FLOW CONFLICT ELIMINATIONS THROUGH MACHINE LEARNING FOR SOFTWARE DEFINED NETWORK

MUTAZ HAMED HUSSIEN KHAIRI

UNIVERSITI TEKNOLOGI MALAYSIA

FLOW CONFLICT ELIMINATIONS THROUGH MACHINE LEARNING FOR SOFTWARE DEFINED NETWORK

MUTAZ HAMED HUSSIEN KHAIRI

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Doctor of Philosophy

School of Electrical Engineering
Faculty of Engineering
Universiti Teknologi Malaysia

NOVEMBER 2021

ACKNOWLEDGEMENT

Praise is to Allah, the Lord of the world. My profound gratitude is to Al-Mighty Allah first, by Praise is to Allah, the Lord of the world. My profound gratitude is to Al- Mighty Allah first, by whose will and power this thesis report came into being.

I would like to express my sincere appreciation and gratitude to my supervisor Assoc. Prof. Dr. Sharifah Hafizah Binti Syed Arifin for her support, guidance, encouragement, and patient throughout this research period. Without his unwavering guidance, support, and valuable advice during the research and writing, this thesis would have been completed. Her dedication and technical expertise proved to be the key elements to my doctoral research. Furthermore, I would like to extend my gratitude to my co-supervisor, Assoc. Prof. Dr. Nurul Mu'azzah Binti Abdul Latiff and Dr. Kamaludin Mohamed Yusuf for their generous time, fruitful discussions, motivation, and patience to attend to my numerous questions during this study.

My appreciation goes to my university Future University in Sudan, especially the President Dr. Abubaker Mustafa and the vice President Mr. Ghassan Abubaker. Same goes to my friends and brothers such as Osman Ali, Ashraf Hassan, Omer Fagirey, Mohamed Kallafallah, Mohamed Kannon, Yassin Abdelkarim, Mohamed Shawky, Mosab Hamdan, Sami Eltype and many others.

Finally, my special thanks to my beloved parents, and my brother and sister for their unending love, sacrifice, encouragement, and support. The same goes to my wife and my kid Yousef for their unreserved support, love, and patient towards the success of this thesis.

ABSTRACT

Software-Defined Network (SDN) is a modern approach in networking technologies that enables dynamic and programmatically efficient network configuration for improved performance and network monitoring. Similar to the traditional networks, the SDN system is susceptible to conflicts in flows within the network. Flow conflict in SDN occurs in response to adjustment of certain features of flows such as priority, match field, and action. While efforts have been made to address these challenges, the current flow of conflict solutions in SDN has several limitations. First, the control layer does not show nor collect the conflict flows that are affected in the OpenFlow switch. Second, the flow entry detection and classification process are relatively time-consuming. Third, there are no studies on detection methods to avoid flow conflicts using artificial intelligence methods such as Machine Learning (ML) as a solution to flow conflict in SDN. This thesis aims to eliminate flows conflict in SDN by using ML algorithms to detect and classify all flow conflicts in the OpenFlow switch. This thesis aims to develop the flow construction model in the SDN controller, detect the conflict flow using ML algorithm, and classify all the conflict types in the flow table using a classification algorithm. In this work, simulation works were conducted in Mininet software using two types of topologies. Decision trees (DT), support vector machine (SVM), hybrid DT- SVM, and extreme fast decision trees (EFDT) ML algorithms were used to detect the conflicts. The main contribution of this thesis is the development of a flow construction model with conflict rules in the OpenFlow table that enhanced the SDN process. By using accurate and effective ML algorithms designed and implemented in the controller layer, flow conflicts are detected and classified to reduce the adverse effects of conflict in the SDN. The performance of the proposed algorithms was evaluated for their efficiency and effectiveness across a variety of evaluation metrics. The EFDT algorithm produced the best results with a performance accuracy above 90% and 95% in detection and classification respectively for all sizes of flows between 1,000 and 100,000. The proposed algorithms for detection and classification show performance improvements over two different algorithms used as benchmarks.

ABSTRAK

Rangkaian Takrif Perisian (SDN) adalah pendekatan moden dalam teknologi rangkaian yang membolehkan konfigurasi rangkaian yang dinamik dan cekap secara terprogram untuk prestasi dan pemantauan rangkaian yang lebih baik. Sama seperti rangkaian tradisional, sistem SDN terdedah kepada konflik aliran dalam rangkaian. Konflik aliran dalam SDN berlaku sebagai tindak balas kepada pelarasan ciri-ciri aliran tertentu seperti keutamaan, medan padanan dan tindakan. Walaupun usaha telah dibuat untuk menangani cabaran ini, penyelesaian konflik aliran semasa dalam SDN mempunyai beberapa batasan. Pertama, lapisan kawalan tidak menunjukkan atau mengumpulkan aliran konflik yang terjejas dalam suis OpenFlow. Kedua, pengesanan kemasukan aliran dan proses pengelasan agak memakan masa. Ketiga, tiada kajian berkaitan kaedah pengesanan untuk mengelakkan konflik aliran menggunakan kaedah kecerdasan buatan seperti Pembelajaran Mesin (ML) sebagai penyelesaian kepada konflik aliran di SDN. Tesis ini bertujuan untuk menghapuskan konflik aliran dalam SDN menggunakan algoritma ML untuk mengesan dan mengelaskan semua konflik aliran dalam suis OpenFlow. Matlamat tesis ini ialah membangunkan model pembinaan aliran dalam pengawal SDN, mengesan konflik aliran menggunakan algoritma ML dan mengelaskan semua jenis konflik yang berlaku dalam jadual aliran menggunakan algoritma pengelasan. Selanjutnya, simulasi telah dijalankan menggunakan perisian Mininet dua jenis topologi. Algoritma pepohon keputusan (DT), mesin vektor sokongan (SVM), hibrid DT- SVM dan pepohon keputusan pantas ekstrim (EFDT) ML digunakan untuk mengesan konflik. Sumbangan utama tesis ini adalah pembangunan model pembinaan aliran dengan peraturan konflik dalam jadual OpenFlow yang meningkatkan proses SDN. Dengan menggunakan algoritma ML yang tepat dan berkesan yang direka dan dilaksanakan dalam lapisan pengawal, konflik aliran dikesan dan dikelaskan untuk mengurangkan kesan buruk konflik di SDN. Prestasi algoritma yang dicadangkan telah dinilai untuk kecekapan dan keberkesanannya dalam pelbagai metrik penilaian. Algoritma EFDT memperoleh hasil terbaik dengan prestasi ketepatan melebihi 90% dan 95% masing-masing dalam pengesanan dan klasifikasi untuk semua saiz aliran antara 1,000 and 100,000. Algoritma yang dicadangkan untuk pengesanan dan klasifikasi menunjukkan peningkatan hasil apabila dibandingkan dengan dua algoritma yang berbeza yang digunakan sebagai penanda aras.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	-	Artificial Neural Network
API	-	Application Programming Interfaces
ARP	-	Address Resolution Protocol
AUC	-	Area Under Curve
BGP	-	Border Gateway Protocol
BML	-	Bayesian Machine Learning
CCM	-	Conflict Classification Model
CVB	-	Compact Bit Vector
DAC	-	Dynamic Access Control
DDoS	-	Distributed Denial-Of-Service
DoS	-	Denial-Of-Service
DT	-	Decision Tree
EFDT	-	Extremely Fast Decision Tree
ESCA	-	Efficient Sampling and Classification Approach
FCM	-	Flow Construction Model
FDM	-	Flow Detection Model
FEC	-	Flow Conflict Eliminations
FLD	-	Flood Light Detection
FTP	-	File Transfer Protocol
FNs	-	False Negatives
FPs	-	False Positives
HFT	-	Hierarchical Flow Tables

