Software-Defined Networking (SDN) Traffic Analysis Using Big Data Analytic Approach

I. Made Suartana[1](✉) and Ricky Eka Putra[2]

[1] Department of Electrical Engineering, Faculty of Engineering, Universitas Negeri Surabaya, Surabaya, Indonesia
madesuartana@unesa.ac.id

[2] Department of Informatics Engineering, Faculty of Engineering, Universitas Negeri Surabaya, Surabaya, Indonesia

Abstract. Network Traffic Analysis is an essential component of network management, especially for ensuring the proper operation of large-scale networks like the Internet. Monitoring and analyzing network traffic has become increasingly difficult due to the growing complexity of Internet services and the volume of network traffic. The development of the network concept with a virtualization approach such as Software-Defined Networking provides new problems in monitoring and analyzing traffic on the network. The SDN concept that distinguishes data flow and control makes network traffic data different from conventional networks. On the other hand, applications such as traffic classification and policy making in computer networks require a real-time approach and scalability. Anomaly detection and security techniques must rapidly recognize and respond to unanticipated events while processing millions of heterogeneous events extracted from computer network traffic data. Finally, the system must gather, store, and process enormous historical data sets in preparation for analysis. Volume, Velocity, Variety, and Veracity are challenges that must be faced in managing traffic analysis. This study explains how traffic analysis on SDN networks and Big Data analytics are combined to take full advantage of the potential of network data. This study aims to discuss the extent to which traffic analysis can take advantage of the potential of Big Data technology and describe the challenges and opportunities of using Big Data and machine learning technology for traffic analysis. A prototype approach is used to build Big Data Analytics architecture and apply machine learning methods for data analysis. Based on the result big data approach can be used to classify attack traffic on SDN networks. The results of the train scores and test scores on the classification using the decision tree are as follows: the training score is: 0.998265782638848, and the Test score is: 0.9982486670135102.

Keywords: Software Define Networking · Traffic Analysis · Analytic Approach

1 Introduction

The complexity of contemporary networks is growing. New technologies like as artificial intelligence, the Internet of things, multi-cloud, network virtualization, and software-defined networking (SDN) are increasing the entire network infrastructure's complexity.

H. P. Agustin et al. (Eds.): IJCSE 2022, AER 218, pp. 243–250, 2022.
https://doi.org/10.2991/978-94-6463-100-5_25

As a result, network visibility and monitoring have become more sophisticated, and end-to-end real-time monitoring has become challenging. In contrast, network monitoring is essential and provides managers with numerous benefits. For instance, monitoring and analysis can assist with network troubleshooting, configuration validation, and network security enhancement.

Monitoring and analysis of traffic on a computer network is a complex task. Some challenges faced by network traffic analysis applications are large traffic volumes, the speed of transmission systems, the natural evolution of services and attacks, and multiple data sources and methods for obtaining measurements. As the complexity of the network continues to increase, more points are needed for observations and potentially generate heterogeneous data that must be collected and evaluated. In addition, implementing efficient mechanisms for the online analysis of large-scale data streams makes it challenging. The complexity and affordability of storage in terms of price have also resulted in the increase in large-scale historical data sets for retrospective analysis.

With the advent of big data technology, where data volume, speed, correctness, and variation are the main challenges, it is possible to extract value from data. It is possible to be a solution for complex network traffic analysis data. Then using machine learning, optimization of traffic analysis becomes an intriguing topic of study. Several researchers use big data and machine learning technology to support traffic analysis, Such as [1] examine the factors influencing network security platforms in the era of big data. [2] Applying duplicate data detection and deletion, detection of missing value, and data quality analysis to the big data methodology. k-means is also used to do variable correlation analysis and grouping analysis. [3] Developing network traffic measurement and analysis, [4] developing an analytical framework called Big-DAMA, [5] using A Big Data Architecture for Large Scale Security Monitoring, [5] create Architecture for Big Data Security in Cloud Computing. [6] Classifying distributed network traffic flow with machine learning. [7] Using IP addresses, routing statistics, and hop count distance matrices, examines vector representations for Internet nodes using deep learning.

Big Data Platforms and machine learning are technologies that help applications handle data sets. The network monitoring applications of big data analytics include traffic prediction, traffic classification, fault management, and network security. Several studies use a big data analytics-based technique for network monitoring and security. Further exploration is needed on whether traffic analysis applications can fully exploit the potential of Big Data technology.

