attacks' detection. That list is used by both Algorithms 2 and 3, in particular ISARPPOISON and ISARPFLOOD, to find the assault. The log Detail file is deleted by the REMOVE function. ARP entries are examined by ISARPPOISON for the same source IP address and distinct MAC addresses. The ISARPFLOOD function's GETOBSERVEDTHRESHOLD function returns the average of historical observed thresholds. This function

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

analyzes the probability of an ARP flood if the number of ARP requests for which no IP communication exists exceeds 'tsld. If so, the threat port should be logically blocked in order to filter ARP traffic.

Aldabbas H et al. developed a secure SDN-based IoT architecture in [20] to manage and reduce ARP spoofing threats by establishing a new machine close to the SDN controller to handle address resolution queries. ARP traffic is transferred to this new machine so that address spoofing threats can be examined. A system that automatically recognizes ARP assaults in an IoT and mitigates the attacks using a single computer running the custom code is offered to address the address spoofing issue. The Controller ARP Table (CAT) and network topology data are both kept up-to-date by the SDN controller. The controller and the dynamic host configuration protocol (DHCP) are used to obtain the network topology data. Together with the controller, this module gathers topology information and ARP queries to identify probable attack scenarios. Specialized techniques are used to analyze the ARP data. This novel system is made up of three primary parts. The first elements of the suggested approach retrieve network topology data from the entire network. The establishment of flow rules on network devices is covered in the second section in order to force all ARP traffic to go through a specific computer. ARP spoofing mitigation and traffic analysis modules are found in the third component.

Rangisetti A et al. in [21] developed a Denial of ARP Spoofing (D-ARPSpoof) solution to prevent ARP spoofing in SDN and NFV enabled Cloud-Fog-Edge platforms. D-ARPSpoof, in contrast to current IDS and anomaly detection systems, stops ARP spoofing attacks by lowering the burden on the centralized controllers and OpenFlow switches. To ensure that D-ARPSpoof only receives authentic ARP packets from the flow table, host_certification proactive flow rules are used. Continuously monitoring for ARP_REQ and ARP_REP messages, the D-ARPSpoof module obtains the attachment point (SWID, port) of the destination from the Host_Cert map when an ARP_REQ arrives and the destination host (IP) is verified. ARP_REP messages cannot be spoof by any malicious hosts in the network since they are authenticated using proactive flow rules set up by the switch's Host_Certification module. D-ARPSpoof also manages DDoS assaults, in which numerous hosts connected to multiple switches attempt to flood the network with ARP_REQ messages in order to waste controller resources and network bandwidth.

In [22], Tchendji V et al. proposed the Efficient Bayes Based Security Protocol (E2BaSeP), which employs a Bayes-based algorithm to identify potential attackers. Both networks with dynamic and static addressing can use this method. Describe a method for securely protecting SDN-based virtual networks from ARP spoofing attacks. Thus, we suggested the E2BaSeP security protocol. This protocol is based on updating the Global ARP Cache (GAC) after a network reallocation of IP addresses. It uses a Bayes-based detection method to find intruders. This approach makes advantage of the IP-MAC pairs acquired and stored in the GAC as help for identifying attackers. The E2BaSeP solution operates in both dynamic and static addressing networks and assures GAC consistency. A new network security solution based on routing trace that may defend the internal network from ARP spoofing attacks was proposed by Moon D et al. in [23]. A server and an agent make up the planned RTNSS. Agent protection (AP), agent manager (AM), and agent detection (AD) modules make up the agent. When a client updates its ARP cache table, the AD Module does a routing trace of the modified <IP and MAC and compares it with the information from the ARP cache list before sending the information to the AM. When the information for the <IP, MAC> pair in the ARP table changes, the system cycles through the ARP table and detects ARP spoofing attacks through a routing trace. By sending two units of the ICMP packet with an altered TTL value as the reply message, it is possible to confirm network movement paths through a routing trace. Additionally, the suggested system employs the idea of an increase of "1" in the number of HOPs to enable the transmission of the ICMP packet to the hosts and the gateway on the same Ethernet with 1 HOP each time it goes via a router.

In the last, these various solutions are used as alternatives in the neutrosophic TOPSIS method. There are 13 solutions used. These solutions are ranked and select best one based on the single-valued neutrosophic TOPSIS method.

