DOI: https://doi.org/10.33103/uot.ijccce.23.3.6

A Review on Utilizing Big Data Techniques for Performance Improvement in Software Defined Networking

Sama Salam Samaan[1], Hassan Awheed Jeiad[2]
[1,2]*Computer Engineering Department, University of Technology, Baghdad, Iraq*
[1]*sama.s.samaan@uotechnology.edu.iq*, [2]*hassan.a.jeiad@uotechnology.edu.iq*

Abstract— *Traditional network abilities have a drastic shortage in the current networking world. Software-Defined Networking (SDN) is a revival development in the networking domain that provides separation of control and data planes, enlarges the data plane granularity, and simplifies the network devices. All these factors accelerate and automate the evolution of new services. However, when the SDN network topology becomes large, it poses new challenges in security, traffic management, and scalability due to the vast amounts of traffic data generated and the need for additional controllers to manage the significant number of networking devices. On the other hand, big data has become an attractive trend that can enhance network performance in general, specifically SDN. Both SDN and big data have gained great attraction from industry and academia. Traditionally, these two subjects have been studied separately in most of the preceding works. However, big data can impact the design and implementation of SDN thoroughly. This paper presents how big data can support SDN in various aspects, including intrusion detection, traffic monitoring, and controller scalability and resiliency. We suggest several approaches toward deeper cooperation between big data and SDN.*

Index Term – *SDN, big data, Spark, graph database, traffic engineering, intrusion detection.*

I. INTRODUCTION

Big data has emerged as one of the trending topics in both industry and academia. It is defined as vast and complex datasets that conventional data processing and management tools cannot handle. Big data is characterized by its 5 Vs.; Volume, Velocity, Variety, Veracity, and Value [1].

The quantity of data from numerous sources such as social media (Twitter, Facebook, Instagram, etc.), the Internet of Things (IoT), and scientific research is expanding exponentially. In addition, big data is considered an essential network application that intensely affects the design and operation of networking, specifically SDN.

Mainly, with the global network view, the logically centralized SDN controller collects various data types with a different granularity from both data and application planes [2]. Using this data, big data analytics can gain valuable insights to make decisions related to SDN design and operation. For example, the SDN controller can utilize big data analytics to accomplish traffic monitoring to enhance SDN overall performance. Besides, big data techniques are used to detect Distributed Denial of Service (DDoS) attacks on control and data planes since the SDN architecture has some security holes that intruders can exploit [3].

Another benefit of big data in SDN is controller optimization. An essential problem in SDN is to specify the required number of controllers and their placement within the network [4]. Big data techniques can solve this problem by modeling the network topology as a graph and using graph theory to compute the required number of controllers and their locations within the network.

Received 14/December/2022; Accepted 09/January/2023

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.6

The main contributions of this paper are as follows:

- Present a review of two trending topics, SDN and big data, and show how big data can improve SDN in multiple areas, such as intrusion detection, traffic monitoring, and controller optimization.
- Propose a real-time intrusion detection system for SDN using a scalable machine learning model and a stream processing platform to detect malicious traffic flows.
- Suggest a traffic monitoring system for SDN by integrating graph database and graph processing platform as big data tools to measure and store the traffic load of each networking device and rank these devices based on their load for a specific period.
- Suggest using the Minimum Connected Dominating Set (MCDS) algorithm in a scalable and distributed platform to solve the controller placement problem.

The rest of this paper is organized as follows. Section II presents the state-of-the-art related to the collaboration between big data and SDN. Section III briefly details this work's basic concepts, including big data and SDN. Then, section IV suggests the main aspects in which big data can assist SDN, including intrusion detection, traffic monitoring, and controller optimization. Lastly, section V concludes the paper.

II. RELATED WORK

This section presents several works concerning how big data techniques can enhance SDN overall performance. The other side is how SDN can be exploited to improve big data issues, which is out of the scope of this paper. L. Cui *et al.* [5] studied the cooperation between big data and SDN and how big data can assist SDN in different aspects, such as traffic engineering, cross-layer design, and defeating security attacks. In addition, they presented how SDN can solve some critical issues in big data, such as data transportation for big data applications, data processing in cloud data centers, and scheduling in Hadoop for big data applications. However, the work in [5] gives a high-level description of the proposed system where big data assists SDN in multiple traffic engineering issues. The implementation of the two newly added components, Big Data Application and Traffic Engineering Manager, is not explained.

Efficient quality of service management needs an accurate and timely collection of massive data across the network. S. Jain *et al.* [6] showed the use of big data analytics to improve SDN's quality of service management. They applied multi-dimensional analysis and machine learning to find new correlations, achieve root cause analysis, and forecast traffic congestion. They used R, a big data tool for cleansing, processing, mining, computation, and visualization. Since R is a low-level language, It requires more extended codes and processing time compared to other big data languages, such as Python. In addition, it is a complicated language and lacks essential security requirements.

The growth of cloud services and Content Delivery Networks (CDN), paired with the power of encryption, becomes a crucial challenge to per-flow management and requires a more thorough approach to managing web traffic. M. Trevisan *et al.* [7] proposed a new paradigm based on a "per service" management that allows for analyzing and prioritizing the traffic of critical web services while separating others, even if they are served by the identical CDN or executing on the same cloud platform. They designed an automatic web service manager, an SDN application leveraging big data techniques to build models that automatically describe enormous web service traffic. The created models are used to set rules in the SDN switches to guide all the flows according to the originating services. However, the work didn't address the velocity and variety of the incoming flows.