HGW	-	Home Gateway
ICMP	-	Internet Control Message Protocol
IDS	-	Intrusion Detection System
IDPS	-	Intrusion Detection/Prevention System
IoT	-	Internet of Things
IP	-	Internet Protocol
IPv4	-	Internet Protocol version 4
IPv6	-	Internet Protocol version 6
IPS	-	Intrusion Prevention System
KNN	-	K-Nearest Neighbour
LFA	-	Link Flooding Attack
L4	-	Transport Layer
MAC	-	Media Access Control
MATLAB	-	Matrix Laboratory
MC	-	Multi-Controller
MFT	-	Multiple Flow Tables
ML	-	Machine Learning
MTBDD	-	Multi Terminal Binary Decision Diagram
MTD	-	Mobile Threat Defence
NOX	-	Original OpenFlow Controller
NSL-KDD	-	Network Security Laboratory- Knowledge Discovery and Data Mining
OCR	-	Optical Character Recognition
ODL	-	Open Daylight
OF	-	OpenFlow

ONF	-	Open Networking Foundation
ONOS	-	Open Network Operating System
OSI	-	Open Systems Interconnection
OVSDB	-	Open vSwitch Database Management Protocol
PANDA	-	Platform for Anomaly Detection Applications
PCA	-	Principal Component Analysis
QoS	-	Quality of service
RAM	-	Random-Access Memory
ROC	-	Receiver Operating Characteristic
RVB	-	Redundant Bit Vectors
SaaS	-	Software as a Service
SCTP	-	Stream Control Transmission Protocol
SDDC	-	Software-Defined Data Center
SDN	-	Software-Defined Network
SD-WMN	-	Software-Defined Network -Wireless Mesh Network
SDR	-	Software Defined Radios
SeLeCT	-	Self-Learning Classifier for Internet Traffic
SQL	-	Structured Query Language
SVM	-	Support Vector Machine
TCDR	-	Transaction Conflict Detection and Resolution
(TLA+)	-	Top Layer Aggregation
TCP	-	Transmission Control Protocol
TLC	-	Tools of Checking
TNs	-	True Negatives

TPs	-	True Positives
TSS	-	Tuple Space Search
TSVM	-	Transudative Support Vector Machine
VEFD	-	Very Fast Decision Tree
VNs	-	Virtual Networks
VNF	-	Virtual Network Function
VPN	-	Virtual Private Network
UDP	-	User Datagram Protocol
UI	-	User Interface
WMN	-	Wireless Mesh Network

LIST OF SYMBOLS

r	-	Flow rule
ϵs	-	Source MAC address
ϵ_d	-	Destination MAC address
ζ_s	-	Source IP address
ζ_d	-	Destination IP address
η_s	-	Source protocol
η_d	-	Destination protocol
n	-	Segment space
N	-	Universal set of space addresses
f	-	Transformation function
a	-	Flow table
x	-	Features
Y	-	Dependency variable
w	-	Normal vector
b	-	Normal Variable
n	-	Independent random variables
R	-	Range
r_n	-	Variable number
ř	-	The mean
δ	-	Probability
ϵ	-	Hoeffding Bound
TP	-	Number of conflict flow records correctly detected.

TN	-	Number of normal flow records correctly detected.
FP	-	Number of normal flow records falsely detected.
FN	-	Number of conflict flow records falsely detected.
FPR	-	False Positive Fraction
TPR	-	True Positive Fraction
t	-	Time
L	-	Flow Required
R	-	Host
D	-	Attack
ð	-	Flow1
ñ	-	Flow2
P	-	Priority
T	-	Protocol
Á	-	Action

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Software-defined networking (SDN) is a network management technique that has enhanced the performance of the network by making its structure and configuration more dynamically and programmatically efficient (Bera et al., 2017; Cui et al., 2018; Pisharody et al., 2017). The application of this approach has led to several benefits, such as addressing changing business requirements by allowing administrators and network engineers to make these changes via a centralized control console (Cui et al., 2016). SDN's implementation on a network allows it to become more flexible and agile by combining a multitude of network technologies specifically designed for such a purpose. SDN's main structure involves separating the network control from the forwarding planes, which would be the same as separating the brain from the muscle (Karakus & Durresi, 2017). This separation would theoretically enable the network's control plane (or brain) to be programmable on its own and thus provide network engineers with direct control over the underlying network infrastructure(Hong & Wey, 2017a). SDN also has other underlying benefits, such as being manageable, dynamic, adaptable, and, more importantly, cost-effective, which make it the ideal solution for the ever-growing nature of the internet and its application's high-bandwidth requirements (Ray & Kumar, 2021; Xia et al., 2015).

SDN has many new standards and OpenFlow (OF) is one of them. One of OpenFlow's key elements in the framework of SDN is the controller, which allows all application development via an Application Programming Interfaces (API) with a northbound connection. SDN is highly determined by the controller's actions, and its performance is directly proportional to the SDN's efficiency. The OpenFlow switch consists of several flow tables that are connected to the controller using the OpenFlow protocol. This protocol is used to communicate between switches and the controller

while employing flow tables as an abstraction. The flow tables aim to ensure that packets are correctly allocated, sorted, and distributed in adherence to the flow entries (Hao et al., 2017; Tok & Demirci, 2021; Tran & Danciu, 2019b). A flow table is composed of multiple flow entries, which include: Match fields that are used to match the flow entries; Priority, used to match the flow's priority address; Counters, to be modified to correspond with the packets; Instructions, to modify the action taken or to deal with the flow; Timeouts, complete or remaining time before the session expires and the implicit value of the data selected by the controller is referred to as a cookie(Maldonado-Lopez et al., 2015).

1.2 Flow Conflict

Both traditional networks and SDNs are affected by several types of conflict that negatively impacts network performance (Kim & Kang, 2020; Pisharody et al., 2017). These conflicts can be classified according to their rules and effects into two main types: Intelligible Conflicts (redundancy, shadowing, overlapping) and Interpretative Conflicts (generalization, correlation, imbrication). Since packet counters and timeout values are not important for managing flow conflicts, consideration given to limitation in flow entries is in relation to priority, match fields, and action fields in the remainder of this study. SDN conflicts occur depending on the impact and adjustment of certain features, such as priority and action. Depending on the changes in the flow rule policy or flow entry, conflict forms appear in the controller and flow table. Priority and actions are the key components for developing the rule and flow entry in SDN. It is also well known that priority and action are some of the major differences between the traditional network and the SDN in terms of features.