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

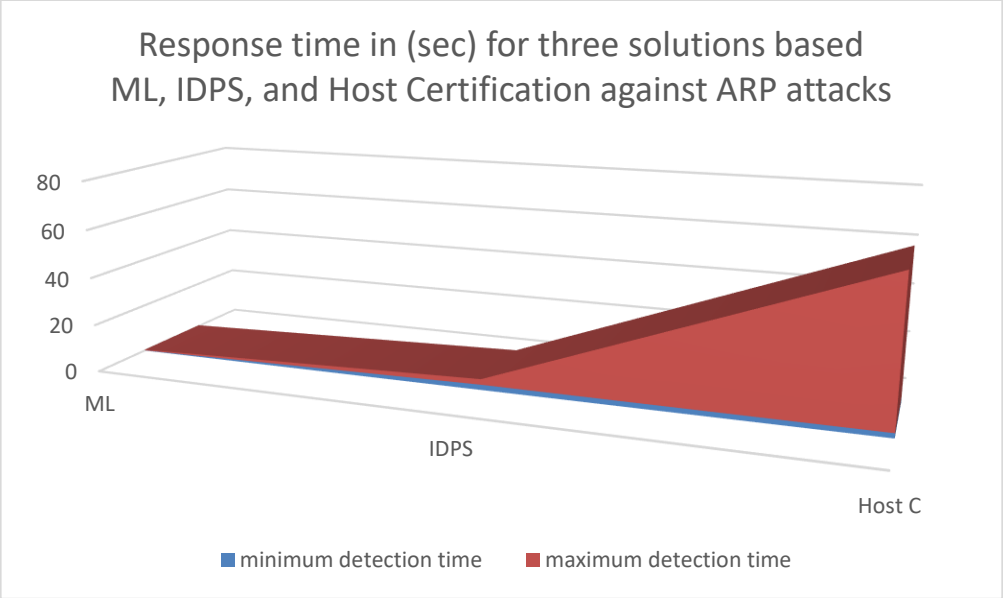

Figure 4: Comparing response time for three existing solutions.

5 Metrics & Discussion

The detection rates were 100% as in [7], much like in Fig. 4. The average additional response time is 02.01465, the minimum is 02.00890, and the maximum is 02.03694 in the measurement. This difference was found to be quite minimal, demonstrating that the intrusion prevention capabilities did not significantly increase the IDS's overhead. We notice that the timeframes for IDPS mitigation are essentially the same as those for the prior trial. The maximum packet response increased because, similar to other trials, a number of packets with longer than usual response times were identified. The extra time required to mitigate the assault by the intrusion prevention system is: Avg. = 0.001092, Min. = 0.000674, and Max. = 0.002004. The ML-based approach in [8] achieved an accuracy score of 99.73%, and it was noted that it was not noteworthy. The detection times for poison attacks and ARP flooding are roughly 0.1 sec and 0.3 sec, respectively. The proposed method in [9] has a true positive ratio for attack detection that is 57.14% for the first stage, 66.66% for the second stage, and in some situations, 100% for the final step. Spoofing case time (first attempt) is 300 seconds, while storm case time (first attempt) is 180 seconds. According to KHALID H. et al.'s discussion in [10], the faked packet's detection time is approximately 6 ms for the ARP request and 5 ms for the ARP reply. CPU consumption for the Pox controller was 3.4%. By extending the controller with the mitigation technique, the CPU load rose to 4.2%. On the CPU, mitigation functionality comes at a cost. Sending every ARP packet to the controller in order to check the infected packet has the consequence of increasing CPU load. the frequency of flooding the controller with messages from nearby switches (20 times). The average time interval in [11] is 2.09ms. According to Fig. 5, the number of ARP request messages the controller has received from the first to the last is used to indirectly determine the flooding time. The two controllers in [12] that the detection algorithm was installed on showed two distinct behaviors. A comparison of the CPU usage of the POX and RYU controllers revealed that the RYU controller uses 15% less space overall. CPU Utilization (in%) for RYU is 17.3 during an attack and 6.3 following mitigation. CPU Utilization (in%) for POX is 20.2 during an attack and 17.3 following mitigation. Compared to the RYU controller, POX's execution time is 9% longer. Execution times for POX and RYU are 0.0023 and 0.0011, respectively, for hosts with a total of 20. In [13], Ryu 3.7 was used to simulate the controller, while Mininet was utilized to simulate the SDN OpenFlow switch 1.3. When the network was attacked without protection, the bandwidth was reduced by 7.7 GBit/sec. However, when the network was protected by the suggested defense mechanism, the bandwidth was only reduced from the baseline condition by 0.4 Gbits/sec. 1.9% of the CPU is used by the Original status, 14.4% by the Attack (570 packets per second), and 2.1% by the Defense. Attack (570 packets per second): 11.8 ms; Attack (original status): 8.05 ms; and Defense: 8.53 ms.

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

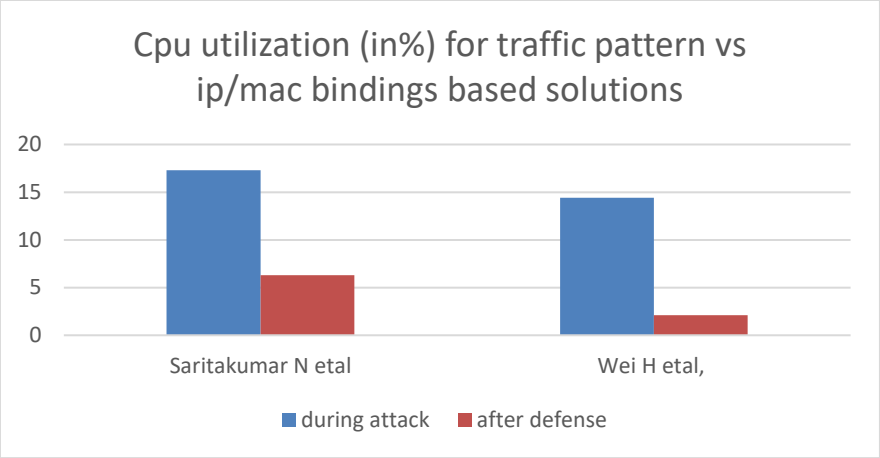

Figure 5: Comparing CPU utilization for two different solutions.

