International Journal of Engineering Sciences Paradigms and Researches (IJESPR)
Volume 52, Issue 04 and Publication Date: 19th June, 2023
An Indexed, Referred and Peer Reviewed Journal
ISSN: 2319-6564
www.ijesonline.com

A Review on Role of ML Models to Software Defined Networks

Hanni[1], Dr. Vinit Kumar Lohan[2], Dr. Deepak Goyal[3], Dr. Pankaj Gupta[3] and Dr. Bijender Bansal[4]

[1]M.Tech. Scholar, Department of CSE, Vaish College of Engineering, Rohtak, Haryana, India

[2]Assistant Professor, Department of CSE, Vaish College of Engineering, Rohtak, Haryana, India

[3]Professor, Department of CSE, Vaish College of Engineering, Rohtak, Haryana, India

[4]Associate Professor, Department of CSE, Vaish College of Engineering, Rohtak, Haryana, India

Abstract

The control plane and data plane are separated by an emerging technology called software defined networking (SDN). The data plane transmits the traffic in line with the decisions made by the control plane for how to manage it. The SDN also unifies the control plane, allowing different data plane components to operate under the control of a single software control program. Through a well-defined Application Programming Interface (API), the SDN control plane has direct control over the state of the network's data-plane components. Regrettably, granular security measures were not effectively supported by the SDN network architecture. This is brought on by the absence of strongly linked control and information planes. The ability to handle network packet abnormalities successfully is also one of the technology's major problems since SDN is logically centralized, making it susceptible to anomalous control plane attacks that might bring down the whole network system. Several initiatives are presently underway to make existing network technology programmable in a manner that follows SDN principles. The present SDN anomaly detection methods, on the other hand, are infamous for their poor overall performance and accuracy. They do not, however, encourage progressive learning. Therefore, it is essential to build high-precision real-time classification models for aberrant flows and optimize existing methodologies. In addition, the flow detection algorithm must handle gigabit level networks in real-time. The objective of this project is to provide a powerful anomaly detection technique for SDN flows. This is shown in the research by employing Transudative learning and Deep Neural Networks to identify irregular flows. Deep neural networks are being improved to have a higher learning capacity while using less time and resources.

Keywords: *SVM, TL-DNN, KNN.*

Introduction

Today's applications emigrate from traditional networking structures to Software Defined Networking (SDN) due to flexibility property of SDN that allows user applications to run faster and in appropriate network control. SDN provides a virtualized network for storage of data there by separating forwarding mechanisms from controlling mechanisms. SDN controller is the primary central core part in SDN who is responsible for handling network and flow control management and so it is more vulnerable to attacks. These attacks must be discovered by an Intrusion Detection System (IDS), which is the main aim behind our work. This chapter describes Software Defined Networking as an idea, its architecture, and its applications in various areas. Various challenges over SDN are describes the most important key challenge i.e., security in terms of various attacks; various types of Intrusion Detection Systems (IDS) that can be designed for detection of anomalies.

SDNs: Software Defined Networks

Software Defined Networking (SDN) is a developing technology that enables us to increase the administration and design of networks more effectively. It is a framework developed by changing tried-and-true techniques for organizing and controlling networks. It has two essential qualities [1]. The division of the Control Plane (CP) and Data Plane (DP) is the initial feature of SDN. DP delivers the traffic while CP manages it in accordance with commands from the control plane. The second feature of SDN that unifies the control plane is the ability to handle switches, routers, and various middleboxes with a single software control program.

