Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Volume 13, Issue 03 March 2024

Enhancing IOT Security Using Software Defined Networks

Addressing Vulnerabilities and Enhancing Resilience in IoT Networks with SDN

Muthulakshmi K
Computer Science & Engineering
Kamaraj College of Engineering and Technology
Virudhunagar, India

Sivaranjani P
Computer Science & Engineering
Kamaraj College of Engineering and Technology
Virudhunagar, India

Kaaviya M
Computer Science & Engineering
Kamaraj College of Engineering and Technology
Virudhunagar, India

Kaleeswari D
Computer Science & Engineering
Kamaraj College of Engineering and Technology
Virudhunagar, India

Abstract—**Enhancing IOT Security using SDN is an innovative solution leveraging machine learning techniques to address the persistent challenge of IOT security in networking. The primary goal of this project is to develop a comprehensive and effective solution for enhancing the security of Internet of Things (IoT) networks. Leveraging the power of Software Defined Networking (SDN), the project aims to address the challenges posed by the dynamic and diverse nature of IoT ecosystems. Internet of Things is an upcoming technology, where IoT devices are inter-acting with cloud over Internet. The network security issue like distributed denial of service [DDoS]attacks are of major concern, and its mitigation at the earliest remains vital. In IoT-related environment, the security issues of traditional network have major impact in IoT application domain. The IoT-related data that are highly confidential and there arises the need to change the paradigm of traditional network. The expectant network should be more secure and flexible to detect and mitigate the network attacks. The recent developments in IoT botnets contributes a major part in launching DDoS attacks on the IoT networks. In this project a Software-defined IoT gateway model is presented to provide a secured IoT gateway and then a DDoS detection and mitigation monitoring system is proposed to defend the network from DDoS attack. IoT environment with software-defined network seems to be promising enough to reduce many security issues with respect to IoT in traditional network environment. The proposed project work has created a test bed that collects IoT live data and sends it through secure SDN into the cloud platform. We are using an ensemble learning model which combines all the best classifiers which can detect the DDOS attack more precisely. By ensemble the supervised and unsupervised learning algorithms like Multi-Layer Perceptron (MLP) Classifier, Decision Tree, Random Forest are used to boosts the performance of detection of DDOS attacks in SDN.**

Keywords— **SDN, IoT, Machine Learning, DDOS attack**.

I. INTRODUCTION

Enhancing the security of the Internet of Things (IoT) is of paramount importance in today's interconnected world. IoT devices, ranging from smart thermostats to industrial sensors, have become integral to our daily lives and critical infrastructure. However, they also present attractive targets for cyberattacks. To bolster IoT security, a multifaceted strategy is essential. IoT ecosystem is dynamic and ever-expanding, demanding ongoing risk assessments and a collaborative approach with a focus on staying ahead of emerging threats and regulatory requirements. In our project, we employ an innovative approach centred around Software-Defined Networking (SDN) to counteract Distributed Denial of Service (DDoS) attacks. This cutting-edge framework relies on the principles of SDN to effectively manage the security dynamics in a network facing DDoS threats. Software Defined Networking (SDN) is a network architecture that centralizes control and enables software-based management of network resources. SDN separates the control plane from the data plane, allowing for dynamic, programmable, and efficient network management. Introduction to Enhancing IOT Security using SDN concept is discussed in this chapter. The convergence of IoT and SDN presents a unique opportunity to address security concerns in interconnected environments. Traditional network architectures struggle to cope with the dynamic nature of IoT devices, their diverse communication patterns, and the need for real-time threat detection. SDN, with its centralized control and programmability, offers a promising framework to enhance security, scalability, and manageability. This project focuses on securing a heterogeneous network of IoT devices. These devices, ranging from sensors to actuators, collect critical data and play a pivotal role in various applications. By integrating SDN, we can enforce access control policies, segment the network, and authenticate devices to prevent unauthorized access. DDoS attacks pose a significant threat to IoT networks. Malicious actors exploit vulnerabilities in IoT devices to launch large-scale attacks, disrupting services and compromising data integrity.

Our solution employs SDN-based traffic analysis and anomaly detection techniques to identify and mitigate DDoS attacks in real time. ThinksBoard, a powerful open-source IoT platform, provides real-time data visualization and analytics.

By securely transmitting data from IoT devices to ThinksBoard through the SDN controller, we enable stakeholders to monitor device behaviour, identify trends, and make informed decisions.