1.3 Problem Statement

Flow conflict manifests in different forms, such as the case of designing the SDN. Since flow rules can check further than only layer-3 and layer-4 prefixes, they are fundamentally more complicated than traditional network matching rules as there are more variables involved. Many stream attributes can be dynamically modified because cross-layer communication is reinforced in SDN by flow rules that allow set-field actions. Furthermore, as wildcard entries are permitted, a partial conflict of flow policies may arise, thereby increasing the difficulty of resolving conflicting flow rules. Flow rules in SDN, unlike traditional networks, might have the same priority and match on several packet headers, leading to indirect dependence (Pisharody et al., 2017). The changes in the flow rule policy or flow entry led to varying forms of conflict appearing in the controller and flow table. Priority and actions are the key components in developing the rules and flow entries in SDN. Traditional networks have been documented to have important features, such as priority and action, which set them apart from SDNs. In the first mechanism, packets are matched with flow entries according to the priority of the flow entries. Since the flow entries are similar to each other, a packet can match more than one flow entry: resulting in the flow entry conflict (Cui et al., 2018; Lin & Sun, 2018). In previous research (Danciu & Tran, 2020), it was discovered that there are special kinds of anomalies that generate a hidden conflicts which appears primarily due to side effects of the application activities that are beyond the class in conflict with the SDN structures. Therefore, in order to prevent network paralysis, SDN modules must be properly implemented, especially among varied devices and applications such as the network devices, SDN controllers, and applications (Kim & Kang, 2020). Policies of the SDN network should be dynamically modified at a fast rate. The identification of these conflicts is an arduous and complicated operation, because of the high number of switches and heterogeneous policies in a common SDN network (Aryan et al., 2017; Danciu & Tran, 2020). In line with these issues, the problem statement of this work is formulated as follows:

i. Previous algorithms and detection methods used for avoiding flow conflict have not used artificial intelligence approaches like Machine Learning (ML) as a solution for flow conflict resolution in SDN. The use of ML

algorithms can considerably improve the efficiency and accuracy of conflict flow detection in the OpenFlow table. To identify conflicting flows in SDN, the ML algorithm is fitted with relevant features of a pre-processed labeled dataset. However, most of the previous research studies on conflict detection did not collect and save the flow conflicts created in OpenFlow switches (Danciu & Tran, 2020; Lin & Sun, 2018; Maldonado-Lopez et al., 2015; Pisharody, 2017). Furthermore, previous research studies on the detection of conflict in SDN have implemented and designed algorithms for use with a maximum number of 10000 flows (Cui et al., 2018; Tran & Danciu, 2019b).

ii. Both traditional and SDNs are affected by several types of conflict that can negatively affect network performance. Thus, it is important to define and classify the types of conflict that occur in the network to resolve and avoid these conflicts. Most of the research studies conducted on flow detection have not identified the effects of conflict in SDN as they have often avoided the classification of the conflict types that can occur in the OpenFlow table (Cui et al., 2018; Danciu & Tran, 2020; Wang et al., 2016b). The few other works that have attempted to classify the types of conflicts have done without a comprehensive classification of all conflict types. The most recent study on conflict detection in SDN classifies the types of conflict into four classes of redundancy, shadowing, correlation, and generalization (Aryan et al., 2017; Lu et al., 2019).

iii. Besides, the numbers and types of algorithms and detection methods used for avoiding flow conflict takes a long time (over 9 ms for 10000 flows and over 42 ms for 100000 flows) to apply their instructions in the flow table (Lo et al., 2015; Metter et al., 2017; Pisharody et al., 2017). The lengthy time taken by the algorithms to apply their instructions can slow down the speed at which the controller adds or modifies the flow entries in the flow table which in turn affects the performance and security of the SDN.

Therefore, a solution is required to enhance the accuracy of conflict detection while remaining relatively efficient in terms of time.

1.4 Research Objectives

This study aims to propose an algorithm for eliminating flow conflict in SDN. The proposed algorithms will detect and classify all flows in the OpenFlow switch to reduce the conflict between them. The specific objectives of this research are:

i. To develop a flow construction model for the SDN controller.

ii. To propose a machine learning algorithm for detecting and identifying the conflict flows in terms of accuracy, precision, recall, F1 measure (weight average of Precision and Recall) and running time[1].

iii. To implement a conflict classification algorithm for reducing the flow conflict in the SDN.

1.5 Scope of Research

SDN is a network architecture where network traffic may be operated and managed dynamically based on user requirements and demands. This research focuses on flow conflict elimination between the data and control planes in SDN. To achieve the research objectives, it is necessary to outline its scope and limitations.

i. The effectiveness of the proposed method is verified through the network topologies referenced by GitHub and some previous studies conducted on flow conflicts in SDN. Mininet software with Ryu controller is used to implement and apply conflict rules within the topologies.

ii. The performance is measured based on the flow conflict classification and detection. Conflict is reduced and eliminated by using ML algorithms.

[1] accuracy, precision, recall, F1 measure and running time are metrics used to evaluate performance of algorithm in ML (this is detailed in section 2.12.2).

iii. The OpenFlow protocol that is used in this study is OpenFlow 1.3, whereas experimental simulations is carried out in the Mininet simulation environment

iv. The evaluation metrics used to evaluate the performance of the proposed approach are accuracy, precision, recall, F1 measure, running time, and flow setup rate (throughput).

1.6 Research Contributions

This research contributes to the elimination of flow conflicts in SDN. It enhances the SDN by providing a flow construction model with conflict rules implemented in the OpenFlow table. An effective algorithm is designed and implemented to detect and classify flow conflicts in a bid to reduce the adverse effects of conflict in SDN. The main contributions are summarized as follows:

i. Provide an approach of flow construction model in SDN controller with conflicting flows implemented in the OpenFlow table for different types of topologies of varying flow sizes ranging from 1,000 to 100,000 flows.

ii. This study produces implementations of ML algorithms to distinguish and identify the different types of flows in open flow switches within the flow table.

iii. Implement the flow rules using ML in an algorithm in the control plane that is used to identify conflict types that appear in the flow.

1.7 Significance of Research

The significance of this research is shown in the comprehensive practical solution it profers as it eliminates and reduces flow conflicts in the OpenFlow table for small and large flows. The benefits of the solution implemented and developed in this research are:

i. Data from the flow construction model can be collected with flow entries information.

ii. The anomaly detection techniques used to detect conflicting flows in this study can be produced with the high performance of SDN and measured using accuracy, precision, recall, F1 measure, and running time for varying flow sizes.

iii. The implemented and developed Extremely Fast Decision Tree (EFDT) algorithm classifies and distinguishes between seven types of flow conflicts in SDN.

iv. The detection and classification algorithms show a significant improvement in the running time of the detection and classification of conflicting flows compared to other available solutions.

1.8 Thesis Outline

The outline and organization of this thesis is as follows. Chapter 1 presents the problem statement as well as the aims and specific objectives of this thesis. Chapter 2 presents a literature review of studies and research on flow conflicts as well as the previous approaches that have been presented in the relevant literature to resolve the problem in SDN. The chapter begins with a short overview of SDN concepts and architectures followed by flow conflict detection mechanisms and solutions in forward and control planes. The existing studies related to ML used for detection and classification in SDN are reviewed and discussed. The chapter concludes with the

definition of the metrics used to evaluate and test the performance of the implemented methods.