Through the use of a well-defined application programming interface, or API, such as OpenFlow,

CP has immediate influence over the state of the network's DP components. Fine-grained and QoS-aware traffic engineering is not allowed by the present network architecture due to its challenging and poor design [2]. The tight integration of the CP and DP makes tracking network traffic more efficient. Problematic, resulting in insufficient QoS-aware flow control and inadequate resource allocation, managing network devices separately is difficult, time-consuming, and costly. According to the Enterprise Strategy Group (ESG), 40% of the most frequent network operations issues may be the result of network management operators having to do more manual chores and reconfigure their networks as data centers grow [3]. Traditional networks lack self-control and the ability to adapt. This causes activation via erratic failures and load fluctuations in extended networks, in addition to the difficulties in setup. The underlying network architecture cannot instantly satisfy the requirements of the application layer since it is not programmable. On the other side, a novel network architecture known as SDN promises easier network management, better resource usage, and improved network development and innovation. In contrast to traditional network systems, SDN offers design freedom and programmability by isolating network devices from management and enabling both planes to evolve independently. A centralized SDN controller makes it easier to install, update, and manage traffic-forwarding devices in SDN architecture [2]. Network monitoring and setup become easier since they can be controlled from a single location inside the network and significantly lower operating costs.

Benefits of SDN

In addition to the ability to divide planes, SDN also provides a lot of wonderful advantages, including design flexibility and simplicity in updating [4]. These are a few justifications for these characteristics.

Abstraction

Location flexibility is described in this SDN feature. While accessing network assets, users don't have to be concerned about their real charges, setups, or locations.

Dynamic Scaling

SDNs are particularly scalable since they may change in size and quantity according on the situation. If there are more workloads, locations, devices, and resources than the network can handle. The virtualization concept makes SDN scalability viable.

Orchestration and Programming

SDNs may swiftly change their behavior and are easy to program. They can also control and manage a variety of devices with a single command.

Automated and Economic

SDNs are largely automated, meaning there is no need for human intervention or troubleshooting, which lowers operating costs and downtime. When adding additional workloads, locations, or devices, SDN may automatically deliver, re-provision, and partition the resources etc.

Performance

SDNs optimize network device use to raise network performance. SDN conducts traffic management, load balancing, bandwidth management, and can quickly recover from a few errors.

Service Integration

SDN can efficiently place resources like as load balancers, firewalls, intrusion detection systems (IDS), and other resources on the traffic path as needed.

SDN Architecture

SDN is a concept that aims to move away from the traditional completely dispersed architecture and toward a more centralized one. This is accomplished by dividing the functionalities associated with both planes into separate pieces [5]. Switches in SDN are detached from the CP and simply assist the data plane with controllers handling the manipulation. In this situation, control decisions are made by looking at the network's broader perspective status. The CP in SDN operates as a single and conceptually centralized network operating system for abstracting low-level device characteristics, scheduling, and resolving resource conflicts. But it doesn't mean that the controller is physically centralized. This logically centralized SDN controller can be distributed for scalability, consistency, performance, and reliability reasons,

allowing various physical controller instances to work together to manage the network as well as applications. As the controller knows the entire network design, it can easily swiftly scale and serve the requirements. The problem of bandwidth allotment can be overcome by dynamically configuring the controller via the disclosed northbound API.

In contrast to conventional, where the network is not aware of applications, SDN design provides applications with more information from the controller about the entire network state. Many data layer devices can be attached to centralized CP allowing the controller to see the entire network topology and allowing traffic engineers to create and implement services such as routing and security. A hierarchy, star with a single controller or even a dynamic ring type of network are all forms of control networks for SDNs.

The broad architecture of SDN with three different layers and two Application Programming Interfaces (APIs) that are used for communication between three layers as represented in Figure 1.1.
SDN separates the CP (networking logic) from the DP components and gives them to an outside entity to abstract the network layers (controller). Southbound and northbound interfaces are terms used to describe the SDN architectural APIs.