This study describes how traffic analysis and Big Data analytics are combined to take full advantage of the potential of network data. This study aims to discuss the extent to which traffic analysis can take advantage of the potential of Big Data technology and describe the challenges and opportunities of using Big Data and machine learning technologies for traffic analysis applications. A prototype approach is used to build Big Data Analytics architecture and apply machine learning methods for data analysis.

2 Research Methodology

This study used a prototype to build a Big Data Analytic architecture for dataset storage of computer network traffic and trials of applying machine learning methods for data analysis based on the created datasets.

2.1 Datasets

The system simulation dataset was created and made available by [8] This dataset contains shuffle data, which combines attack and normal data. There are 65 characteristics in the dataset. Data sets generated with the Mininet emulator Both regular traffic, such TCP, UDP, and ICMP, as well as malicious traffic, like TCP Syn attack, UDP flood attack, and ICMP assault, are simulated on the network. There are 23 features in all, some of which were calculated and some of which were collected from the switches, in the data collection. Switch-id, Packet count, Byte count, Duration sec, Duration nsec, Duration in Nanoseconds, Source IP, Destination IP, and Total Duration are a few of the features that were extracted. Iteration number Tx bytes stand for the number of bytes sent from the switch port, and rx bytes for the number of bytes received on the switch port. After being converted to numbers, the date and time are shown in the dt field, and a flow is seen every 30 s. Packet per flow, or the number of packets transmitted in a single flow, byte per flow, the number of Packet ins messages, the total number of flow entries in the switch, and packet rate—or the number of packets sent per second—are the features that are calculated. The sum of the data transfer and reception rates, or tx kbps and rx kbps, is known as port bandwidth. The last column's class name, which indicates whether the traffic is malicious or benign, is displayed. Label 1 designates malicious traffic, while Label 0 designates benign traffic. We perform a 250-min network simulation and collect 1,04,345 rows of data. More data can be gathered by rerunning the simulation for a predetermined interval.

2.2 Big Data Analytic Approach

Big data analytics identifies trends, patterns, and correlations in huge quantities of unstructured data in order to enhance data-driven decision making. These procedures employ modern technologies to apply well-known statistical analysis techniques, such as clustering and regression, to larger data sets. Figure 1 depicts the processes involved in big data analytics, which include Data Acquisition: the collection of structured and unstructured data from multiple sources, such as cloud storage, mobile applications, in-store IoT sensors, and more. Some data will be housed in data warehouses, giving business intelligence tools and solutions easy access. Raw or unstructured data that is too complex or diverse for a warehouse might be labeled with metadata and kept in a data lake. Data Preprocessing: After data has been collected and stored, it must be adequately organized for analytical queries to produce accurate results, particularly when the data is huge and unstructured. The exponential rise of available data poses a problem for data processing in companies. Batch processing is a method for processing that examines huge data blocks progressively. Long intervals between data gathering and analysis call for batch processing. Stream processing examines tiny data batches concurrently, shortening the time between data collection and analysis so that decisions can be made more quickly. Nevertheless, stream processing is often more complicated and costly. Data Cleansing: All data, regardless of size, must be sanitized and formatted correctly in order to enhance data quality and produce more substantial results. Any redundant or irrelevant information must be removed or accounted for.

Unclean data can obscure and mislead, leading to faulty conclusions. Data Analysis: Getting huge data into an useful condition takes time. Once ready, however, modern analytics algorithms can translate vast quantities of data into significant insights. Among these strategies for analyzing huge data are: Data mining is the process of sifting through enormous data sets to identify patterns and associations. This is accomplished by searching for outliers and grouping related data. Predictive analytics use an organization's historical data to develop forecasts, which helps to identify upcoming dangers and opportunities. Lastly, deep learning imitates how humans learn by employing AI and machine learning to layer algorithms and discover patterns in the most complex and abstract data.

2.3 Big Data Analytic Support (Spark)

Spark is an in-memory cluster computing framework for processing and analyzing massive volumes of data. It is a general-purpose advanced execution engine that can handle batch processing, interactive analysis, streaming data, machine learning, and graph computing [4]. Spark is a powerful execution engine capable of batch processing, interactive analysis, streaming data, machine learning, and graph computing. This type of programming interface has become indispensable as the demand for handling massive datasets has grown. Spark caches data in memory rather than repeatedly writing to the disk, and only writes to the disk once. Spark is additionally effective due to its usability, velocity, general purpose, scalability, and fault tolerance. Spark was meant to be scalable since it can perform a wide variety of data processing jobs. To increase the capacity of a Spark cluster, it is only necessary to add nodes [4]. As stated previously, Spark is fault-tolerant and handles node failure automatically without affecting application failure. As a result, Spark can analyze massive volumes of data fast and efficiently.