A traffic matrix gives the traffic throughput between any source-target pairs in a network over a particular period [8]. W. J. Queiroz *et al.* [9] presented a near real-time paradigm for traffic matrix

Received 14/December/2022; Accepted 09/January/2023

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.6

estimation for SDN networks that employ OpenFlow (OF) as a southbound protocol. The presented method uses traffic statistics that the SDN controller had collected from the OF switches. The collected data are aggregated in a big data stream processing platform (Spark Streaming). They used a distributed computing model of MapReduce to estimate the traffic matrix for all source-target pairs of hosts. Elasticsearch, a NoSQL database, was used to persist the generated traffic matrix to be available for diverse traffic monitoring applications.

Nowadays, SDN is widely used in diverse practical domains, providing a new way to manage networks by separating the control plane from its data plane. However, it becomes vulnerable to DDoS attacks because of its centralized control. P. T. Dinh and M. Park [10] utilized big data tools in building a framework to overcome conventional limitations in data processing and detect DDoS attacks in a large-scale SDN network. The suggested framework consists of three sequential stages; data ingestion, data preprocessing, and machine learning model training and deployment. Apache Kafka, Spark Streaming, Spark core, Hadoop Distributed File System (HDFS), and Spark MLlib, are the big data tools used in the proposed framework.

Graph databases are utilized as a big data tool to handle SDN networks from a graph computing perspective. Souza et al. [11] presented a technique to augment the state of SDN networks by introducing a framework that consists of a semantic model and a graph database. They suggested importing the Network Markup Language (NML) model into the Neo4j database. However, the work in [11] didn't implement the feeding of the graph database with the required information, such as using Link Layer Discovery Protocol (LLDP) or similar methods by the SDN controller to find the underlying topology or to set the edge weights depending on the available bandwidth or link capacities.

The work in [12] applied Big Data analytics to assist traffic management in SDN networks using big graph analytics. However, the work has some limitations since they didn't address the dynamic workloads of SDN in real-time. For instance, when a new device (storage, computing, etc.) is added to the SDN network, all services should be accessible without delay. To accomplish this goal, the SDN controller needs to monitor the entire network and instantly update the infrastructure when the network changes. This update should be reflected immediately in the dynamic graph, representing a challenge in a large-scale SDN network. In addition, the authors in [12] didn't consider requesting the traffic statistics in real-time. They deployed a synthetic square grid for performance evaluation in which each node is connected to its four neighbouring nodes. The edge weights are randomly generated and uniformly distributed from 0 to 10. Table I shows the big data tool used and its aim in each of the presented works.

TABLE I. PREVIEW OF THE RELATED STATE OF THE ARTS

Ref #	Publication Year	Big Data Technique	Aim of the Work
[5]	2016	—	Review of big data and SDN cooperation
[6]	2016	R	Quality of Service management
[7]	2017	Apache Spark	Automatic web management
[9]	2020	Apache Kafka, Spark Streaming, Elasticsearch	Traffic monitoring
[10]	2021	Apache Kafka, Spark Streaming, Spark MLlib	DDoS attack detection
[11]	2015	Graph database	Traffic engineering
[12]	2018	Graph database	Traffic management

Received 14/December/2022; Accepted 09/January/2023

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.6

III. OVERVIEW OF SDN AND BIG DATA

This section concisely explains the main concepts in this work, SDN and big data.

A. SDN

Although traditional networks provide a wide range of services in different domains, it suffers from several problems [13], [14]. Some of these problems are duplication of tasks and inequality in the distribution between the control and data planes, complexity and replication of software in network devices, limitation of TCP/IP stack, overprovisioning of network resources, and management and monitoring constraints.

SDN represents the response to these problems. Data and control planes are separated in SDN, so the management functions are isolated from the forwarding devices to the applications running in the control and application planes. Thus, software complexity and duplication are reduced in the network devices, and network management and control become centralized [15]. The basic principle of SDN is the physical separation of data and control planes in networking devices. The forwarding devices transmit packets according to the rules installed into them by the SDN controller, which has all the logic to make decisions and build these rules.

In a traditional network, management is done for each device individually. In contrast, the entire network is managed at once in SDN. *Fig. 1* shows the architecture of SDN (part a) and traditional (part b) networks. We can see that the control and data planes are combined in conventional network devices while separated in SDN.

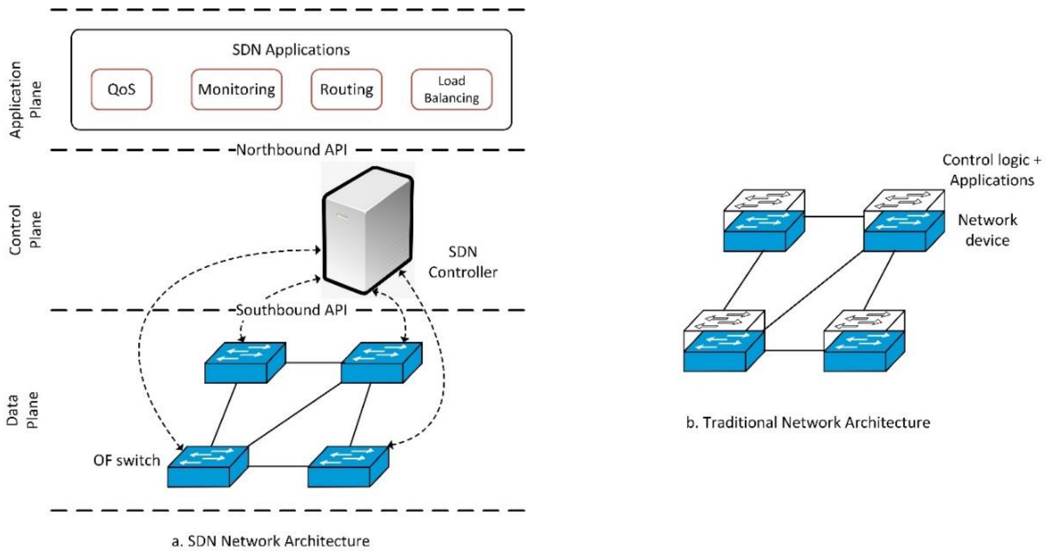

FIG. 1. SDN AND TRADITIONAL NETWORK ARCHITECTURES.