By leveraging the capabilities of SDN technology, an innovative approach that harnesses the power of SDN to address the complex challenges posed by the dynamic landscape of IoT ecosystems. In the realm of IoT, where interconnected devices communicate with each other and the cloud over the Internet, ensuring robust security measures is paramount.

Traditional network security issues, such as DDoS attacks, pose significant threats to IoT networks, necessitating proactive measures for detection and mitigation. SDN technology offers a flexible and programmable framework that enables the creation of secure IoT gateways and the implementation of advanced threat detection and mitigation systems. By centralizing network control and management, SDN allows for dynamic adjustments to network configurations and traffic prioritization, enhancing resilience against evolving security threats. Moreover, the emergence of IoT botnets has further underscored the urgency of bolstering security measures in IoT environments. Through the integration of SDN and machine learning techniques, such as ensemble learning models, the project aims to enhance the accuracy and efficiency of DDoS attack detection in SDN-based IoT networks. Furthermore, the project has developed a comprehensive test bed that collects real-time IoT data and securely transmits it to cloud platforms using SDN infrastructure.

II. ARCHITECTURE OVERVIEW

Our proposed architecture consists of the following components:

A. IoT Devices: These include sensors, actuators, and edge devices deployed across the network. They generate data related to environmental conditions, health parameters, industrial processes, etc.

B. SDN Controller (Ryu): The Ryu controller acts as the brain of our network. It dynamically manages network flows, enforces security policies, and orchestrates communication between IoT devices and ThinksBoard.

C. ThinksBoard: ThinksBoard serves as the visualization platform. It aggregates data from IoT devices, provides customizable dashboards, and facilitates real-time monitoring.

SDN suggests to centralize network intelligence in one network component by disassociating the forwarding process of network packets (data plane) from the routing process (control plane). The control plane consists of one or more controllers which are considered as the brain of SDN network where the whole intelligence is incorporated.

III. SDN TECHNOLOGY

Software-Defined Networking (SDN) is a network architecture that centralizes control and enables software-based management of network resources. SDN separates the control plane from the data plane, allowing for dynamic, programmable, and efficient network management. Introduction to Enhancing IOT Security using SDN concept is discussed in this chapter.

Enhancing the security of the Internet of Things (IoT)is of paramount importance in today's interconnected world. IoT devices, ranging from smart thermostats to industrial sensors, have become integral to our daily lives and critical infrastructure. However, they also present attractive targets for cyberattacks. To bolster IoT security, a multifaceted strategy is essential. IoT ecosystem is dynamic and ever- expanding, demanding ongoing risk assessments and a collaborative approach with a focus on staying ahead of emerging threats and regulatory requirements. In our project, we employ an innovative approach centred around Software-Defined Networking (SDN) to counteract Distributed Denial of Service (DDoS) attacks. This cutting- edge framework relies on the principles of SDN to effectively manage the security dynamics in a network facing DDoS threats.

Software-Defined Networking (SDN) is a transformative technology that revolutionizes the way networks are managed and operated. Unlike traditional networks, where network devices are typically controlled and configured individually, SDN centralizes network control through a software-based controller. This controller communicates with network devices and orchestrates their behaviour, making it possible to manage the entire network from a single point.

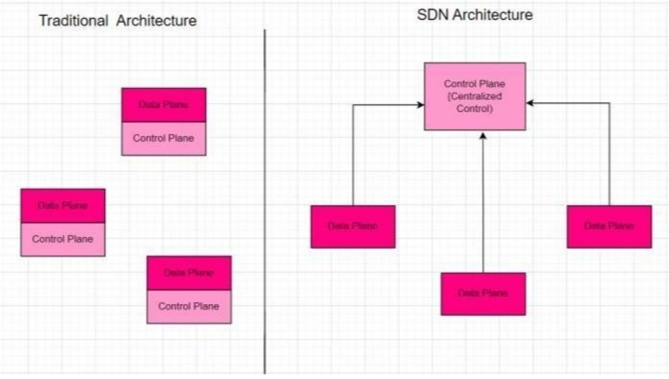

Fig. 1: Traditional Network VS SDN

SDN stands for Software Defined Network which is a networking architecture approach. It enables the control and management of the network using software applications. Through Software Defined Network (SDN) networking behaviour of the entire network and its devices are programmed in a centrally controlled manner through software applications using open APIs.