Chapter 3 provides the methodology of this research. The chapter starts by showing how conflict rules are generated and implemented in the OpenFlow switch. The experimental setup and development of the modules in the Ryu controller are then explained. All steps and processes used to develop the respective detection and classification algorithms are shown.

Chapter 4 presents the proposed algorithm used to collect and save flow conflicts in the OpenFlow switch. This chapter begins with a block diagram and a pseudo algorithm code. The results of the stream flows of all sizes are presented in this chapter, including graphs of flow data for Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

Chapter 5 presents the proposed detection and classification algorithms. The first section discusses the four algorithms used for detection, followed by block diagrams and the results of each of the four algorithms. The results are presented using six metrics, namely accuracy, recall, precision, F1 score measure, Receiver Operating Characteristic (ROC) curve, and running time. The detection performance of algorithms is further validated by comparison with other existing detection techniques.

Chapter 6 presents a discussion and explanation of the classification algorithm followed by a discussion of the results. The results are discussed relative to six metrics, namely accuracy, recall, precision, F1 score measure, Receiver Operating Characteristic (ROC) curve, and running time. The classification performance of algorithms is further validated by comparison with other existing classification techniques.

Finally, Chapter 7 shows the main achievements of the proposed algorithms in this study and highlights open areas of future work.

REFERENCES

Abar, T., Letaifa, A. B., & El Asmi, S. (2017). Machine learning based QoE prediction in SDN networks. 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC),

Ahmad, S., & Mir, A. H. (2021). Scalability, consistency, reliability and security in sdn controllers: A survey of diverse sdn controllers. *Journal of Network and Systems Management, 29*(1), 1-59.

Akyildiz, I. F., Lee, A., Wang, P., Luo, M., & Chou, W. (2014). A roadmap for traffic engineering in SDN-OpenFlow networks. *Computer Networks, 71*, 1-30.

Alomari, A., Subramaniam, S. K., Samian, N., Latip, R., & Zukarnain, Z. (2021). Resource Management in SDN-Based Cloud and SDN-Based Fog Computing: Taxonomy Study. *Symmetry, 13*(5), 734.

Alraawi, A. A. M., & Adam, S. A. N. Performance Evaluation of Controller Based SDN Network Over Non-controller Based Network in Data Center Network. 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE),

Amaral, P., Pinto, P. F., Bernardo, L., & Mazandarani, A. (2018). Application Aware SDN Architecture using Semi-supervised Traffic Classification. 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN),

Amin, R., Rojas, E., Aqdus, A., Ramzan, S., Casillas-Perez, D., & Arco, J. M. (2021). A survey on Machine Learning Techniques for Routing Optimization in SDN. *IEEE Access*.

Arbettu, R. K., Khondoker, R., Bayarou, K., & Weber, F. (2016). Security analysis of OpenDaylight, ONOS, Rosemary and Ryu SDN controllers. 2016 17th International telecommunications network strategy and planning symposium (Networks),

Arjunan, A., & Kaviarasan, R. (2021). Weighted distance hyperbolic prediction-based detection scheme for non line of sight nodes in VANETs. *Journal of King Saud University-Computer and Information Sciences, 33*(4), 489-496.

Aryan, R., Yazidi, A., Engelstad, P. E., & Kure, Ø. (2017). A general formalism for defining and detecting openflow rule anomalies. 2017 IEEE 42nd Conference on Local Computer Networks (LCN),

Asadollahi, S., Goswami, B., & Sameer, M. (2018). Ryu controller's scalability experiment on software defined networks. 2018 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC),

B. R. Granby, B. A. a. A. K. M., "SDN-PANDA: Software-Defined Network Platform for ANomaly Detection Applications," 2015 IEEE 23rd International Conference on Network Protocols (ICNP), San Francisco, CA, 2015, pp. 463-466, doi: 10.1109/ICNP.2015.58.

Babbar, H., & Rani, S. (2021). Performance Evaluation of QoS metrics in Software Defined Networking using Ryu Controller. IOP Conference Series: Materials Science and Engineering,

Balasubramanian, V., Aloqaily, M., & Reisslein, M. (2021). An SDN architecture for time sensitive industrial IoT. *Computer Networks, 186*, 107739.

Balta, M., & Özçelik, İ. (2020). A 3-stage fuzzy-decision tree model for traffic signal optimization in urban city via a SDN based VANET architecture. *Future Generation Computer Systems, 104*, 142-158.

Bao, K., Matyjas, J. D., Hu, F., & Kumar, S. (2018). Intelligent software-defined mesh networks with link-failure adaptive traffic balancing. *IEEE Transactions on Cognitive Communications and Networking, 4*(2), 266-276.

Bardalai, P., Medhi, N., Bargayary, B., & Saikia, D. K. (2021). OpenHealthQ: OpenFlow based QoS management of Healthcare Data in a Software-Defined Fog environment. ICC 2021-IEEE International Conference on Communications,

Barrett, R., Facey, A., Nxumalo, W., Rogers, J., Vatcher, P., & St-Hilaire, M. (2017). Dynamic traffic diversion in SDN: Testbed vs mininet. 2017 International Conference on Computing, Networking and Communications (ICNC),

Baz, A. (2018). Bayesian machine learning algorithm for flow prediction in sdn switches. 2018 1st International Conference on Computer Applications & Information Security (ICCAIS),

Bera, S., Misra, S., & Vasilakos, A. V. (2017). Software-defined networking for internet of things: A survey. *IEEE Internet of Things Journal, 4*(6), 1994-2008.

Bhardwaj, S., & Panda, S. (2021). Performance Evaluation Using RYU SDN Controller in Software-Defined Networking Environment. *Wireless Personal Communications*, 1-23.

Bowles, M. (2015). *Machine learning in Python: essential techniques for predictive analysis*. John Wiley & Sons.

Burkart, N., & Huber, M. F. (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research, 70*, 245-317.

Cam, E., & Ozdag, M. E. (2021). Discovery of Course Success Using Unsupervised Machine Learning Algorithms. *Malaysian Online Journal of Educational Technology, 9*(1), 26-47.

Carvalho, L. F., Abrão, T., de Souza Mendes, L., & Proença Jr, M. L. (2018). An ecosystem for anomaly detection and mitigation in software-defined networking. *Expert Systems with Applications, 104*, 121-133.

Chen, J. I. Z., & Smys, S. (2020). Social Multimedia Security and Suspicious Activity Detection in SDN using Hybrid Deep Learning Technique. *Journal of Information Technology, 2*(02), 108-115.

Chen, Y., Chen, W., Hu, Y., Zhang, L., & Wei, Y. (2016). Dynamic load balancing for software-defined data center networks. International Conference on Collaborative Computing: Networking, Applications and Worksharing,

Chen, Y., Pei, J., & Li, D. (2019). DETPro: A High-Efficiency and Low-Latency System Against DDoS Attacks in SDN Based on Decision Tree. ICC 2019-2019 IEEE International Conference on Communications (ICC),

Cheng, M., et al., Flow Setup Rate Test for OpenFlow Controller June 25, 2017.

Cherian, M., & Verma, S. (2021). Integration of IoT and SDN to Mitigate DDoS with RYU Controller. In *Computer Networks, Big Data and IoT* (pp. 673-684). Springer.