The SDN architecture is divided into three planes:

1. The data plane consists of networking devices (mainly switches and links).
2. The controller represents the control plane, which provides network management services and Application Programming Interfaces (APIs) for controlling the network.
3. The application plane consists of applications for efficient and flexible network control and management.

As seen in part (a) of *Fig. 1*, the API between the control plane and the application plane is known as Northbound API, and the API between the control plane and the data plane is known as Southbound API. The most popular and actively evolving standard for Southbound API is OpenFlow (OF) [16].

Received 14/December/2022; Accepted 09/January/2023

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.6

However, a standard for the Northbound API has still been unelaborated, eliminating application portability between various controllers.

B. Big Data

In the last decade, the amount of generated data has been phenomenal, directly resulting from technological advancement in many fields. Examples are social media superiority, the pervasive equipment connected to the Internet, and space exploration. NASA, for example, is considered one of the biggest data generators, with 12.1 TB per day from thousands of systems and sensors and 100 active missions around space and Earth [17]. The James Webb telescope that NASA designed enables a wide range of discoveries in cosmology and astronomy, like monitoring the beginning stars and galaxies. It can transmit a minimum of 57.2 GB per day of science data [18].

Therefore, it is accurate to say that this is the big data era. Big data can be defined as massive data flows or datasets that have raised the capacity to process and store. Such data can’t be analyzed in conventional ways. Mainly, challenges come to light due to one or more of the following reasons:

- Volume: recently, vast amounts of data have been generated from diverse sources, such as transactional data, healthcare data, and data generated from web crawling. Thus, conventional storage and processing systems may collapse [19]. Therefore, new algorithms and systems are built that can store, recover, and process massive amounts of data [20].
- Velocity: another indication of big data is the high data rate generation [21]. For example, Twitter requires analyzing data streams generated continuously at a very high rate in real-time [22], [23].
- Variety: occasionally, data come from various sources and in diverse forms, such as structured, unstructured, and semi-structured data [24]. As a result, it is vital to have systems that can handle various data models while preserving performance [25].
- Veracity: it indicates the quality of data to be analyzed. Data has high integrity when it consists of accurate records that contribute positively to the overall performance. In contrast, low-integrity data that contain meaningless information are considered noise.
- Value: Finally, this V indicates the usefulness of the collected data for a specific domain. It directly depends on what organizations can do with that data.

C. Big Data Platforms

Big data platforms have a set of characteristics like parallelism, transparency, and fault tolerance. Therefore, selecting the best one to perform a specific task becomes difficult. Although the comprehensive architecture of these platforms has many features in common, they can be integrated into a hierarchal stack, as shown in *Fig. 2*, that is composed of the following layers [26]:

- **Resource management:** this layer consists of platforms deployed to manage and share cluster resources among the platforms in the higher layers.
- **Data storage:** this layer has a set of platforms for storing and retrieving massive amounts of data. It includes distributed file systems that preserve data in distributed disks, messaging systems to handle real-time data, and databases to persist data in a scalable manner.
- **Data processing:** this layer consists of platforms designed for parallel data processing across a set of nodes. These platforms are classified into subgroups according to the input model and the target, such as machine learning [27], graph data processing, streaming data, etc.

Received 14/December/2022; Accepted 09/January/2023

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.6

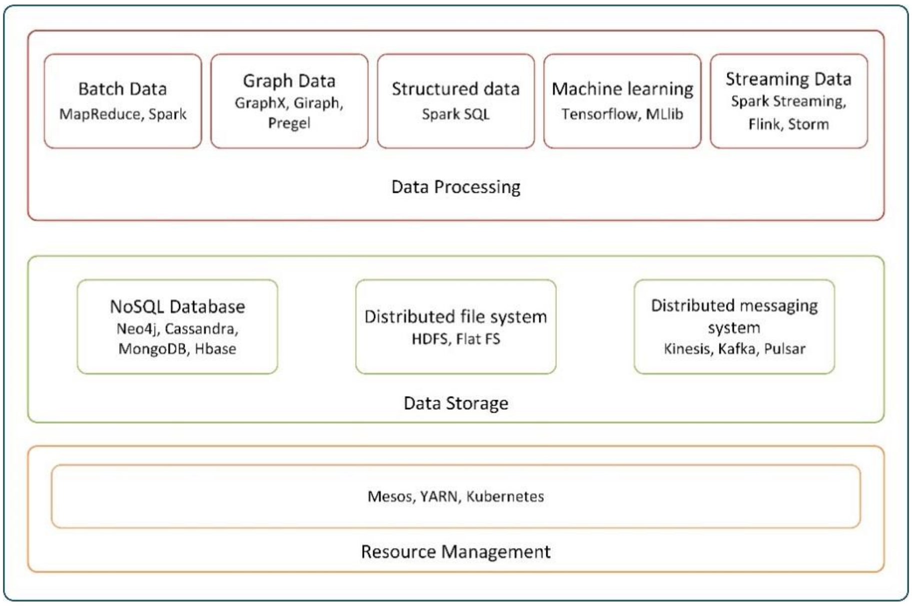

FIG. 2. HIERARCHAL STACK OF BIG DATA PLATFORMS [23].