A. Data plane: All the activities involving as well as resulting from data packets sent by the end-user belong to this plane.
B. Control plane: All activities necessary to perform data plane activities but do not involve end-user data packets belong to this plane.

Software-defined networking (SDN) technology in combination with Internet of Things (IoT) devices presents an intriguing blend of capabilities, offering enhanced flexibility, scalability, and management in network infrastructures. Here's a breakdown of how SDN intersects with IoT devices:

Dynamic Network Configuration: SDN allows for centralized control of network resources through programmable interfaces. This enables dynamic configuration and reconfiguration of network elements to accommodate the changing requirements of IoT devices. For instance, SDN controllers can adjust network policies in real-time based on IoT device data or traffic patterns. Traffic Prioritization and Quality of Service (QoS): With SDN, administrators can prioritize traffic based on IoT device requirements. This ensures that critical IoT data, such as real-time sensor readings or control signals, receives priority treatment over less time- sensitive traffic. QoS policies can be easily enforced and adapted as needed.

Segmentation and Isolation: IoT deployments often involve diverse devices with varying security and performance requirements. SDN facilitates network segmentation and isolation, allowing administrators to create virtual network slices for different IoT applications or device types. This helps in containing security breaches and optimizing network performance. Enhanced Security: By centralizing network control, SDN enables more robust security measures for IoT deployments. Security policies can be enforced at the network level, ensuring consistent protection across all IoT devices. Additionally, SDN platforms often integrate with security solutions, enabling threat detection and response mechanisms.

Scalability and Resource Optimization: IoT deployments can rapidly scale in terms of device count and data volume. SDN's scalability features, such as dynamic provisioning and efficient resource utilization, help accommodate the growth of IoT networks without sacrificing performance or reliability. Traffic Engineering and Optimization: SDN provides granular control over network traffic flows, allowing administrators to optimize routing paths and resource utilization for IoT applications. This can improve overall network efficiency and reduce latency, enhancing the responsiveness of IoT services. Centralized Management and Orchestration: SDN simplifies the management of complex IoT deployments by providing a centralized interface for network configuration, monitoring, and troubleshooting. This streamlines administrative tasks and reduces operational overhead, particularly in large-scale IoT environments.

SDN Architecture: In a traditional network, each switch has its own data plane as well as the control plane. The control plane of various switches exchange topology information and hence construct a forwarding table that decides where an incoming data packet has to be forwarded via the data plane. Software-defined networking (SDN) is an approach via which we take the control plane away from the switch and assign it to a centralized unit called the SDN controller. Hence, a network administrator can shape traffic via a centralized console without having to touch the individual switches.

The data plane still resides in the switch and when a packet enters a switch, its forwarding activity is decided based on the entries of flow tables, which are pre-assigned by the controller. A flow table consists of match fields (like input port number and packet header) and instructions. The packet is first matched against the match fields of the flow table entries. Then the instructions of the corresponding flow entry are executed. The instructions can be forwarding the packet via one or multiple ports, dropping the packet, or adding headers to the packet. If a packet doesn't find a corresponding match in the flow table, the switch queries the controller which sends a new flow entry to the switch. The switch forwards or drops the packet based on this flow entry.

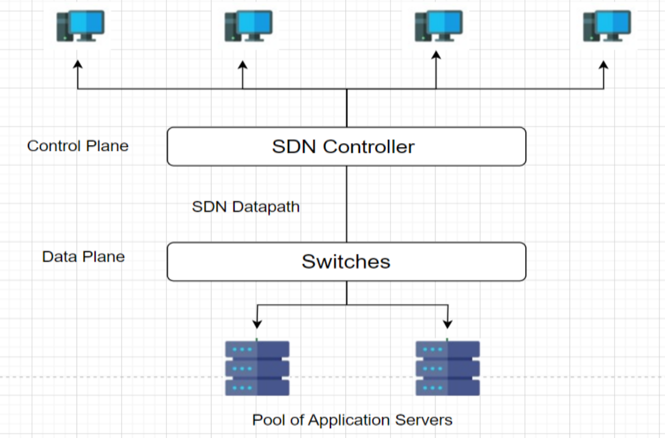

Fig. 2: SDN Architecture movement of data packets

A typical SDN architecture consists of three layers.