SDN Planes

In SDN architecture, the dimensions of network can be determined with the help of planes. In SDN, there are three types of planes used for data forwarding and controlling purposes. These planes are described as follows:

Forwarding Plane

Depending on instructions from the control plane, the Forwarding Plane (FP) manages the processing of packets in the data stream. This plane performs a variety of operations, including replicating, tossing out, forwarding, and changing packets. The forwarding plane often terminates the services and applications of the control plane. In order to do simple forwarding, the device looks up the address in the address database to determine the proper output port.
An IDS is designed to keep track of network activity after data collection and match patterns to known attacks. This method, additionally referred to as pattern correlation, may be used by an intrusion defense system to determine if unusual behavior qualifies as a cyberattack. When it identifies suspicious or hazardous behavior, an intrusion detection system will notify certain specialists or IT administrators. You may quickly begin debugging, identify the root of issues, or locate and eliminate harmful agents thanks to IDS alerts. Anomaly-based intrusion detection, state-full protocol analysis, while signature-based intrusion detection are the three primary intrusion detection methods used by systems for detecting intrusions (IDS).Network traffic and log data are matched to known attack patterns in signature-based intrusion detection to identify potential threats. Sequences are the name given to these patterns, and they may include byte sequences, sometimes known as risky instruction sequences. By using signature-based detection, you may precisely locate and classify potential known threats.

Anomaly-Based System for Detecting Intrusions

Anomaly-based IDS was developed to detect attacks from unknown malware. Machine learning is used in anomaly-based IDS to create a trustworthy activity model that is compared to anything incoming and is classified as suspicious if it is not included in the model. Machine learning-based IDS offers a stronger general characteristic than signature-based IDS since these models may be trained based on the applications and hardware settings. An overview of some of the anomaly-based intrusion detection systems developed recently is provided below: The SDN architecture is a crucial component of this technology, which has drawn significant attention from academics and business experts. The control plane and the data plane are kept separate in an SDN configuration. It enables network administrators to more effectively and creatively control network traffic. A unified picture of the network is provided by this networking technique, which may, among other advantages, make network administration simpler and increase network security. There is an urgent need for a detailed investigation of the architecture of the SDN technology, which should include a look at its layers. This will make it possible to understand the limitations and potential of the technology better.

Infrastructure Layer

The infrastructure layer is provides the physical network infrastructure, including routers, switches,

as well as other network devices, as the lowest layer in the SDN architecture. The interfaces needed to connect with the underlying physical network infrastructure are also provided by this layer. The separation of the control and data planes is made possible by it, which serves as the core of the SDN architecture. Traditional networking devices are replaced at this layer with switches that support OpenFlow and may be controlled from a central location by the SDN controller. To regulate traffic flow, the controller may connect with the underlying network devices using the OpenFlow protocol, enabling centralized network administration and control. The infrastructure layer also offers interfaces for managing network devices, such as configuring, monitoring, and reporting status. The infrastructure layer's capacity to streamline network administration and boost network flexibility is one of its main advantages. Without having to manually configure each network device, the network administrator may simply alter network rules, traffic routing, and other network parameters using SDN. Additionally, the infrastructure layer supports dynamic provisioning of network resources, allowing for the as-needed allocation of resources. The SDN infrastructure layer is a crucial element of contemporary networking topologies because of its flexibility and agility. However, the infrastructure layer also presents several challenges. First, SDN requires significant investments in new hardware and software infrastructure to enable OpenFlow-enabled switches and SDN controllers. Additionally, network administrators must be trained to manage the new SDN infrastructure, which may require new skills and knowledge. Finally, SDN infrastructure must be carefully planned and designed to ensure that it is scalable, secure, and reliable. Overall, the infrastructure layer is critical to the success of the SDN architecture and requires careful consideration in the design and implementation of SDN networks.

Control Layer

Centralized network control, a crucial element of the SDN architecture, is the responsibility of the Control Layer. It is made up of the components that regulate network configuration, traffic forwarding policies, and topology. The main element of the control layer is the SDN controller, which acts as the network operating system. It makes use of software that manages and steers network traffic in line with established rules. A logical representation of the network is provided to the applications running on top of the SDN controller. The Infrastructure Layer controller communicates with the switches using a standard protocol, such as OpenFlow. The control layer also includes the network applications that run on top of the SDN controller. To carry out network services and regulations, the controller and these applications converse. A few examples of SDN applications include load balancers, firewalls, intrusion detection systems, and other traffic engineering modules. The Control Layer's ability to keep the control plane and data plane apart is one of its key features. This division enables network administration to be centralized, improving network programmability and automation. Additionally, it makes it possible for finer-grained network traffic management, enhancing network security and resilience. Additionally, the Control Layer is in charge of preserving the dependability and scalability of the network. Distributed deployment of SDN controllers enables redundant and fault-tolerant setups. The Control Layer also offers methods for managing and recovering from network faults.