Comaneci, D., & Dobre, C. (2018). Securing networks using SDN and machine learning. 2018 IEEE International Conference on Computational Science and Engineering (CSE),

Cui, J., Zhou, S., Zhong, H., Xu, Y., & Sha, K. (2018). Transaction-based flow rule conflict detection and resolution in SDN. 2018 27th International Conference on Computer Communication and Networks (ICCCN),

Cui, L., Yu, F. R., & Yan, Q. (2016). When big data meets software-defined networking: SDN for big data and big data for SDN. *IEEE network, 30*(1), 58-65.

Czech, J. (2021). Distributed methods for reinforcement learning survey. In *Reinforcement Learning Algorithms: Analysis and Applications* (pp. 151-161). Springer.

Danciu, V., & Tran, C. N. (2020). Side-Effects Causing Hidden Conflicts in Software-Defined Networks. *SN Computer Science, 1*(5), 1-16.

Dash, S. S., Naidu, P. C. B., Bayindir, R., & Das, S. (2018). *Artificial Intelligence and Evolutionary Computations in Engineering Systems: Proceedings of ICAIECES 2017* (Vol. 668). Springer.

Dey, A. (2016). Machine learning algorithms: a review. *International Journal of Computer Science and Information Technologies, 7*(3), 1174-1179.

Díaz-Montiel, A. A., Lantz, B., Yu, J., Kilper, D., & Ruffini, M. (2021). Real-Time QoT Estimation through SDN Control Plane Monitoring Evaluated in Mininet-Optical. *IEEE Photonics Technology Letters*.

Eftimie, A., & Borcoci, E. (2020). SDN controller implementation using OpenDayLight: experiments. 2020 13th International Conference on Communications (COMM),

Eljack, A. H., Hassan, A. H. M., & Elamin, H. H. (2019). Performance Analysis of ONOS and Floodlight SDN Controllers based on TCP and UDP Traffic. 2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE),

Elsayed, M. S., Le-Khac, N.-A., & Jurcut, A. D. (2020). InSDN: A Novel SDN Intrusion Dataset. *IEEE Access, 8*, 165263-165284.

Fan, Z., & Liu, R. (2017). Investigation of machine learning based network traffic classification. 2017 International Symposium on Wireless Communication Systems (ISWCS),

Fang, Y., & Lu, Y. (2019). Checking Intra-Switch Conflicts of Rules During Preprocessing of Network Verification in SDN. *IEEE Communications Letters, 23*(9), 1547-1550.

Fathul Arif Kamarudin, M. N. M. M. N., Fuead Ali. (2020). A comparative study for bandwidth on demand using ONOS Reactive and Intent forwarding. *Indonesian Journal of Electrical Engineering and Computer Science, 17*(3), 1410~1421.

G. Garg and R. Garg, A. c. s. f. a. o. a. d. m. o. a. f. c. i. S., " 2015 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS), Chandigarh, 2015, pp. 1-4, doi: 10.1109/RAECS.2015.7453293.

Gao, P., Li, H., Qu, X., & Cheng, Y. (2021). Research on Virtual Machine Performance Test Based on Cloud Platform. 2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC),

Garg, G., & Garg, R. (2015). Detecting anomalies efficiently in SDN using adaptive mechanism. 2015 Fifth International Conference on Advanced Computing & Communication Technologies,

Garrich, M., Hernández-Bastida, M., San-Nicolás-Martínez, C., Moreno-Muro, F.-J., & Pavon-Marino, P. (2019). The Net2Plan-OpenStack Project: IT Resource Manager for Metropolitan SDN/NFV Ecosystems. Optical Fiber Communication Conference,

Gedia, D., & Perigo, L. (2019). Latency-Aware, Static, and Dynamic Decision-Tree Placement Algorithm for Containerized SDN-VNF in OpenFlow Architectures. 2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN),

Gong, C., Liu, T., Tang, Y., Yang, J., Yang, J., & Tao, D. (2017). A regularization approach for instance-based superset label learning. *IEEE transactions on cybernetics, 48*(3), 967-978.

Gong, R., Li, W., Li, F., & Wang, Y. (2021). On the Reliability of State-of-the-art Network Testbed Components. IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS),

Hamdan, M., Mohammed, B., Humayun, U., Abdelaziz, A., Khan, S., Ali, M. A., Imran, M., & Marsono, M. N. (2020). Flow-Aware Elephant Flow Detection for Software-Defined Networks. *IEEE Access, 8*, 72585-72597.

Hande, Y., & Muddana, A. (2021). A survey on intrusion detection system for software defined networks (SDN). In *Research Anthology on Artificial Intelligence Applications in Security* (pp. 467-489). IGI Global.

Hao, W., Jiang, Y., & Gao, J. (2017). Detection mechanisms of rule conflicts in SDN based on a path-tree model. 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS),

Hauser, F., Schmidt, M., Häberle, M., & Menth, M. (2020). P4-MACsec: Dynamic Topology Monitoring and Data Layer Protection With MACsec in P4-Based SDN. *IEEE Access, 8*, 58845-58858.

He, D., Chan, S., Ni, X., & Guizani, M. (2017). Software-defined-networking-enabled traffic anomaly detection and mitigation. *IEEE Internet of Things Journal, 4*(6), 1890-1898.

Hong, E. T. B., & Wey, C. Y. (2017a). An optimized flow management mechanism in OpenFlow network. 2017 International Conference on Information Networking (ICOIN),

Hong, E. T. B., & Wey, C. Y. (2017b). An optimized flow management mechanism in OpenFlow network. Information Networking (ICOIN), 2017 International Conference on,

Hu, D., Hong, P., & Chen, Y. (2017). FADM: DDoS flooding attack detection and mitigation system in software-defined networking. GLOBECOM 2017-2017 IEEE Global Communications Conference,

Hu, H., Han, W., Ahn, G.-J., & Zhao, Z. (2014). FLOWGUARD: building robust firewalls for software-defined networks. Proceedings of the third workshop on Hot topics in software defined networking,

Huang, C., Hatano, T., Yamada, T., Shimada, T., & Yoshida, T. (2021). A low workload operation method for SDN switch replacement that prevents wiring mistakes. *IEICE Communications Express, 10*(7), 368-373.

Humayun, U., Hamdan, M., & Marsono, M. (2021). Early Flow Table Eviction Impact on Delay and Throughput in Software-Defined Networks. 2021 11th IEEE International Conference on Control System, Computing and Engineering (ICCSCE),

Irawati, I. D., Hadiyoso, S., & Hariyani, Y. S. (2017). Link Aggregation Control Protocol on Software Defined Network. *International Journal of Electrical and Computer Engineering, 7*(5), 2706.

Jafarian, T., Masdari, M., Ghaffari, A., & Majidzadeh, K. (2021). A survey and classification of the security anomaly detection mechanisms in software defined networks. *Cluster Computing, 24*(2), 1235-1253.

Jarschel, M., Zinner, T., Hoßfeld, T., Tran-Gia, P., & Kellerer, W. (2014). Interfaces, attributes, and use cases: A compass for SDN. *IEEE Communications Magazine, 52*(6), 210-217.

Jasinski, A., Qiao, Y., Fallon, E., & Flynn, R. (2021). A framework for the dynamic generation of workflows for network management. 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM),

Karakus, M., & Durresi, A. (2017). A survey: Control plane scalability issues and approaches in software-defined networking (SDN). *Computer Networks, 112*, 279-293.