A. Application layer: It contains the typical network applications like intrusion detection, firewall, and load balancing

A. Control layer: It consists of the SDN controller which acts as the brain of the network. It also allows hardware abstraction to the applications written on top of it.

B. Infrastructure layer: This consists of physical switches which form the data plane and carries out the actual

I. RYU CONTROLLER

The Ryu controller is an open-source SDN controller framework written in Python. It serves as a critical component in SDN architectures, enabling centralized control and dynamic management of network resources. Here are the key aspects of the Ryu controller,

The Ryu controller follows a modular architecture, allowing developers to create custom applications and extensions. It consists of several components, including, Handles asynchronous events such as switch connections, disconnections, and packet arrivals. Provides APIs for communication with OpenFlow-enabled switches. Allows external applications to interact with the controller. Collects information about network topology. Manages flow rules and forwarding decisions. Ryu primarily supports the OpenFlow protocol, which defines the communication between the controller and switches.

It allows the controller to instruct switches on how to process packets, set up flow rules, and manage network traffic.Ryu's OpenFlow library provides a Python interface for creating, modifying, and deleting flow entries in switches. Developers can build custom SDN applications using Ryu's APIs. These applications can implement various network functions, such as load balancing, traffic engineering, security, and monitoring. Ryu's extensible architecture encourages innovation and experimentation.

Use Cases - Ryu can collect real-time network statistics, monitor link utilization, and detect anomalies. It enables dynamic path selection and load distribution. Ryu- based applications can enforce access control policies. Ryu can prioritize traffic based on requirements. Ryu has an active community of contributors and users

.It integrates well with other SDN tools, libraries, and platforms. Developers can extend Ryu by writing custom modules or leveraging existing ones. Thus, the Ryu controller empowers SDN networks by providing a flexible, programmable, and efficient control plane. Its modular design, OpenFlow support, and application development capabilities make it a valuable asset for network administrators and researchers alike.

II. DENIAL-OF-SERVICE (DOS) ATTACKS

Distributed Denial of Service (DDoS) is a type of DOS attack where multiple systems, which are trojan infected, target a particular system which causes a DoS attack. A DDoS attack uses multiple servers and Internet connections to flood the targeted resource. A DDoS attack is one of the most powerful weapons on the cyber platform.

When you come to know about a website being brought down, it generally means it has become a victim of a DDoS attack. This means that the hackers have attacked your website or PC by imposing heavy traffic. Thus, crashing the website or computer due to overloading. DDoS attacks can be launched using various techniques, including UDP flooding, ICMP flooding, SYN flooding, and HTTP flooding, among others.

Attackers may exploit vulnerabilities in network protocols or use amplification techniques to magnify the volume of traffic directed at the target. Organizations often deploy various defence mechanisms to mitigate the impact of DDoS attacks, such as traffic filtering, rate limiting, and deploying DDoS mitigation services or appliances. Additionally, network administrators may monitor traffic patterns and employ intrusion detection and prevention systems to identify and block malicious traffic in real-time.

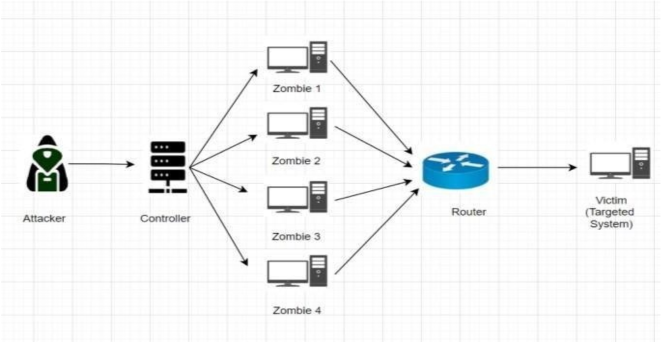

Fig. 3: DDOS Attack

A. Working of DDOS Attack*:* DDoS attacks are carried out with networks of Internet-connected machines. These networks consist of computers and other devices (such as IoTdevices) which have been infected with malware, allowing them to be controlled remotely by an attacker. These individual devices are referred to as bots (or zombies), and a group of bots is called a botnet. Once a botnet has been established, the attacker is able to direct an attack by sending remote instructions to each bot. When a victim's server or network is targeted by the botnet, each bot sends requests to the target's IP address, potentially causing the server or network to become overwhelmed, resulting in a denial-of-service to normal traffic. Because each bot is a legitimate Internet device, separating the attack traffic from normal traffic can be difficult.