IV. UTILIZATION OF BIG DATA TECHNIQUES IN SDN

This section presents how big data can assist SDN in multiple aspects, including intrusion detection, traffic monitoring, and controller optimization.

A. Big Data assists SDN in intrusion detection

Although SDN has gained countless attractions in the networking domain, new challenges in security have been raised [28]. Compared to traditional IP networks, SDN is identified by its centralized control and network programmability. However, centralization in SDN design may lead to critical security issues since the controller may become a single point of failure [2]. Currently, security protections are not obliged because of implementation complexity in the current SDN standards. Consequently, SDN popularity is affected in the long run. As a result, malicious attackers are attracted by these security breaches.

Even though SDN faces different threats due to its network infrastructure, it has the potential to enhance network security. Notably, SDN improves traffic monitoring using security applications executing on the application plane.

In traditional networks, security services are applied manually on network devices and considered additional features. Thus, to update a specific security policy, a network operator should alter the low-level configurations at each device. Therefore, such services have limited capabilities since there is a lack of global network information and complications in updating the network policies since it is expected to have configuration errors. In contrast, these services in SDN are provided based on traffic statistics collected by the SDN controller. Subsequently, intelligent security services are implemented in SDN by analyzing big traffic data and developing suitable security policies.

The big data analytics approach depends on the ability to store and process massive amounts of data. Big data benefits arise from learning various features to determine the optimal configurations. Besides, machine learning is considered one of the most effective tools to achieve complicated learning and make decisions using big data analytics. By creating systems that can learn from data, machine learning can intercept the variations in data, classify events and predict upcoming challenges independently.

Generally speaking, the cooperation of big data analytics and machine learning can enhance SDN security in the following issues:

Received 14/December/2022; Accepted 09/January/2023
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.6

- Sensing: intrusion detection system needs to sense anomalies from different network devices. These systems can compare the flow data collected by the SDN controller and achieve real-time anomaly detection utilizing big data analytics and supervised learning techniques. This is illustrated in *Fig. 3*, in which a DDoS intrusion detection system is proposed for the SDN network. The system is composed of two phases; offline and online. A machine learning model is created in the offline phase using a machine learning pipeline built and trained particularly for this purpose using historical data [29]. The online phase consists of three components; a messaging system, a big data stream processing platform, and the machine learning model built in the offline phase. The messaging system is responsible for transmitting traffic data from the SDN controller to the stream processing platform to perform the required analysis. The chosen messaging system should be scalable, fault-tolerant, elastic, and can transfer high volumes of data in real-time with low latency. In addition, it should be integrated conveniently with the deployed SDN controller. All these specifications exist in Apache Kafka. Practically, OpenDayLight (ODL) controller has a northbound plugin that allows real-time event streaming into Kafka. The ODL controller publishes traffic flow data as messages on Kafka using a common topic. Then, the big data stream processing platform (e.g., Spark Streaming) subscribes to that topic and acquires the message streams from Kafka. The stream processing platform represents the analytics point that performs data cleaning and preprocessing to generate the required information for the machine learning model that detects the traffic type (normal or attack) in real-time.
- Mining: hidden network service patterns can be concluded by mining the collected network state information [30]. Thus, the service operator can categorize services based on the required resources. Unsupervised learning algorithms are frequently used to conclude and explain essential attributes of the sensed information [31].
- Forecasting: big data analytics can predict network behaviours and abnormalities. Therefore, the complication of network design and management can be eliminated. Deep learning is ideal for predicting future intrusions by automatically discovering hidden traffic flow correlations [26].

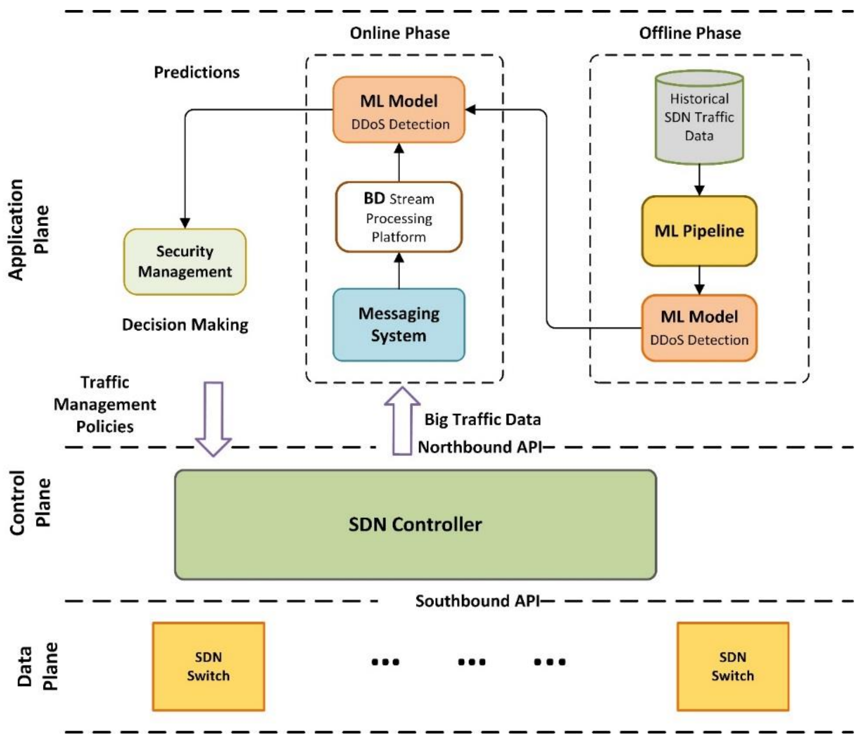

FIG. 3. THE PROPOSED MODEL FOR INTRUSION DETECTION IN SDN WITH BIG DATA ANALYTICS.