Keti, F., & Askar, S. (2015). Emulation of software defined networks using mininet in different simulation environments. 2015 6th International Conference on Intelligent Systems, Modelling and Simulation,

Khairi, H., Ariffin, S. H., Latiff, N. A., Yusof, K. M., Hassan, M., & Rava, M. (2020). The impact of firewall on TCP and UDP throughput in an openflow software defined network. *Indonesian Journal of Electrical Engineering and Computer Science, 20*(1), 256-263.

Khairi, M. H., Ariffin, S. H., Latiff, N. A., Abdullah, A., & Hassan, M. (2018). A review of anomaly detection techniques and distributed denial of service (DDoS) on software defined network (SDN). *Engineering, Technology & Applied Science Research, 8*(2), 2724-2730.

Khamaiseh, S., Serra, E., Li, Z., & Xu, D. (2019). Detecting saturation attacks in sdn via machine learning. 2019 4th International Conference on Computing, Communications and Security (ICCCS),

Khin, C. S., Oo, M. Z., & Kyaw, A. T. (2020). Packet-in Messages Handling Scheme to Reduce Controller Bottlenecks in OpenFlow Networks. 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON),

Kiani, R., & Bohlooli, A. (2020). An Accelerated Method for Rules Anomaly Detection in Software Defined Networks. *Electronic and Cyber Defense, 8*(4), 31-39.

Kim, Y.-M., & Kang, M. (2020). Formal Verification of SDN-Based Firewalls by Using TLA+. *IEEE Access, 8*, 52100-52112.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., & Pérez, P. (2021). Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*.

Kocoloski, B., Hussain, A., Troglia, M., Ardi, C., Cheng, S., DeAngelis, D., Symonds, C., Collins, M., Goodfellow, R., & Schwab, S. (2021). Case Studies in Experiment Design on a minimega Based Network Emulation Testbed. Cyber Security Experimentation and Test Workshop,

Kuo, Y.-H., Tsai, J.-S., & Leung, T. (2020). A multilevel Bit Vector minimization method for fast online detection of conflicting flow entries in OpenFlow table. *Computer Communications*.

L. Grimaudo, M. M., E. Baralis and R. Keralapura, "SeLeCT: Self-Learning Classifier for Internet Traffic," in IEEE Transactions on Network and Service Management, vol. 11, no. 2, pp. 144-157, June 2014, doi: 10.1109/TNSM.2014.011714.130505.

Lai, Y.-C., Ali, A., Hossain, M. S., & Lin, Y.-D. (2019). Performance modeling and analysis of TCP and UDP flows over software defined networks. *Journal of Network and Computer Applications, 130*, 76-88.

Lama Ruano, F. (2017). *Creation of a virtual overlay network with SDN and VXLAN* Universitat Politècnica de Catalunya].

Lantz, B., Díaz-Montiel, A. A., Yu, J., Rios, C., Ruffini, M., & Kilper, D. (2020). Demonstration of Software-Defined Packet-Optical Network Emulation with Mininet-Optical and ONOS. Optical Fiber Communication Conference,

Latah, M., & Toker, L. (2018). Towards an efficient anomaly-based intrusion detection for software-defined networks. *IET networks, 7*(6), 453-459.

Lee, S., Ali, J., & Roh, B.-h. (2019). Performance comparison of software defined networking simulators for tactical network: Mininet vs. OPNET. 2019 International Conference on Computing, Networking and Communications (ICNC),

Li, W., Li, X., Li, H., & Xie, G. (2018). Cutsplit: A decision-tree combining cutting and splitting for scalable packet classification. IEEE INFOCOM 2018-IEEE Conference on Computer Communications,

Liaoruo, H., Qingguo, S., & Wenjuan, S. (2016). A source routing based link protection method for link failure in SDN. Computer and Communications (ICCC), 2016 2nd IEEE International Conference on,

Lin, L., & Sun, X. (2018). A Case for Systematic Detection and Rigorous Location of SDN Control Conflicts. 2018 IEEE 43rd Conference on Local Computer Networks (LCN),

Lin, Y.-D., Lai, Y.-K., Tsou, Y.-L., Lai, Y.-C., Liou, E.-C., & Chiang, Y. (2019). Generic Validation Criteria and Methodologies for SDN Applications. *IEEE Systems Journal, 13*(4), 3909-3920.

Liu, B., Ding, M., Shaham, S., Rahayu, W., Farokhi, F., & Lin, Z. (2021). When machine learning meets privacy: A survey and outlook. *ACM Computing Surveys (CSUR), 54*(2), 1-36.

Liu, C.-C., Chang, Y., Tseng, C.-W., Yang, Y.-T., Lai, M.-S., & Chou, L.-D. (2018). Svm-based classification mechanism and its application in sdn networks. 2018 10th International Conference on Communication Software and Networks (ICCSN),

Liu, S., Benson, T. A., & Reiter, M. K. (2019). Efficient and safe network updates with suffix causal consistency. Proceedings of the Fourteenth EuroSys Conference 2019,

Lo, C.-C., Wu, P.-Y., & Kuo, Y.-H. (2015). Flow entry conflict detection scheme for software-defined network. Telecommunication Networks and Applications Conference (ITNAC), 2015 International,

Lu, Y., Fu, Q., Xi, X., Chen, Z., Zou, E., & Fu, B. (2019). A policy conflict detection mechanism for multi-controller software-defined networks. *International Journal of Distributed Sensor Networks, 15*(5), 1550147719844710.

Maldonado-Lopez, F. A., Calle, E., & Donoso, Y. (2015). Detection and prevention of firewall-rule conflicts on software-defined networking. 2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM),

Meena, R. C., Bundele, M., & Nawal, M. (2020). RYU SDN Controller Testbed for Performance Testing of Source Address Validation Techniques. 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE),

Mehr, S. Y., & Ramamurthy, B. (2019). An SVM Based DDoS Attack Detection Method for Ryu SDN Controller. Proceedings of the 15th International Conference on emerging Networking EXperiments and Technologies,

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR), 54*(6), 1-35.

Menuka, P., Piamrat, K., & Hamma, S. (2019). Network traffic classification using machine learning for software defined networks. IFIP International Conference on Machine Learning for Networking (MLN'2019),

Merayo, N., de Pintos, D., Aguado, J. C., de Miguel, I., Durán, R. J., Fernández, P., Lorenzo, R. M., & Abril, E. J. (2021). An Experimental OpenFlow Proposal over Legacy GPONs to Allow Real-Time Service Reconfiguration Policies. *Applied Sciences, 11*(3), 903.

Metter, C., Seufert, M., Wamser, F., Zinner, T., & Tran-Gia, P. (2017). Analytical Model for SDN Signaling Traffic and Flow Table Occupancy and Its Application for Various Types of Traffic. *IEEE Transactions on Network and Service Management, 14*(3), 603-615.

Mininet – An Instant Virtual Network on your Laptop (or other PC), & http://mininet.org/, a. a.