B. Identification of DDOS Attack: Distributed Denial of Service (DDoS) attacks can cripple networks, disrupt services, and cause significant damage. Identifying these attacks promptly is crucial for effective mitigation. Monitor your server logs or use web analytics tools. If you notice a sudden surge in traffic from a specific location or IP address, it could indicate a DDoS attack. Slow Loading Times for Your Website – Attackers flood your server with requests, overloading it. If your site takes longer than usual to load, consider a DDoS attack. Unexplained Errors and Timeouts - Too many requests overwhelm your server. Users may encounter HTTP 503 Service Unavailable errors or timeouts. In severe cases, your website might become completely inaccessible. Decreased Performance for Other Services - If services on the same network suffer performance hits, your site could be under attack. The attacker's requests consume bandwidth, affecting other services. Increased CPU or Memory Usage on Your Server: Monitor resource utilization. A sudden spike in CPU or memory usage may indicate a DDoS attack.

C. Mitigation of DDOS Attack*:* We utilize the capabilities of Software Defined Networking (SDN) to monitor and analyse network traffic in real-time. By leveraging the Ryu controller as the SDN controller, we have the flexibility to programmatically manage the network and implement sophisticated detection mechanisms for identifying DDoS attacks. Traffic Analysis - Our system continuously monitors incoming and outgoing traffic from IoT devices connected to the network. We analyse various metrics such as packet rate, packet size, and traffic patterns to identify deviations from normal behaviour. Once a DDoS attack is detected, our system takes proactive measures to mitigate its impact and restore normal network operation. Upon detecting a DDoS attack, the Ryu controller dynamically updates the network's flow tables to block the MAC address of the source of the malicious traffic. By instructing network switches to drop packets from the identified MAC address, we effectively mitigate the impact of the attack. This approach to detecting and mitigating DDoS attacks in IoT networks leverages the capabilities of SDN and machine learning techniques for proactive threat management.

III. ENSEMBLING TECHNIQUE:

Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model. The goal of any machine learning problem is to find a single model that will best predict our wanted outcome. Rather than making one model and hoping this model is the best/most accurate predictor we can make, ensemble methods take a myriad of models into account, and average those models to produce one final model.

In this project, we are using an ensemble learning model which combines all the best classifiers which can detect the DDOS attack more precisely. By ensemble the supervised and unsupervised learning algorithms like Multi- Layer Perceptron (MLP) Classifier, Decision Tree, Random Forest are used to boosts the performance of detection of DDOS attacks in SDN.

A. MLP Classifier: The MLP (Multi-Layer Perceptron)

Classifier stands out as a robust tool in the realm of supervised learning, particularly renowned for its prowess in tackling classification tasks within machine learning. This sophisticated neural network architecture comprises multiple layers of interconnected nodes, each layer serving a distinct purpose in processing input data and generating predictions. At its core, the MLP Classifier encompasses an input layer, one or more hidden layers, and an output layer. Within this structure, information flows from the input layer through the hidden layers, where nonlinear transformations are applied via activation functions, culminating in the output layer's generation of class probabilities or confidence scores. This hierarchical arrangement allows the MLP Classifier to capture intricate patterns and relationships within complex datasets, making it adept at handling nonlinearities and achieving high predictive accuracy.

During the training phase, the MLP Classifier undergoes a series of iterative steps to learn from the provided data and optimize its parameters. The process commences with forward propagation, wherein input data traverses the network, and each layer computes its output based on weighted sums and activation functions. Subsequently, the calculated output is compared to the ground truth labels using a chosen loss function, facilitating the quantification of prediction errors. Through the mechanism of backpropagation, gradients of the loss function with respect to the network parameters are computed and leveraged to update the model's weights and biases iteratively. This iterative optimization process, often driven by optimization algorithms like gradient descent, endeavors to minimize the loss function, aligning the model's predictions more closely with the true labels.

While the MLP Classifier boasts numerous advantages, such as its capability to model intricate nonlinear relationships and its applicability across diverse domains, it is not devoid of challenges. The complexity inherent in tuning its architecture and hyperparameters demands meticulous attention, and training can entail significant computational resources. Moreover, the risk of overfitting looms large, necessitating the adoption of regularization techniques to prevent the model from memorizing noise in the training data. Despite these considerations, the MLP Classifier remains a versatile and potent tool in the machine learning toolkit, offering unparalleled flexibility and performance in a wide array of classification tasks.