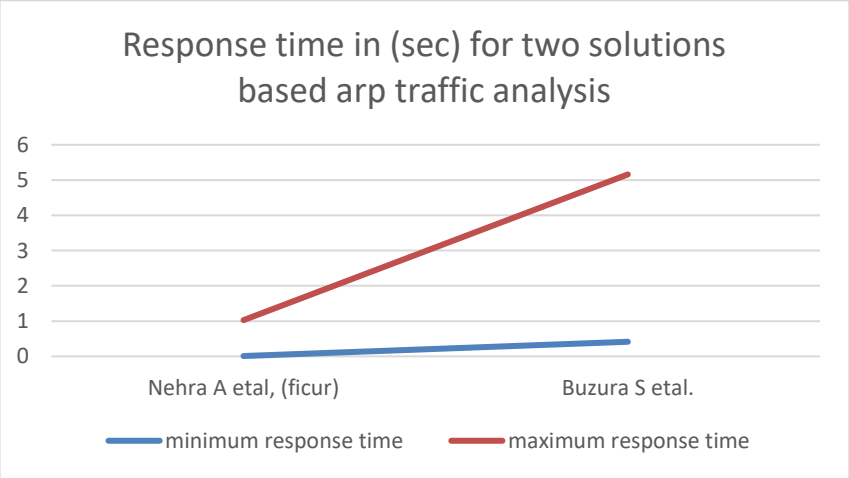

Figure 6: Comparing response time in line chart for two different solutions.

The author of [14], as shown in Fig. 6, observed the attack detection period throughout ten iterations in which various quantities of traffic were generated inside the network of twenty end devices where an attacker was executing the MitM attack. Average detection times ranged from 415.6 milliseconds for low traffic to 1503.2 milliseconds for medium traffic to 5615 milliseconds for high traffic. The time it took to detect the attack across ten iterations in which various amounts of traffic were produced within the network of 10 endpoints when a MitM attack was being conducted by an attacker Average detection times varied depending on the volume of traffic: for low traffic, they were 402.7 ms, for medium traffic, 1086.8 ms, and for heavy traffic, 3239.5 ms. The attacker performed the MitM attack between the end devices, and the attack detection period consisted of ten rounds where various levels of traffic were generated inside the network, which comprises three end devices connected to one switch. The average detection time ranged from 398.8 milliseconds for low traffic to 921.8 milliseconds for medium traffic and 1975.6 milliseconds for high traffic. Additionally, Nehra A et al, created a traffic pattern-based Solution to ARP-related threats (FICUR), and they were able to identify these attacks in under a second [19]. The detection times for an ARP poison and an ARP flood assault are 16056, 42840, and 422517, 1028160 microseconds, respectively. Both the mininet and openflow enable switches were used to collect all of these statistics. These figures represent the mean of various observations. According to [16], Matties M et al. observed a normal ARP service time of 1 ms in the test network. Under comparable network conditions, DR-ARP performance comes extremely close to regular ARP performance while not quite matching it. Approximately 4.5 ms, or roughly 80%, of the execution time for the DR-ARP responder code can be attributed to the libpcap packet capture function pcap_loop() capturing and processing the incoming packets. The performance of the ARP Cache

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

recovery module was captured by simulation results in [17] at three different times: before, during, and after attack detection and mitigation. Results indicate that there is no appreciable difference in host device latency during the pre- and post-attack phases. Regarding bandwidth, there were no appreciable differences between pre-attack and post-attack statistics.

According to [20] Aldabbas H et al., the throughput for the proposed strategies before the assault was 8.738 Mbps; it was 0 Mbps during the attack; it was 7.638 Mbps during the attack; and it was 6.5 Mbps after the attack. The ARP request interval is 2 seconds, the total test period is 300 seconds, the start time of genuine connections is 0 seconds, the start time of malicious connections is 60 seconds, and the average computation time for metrics is 1 minute [21]. Compared to other systems, which have an average precision of less than 70%, E2BaSeP attacker detection, as in [22], has an average precision of roughly 95%. Because only 10 hosts are aware of an IP address reallocation, the attacker's precision with these techniques is greater than 50%. The E2BaSeP solution compares the number of features activated by a virtual host to the predefined threshold number when the attack frequency is low. It was discovered that an ARP spoofing detection error did not occur in [23], the event of monitoring ARP spoofing with a monitoring cycle ranging from 1,000 to 300 ms, but that the number of ARP spoofing detection error occurrences increased as the monitoring cycle shortened to less than 300 ms. This study proved that the CPU share rose according to the monitoring cycle starting at less than 300 ms, and in particular, when the cycle was less than 100 ms, the CPU share climbed by a maximum of 15%. This investigation proved that the ARP spoofing detection error occurrence frequency and CPU share increased in the monitoring cycle by less than 300 milliseconds. The shorter the monitoring cycle, the more the ARP spoofing detection time rose. In an experimental setting, it was discovered that when the monitoring cycle of ARP spoofing is longer than 300 ms, the accuracy and stability of the suggested system can be assured.

Finally, as in Tab. 3, we analyzed the existing solutions based on three serious types of ARP attacks, such as ARP flooding attacks, ARP spoofing attacks, and ARP broadcasting attacks. These three main types of ARP attacks lead to other types of attacks like MITM, DDOS, etc. We discussed some metrics for these solutions in the previous section. And we compared the latest proposed solutions based on ARP flooding attacks, ARP spoofing attacks, ARP broadcasting attacks, ARP request attacks, and ARP reply attacks. This analysis enables future researchers to know the limitations of these solutions and solve them in their future work.