Mohammed, S. S., Hussain, R., Senko, O., Bimaganbetov, B., Lee, J., Hussain, F., Kerrache, C. A., Barka, E., & Bhuiyan, M. Z. A. (2018). A new machine learning-based collaborative ddos mitigation mechanism in software-defined network. 2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob),

Mohanty, N. N., Hemadri, D., Munivenkatarayappa, A., Shetty, N., Subramanyam, V., Biswas, S. K., Chanda, M. M., & Shivachandra, S. B. (2021). Development of recombinant NS1-NS3 antigen based indirect ELISA for detection of bluetongue antibodies in sheep. *Journal of Immunological Methods, 490*, 112959.

Monika, P., Negara, R. M., & Sanjoyo, D. D. (2020). Performance analysis of software defined network using intent monitor and reroute method on ONOS controller. *Bulletin of Electrical Engineering and Informatics, 9*(5), 2065-2073.

Monir, M. F., & Akhter, S. (2019). Comparative Analysis of UDP Traffic With and Without SDN-Based Firewall. 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST),

Myint Oo, M., Kamolphiwong, S., Kamolphiwong, T., & Vasupongayya, S. (2019). Advanced support vector machine-(ASVM-) based detection for distributed denial of service (DDoS) attack on software defined networking (sdn). *Journal of Computer Networks and Communications, 2019*.

Naing, M. T., Khaing, T. T., & Maw, A. H. (2019). Evaluation of TCP and UDP Traffic over Software-Defined Networking. 2019 International Conference on Advanced Information Technologies (ICAIT),

Ono, D., Guillen, L., Izumi, S., Abe, T., & Suganuma, T. (2021). A proposal of port scan detection method based on Packet-In Messages in OpenFlow networks and its evaluation. *International Journal of Network Management*, e2174.

Ontiveros, C. (2019). A SOFTWARE DEFINED NETWORK IMPLEMENTATION USING MININET AND RYU _ A Project Presented.

Othman, W. M., Chen, H., Al-Moalmi, A., & Hadi, A. N. (2017). Implementation and performance analysis of SDN firewall on POX controller. 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN),

Pandiyan, V., Prost, J., Vorlaufer, G., Varga, M., & Wasmer, K. (2021). Identification of abnormal tribological regimes using a microphone and semi-supervised machine-learning algorithm. *Friction*, 1-14.

Peng, H., Sun, Z., Zhao, X., Tan, S., & Sun, Z. (2018). A detection method for anomaly flow in software defined network. *IEEE Access, 6*, 27809-27817.

Phan, T. V., Nguyen, T. G., Dao, N.-N., Huong, T. T., Thanh, N. H., & Bauschert, T. (2020). DeepGuard: Efficient Anomaly Detection in SDN With Fine-Grained Traffic Flow Monitoring. *IEEE Transactions on Network and Service Management, 17*(3), 1349-1362.

Pisharody, S. (2017). *Policy conflict management in distributed SDN environments* Arizona State University].

Pisharody, S., Natarajan, J., Chowdhary, A., Alshalan, A., & Huang, D. (2017). Brew: A security policy analysis framework for distributed sdn-based cloud environments. *IEEE transactions on dependable and secure computing*.

Preamthaisong, P., Auyporntrakool, A., Aimtongkham, P., Sriwuttisap, T., & So-In, C. (2019). Enhanced DDoS detection using hybrid genetic algorithm and decision tree for SDN. 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE),

Putatunda, S. (2021). Supervised Learning for Streaming Data. In *Practical Machine Learning for Streaming Data with Python* (pp. 57-96). Springer.

Raghav, P., & Dua, A. (2017). Enhancing flow security in ryu controller through set operations. 2017 3rd IEEE International Conference on Computer and Communications (ICCC),

Ramdhani, M. F., Hertiana, S. N., & Dirgantara, B. (2016). Multipath routing with load balancing and admission control in Software-Defined Networking (SDN). 2016 4th International Conference on Information and Communication Technology (ICoICT),

Rasool, R. U., Ashraf, U., Ahmed, K., Wang, H., Rafique, W., & Anwar, Z. (2019). Cyberpulse: A machine learning based link flooding attack mitigation system for software defined networks. *IEEE Access, 7*, 34885-34899.

Ray, P. P., & Kumar, N. (2021). SDN/NFV architectures for edge-cloud oriented IoT: A systematic review. *Computer Communications*.

Romanov, O., Saychenko, I., Marinov, A., & Skolets, S. (2021). RESEARCH OF SDN NETWORK PERFORMANCE PARAMETERS USING MININET NETWORK EMULATOR. *Information and Telecommunication Sciences*(1), 24-32.

Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., & Müller, K.-R. (2021). A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*.

Sahu, M., & Dash, R. (2021). A Survey on Deep Learning: Convolution Neural Network (CNN). In *Intelligent and Cloud Computing* (pp. 317-325). Springer.

Samir, M., Azab, M., & Samir, E. (2021). SD-CPC: SDN Controller Placement Camouflage based on Stochastic Game for Moving-target Defense. *Computer Communications, 168*, 75-92.

Santos, D., Gomes, T., & Tipper, D. (2021). SDN controller placement with availability upgrade under delay and geodiversity constraints. *IEEE Transactions on Network and Service Management, 18*(1), 301-314.

Santos, R., Souza, D., Santo, W., Ribeiro, A., & Moreno, E. (2020). Machine learning algorithms to detect DDoS attacks in SDN. *Concurrency and Computation: Practice and Experience, 32*(16), e5402.

Schueller, Q., Basu, K., Younas, M., Patel, M., & Ball, F. (2018). A hierarchical intrusion detection system using support vector machine for SDN network in cloud data center. 2018 28th International Telecommunication Networks and Applications Conference (ITNAC),

Shafi, Q., Basit, A., Qaisar, S., Koay, A., & Welch, I. (2018). Fog-assisted SDN controlled framework for enduring anomaly detection in an IoT network. *IEEE Access, 6*, 73713-73723.

Shan, T., Guo, R., Li, M., Yang, F., Xu, S., & Liang, L. (2021). Application of Multitask Learning for 2-D Modeling of Magnetotelluric Surveys: TE Case. *IEEE Transactions on Geoscience and Remote Sensing*.

Singla, A. (2021). SatNetLab: a call to arms for the next global internet testbed. *ACM SIGCOMM computer communication review, 51*(2), 28-30.

Tang, F., Zhang, H., Yang, L. T., & Chen, L. (2019). Elephant flow detection and differentiated scheduling with efficient sampling and classification. *IEEE Transactions on Cloud Computing*.

Tok, M. S., & Demirci, M. (2021). Security analysis of SDN controller-based DHCP services and attack mitigation with DHCPguard. *Computers & Security, 109*, 102394.

Tran, C. N., & Danciu, V. (2019a, 2019/07/08). A General Approach to Conflict Detection in Software-Defined Networks. *SN Computer Science, 1*(1), 9. https://doi.org/10.1007/s42979-019-0009-9

Tran, C. N., & Danciu, V. (2019b). Hidden Conflicts in Software-Defined Networks. 2019 International Conference on Advanced Computing and Applications (ACOMP),

Tran, C. N., & Danciu, V. (2020). A General Approach to Conflict Detection in Software-Defined Networks. *SN Computer Science, 1*(1), 9.