B. Random Forest*:* The Random Forest algorithm is a formidable ensemble learning technique widely acclaimed for its versatility and efficacy in both classification and regression tasks within the realm of machine learning. At its core, Random Forest comprises an ensemble of decision trees, each trained independently on a random subset of the training data and features. This inherent randomness injected into the training process fosters diversity among the individual trees, mitigating the risk of overfitting and enhancing the ensemble's ability to generalize well to unseen data.

During the training phase, Random Forest operates through a series of key mechanisms that collectively contribute to its robust performance. Firstly, the algorithm employs bootstrapping to create multiple bootstrap samples from the original training data, thereby facilitating the construction of diverse decision trees. Additionally, at each node of every decision tree, a random subset of features is considered for splitting, further enhancing the diversity of the ensemble and preventing individual trees from becoming overly specialized to the training data.

The training process culminates in the aggregation of predictions from all decision trees within the ensemble. For classification tasks, the final prediction is determined through majority voting among the individual tree predictions, while for regression tasks, it is computed as the average of the predictions. This ensemble aggregation mechanism not only fosters robustness by leveraging the collective wisdom of multiple trees but also provides resilience against noise and outliers in the data.

Random Forest's allure lies in its ability to deliver high predictive accuracy while simultaneously offering insights into feature importance. By virtue of its ensemble nature, Random Forests are adept at capturing complex relationships within the data and can handle a wide array of data types and structures. However, it is important to acknowledge the computational complexity associated with training Random Forest ensembles, particularly for large datasets with numerous features.

Thus, the Random Forest algorithm stands as a stalwart in the machine learning landscape, embodying a balance of accuracy, robustness, and interpretability. Its ability to effectively address both classification and regression tasks, coupled with its resistance to overfitting and feature importance analysis, renders it a quintessential tool in the data scientist's arsenal. Nonetheless, practitioners should judiciously weigh the computational resources required against the algorithm's performance benefits, ensuring an optimal fit for their specific use case.

C.Decision Tree: The Decision Tree algorithm is a foundational method in machine learning, celebrated for its simplicity, interpretability, and effectiveness in solving classification and regression problems. At its core, a Decision Tree recursively partitions the feature space based on the most informative attributes, creating a hierarchical structure akin to a tree. Beginning with the entire dataset at the root node, the algorithm iteratively selects features and thresholds to split the data into increasingly homogeneous subsets until a stopping criterion is met. These splits are determined based on criteria such as Gini impurity or information gain, aiming to maximize the purity of the resulting subsets. Each leaf node represents a final decision or prediction, determined by the majority class in classification tasks or the average value in regression tasks.

One of the primary advantages of Decision Trees lies in their interpretability. The resulting tree structure provides clear insights into the decision-making process, allowing users to understand the underlying logic and factors driving predictions. Moreover, Decision Trees can handle both numerical and categorical data, making them versatile for various types of datasets. However, their susceptibility to overfitting is a notable concern, particularly when dealing with complex datasets or trees with excessive depth.

Pruning involves removing unnecessary branches, can help alleviate this issue by promoting simpler tree structures that generalize well to unseen data. Despite their limitations, Decision Trees remain a cornerstone of machine learning, valued for their intuitive nature and ability to deliver reliable predictions across a range of applications.

IV. METHODOLOGY

The topology module lays the groundwork for an efficient network structure, crucial for effective traffic monitoring and control. By strategically arranging switches, routers, and other network devices, emphasis is placed on optimizing security measures while ensuring seamless data flow. The data collection module, highlighting its pivotal role in continuous monitoring of data packets' ingress and egress. This section elaborates on the strategic surveillance techniques employed to collect and analyse both normal and potentially malicious traffic patterns. Subsequently, the detection module is explored in depth, showcasing the integration of machine learning techniques to identify and analyse network anomalies, particularly DDoS attacks.

The module's ability to distinguish between normal and malicious activities, providing timely alerts and insights into potential security threats, is emphasized. The mitigation strategies employed to counter DDoS attacks. It elucidates how the mitigation module dynamically responds to detected threats by leveraging SDN capabilities. Key strategies, such as targeted blocking of MAC addresses associated with malicious traffic, are explained in detail. Additionally, the importance of intelligent and adaptive responses to mitigate the impact of DDoS attacks is underscored. Real-world examples and case studies may be included to illustrate the effectiveness of these mitigation strategies.