Table 3: the latest solutions categorizing based three types of ARP attacks

Authors	Attack area	ARP request attack	ARP reply attack
Girdler T etal, [7]	ARP spoofing attack & ARP broadcasting attack	yes	Yes
Ahuja N etal, [8]	ARP flooding attack & ARP spoofing attack & ARP broadcasting attack	yes	Yes
Munther M etal, [9]	ARP spoofing attack & ARP broadcasting attack	yes	Yes
KHALID H etal, [10]	ARP spoofing attack & ARP broadcasting attack	yes	Yes
Du J etal, [11]	ARP spoofing attack	yes	
Saritakumar etal,[12]	ARP flooding attack & ARP spoofing attack	yes	
Wei H etal, [13]	ARP flooding attack	yes	
Buzura S etal, [14]	ARP spoofing attack	yes	
Hou X etal, [15]	ARP spoofing attack	yes	
Matties M etal, [16]	ARP spoofing attack & ARP broadcasting attack	yes	Yes
Balagopal D etal,[17]	ARP flooding attack & ARP spoofing attack	yes	

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

Zaalouk A etal,[18]	ARP flooding attack & ARP spoofing attack & ARP broadcasting attack	yes	Yes
Nehra A etal,[19]	ARP flooding attack & ARP spoofing attack	yes	
Aldabbas H etal,[20]	ARP spoofing attack	yes	
Rangisetti A etal,[21]	ARP flooding attack & ARP spoofing attack & ARP broadcasting attack	yes	Yes
Tchendji V etal,[22]	ARP spoofing attack & ARP broadcasting attack	yes	Yes
Moon D etal, [23]	ARP spoofing attack	yes	

5.1 Neutrosophic TOPSIS Method

This part introduces the neutrosophic TOPSIS method. The first step compute the weights of criteria. It is crucial to provide relative values to the various aspect groups since their relative importance may vary[24-26]. Patients may use their own judgment to assign these values, but text mining methods like phrase presence and term usage can also be used to generate more objective results. How often a certain phrase appears in the text is represented by the term's frequency[27-29]. The preceding analyses allow for the determination of objective weights for the various aspect groups[30-32].

Step 1. Build the Decsion matrix
Step 2. Compute the positive and negative ideal solution (PIS and NIS)

TOPSIS takes into account both positive and negative factors, or "aspects." Let the set of positives be the benefits and the set of negatives be the costs [30-32]. Next, we define the PIS and the NIS in turn:

$$X^{+} = \{< c_j, A_{x^{+}}(c_j), B_{x^{+}}(c_j), D_{x^{+}}(c_j) > \mid c_j \in C\} \quad (1)$$

$$A_{x^{+}}(c_j) = \{(\max_i A_{x^{+}}(c_j)), (\min_i A_{x^{+}}(c_j))\} \quad (2)$$

$$B_{x^{+}}(c_j) = \{(\min_i B_{x^{+}}(c_j)), (\max_i B_{x^{+}}(c_j))\} \quad (3)$$

$$D_{x^{+}}(c_j) = \{(\min_i D_{x^{+}}(c_j)), (\max_i D_{x^{+}}(c_j))\} \quad (4)$$

$$X^{-} = \{< c_j, A_{x^{-}}(c_j), B_{x^{-}}(c_j), D_{x^{-}}(c_j) > \mid c_j \in C\} \quad (5)$$

$$A_{x^{-}}(c_j) = \{(\min_i A_{x^{-}}(c_j)), (\max_i A_{x^{-}}(c_j))\} \quad (6)$$

$$B_{x^{-}}(c_j) = \{(\max_i B_{x^{-}}(c_j)), (\min_i B_{x^{-}}(c_j))\} \quad (7)$$

$$D_{x^{-}}(c_j) = \{(\max_i D_{x^{-}}(c_j)), (\min_i D_{x^{-}}(c_j))\} \quad (8)$$

Step 3. Compute the distance from ideal solution

$$T(P_{ij}, P_j^{+}) = \frac{1}{3}\left(\left|A_{xi}(c_j) - A_{x^{+}i}(c_j^{+})\right| + \left|B_{xi}(c_j) - B_{x^{+}i}(c_j^{+})\right| + \left|D_{xi}(c_j) - D_{x^{+}i}(c_j^{+})\right|\right) \quad (9)$$

$$T(P_{ij}, P_j^{-}) = \frac{1}{3}\left(\left|A_{xi}(c_j) - A_{x^{-}i}(c_j^{-})\right| + \left|B_{xi}(c_j) - B_{x^{-}i}(c_j^{-})\right| + \left|D_{xi}(c_j) - D_{x^{-}i}(c_j^{-})\right|\right) \quad (10)$$

Step 4. Compute the relative closeness coefficient of every option

$$C(X_i) = \frac{T(X_i, X^{-})}{T\_\max(X_i, X^{-})} - \frac{T(X_i, X^{+})}{T\_\max(X_i, X^{+})} \quad (11)$$

Setp 5. Compute the ordering of the options

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

5.2 Results

This section introduces the results of the neutrosophic TOPSIS method. The goal of this part, ranking the technical solutions of ARP attacks based on different criteria. This paper discussed the set of technical solutions in section 4, these solutions acted as an alternative to the neutrosophic TOPSIS method. Also, there are ten criteria are introduced in part 1.2. Overall, there are ten criteria and 13 alternatives are processed to choose the best one. This paper used single-valued neutrosophic numbers (SVNNs) to evaluate the criteria and alternatives. The experts evaluated the criteria and set of alternatives. Tab.4 shows the single-valued neutrosophic number for the set of solutions and criteria. Then compute the weights of the criteria. Then compute the normalization matrix as shown in Tab. 5. Then multiply the weights of criteria by the proposed solution as shown in Tab. 6. Then compute the positive and negative ideal solution. Then compute the distance from each alternative to the positive and negative ideal solution. Then compute the closeness value as shown in Figure 7.