Received 14/December/2022; Accepted 09/January/2023

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.6

B. Big Data assists SDN in traffic monitoring

Traffic monitoring is a fundamental network mechanism for providing high-performance services that satisfy diverse user requirements [32]. SDN architecture introduces new features that facilitate traffic monitoring. Because the control plane is separated from the data plane, the centralized controller collects the network state information from the network devices in the data plane to construct a global network topology view.

While SDN provides an efficient control platform for traffic management, some issues must be addressed before exploiting this platform. The vast amounts of data for diverse network states and traffic loads that the SDN controller collects must be analyzed thoroughly to make suitable decisions to optimize resource utilization and service performance.

In a dynamic, large-scale SDN, the massive and varied network states and the traffic load fluctuations on the network need additional complicated data analysis and decision-making abilities. These are not offered easily by the traditional network traffic management methods currently deployed [33], [34]. Big data analytics handle data that has one or more of the following characteristics:

- Massive amounts of data (Volume)
- Data streams of a high speed (Velocity)
- Diverse data types and formats (Variety).

Therefore, big data analytics facilitate analyzing the massive, dynamic and diverse network state data to acquire correlation between multiple factors in network behaviours, consisting of network topology, traffic distribution, resource allocation, and attainable service performance. So they can provide network design and operation guidelines [35].

Big data analytics applications for traffic monitoring and control in SDN are as follows.

- Big data analytics can be used to compute the load of each switch of the SDN topology from the huge monitoring data. Distributed graph databases such as neo4j, Titan, and ArangoDB [36], [37] can store the traffic data for each switch in the SDN network. The traffic data is retrieved from the SDN controller through Application Programming Interface (API) and stored in a graph database. Naturally, the network topology can be described as a graph by modeling each switch as a node and the switch connections as edges. The traffic load of each switch is stored, and the required computations are performed, such as Bytes Received Throughput (BRT), Bytes Transmitted Throughput (BTT), and Total Bytes Throughput (TBT), to be used by a graph processing platform [38]. These two techniques can work together seamlessly to perform traffic monitoring tasks. The selection of these techniques depends on the flexible integration between the selected graph database with the deployed SDN controller and the chosen graph processing platform.

  The suggestion above is illustrated in *Fig. 4*, in which an application is developed using a graph processing platform (e.g., Spark GraphX) to compute the traffic load distribution stored in a graph database (e.g., neo4j) for each switch in the SDN data plane for a specific period (week, month) and rank switches according to their traffic load. This can assist in identifying network bottlenecks and underutilized network resources. These identifications offer instructions for network design, e.g., raising or lowering link bandwidth and switches capacity in certain network parts and continuing the network topology improvements. As seen in *Fig. 4* and according to the SDN architecture, the graph database and the graph processing platform are located in the control plane since they require high interaction with the network. Besides, this location allows such applications to respond quickly to network events.
- In addition, big data analytics can be used to analyze real-time SDN network traffic. For example, Spark Streaming is designed to effectively process and analyze streaming data received from multiple sources in a high-speed parallel manner across the available cluster nodes [39]. Using such systems, each switch load is monitored in real-time. When the switch load reaches a predefined

Received 14/December/2022; Accepted 09/January/2023
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.6

threshold, the SDN controller can reroute the flows to bypass the overloaded switches. This way, a flow-level adaptation is achieved in real-time using Spark Streaming [26].

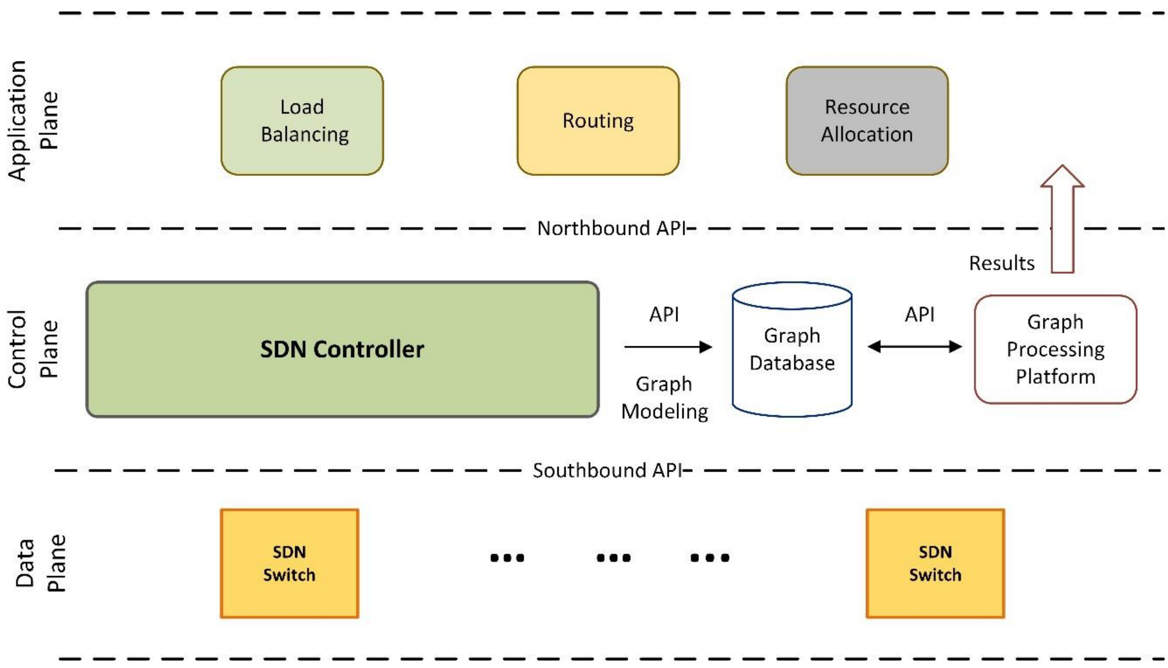

FIG. 4. THE SUGGESTED MODEL FOR TRAFFIC MONITORING IN SDN WITH BIG DATA ANALYTICS.