The evaluation of DDoS attack detection accuracy, showcasing the integration of supervised and unsupervised learning algorithms such as Multi-Layer Perceptron (MLP) Classifier, Decision Tree, and Random Forest. Details are provided on how these algorithms enhance detection performance in the SDN environment. The conclusion encapsulates the project's achievements, underscoring its contribution to bolstering IoT security through advanced methodologies and technologies. Future directions and areas for further research may also be discussed to provide a comprehensive outlook on the project's implications and potential impact.

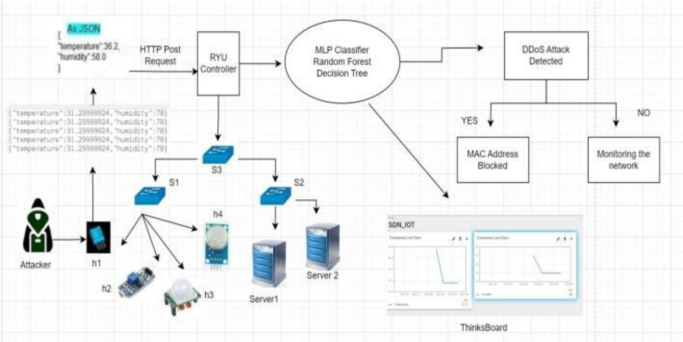
As JSON
{
"temperature":36.2,
"humidity":58.0
}
{"temperature":31.29999924,"humidity":78}
{"temperature":31.29999924,"humidity":78}
{"temperature":31.29999924,"humidity":78}
{"temperature":31.29999924,"humidity":78}
{"temperature":31.29999924,"humidity":78}
HTTP Post Request
RYU Controller
MLP Classifier Random Forest Decision Tree
DDoS Attack Detected
YES
NO
MAC Address Blocked
Monitoring the network
S3
S1
S2
h4
Attacker
h1
h2
h3
Server1
Server 2
SDN_IOT
ThinksBoard

Fig 4: System Design

V. CONCLUSION

In conclusion, this project on "Enhancing IoT Security Using Software-Defined Networks (SDN)" represents a comprehensive and innovative approach to mitigating Distributed Denial of Service (DDoS) attacks in IoT environments. Through meticulous topology design, continuous data collection, advanced detection techniques, and adaptive mitigation strategies, the project addresses the critical security challenges faced by IoT networks. By leveraging the capabilities of SDN and integrating machine learning algorithms, the project achieves significant improvements in detecting and mitigating DDoS attacks, thereby enhancing the overall security posture of IoT deployments. The project's success lies in its holistic approach, which combines network design principles, data analysis techniques, and cutting-edge technologies to fortify IoT security. Notably, the evaluation of DDoS attack detection accuracy using supervised and unsupervised learning algorithms underscores the project's commitment to rigorous assessment and continuous improvement.

REFERENCES

[1] M. Tsagkaropoulos, I. Politis, C. Tselios, T. Dagiuklas, and S. Kotsopoulos, “Service continuity over intertechnology rats,” in 2011 IEEE 16th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), June 2011, pp. 117– 121.

[2] D. Kreutz, F. M. V. Ramos, P. E. Verssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software- defined networking: A comprehensive survey,” Proceedings of the IEEE, vol. 103, no. 1, pp. 14–76, Jan 2015.

[3] D. Kreutz, F. M. Ramos, and P. Verissimo, “Towards secure and dependable software-defined networks,” in Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, ser. HotSDN ’13. New York, NY, USA: ACM, 2013, pp. 55–60. [Online]. Available: http://doi.acm.org/10.1145/2491185. 2491199

[4] D. Athanasopoulos, I. Politis, A. Lykourgiotis, C. Tselios, and T. Dagiuklas, “End-to-end quality aware optimization for multimedia clouds,” in 2016 International Conference on Telecommunications and Multimedia (TEMU), July 2016, pp. 1–5.

[5] C. Tselios, K. Birkos, P. Galiotos, S. Kotsopoulos, and T. Dagiuklas, “Malicious threats and novel security extensions in p2psip,” in 2012 IEEE International Conference on Pervasive Computing and Communications Workshops, March 2012, pp. 746–751.

[6] S. Scott-Hayward, S. Natarajan, and S. Sezer, “A survey of security in software defined networks,” IEEE Communications Surveys Tutorials, vol. 18, no. 1, pp. 623–654, Firstquarter 2016.