Table 4: SVNNs of criteria and technical solutions of ARP attacks.

	ARPC 1	ARPC 2	ARPC 3	ARPC 4	ARPC 5	ARPC 6	ARPC 7	ARPC 8	ARPC 9	ARPC 10
ARPS 1	(0.9,0.1,0.2)	(0.8,0.25,0.30)	(0.8,0.25,0.30)	(0.9,0.1,0.2)	(0.75,0.35,0.35)	(0.8,0.25,0.30)	(0.9,0.1,0.2)	(0.8,0.25,0.30)	(0.8,0.25,0.30)	(0.9,0.1,0.2)
ARPS 2	(0.8,0.25,0.30)	(0.75,0.35,0.35)	(0.3,0.60,0.70)	(0.65,0.45,0.40)	(0.8,0.25,0.30)	(0.65,0.45,0.40)	(0.75,0.35,0.35)	(0.65,0.45,0.40)	(0.75,0.35,0.35)	(0.8,0.25,0.30)
ARPS 3	(0.65,0.45,0.40)	(0.3,0.60,0.70)	(0.3,0.60,0.70)	(0.65,0.45,0.40)	(0.2,0.70,0.80)	(0.65,0.45,0.40)	(0.75,0.35,0.35)	(0.75,0.35,0.35)	(0.65,0.45,0.40)	(0.75,0.35,0.35)
ARPS 4	(0.9,0.1,0.2)	(0.75,0.35,0.35)	(0.8,0.25,0.30)	(0.9,0.1,0.2)	(0.2,0.70,0.80)	(0.65,0.45,0.40)	(0.9,0.1,0.2)	(0.65,0.45,0.40)	(0.3,0.60,0.70)	(0.75,0.35,0.35)
ARPS 5	(0.75,0.35,0.35)	(0.2,0.70,0.80)	(0.3,0.60,0.70)	(0.65,0.45,0.40)	(0.2,0.70,0.80)	(0.3,0.60,0.70)	(0.75,0.35,0.35)	(0.8,0.25,0.30)	(0.65,0.45,0.40)	(0.9,0.1,0.2)
ARPS 6	(0.65,0.45,0.40)	(0.8,0.25,0.30)	(0.3,0.60,0.70)	(0.65,0.45,0.40)	(0.75,0.35,0.35)	(0.65,0.45,0.40)	(0.65,0.45,0.40)	(0.75,0.35,0.35)	(0.3,0.60,0.70)	(0.65,0.45,0.40)
ARPS 7	(0.65,0.45,0.40)	(0.75,0.35,0.35)	(0.9,0.1,0.2)	(0.65,0.45,0.40)	(0.65,0.45,0.40)	(0.8,0.25,0.30)	(0.3,0.60,0.70)	(0.9,0.1,0.2)	(0.65,0.45,0.40)	(0.8,0.25,0.30)
ARPS 8	(0.9,0.1,0.2)	(0.75,0.35,0.35)	(0.3,0.60,0.70)	(0.65,0.45,0.40)	(0.75,0.35,0.35)	(0.2,0.70,0.80)	(0.65,0.45,0.40)	(0.3,0.60,0.70)	(0.3,0.60,0.70)	(0.75,0.35,0.35)
ARPS 9	(0.75,0.35,0.35)	(0.65,0.45,0.40)	(0.8,0.25,0.30)	(0.2,0.70,0.80)	(0.2,0.70,0.80)	(0.9,0.1,0.2)	(0.2,0.70,0.80)	(0.9,0.1,0.2)	(0.75,0.35,0.35)	(0.8,0.25,0.30)
ARPS	(0.8,0.25,0.30)	(0.75,0.35,0.35)	(0.2,0.70,0.80)	(0.65,0.45,0.40)	(0.65,0.45,0.40)	(0.2,0.70,0.80)	(0.65,0.45,0.40)	(0.2,0.70,0.80)	(0.65,0.45,0.40)	(0.75,0.35,0.35)

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

$_{10}$										
ARPS$_{11}$	(0.75, 0.35,0.35)	(0.2,0.70,0.80)	(0.9,0.1,0.2)	(0.2,0.70,0.80)	(0.8,0.25,0.30)	(0.2,0.70,0.80)	(0.75, 0.35,0.35)	(0.3,0.60,0.70)	(0.75, 0.35,0.35)	(0.8,0.25,0.30)
ARPS$_{12}$	(0.8,0.25,0.30)	(0.65, 0.45,0.40)	(0.75, 0.35,0.35)	(0.75, 0.35,0.35)	(0.2,0.70,0.80)	(0.75, 0.35,0.35)	(0.2,0.70,0.80)	(0.75, 0.35,0.35)	(0.3,0.60,0.70)	(0.75, 0.35,0.35)
ARPS$_{13}$	(0.9,0.1,0.2)	(0.75, 0.35,0.35)	(0.8,0.25,0.30)	(0.8,0.25,0.30)	(0.75, 0.35,0.35)	(0.8,0.25,0.30)	(0.65, 0.45,0.40)	(0.75, 0.35,0.35)	(0.8,0.25,0.30)	(0.9,0.1,0.2)

Table 5. Normalization of criteria and technical solutions of ARP attacks.