2.4 Decision Tree

Using a training dataset, the decision tree method derives a set of decision rules. It builds a decision tree that can be used to forecast the numeric label of an observation. The nodes and edges of the tree are structured hierarchically. A decision tree differs from a graph in that there are no loops; non-leaf nodes are referred to as internal or split nodes, and leaf nodes are referred to as terminal nodes. The method for the decision tree begins at the root node and descends to the terminal node. The decision tree algorithm "performs a sequence of tests on the features in order to predict a label" [4]. Although a decision tree can be used for regression and classification, in this case it is used for classification.
Using a training dataset, the decision tree method derives a set of decision rules. It builds a decision tree that can be used to forecast the numeric label of an observation. The nodes and edges of the tree are structured hierarchically. A decision tree differs from a graph in that there are no loops; non-leaf nodes are referred to as internal or split nodes, and leaf nodes are referred to as terminal nodes. The method for the decision tree begins at the root node and descends to the terminal node. The decision tree algorithm "performs a sequence of tests on the features in order to predict a label". Although a decision tree can be used for regression and classification, in this case it is used for classification.

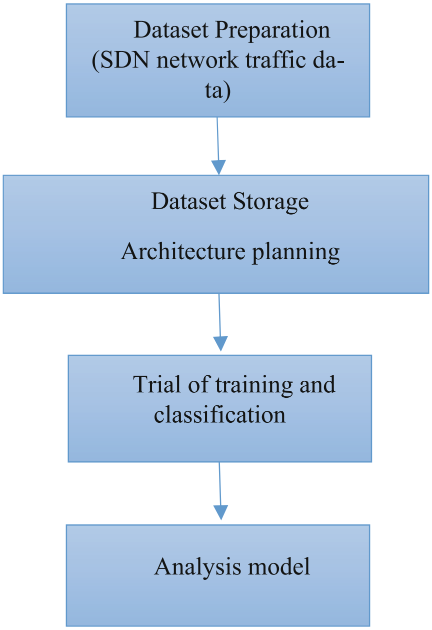
Dataset Preparation (SDN network traffic data)
Dataset Storage
Architecture planning
Trial of training and classification
Analysis model

Fig. 1. Methodology flow diagram

2.5 Methodology

Figure 1 presents a flow diagram for the overall methodology used in this work.

3 Result and Discussion

In this section, we discuss the result analysis of the classification of SDN traffic schemes. Performance is evaluated using the SDN dataset [8].

3.1 Data Preprocessing

We use a correlation-based feature selection scheme for data preprocessing to select the most relevant data attributes for a fast and accurate classification process. By using the correlation-based feature selection method, it can determine which attributes are highly correlated. These attributes include pktcount, bytecount, Protocol, pktrate, pktperflow, src, dst, dur_nsec, switch, packetins, etc.; classification-based model training is performed after removing these highly related features.

3.2 Evaluation Metric

Evaluation of learning process performance and classification based on metrics [9].

Definition of accuracy: the proportion of correctly classified data to the total number of data.

Table 1. Performance evaluation classification SDN datasets

data	precision	recall	f1-score	support
doz	1.00	1.00	1.00	1281483
normal	1,00	1,00	1,00	321021
probe	1.00	0.99	0.99	13594
r21	0.95	0.90	0.93	365
u2r	1.00	0.30	0.46	20
accuracy	0.99	0.84	1.00	1616483
macro avg	1.00	1.00	0.88	1616483
weighted avg			1.00	1616483

Precision (P): Defined as the ratio of the number of true positive (TP) data to the number of true positive (TP) and false positive (FP) data that have been categorized.

$$P = \frac{TP}{(TP + FP)} \times 100\% \tag{1}$$

Recall (R): Defined as the proportion of true positive (TP) data to true positive (TP) and false negative (FN) categorized records.

$$P = \frac{TP}{(TP + FN)} \times 100\% \tag{2}$$

F-Measure (F): Defined as the average of the harmonics of Precision (P) and Recall (R), F-Measure (F) shows the equilibrium between Precision (P) and Recall (R).

$$F \frac{2.P.R}{(P + R)} \times 100\% \tag{3}$$

3.3 Performance Analysis

From Table 1, it can be seen the performance of the results of the classification of each attack contained in the SDN dataset. Performance is seen from the evaluation matrix. From the results of the evaluation matrix, the dos attack has a precision, recall, and f1-score with a score of 1. For probe attacks, it has a precision with a score of 1, while recall and f1-score with a score of 0.99. The r2l attack has precision with a score of 0.95, recall with a score of 0.90, and f1-score with a score of 0.93. The u2r attack has precision with a score of 1.00, recall with a score of 0.30, and f1-score with a score of 0.46. At the same time, normal data has precision, recall, and f1-score with a score of 1.