Application Layer

The uppermost layer in the software-defined networking architecture, known as the Application Layer, is in charge of delivering services to network-based applications. It is made up of many application programming interfaces (APIs), which let network applications communicate with the SDN controller and manage the network. Network applications may function on an abstracted representation of the network topology thanks to the Application Layer, which offers a centralized view of the network and exposes this view to users. The SDN controller provides its northbound application-programming interface (API) to the different network applications at the application layer. Network applications may connect with the controller and program the network via the northbound application-programming interface (API), which is a set of interfaces made accessible to them. The northbound application-programming interface (API) facilitates the development of network applications that are independent of the underlying network hardware since it is vendor-neutral by design.

Network operations like load balancing, traffic engineering, intrusion detection, and firewalling may all be implemented using network apps at the application layer. It is possible to expand the capability of the SDN infrastructure beyond what is

offered by the controller manufacturer by having third-party developers create these applications. In general, the SDN architecture's Application Layer offers a flexible and expandable framework for network application development, allowing the development of novel and unique network services

SDN Features

SDN has emerged as a new approach to designing and managing networks. It offers several unique features that set it apart from traditional networking. Firstly, The control plane and data plane are separated by SDN. This indicates that sophisticated routing choices are no longer made by network devices, but rather by a centralized controller. This makes managing the network easier and gives you more freedom and control over how the network behaves. Second, SDN bases forwarding choices on flows rather than the origin and objective addresses. A flow is a group of packets with similar properties, such as the same protocol type, source and destination ports, and time interval. SDN may provide more effective and efficient network use by basing forwarding choices on flows rather than individual packets. Thirdly, SDN offers a software encapsulation of network logic and a global network view. SDN enables increased network programmability and automation by shifting the control logic to a third-party entity, such as a network operating system or an SDN controller. This enables network operators to write software applications that can communicate with the underlying data layer hardware, and automate network management tasks, such as traffic engineering, load balancing, and security. Finally, SDN allows networks to be programmed using software applications that communicate with the underlying data layer hardware that runs on top of the network operating system. This software-defined approach to network management provides greater agility and flexibility, enabling operators to quickly and easily adapt to changing network conditions and user needs. Additionally, the network can be set up with SDN's software/hardware links, allowing forwarding devices to be changed to execute applications, further increasing the network's flexibility and adaptability.

Software Defined Networking Controller

In software-defined networking (SDN), the controller serves as the "brain" of the network, as it determines the routing path for each new flow. Without the controller, the network would be vulnerable to attacks and potential destruction. There are different types of controllers that can manage the control plane, including a single controller or multiple controllers. However, the architecture of the controller greatly impacts the performance of the SDN. Recently, three new controller architectures were introduced, which include distributed, multi-core, and logically centralized controllers. Distributed controllers synchronize their conditions with other controllers to produce an optimal global solution and maintain a comprehensive network picture. On the other hand, logically centralized controllers are composed of multiple distributed controllers, enabling them to respond more quickly to each flow request and handle more requests per second. Although exchanging data between controllers using network services is necessary, it can also create a heavy load on the network. To improve scalability and maintain data consistency, efforts must be made to lessen the load of state synchronization between controllers.