	ARP C_1	ARP C_2	ARP C_3	ARP C_4	ARP C_5	ARP C_6	ARP C_7	ARP C_8	ARP C_9	ARP C_{10}
ARP S_1	6.766775	38.77405	60.05988	35.78063	123.656	64.05016	44.15912	25.57802	44.25825	6.387027
ARP S_2	5.85584	35.32729	26.69328	24.77111	135.7206	51.24013	34.81746	20.46241	40.32399	5.527214
ARP S_3	4.684672	17.23291	26.69328	24.77111	42.22419	51.24013	34.81746	23.3043	35.4066	5.035881
ARP S_4	6.766775	35.32729	60.05988	35.78063	42.22419	51.24013	44.15912	20.46241	19.67034	5.035881
ARP S_5	5.335295	12.06304	26.69328	24.77111	42.22419	28.46674	34.81746	25.57802	35.4066	6.387027
ARP S_6	4.684672	38.77405	26.69328	24.77111	123.656	51.24013	30.57158	23.3043	19.67034	4.421771
ARP S_7	4.684672	35.32729	69.4028	24.77111	108.5765	64.05016	16.98421	29.55693	35.4066	5.527214
ARP S_8	6.766775	35.32729	26.69328	24.77111	123.656	19.92672	30.57158	11.36801	19.67034	5.035881
ARP S_9	5.335295	31.01924	60.05988	9.633209	42.22419	74.0138	11.88895	29.55693	40.32399	5.527214
ARP S_{10}	5.85584	35.32729	18.6853	24.77111	108.5765	19.92672	30.57158	7.957605	35.4066	5.035881
ARP S_{11}	5.335295	12.06304	69.4028	9.633209	135.7206	19.92672	34.81746	11.36801	40.32399	5.527214
ARP S_{12}	5.85584	31.01924	54.72096	28.2114	42.22419	58.35653	11.88895	23.3043	19.67034	5.035881
ARP S_{13}	6.766775	35.32729	60.05988	30.96389	123.656	64.05016	30.57158	23.3043	44.25825	6.387027

Table 6. Weighted normalization of criteria and technical solutions of ARP attacks.

	ARP C_1	ARP C_2	ARP C_3	ARP C_4	ARP C_5	ARP C_6	ARP C_7	ARP C_8	ARP C_9	ARP C_{10}
ARP S_1	0.742349	3.681076	5.70188	3.925311	10.69591	6.080703	4.844473	2.42829	4.201728	0.700688
ARP S_2	0.642415	3.353853	2.534169	2.717513	11.73947	4.864563	3.819647	1.942632	3.828222	0.606363
ARP S_3	0.513932	1.636034	2.534169	2.717513	3.652281	4.864563	3.819647	2.212431	3.361382	0.552461
ARP	0.742	3.353	5.701	3.925	3.652	4.864	4.844	1.942	1.867	0.552

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

S_4	349	853	88	311	281	563	473	632	435	461
ARP S_5	0.585 308	1.145 224	2.534 169	2.717 513	3.652 281	2.702 535	3.819 647	2.428 29	3.361 382	0.700 688
ARP S_6	0.513 932	3.681 076	2.534 169	2.717 513	10.69 591	4.864 563	3.353 853	2.212 431	1.867 435	0.485 09
ARP S_7	0.513 932	3.353 853	6.588 865	2.717 513	9.391 58	6.080 703	1.863 251	2.806 034	3.361 382	0.606 363
ARP S_8	0.742 349	3.353 853	2.534 169	2.717 513	10.69 591	1.891 774	3.353 853	1.079 24	1.867 435	0.552 461
ARP S_9	0.585 308	2.944 861	5.701 88	1.056 81	3.652 281	7.026 618	1.304 276	2.806 034	3.828 222	0.606 363
ARP S_{10}	0.642 415	3.353 853	1.773 918	2.717 513	9.391 58	1.891 774	3.353 853	0.755 468	3.361 382	0.552 461
ARP S_{11}	0.585 308	1.145 224	6.588 865	1.056 81	11.73 947	1.891 774	3.819 647	1.079 24	3.828 222	0.606 363
ARP S_{12}	0.642 415	2.944 861	5.195 021	3.094 93	3.652 281	5.540 169	1.304 276	2.212 431	1.867 435	0.552 461
ARP S_{13}	0.742 349	3.353 853	5.701 88	3.396 891	10.69 591	6.080 703	3.353 853	2.212 431	4.201 728	0.700 688

The solution 1 is the best and solution 5 is the worst and least importance.