[7] S. Shin and G. Gu, “Attacking software-defined networks: A first feasibility study,” in Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, ser. HotSDN ’13. New York, NY, USA: ACM, 2013, pp. 165–166. [Online].
Available: http://doi.acm.org/10.1145/2491185.2491220

[8] P. Fonseca, R. Bennesby, E. Mota, and A. Passito, “A replication component for resilient openflow-based networking,” in 2012 IEEE Network Operations and Management Symposium, April 2012, pp. 933–939.

[9] G. Yao, J. Bi, and L. Guo, “On the cascading failures of multi-controllers in software defined networks,” in 2013 21st IEEE International Conference on Network Protocols (ICNP), Oct 2013, pp. 1–2.

[10] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: Enabling innovation in campus networks,” SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp.69–74,March 2008.[Online]. Available: http://doi.acm.org/10.1145/1355734.1355746

[11] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, “Security in software defined networks: A survey,” IEEE Communications Surveys Tutorials, vol. 17, no. 4, pp. 2317–2346, Fourthquarter 2015.

[12] F. Callegati, W. Cerroni, and M. Ramilli, “Man-in-the-middle attack to the https protocol,” IEEE Security Privacy, vol. 7, no. 1, pp. 78–81, Jan 2009.

[13] T. Dierks, “The Transport Layer Security (TLS) protocol version 1.2,” https://tools.ietf.org/rfc/rfc5246.txt, [Online].

[14] M. Liyanage and A. Gurtov, “Secured vpn models for lte backhaul networks,” in 2012 IEEE Vehicular Technology Conference (VTC Fall), Sept2012, pp. 1–5.

[15] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,”
Comput. Netw., vol. 54, no. 15, pp. 2787–2805, Oct. 2010. [Online].
Available: http://dx.doi.org/10.1016/j.comnet.2010.05.010

[16] Gartner Inc., “Gartner Identifies the Top 10 IoT Technologies for 2017 and 2018,” http://www.gartner.com/newsroom/id/3221818, [Online].

[17] M. Chiang and T. Zhang, “Fog and iot: An overview of research opportunities,” IEEE Internet of Things Journal, vol. 3, no. 6, pp. 854– 864, Dec 2016.

[18] A. Francillon, Q. Nguyen, K. B. Rasmussen, and G. Tsudik, “A minimalist approach to remote attestation,” in Proceedings of the Conference on Design, Automation & Test in Europe, ser. DATE ’14. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 2014, pp. 244:1–244:6. [Online].
Available: http://dl.acm.org/citation.cfm?id=2616606.2616905

[19] K. Christidis and M. Devetsikiotis, “Blockchains and smart contracts for the internet of things,” IEEE Access, vol. 4, pp. 2292–2303, 2016.

[20] C. Tselios and G. Tsolis, “On QoE-awareness through Virtualized Probes in 5G Networks,” in Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2016 IEEE 21st International Workshop on, 2016, pp. 1–5.

[21] I. Politis, C. Tselios, A. Lykourgiotis, and S. Kotsopoulos, “On optimizing scalable video delivery over media aware mobile clouds,” in IEEE International Conference on Communications (ICC), 2017, pp. 1–6.

[22] G. Bianchi, E. Biton, N. Blefari-Melazzi, I. Borges, L. Chiaraviglio, P. Cruz Ramos, P. Eardley, F. Fontes, M. J. McGrath, L. Natarianni et al., “Superfluidity: a flexible functional architecture for 5g networks,” Transactions on Emerging Telecommunications Technologies, vol. 27, no. 9, pp. 1178–1186, 2016.

[23] IBM Corp., “Blockchain benefits for electronics - White Paper,”https://public.dhe.ibm.com/common/ssi/ecm/gb/en/gbe0380 9 usen/ GBE03809USEN.PDF, [Online].

[24] Microsoft Corp.,“Blockchain as a Service,”
https://azure.microsoft.com/ en-us/solutions/blockchain/?, [Online].

[25] The Linux Foundation, “Hyperledger project,”
https://www.hyperledger.org/, [Online].

[26] Ericsson,“Data-centric security,” http://cloudpages.ericsson.com/ data- centric-security-ebook, [Online].

[27] Citrix Systems Inc., “Netscaler: Secure Event Delivery Controller,”[Online].