Tseng, Y., Pattaranantakul, M., He, R., Zhang, Z., & Naït-Abdesselam, F. (2017). Controller DAC: Securing SDN controller with dynamic access control. Communications (ICC), 2017 IEEE International Conference on,

Usman, S., Winarno, I., & Sudarsono, A. (2020). Implementation of SDN-based IDS to protect Virtualization Server against HTTP DoS attacks. 2020 International Electronics Symposium (IES),

Wang, A., Mei, X., Croft, J., Caesar, M., & Godfrey, B. (2016a). *Ravel: A Database-Defined Network* Proceedings of the Symposium on SDN Research, Santa Clara, CA, USA. https://doi.org/10.1145/2890955.2890970

Wang, A., Mei, X., Croft, J., Caesar, M., & Godfrey, B. (2016b). Ravel: A database-defined network. Proceedings of the Symposium on SDN Research,

Wang, B., Sun, Y., & Xu, X. (2020). A Scalable and Energy-efficient Anomaly Detection Scheme in Wireless SDN-based mMTC Networks for IoT. *IEEE Internet of Things Journal*.

Wang, C., & Youn, H. Y. (2019). Entry Aggregation and Early Match Using Hidden Markov Model of Flow Table in SDN. *Sensors, 19*(10), 2341.

Wang, M.-H., Chen, L.-W., Chi, P.-W., & Lei, C.-L. (2017). SDUDP: A reliable UDP-Based transmission protocol over SDN. *IEEE Access, 5*, 5904-5916.

Wang, P., Ye, F., Chen, X., & Qian, Y. (2018). Datanet: Deep learning based encrypted network traffic classification in sdn home gateway. *IEEE Access, 6*, 55380-55391.

Wang, S.-Y. (2014). Comparison of SDN OpenFlow network simulator and emulators: EstiNet vs. Mininet. 2014 IEEE Symposium on Computers and Communications (ISCC),

Wazirali, R., Ahmad, R., & Alhiyari, S. (2021). SDN-OpenFlow Topology Discovery: An Overview of Performance Issues. *Applied Sciences, 11*(15), 6999.

Weber, J. S., Neves, M., & Ferreto, T. (2021). VANET simulators: an updated review. *Journal of the Brazilian Computer Society, 27*(1), 1-31.

Xia, W., Wen, Y., Foh, C. H., Niyato, D., & Xie, H. (2015). A survey on software-defined networking. *IEEE Communications Surveys & Tutorials, 17*(1), 27-51.

Xiao, P., Qu, W., Qi, H., Xu, Y., & Li, Z. (2015). An efficient elephant flow detection with cost-sensitive in SDN. 2015 1st International Conference on Industrial Networks and Intelligent Systems (INISCom),

Xu, J., Wang, J., Qi, Q., Sun, H., & He, B. (2018). Deep neural networks for application awareness in SDN-based network. 2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP),

Yao, J. (2018). Event-based anomaly detection for non-public industrial communication protocols in SDN-based control systems.

Yingchareonthawornchai, S., Daly, J., Liu, A. X., & Torng, E. (2018). A sorted-partitioning approach to fast and scalable dynamic packet classification. *IEEE/ACM Transactions on Networking, 26*(4), 1907-1920.

Yoshioka, K., Hirata, K., & Yamamoto, M. (2017a). Routing method with flow entry aggregation for software-defined networking. Information Networking (ICOIN), 2017 International Conference on,

Yoshioka, K., Hirata, K., & Yamamoto, M. (2017b). Routing method with flow entry aggregation for software-defined networking. 2017 International Conference on Information Networking (ICOIN),

Yu, C., Lan, J., Xie, J., & Hu, Y. (2018). QoS-aware traffic classification architecture using machine learning and deep packet inspection in SDNs. *Procedia computer science, 131*, 1209-1216.

Yu, Y., Guo, L., Liu, Y., Zheng, J., & Zong, Y. (2018). An efficient SDN-based DDoS attack detection and rapid response platform in vehicular networks. *IEEE Access, 6*, 44570-44579.

Zaw, H. T., & Maw, A. H. (2019). Traffic management with elephant flow detection in software defined networks (SDN). *International Journal of Electrical & Computer Engineering (2088-8708), 9*.

Zhang, F., Wang, N., Hu, Z., Wu, M., Wang, D., Zhou, Z., & Wang, Y. (2021). A study of UDP and TCP FPGA implementation for data acquisition system. *Journal of Instrumentation, 16*(07), P07044.

Zhang, Y., Tiňo, P., Leonardis, A., & Tang, K. (2021). A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Zhou, J. (2014). *Multicatalyst system in asymmetric catalysis*. John Wiley & Sons.

Zhou, Q., Yu, J., & Li, D. (2021). TSSBV: A Conflict-Free Flow Rule Management Algorithm in SDN Switches. 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring),

Zhou, Z.-H. (2021). Ensemble learning. In *Machine Learning* (pp. 181-210). Springer.

LIST OF PUBLICATIONS

The following are the list of publications by the author relevant to this thesis:

Indexed Journal

Mutaz H.H.Khairi, Sharifah H. S. Ariffin, N. M. Abdul Latiff, Kamaludin Mohamad Yusof, Mohamed Khalafalla Hassan, Fahad Taha AlDhief, Mosab Hamdan, Suleman Khan, Muzaffar Hamzah (2021), Detection and Classification of Conflict Flows in SDN Using Machine Learning Algorithms, IEEE Access, Volume 9, ISSN: 2169-3536, **(ISI Q1).**

Indexed Journal:

Mutaz H.H.Khairi, Sharifah H. S. Ariffin, N. M. Abdul Latiff, Kamaludin Mohamad Yusof,(2021), Generation and Collection of Data for Normal and Conflicting Flows in Software Defined Network Flow Table , *Indonesian Journal of Electrical Engineering and Computer Science*, Volume 22, No 1, ISSN: 2502-4752, (**Indexed by SCOPUS, Q3**).

Indexed Journal:

Mutaz H.H.Khairi, Sharifah H. S. Ariffin, N. M. Abdul Latiff, Kamaludin Mohamad Yusof, Mohammed Kallafallah Hassan (2021), A Review of Flow Conflicts And Solutions In Software Define Network (SDN), *IIUM Engineering Journal*, (Accepted) will published in Volume 22, No. 2, 2021, ISSN: 2289-7860, (**Indexed by SCOPUS, Q3**).

Indexed Conference and Journal:

Mutaz H.H.Khairi, Sharifah H. S. Ariffin, N. M. Abdul Latiff, Kamaludin Mohamad Yusof, M. K. Hassan, Mohammad Rava, (2020), The Impact of Firewall on TCP and UDP Throughput in an OpenFlow Software Defined Network, *Indonesian Journal of Electrical Engineering and Computer Science*, Volume 20, No 1, ISSN: 2502-4752, (**Indexed by SCOPUS, Q3**).

Indexed Journal:

Mutaz H.H.Khairi, Sharifah H. S. Ariffin, N. M. Abdul Latiff, A. S. Abdullah, M. K. Hassan, (2018), A Review of Anomaly Detection Techniques and Distributed Denial of Service (DDoS) on Software Defined Network (SDN), *Engineering, Technology & Applied Science Research,* Volume. 8, No. 2, ISSN 2241-4487.