C. Big Data assists SDN in controller optimization

For several reasons, fundamental performance metrics, including resiliency and scalability, are vulnerable to drastic degradation in a centralized controller architecture like SDN [40]. First, with the continuous growth in network size and the rapid changes in the network state, the processing requirement increases in the controller [41]. It may reach a level that can't be manipulated in a timely way [42]. Second, with the inadequate number of controllers or inappropriate controller(s) placement within the network, several devices cannot reach the controller assigned to them. Unbearable delay may introduce even when the network devices can reach a controller physically [43]. Sometimes, the controller may become a single point of failure when there is only one controller in deployment that is placed inappropriately [44].

An urgent solution to the resiliency and scalability of the SDN control plane is employing numerous controllers in a distributed way. Nevertheless, new challenges must be considered, like determining the specific number of controllers required [45], their placement, and the network devices assigned to each controller [46]. All the solutions to the challenges above rely on the network state, which is naturally dynamic [47].

On the one hand, SDN must depend on the network traffic data to adapt to the network state and use it to optimize the controller behaviour. On the other hand, the network traffic data can be massive and fast-changing, so traditional approaches can't handle such real-time ever-growing data. Thus, the network state is qualified as a big data scenario. As a result, big data tools can support SDN in making optimal decisions concerning control plane topology and behaviour.

The Minimum Connected Dominating Set (MCDS) is a graph algorithm that can be deployed as a practical approach for determining the required number of controllers and solving the SDN controller placement to eliminate the communication time between the distributed controllers and between the controller and their assigned switches. However, when the network becomes large, the deployment of this algorithm requires a distributed graph processing platform (e.g., Spark GraphX) or a graph database

Received 14/December/2022; Accepted 09/January/2023

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.6

that can handle a graph with a vast number of nodes and edges. Neo4j is a fast and scalable graph database that can execute a graph algorithm on a large-scale graph stored in the database. The suggestion is to build the MCDS algorithm in neo4j as a user-defined function and execute this function on the network graph, which is considered an abstract to the SDN data plane. Switches are represented as graph nodes in the network graph, and the connections between the switches as graph edges. Using such big data tools can accelerate the execution of the MCDS algorithm with high reliability and parallelism. *Fig. 5* shows the proposed model for SDN network segmentation, where a single SDN controller controls each administrative domain. The SDN controllers can communicate through east-west interfaces to exchange network information and statistics.

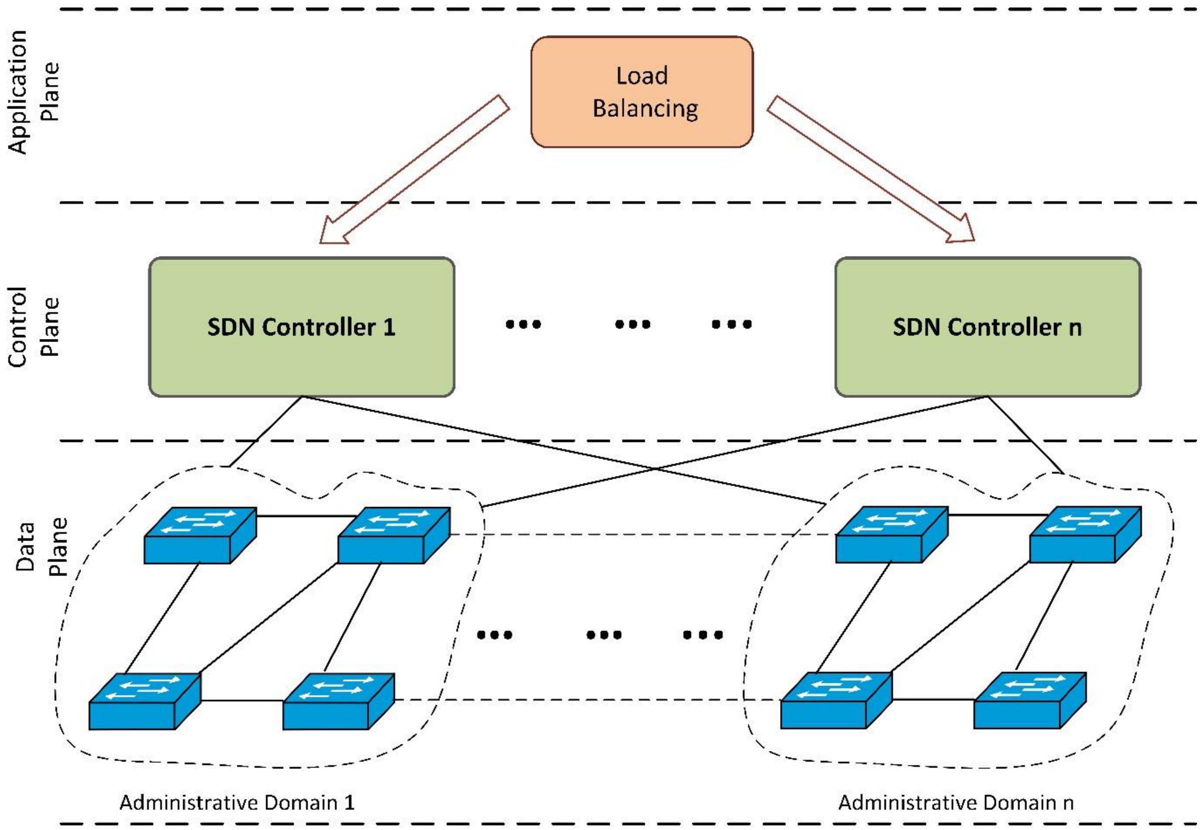

FIG. 5. THE PROPOSED MODEL FOR SDN NETWORK SEGMENTATION.