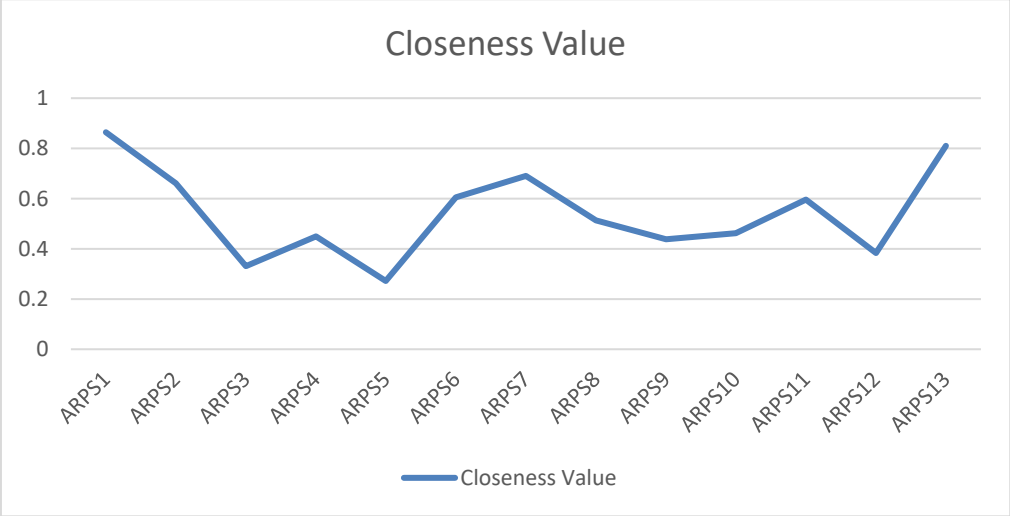

Figure 7: The closeness value by the neutrosophic TOPSIS method.

6. Conclusion

In this study, we concentrated on the most recent techniques put forth by different researchers to identify and counteract different ARP spoofing attacks in SDN. These techniques are assessed based on three different ARP attack types: ARP spoofing, ARP flooding, and ARP broadcasting attacks. The SDN's ARP (Address Resolution Protocol) is vulnerable to numerous assaults despite being essential for communication. Many such attacks were discussed in this survey. Various detection and mitigation techniques have also been proposed. We also showed in detail the background of SDN, discussed SDN architecture and vulnerabilities, briefly showed ARP handling in SDN with an example of the ARP protocol, and mentioned three types of ARP attacks. Finally, we listed the latest technical proposed solutions against these types of ARP attacks,we deeply discussed these solutions, and we analyzed and compared these different solutions based on some criteria like attack area and ARP request/reply analysis to facilitate future researchers in overcoming these three types of ARP attacks. We also discussed performance metrics for these solutions, comparing the results of some existing solutions based on detection and mitigation times, CPU utilization, and other metrics. And also, we proposed the MCDM model to rank and select

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

the best solution in ARP attack. The TOPSIS method is used to compute the rank of alternatives by identifying the set of positive and negative ideal solutions. The TOPSIS method is integrated with the single-valued neutrosophic set to deal with uncertain data. The neutrosophic TOPSIS method operated with ten criteria and 13 solutions to select the best one. We obtained the first solution is the best and the fifth solution is the worst solution.

The future study must retain classification based on additional criteria and other performance measures and address the pros and cons of the suggested solutions to help future researchers avoid the drawbacks of the existing solutions.

References

[1] S. Sandraascott, H. Ludoviccjacquin, and R. Editors, "Computer Communications and Networks Guide to Security in SDN and NFV Challenges, Opportunities, and Applications." [Online]. Available: http://www.springer.com/series/4198

[2] D. Kumar and J. Thakur, "Handling Security Issues in Software-defined Networks (SDNs) Using Machine Learning," in *Computational Vision and Bio-Inspired Computing: Proceedings of ICCVBIC 2021*, Springer, 2022, pp. 263–277.

[3] A. Pradhan and R. Mathew, "Solutions to vulnerabilities and threats in software defined networking (SDN)," *Procedia Comput Sci*, vol. 171, pp. 2581–2589, 2020.

[4] Z. Shah and S. Cosgrove, "Mitigating ARP cache poisoning attack in software-defined networking (SDN): a survey," *Electronics (Basel)*, vol. 8, no. 10, p. 1095, 2019.

[5] V. Rohatgi and S. Goyal, "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)*, IEEE, 2020, pp. 352–356.

[6] J. S. Meghana, T. Subashri, and K. R. Vimal, "A survey on ARP cache poisoning and techniques for detection and mitigation," in *2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN)*, IEEE, 2017, pp. 1–6.

[7] T. Girdler and V. G. Vassilakis, "Implementing an intrusion detection and prevention system using software-defined networking: defending against ARP spoofing attacks and blacklisted MAC addresses," *Computers & Electrical Engineering*, vol. 90, p. 106990, 2021.

[8] N. Ahuja, G. Singal, D. Mukhopadhyay, and A. Nehra, "Ascertain the efficient machine learning approach to detect different ARP attacks," *Computers and Electrical Engineering*, vol. 99, p. 107757, 2022.

[9] M. N. Munther, F. Hashim, N. A. A. Latiff, K. A. Alezabi, and J. T. Liew, "Scalable and secure SDN based ethernet architecture by suppressing broadcast traffic," *Egyptian Informatics Journal*, vol. 23, no. 1, pp. 113–126, 2022.

[10] H. Y. I. KHALID, P. M. ISMAEL, and A. B. AL-KHALIL, "Efficient mechanism for securing software defined network against arp spoofing attack," *Journal of Duhok University*, vol. 22, no. 1, pp. 124–131, 2019.

[11] J. Du *et al.*, "Research on An Approach of ARP Flooding Suppression in Multi-Controller SDN Networks," in *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, IEEE, 2021, pp. 1159–1166.

[12] N. Saritakumar, K. V Anusuya, and S. Ajitha, "Detection and Mitigation of ARP Poisoning Attack in Software Defined Network," in *Proceedings of the First International Conference on Combinatorial and Optimization, ICCAP 2021, December 7-8 2021, Chennai, India*, 2021.

[13] H.-C. Wei, Y.-H. Tung, and C.-M. Yu, "Counteracting UDP flooding attacks in SDN," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, IEEE, 2016, pp. 367–371.