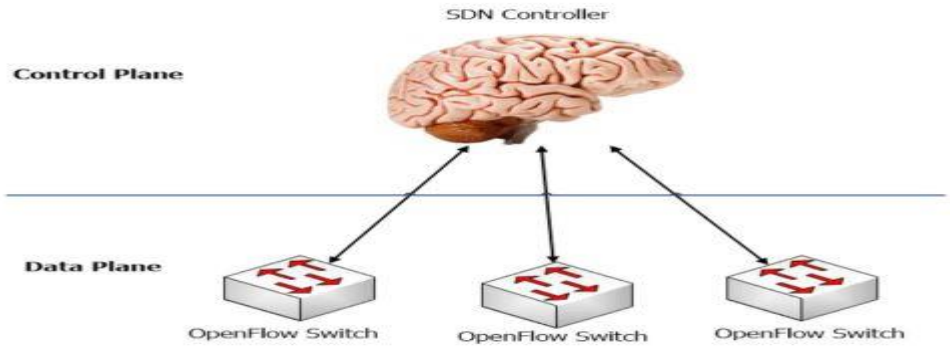

Figure 1: Controller as Brain of SDN

Benefits of SDN Framework

- Operational Saving: SDNs cut operational costs by automating configuration and management activities, which frees up network administrators to focus on other priorities. Application owners receive pre-packaged network services, which frees up the networking team to focus on other tasks.
- Flexibility: SDNs make it possible for the network to be utilized and operated in a number of different manners. Resellers have the ability to design their very own network services by utilizing the standard development tools.
- Increased Uptime: SDNs make it possible for resellers to prevent configuration and deployment problems, which could potentially interrupt the network, by doing away with the need for manual involvement.
- Improved Management: Managed Service Providers (MSPs), also known as cloud service providers, are able to manage virtual networking, computing, and storage resources using a unified point of view and toolkit.
- Planning: With a more complete understanding of their customers' network, compute, and storage capabilities, resellers are better able to plan IT efforts on their clients' behalf.
- Infrastructure Cost Savings: By decoupling, route/switching knowledge from packet forwarding, routers and switches can compete on price-performance attributes.

Conclusion

The evaluation of AI algorithms for intrusion detection systems, which is assessed using common evaluation metrics provided in Table 7. These evaluation measures are based on the various attributes employed in the algorithms.
Several typical assessment metrics, such as the true positive (TP) rate, the true negative (TN) rate, the false positive (FP) rate, and the false negative (FN) rate are used in the process of evaluating artificial intelligence algorithms for use in intrusion detection systems. The term "TP rate" refers to the proportion of actual positive events that the algorithm properly identifies as positive. The TN rate is the proportion of actual negative instances that the algorithm properly identifies as being negative. It is also abbreviated as "TN rate." The false positive rate (FP rate) is the proportion of genuine negative cases that are wrongly identified as positive, while the false negative rate (FN rate) is the proportion of actual positive instances that are incorrectly identified as negative. These metrics are necessary for determining whether or not AI algorithms are successful in recognizing and locating potential dangers to a network's security.

References

[1] Suri, M., Gupta, A., Parmar, V., & Lee, K. H. (2019, May). Performance enhancement of edge-ai-inference using commodity MRAM: Iot case study. In 2019 IEEE 11th International Memory Workshop (IMW) (pp. 1-4). IEEE.

[2] X. Lin, J. Li, J. Wu, H. Liang and W. Yang, "Making Knowledge Tradable in Edge-AI Enabled IoT: A Consortium Blockchain-Based Efficient and Incentive Approach," in IEEE Transactions on Industrial Informatics, vol. 15, no. 12, pp. 6367-6378, Dec. 2019, doi: 10.1109/TII.2019.2917307.

[3] T. A. Al-Janabi and H. S. Al-Raweshidy, "A Centralized Routing Protocol With a Scheduled Mobile Sink-Based AI for Large Scale I-IoT," in IEEE Sensors Journal, vol. 18, no. 24, pp. 10248-10261, 15 Dec.15, 2018, doi: 10.1109/JSEN.2018.2873681.