V. CONCLUSIONS

This paper presents the main characteristics and recent trends in SDN and big data. We discuss how big data can assist SDN in various aspects, including intrusion detection, traffic monitoring, and controller optimization. To sum up, the collaborative design of big data and SDN is considered a promising way in big data networking that is still an open research challenge. How to utilize big data techniques to enhance the overall performance of SDN is a demanding domain that should be addressed as a significant subject. This paper investigates the collaboration between big data and SDN to enlighten the way towards more areas in SDN that can be enhanced using big data techniques.

REFERENCES

[1] S. D. Khudhur and H. A. Jeiad, "LSDStrategy: A Lightweight Software-Driven Strategy for Addressing Big Data Variety of Multimedia Streaming," IEEE Access, Vol. 10, 2022.

[2] T. A. Assegie and P. S. Nair, "A review on software defined network security risks and challenges," TELKOMNIKA Telecommunication, Computing, Electronics and Control, Vol.17, No.6, December 2019.

Received 14/December/2022; Accepted 09/January/2023

DOI: https://doi.org/10.33103/uot.ijccce.23.3.6

[3] J. Singh and S. Behal, "Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges, and future directions," *Computer Science Review 37*, 2020.
[4] B. Isong, R. R. S. Molose, A. M. Abu-Mahfouz, and N. Dladlu, "Comprehensive Review of SDN Controller Placement Strategies," IEEE Access, vol. 8, 2020.
[5] L. Cui, F. R. Yu, and Q. Yan, "When big data meets software-defined networking: SDN for big data and big data for SDN," IEEE Network, vol. 30, no. 1, 2016.
[6] S. Jain, M. Khandelwal, A. Katkar, and J. Nygate, "Applying big data technologies to manage QoS in an SDN," 2016 12th International Conference on Network and Service Management (CNSM), 2016.
[7] M. Trevisan, I. Drago, M. Mellia, H. H. Song, and M. Baldi, "AWESoME: Big Data for Automatic Web Service Management in SDN," IEEE Transactions on Network and Service Management, vol. 15, no. 1, March 2018.
[8] Y. Tian, W. Chen, and C. Lea, "An SDN-Based Traffic Matrix Estimation Framework," IEEE Transactions on Network and Service Management, vol. 15, no. 4, December 2018.
[9] W. J. Queiroz, M. A. M. Capretz, and M. A. R. Dantas, "A MapReduce Approach for Traffic Matrix Estimation in SDN," IEEE Access, vol. 8, 2020.
[10] P. T. Dinh and M. Park, "BDF-SDN: A Big Data Framework for DDoS Attack Detection in Large-Scale SDN-Based Cloud," 2021 IEEE Conference on Dependable and Secure Computing (DSC), 2021.
[11] T. Souza, C. Rothenberg, M. Santos, and L. Paula, "Towards Semantic Network Models via Graph Databases for SDN Applications," 2015 Fourth European Workshop on Software Defined Networks, 2015.
[12] Q. Wen, R. Chen, Y. Xia, L. Zhou, J. Deng, J. Xu, and M. Xia, "Big-Data helps SDN to verify integrity of control/data planes," Big-Data and Software Defined Networking, *IET Book Series on Big Data,* March 2018.
[13] O. Marzuqi, A. Virgono, and R. M. Negara, "Implementation model architecture software defined network using raspberry Pi: a review paper," TELKOMNIKA Telecommunication, Computing, Electronics and Control, Vol.17, No.3, June 2019.
[14] O. S. Al-Heety, Z. Zakaria, M. Ismail, M. M. Shakir, S. Alani, and H. Alsariera, "A Comprehensive Survey: Benefits, Services, Recent Works, Challenges, Security, and Use Cases for SDN-VANET," IEEE Access, vol. 8, 2020.
[15] S. Ghaly and M. Z. Abdullah, "Design and implementation of a secured SDN system based on hybrid encrypted algorithms," TELKOMNIKA Telecommunication, Computing, Electronics and Control, Vol. 19, No. 4, August 2021.
[16] H. Tussyadiah, R. M. Negara, and Danu Dwi Sanjoyo, "Distributed gateway-based load balancing in software defined network," TELKOMNIKA Telecommunication, Computing, Electronics and Control, Vol. 18, No. 5, October 2020.
[17] https://jwst.nasa.gov/, (accessed July 23, 2022).
[18] https://webbtelescope.org/, (accessed August 12, 2022).
[19] J. Patel, "An Effective and Scalable Data Modeling for Enterprise Big Data Platform," 2019 IEEE International Conference on Big Data (Big Data), 2019.
[20] A. D'Alconzoz, I. Dragoy, A. Morichettay, M. Melliay, and P. Casas, "A Survey on Big Data for Network Traffic Monitoring and Analysis," IEEE Transactions on Network and Service Management, 2019.
[21] D. Sun, H. Yan, S. Gao, X. Liu, and R. Buyya, "Rethinking elastic online scheduling of big data streaming applications over high-velocity continuous data streams," J Supercomput, 2018.
[22] R. A. Hasan, R. A. I. Alhayali, N. D. Zaki, and A. H. Ali, "An adaptive clustering and classification algorithm for Twitter data streaming in Apache Spark," TELKOMNIKA Telecommunication, Computing, Electronics and Control, Vol.17, No.6, December 2019.
[23] P. V. Desai, "A survey on big data applications and challenges," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018.
[24] K. Adnan and R. Akbar, "Limitations of information extraction methods and techniques for heterogeneous unstructured big data," International Journal of Engineering Business Management, 2019.
[25] S. A. Dheyab, M. N. Abdullah, and B. F. Abed, "A novel approach for big data processing using message passing interface based on memory mapping," J. Big Data, vol. 6, no. 1, pp. 1-17, Dec. 2019.
[26] J. Taheri, "Big Data and Software Defined Networks," 1st edition, The Institution of Engineering and Technology, London, United Kingdom, 2018.
[27] A. H. Ali, M. N. Abbod, M. K. Khaleel, M. A. Mohammed, and T. Sutikno, "Large scale data analysis using MLlib," TELKOMNIKA Telecommunication, Computing, Electronics and Control, Vol. 19, No. 5, October 2021.
[28] L. Fawcett, S. Scott-Hayward, M. Broadbent, A. Wright, and N. Race, "Tennison: A Distributed SDN Framework for Scalable Network Security," IEEE Journal on Selected Areas in Communications, vol. 36, no. 12, December 2018.
[29] S. S. Samaan and H. A. Jeiad, "Architecting a machine learning pipeline for online traffic classification in software defined networking using spark," International Journal of Artificial Intelligence (IJ-AI), Vol. 12, No. 2, June 2023.
[30] A. Mirarab, S. L. Mirtaheri, and S. A. Asghari, "Value creation with big data analytics for enterprises: a survey," TELKOMNIKA Telecommunication, Computing, Electronics and Control, Vol.17, No.6, December 2019.
[31] K. Badapanda, D. P. Mishra, and S. R. Salkuti, "Agriculture data visualization and analysis using data mining techniques: application of unsupervised machine learning," TELKOMNIKA Telecommunication Computing Electronics and Control, Vol. 20, No. 1, February 2022.
[32] M. Hartung and M. K¨orner, "SOFTmon - Traffic Monitoring for SDN," International Workshop on Applications of Software-Defined Networking in Cloud Computing (SDNCC), 2017.
[33] LU. Laboshin, A.A. Lukashin, and V.S. Zaborovsky, "The Big Data approach to collecting and analyzing traffic data in large scale networks," Procedia Computer Science, 2017.
[34] A. S. Dawood and M. N. Abdullah, "Adaptive Performance Evaluation for SDN Based on the Statistical and Evolutionary Algorithms," Iraqi Journal of Computers, Communications, Control & Systems Engineering (IJCCCE), 2019.