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

[14] S. Buzura, M. Lehene, B. Iancu, and V. Dadarlat, "An Extendable Software Architecture for Mitigating ARP Spoofing-Based Attacks in SDN Data Plane Layer," *Electronics (Basel)*, vol. 11, no. 13, p. 1965, 2022.

[15] X. Hou, Z. Jiang, and X. Tian, "The detection and prevention for ARP Spoofing based on Snort," in *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, IEEE, 2010, pp. V5-137.

[16] M. Matties, "Distributed responder ARP: Using SDN to re-engineer ARP from within the network," in *2017 International Conference on Computing, Networking and Communications (ICNC)*, IEEE, 2017, pp. 678–683.

[17] D. Balagopal and X. A. K. Rani, "A technique for a software-defined and network-based ARP spoof detection and mitigation," *International Journal of Applied Engineering Research*, vol. 13, no. 20, pp. 14823–14826, 2018.

[18] A. Zaalouk, R. Khondoker, R. Marx, and K. M. Bayarou, "OrchSec Demo: Demonstrating the Capability of an Orchestrator-based Architecture for Network Security," *Academic Demo, Open Networking Summit*, 2014.

[19] A. Nehra, M. Tripathi, and M. S. Gaur, "FICUR: Employing SDN programmability to secure ARP," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2017, pp. 1–8.

[20] H. Aldabbas and R. Amin, "A novel mechanism to handle address spoofing attacks in SDN based IoT," *Cluster Comput*, vol. 24, no. 4, pp. 3011–3026, 2021.

[21] A. K. Rangisetti, R. Dwivedi, and P. Singh, "Denial of ARP spoofing in SDN and NFV enabled cloud-fog-edge platforms," *Cluster Comput*, vol. 24, no. 4, pp. 3147–3172, 2021.

[22] V. K. Tchendji, F. Mvah, C. T. Djamegni, and Y. F. Yankam, "E2BaSeP: Efficient Bayes based security protocol against ARP spoofing attacks in SDN architectures," *Journal of Hardware and Systems Security*, vol. 5, no. 1, pp. 58–74, 2021.

[23] D. Moon, J. D. Lee, Y.-S. Jeong, and J. H. Park, "RTNSS: a routing trace-based network security system for preventing ARP spoofing attacks," *J Supercomput*, vol. 72, pp. 1740–1756, 2016.

[24] Irina V. Pustokhina,Denis A. Pustokhin, An Intelligent Neutrosophic Model for Evaluation Sustainable Housing Affordability, International Journal of Advances in Applied Computational Intelligence, Vol. 2 , No. 2 , (2022) : 45-53 (Doi : https://doi.org/10.54216/IJAACI.020205)

[25] S.-W. Lin and H.-W. Lo, "An FMEA model for risk assessment of university sustainability: using a combined ITARA with TOPSIS-AL approach based neutrosophic sets," *Ann. Oper. Res.*, pp. 1–27, 2023.

[26] Shereen Zaki,Mahmoud M. Ibrahim,Mahmoud M. Ismail, Interval Valued Neutrosophic VIKOR Method for Assessment Green Suppliers in Supply Chain, International Journal of Advances in Applied Computational Intelligence, Vol. 2 , No. 1 , (2022) : 15-22 (Doi : https://doi.org/10.54216/IJAACI.020102)

[27] Shimaa Said,Mahmoud M. Ibrahim,Mahmoud M. Ismail, An Integrated Multi-Criteria Decision-Making Approach for Identification and Ranking Solar Drying Barriers under Single-Valued Triangular Neutrosophic Sets (SVTNSs), Neutrosophic and Information Fusion, Vol. 2 , No. 1 , (2023) : 35-49 (Doi : https://doi.org/10.54216/NIF.020103)

[28] F. Sbastian, A. Y. Ridwan, and N. Novitasari, "Implementation of Multi Criteria Decision Making (MCDM) Fuzzy Neutrosophic TOPSIS-CRITIC in Determining Sustainability Aspects of the Location of

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023

IoT Based Products Warehouse," in *2021 International Conference on Computer Science and Engineering (IC2SE)*, IEEE, 2021, pp. 1–8.

[29] Abedallah Z. Abualkishik,Rasha Almajed, Triangular Neutrosophic Multi-Criteria Decision Making AHP Method for Solar Power Site Selection, International Journal of Advances in Applied Computational Intelligence, Vol. 2 , No. 2 , (2022) : 08-15 (Doi : https://doi.org/10.54216/IJAACI.020201)

[30] M. Baghel and C. Krishna, "Multi-objective optimization of mechanical and microstructural characteristics in a stir casting process of MWCNTs/Al6082 composites using neutrosophic TOPSIS and GRA," *J. Chinese Inst. Eng.*, vol. 46, no. 4, pp. 345–354, 2023.

[31] H. Sharma, A. Tandon, P. K. Kapur, and A. G. Aggarwal, "Ranking hotels using aspect ratings based sentiment classification and interval-valued neutrosophic TOPSIS," *Int. J. Syst. Assur. Eng. Manag.*, vol. 10, pp. 973–983, 2019.

[32] M. Junaid, Y. Xue, M. W. Syed, J. Z. Li, and M. Ziaullah, "A neutrosophic ahp and topsis framework for supply chain risk assessment in automotive industry of Pakistan," *Sustainability*, vol. 12, no. 1, p. 154, 2019.

Doi: https://doi.org/10.54216/IJNS.210411
Received: February 12, 2023 Revised: May 23, 2023 Accepted: July 18, 2023