[4] I. García-Magariño, R. Muttukrishnan and J. Lloret, "Human-Centric AI for Trustworthy IoT Systems With Explainable Multilayer Perceptrons," in IEEE Access, vol. 7, pp. 125562-125574, 2019, doi: 10.1109/ACCESS.2019.2937521.

[5] T. Sutjarittham, H. Habibi Gharakheili, S. S. Kanhere and V. Sivaraman, "Experiences With IoT and AI in a Smart Campus for Optimizing Classroom Usage," in IEEE Internet of Things Journal, vol. 6, no. 5, pp. 7595-7607, Oct. 2019, doi: 10.1109/JIOT.2019.2902410.

[6] Y. Lin, Y. Lin, C. Liu, J. Lin and Y. Shih, "Implementing AI as Cyber IoT Devices: The House Valuation Example," in IEEE Transactions on Industrial Informatics, vol. 16, no. 4, pp. 2612-2620, April 2020, doi: 10.1109/TII.2019.2951847.

[7] F Zafari, A Gkelias and K.K Leung, "A Survey of Indoor Localization Systems and Technologies", IEEE Communications

Surveys & Tutorials, vol. 21, pp. 2568-2599, 2019.

[8] A. Longo, M. Rizzi, D. Amendolare, S. Stanisci, R. Russo, G. Cice, et al., "Localization and monitoring system based on ble fingerprint method", CEUR Workshop Proceedings, vol. 1982, pp. 25-32, 2017.

[9] S. Xia, Y. Liu, G. Yuan, M. Zhu and Z. Wang, "Indoor fingerprint positioning based on Wi-Fi: an overview", ISPRS Int. J. Geo-Inf., vol. 6, pp. 135, 2017.

[10] H. Suining and ChanS. H. Gary, "Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons", IEEE Commun. Surv Tut, vol. 18, pp. 466-490, 2016.

[11] J.S. Lee, M. F. Dong and Y.H. Sun, "A preliminary study of low power wireless technologies: ZigBee and Bluetooth Low Energy", 10th IEEE Conf. on. Industrial Electronics and Applications (ICIEA), 2015.

[12] M D'Aloia, F Cortone, G Cice, R Russo, M Rizzi and A Longo, "Improving energy efficiency in building system using a novel people localization system", 2016 IEEE Workshop on Environmental Energy and Structural Monitoring Systems EESMS 2016, 2016.

[13] Paterna Cantón, Vicente et al., "A bluetooth low energy indoor positioning system with channel diversity weighted trilateration and kalman filtering", Sensors, vol. 17.12, 2017.

[14] M. D'Aloia, A. Longo, R. Ruggero, S. Stanisci, D. Amendolare, M. Rizzi, et al., "An innovative LPWA network scheme to increase system reliability in remote monitoring", 2017 IEEE Workshop on Environmental Energy and Structural Monitoring Systems EESMS 2017, 2017.

[15] Zhang Zhongheng, "Introduction to machine learning: k-nearest neighbors", Annals of translational medicine, vol. 4.11, 2016.

[16] C.H. Chen and R.S. Cheng, "Improving Indoor Localization Based on Artificial Neural Network Technology", EAI Endorsed Transactions on Internet of Things, vol. 4, 2018.

[17] M. D'Aloia, M. Rizzi, R. Russo, M Notarnicola and L. Pellicani, "A marker-based image processing method for detecting available parking slots from UAVs", Lecture Notes in Computer Science, vol. 9281, pp. 275-281, 2015.

[18] A.B. Adege, Y. Yayeh, G Berie, H. Lin, L. Yen and Y.R. Li, "Indoor localization using K-nearest neighbor and artificial neural network back propagation algorithms", the 27th Wireless and Optical Communications Conference (WOCC2018), 2018.

[19] A. Gholoobi and S. Stavrou, "RSS based localization using a new WKNN approach", Computational Intelligence Communication Systems and Networks (CICSyN) 2015 7th International Conference on, pp. 27-30, 2015.