Received 14/December/2022; Accepted 09/January/2023
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.6

[35] A. Guo, C. Yuan, G. He, and L. Xu, "Research on SDN/NFV Network Traffic Management and Optimization based on Big Data and Artificial Intelligence," The 18th International Symposium on Communications and Information Technologies (ISCIT 2018), 2018.

[36] Patil N. S., Kiran P, Kavya N. P., and Naresh Patel K. M., "A Survey on Graph Database Management Techniques for Huge Unstructured Data," International Journal of Electrical and Computer Engineering (IJECE), Vol. 8, No. 2, April 2018.

[37] M. Sharma, V. D. Sharma, and M. M. Bundele, "Performance Analysis of RDBMS and No SQL Databases: PostgreSQL, MongoDB, and Neo4j," 2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE), 2018.

[38] W. M. S. Mohammed, and A. K. Jumaa, "Storage, Distribution, and Query Processing RDF Data in Apache Spark and GraphX: A Review Survey," International Journal of Mechanical Engineering, Vol. 6 No. 3, December 2021.

[39] F. C. Yayah, K. I. Ghauth, and C. Ting, "Parallel classification and optimization of telco trouble ticket dataset," TELKOMNIKA Telecommunication, Computing, Electronics and Control, Vol. 19, No. 3, June 2021.

[40] M. Alsaeedi, M. M. Mohamad, and A. A. Al-Roubaiey, "Toward Adaptive and Scalable OpenFlow-SDN Flow Control: A Survey," IEEE Access, 2019.

[41] A. Abuarqoub, "A Review of the Control Plane Scalability Approaches in Software Defined Networking," Future Internet, 2020.

[42] M. R. Belgaum, Z. Alansari, S. Musa, M. M. Alam, and M. S. Mazliham, "Role of artificial intelligence in cloud computing, IoT and SDN: Reliability and scalability issues," International Journal of Electrical and Computer Engineering (IJECE), Vol. 11, No. 5, October 2021.

[43] Q. Qin, K. Poularakis, G. Iosifidis, and L. Tassiulas, "SDN Controller Placement at the Edge: Optimizing Delay and Overheads," IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, 2018.

[44] S. Yang, L. Cui, Z. Chen, and W. Xiao, "An Efficient Approach to Robust SDN Controller Placement for Security," IEEE Transactions on Network and Service Management, vol. 17, no. 3, September 2020.

[45] L. Mamushiane, J. Mwangama, and A. A. Lysko, "Given a SDN Topology, How Many Controllers are Needed and Where Should They Go?," 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2018.

[46] K. Choumas, D. Giatsios, P. Flegkas, and T. Korakis, "SDN Controller Placement and Switch Assignment for Low Power IoT," Electronics, 2020.

[47] T. Das and M. Gurusamy, "Controller Placement for Resilient Network State Synchronization in Multi-Controller SDN," IEEE Communications Letters, vol. 24, no. 6, June 2020.

Received 14/December/2022; Accepted 09/January/2023

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352