The results of the train scores and test scores on the classification using the decision tree are as follows: the training score is: 0.998265782638848, and the Test score is: 0.9982486670135102. Figure 2 explains the matrix comparison.

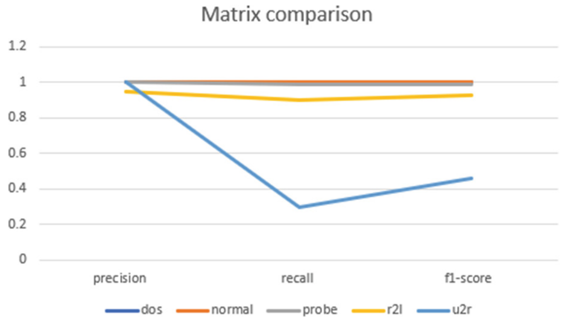

Fig. 2. Matrix comparison

4 Conclusions

This paper proposes a framework for a fast and efficient cybersecurity intrusion detection system using Big Data processing tools and machine learning algorithms. We have used the SDN Dataset for performance evaluation of the proposed framework using feature selection and classification with a decision tree model. Removing features on SDN datasets affects accuracy by a very low margin, but it reduces the time it takes to train models or predict data. The results of the classification of attacks with metrics of accuracy, precision, and recall produce good values for all types of attacks. This research shows that the big data analytic approach can be used to manage NTMA data based on machine learning. The big data approach can be used to classify attack traffic on SDN networks.

Authors' Contributions. All author contributed equally to this work. I Made as researcher writing manuscript and Ricky Eka editing this article.

References

1. Alaoui, Imane el, and Youssef Gahi. 2020. "Network Security Strategies in Big Data Context." *Procedia Computer Science* 175: 730–36. https://doi.org/10.1016/j.procs.2020.07.108.
2. Wang, Lidong, and Randy Jones. 2021. "Big Data Analytics in Cyber Security: Network Traffic and Attacks." *Journal of Computer Information Systems* 61 (5): 410–17. https://doi.org/10.1080/08874417.2019.1688731.
3. Mistry, Devang, Prasad Modi, Kaustubh Deokule, Aditi Patel, Harshagandha Patki, and Omar Abuzaghleh. 2016. "Network Traffic Measurement and Analysis." In *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, 1–7. IEEE. https://doi.org/10.1109/LISAT.2016.7494141.
4. Casas, Pedro, Alessandro D'Alconzo, Tanja Zseby, and Marco Mellia. 2016. "Big-DAMA." In *Proceedings of the 2016 Workshop on Fostering Latin-American Research in Data Communication Networks*, 1–3. New York, NY, USA: ACM. https://doi.org/10.1145/2940116.2940117.
5. Marchal, Samuel, Xiuyan Jiang, Radu State, and Thomas Engel. 2014. "A Big Data Architecture for Large Scale Security Monitoring." In *2014 IEEE International Congress on Big Data*, 56–63. IEEE. https://doi.org/10.1109/BigData.Congress.2014.18.

6. Manogaran, Gunasekaran, Chandu Thota, and M. Vijay Kumar. 2016. “MetaCloudDataStorage Architecture for Big Data Security in Cloud Computing.” *Procedia Computer Science* 87: 128–33. https://doi.org/10.1016/j.procs.2016.05.138.
7. Garcia, Johan, and Topi Korhonen. 2018. “Efficient Distribution-Derived Features for High-Speed Encrypted Flow Classification.” In *Proceedings of the 2018 Workshop on Network Meets AI & ML - NetAI’18*, 21–27. New York, New York, USA: ACM Press. https://doi.org/10.1145/3229543.3229548.
8. Li, Mingda, Cristian Lumezanu, Bo Zong, and Haifeng Chen. 2018. “Deep Learning IP Network Representations.” In *Proceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*, 33–39. New York, NY, USA: ACM. https://doi.org/10.1145/3229607.3229609.
9. Ahuja, Nisha, Gaurav Singal, and Debajyoti Mukhopadhyay. 2020. “DDOS Attack SDN Dataset.” Mendeley Data.
10. Shone, Nathan, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. 2018. “A Deep Learning Approach to Network Intrusion Detection.” *IEEE Transactions on Emerging Topics in Computational Intelligence* 2 (1): 41–50. https://doi.org/10.1109/TETCI.2017.2772792.