A Machine Learning Approach to Implementation of Link Aggregation Control Protocol over Software Defined Networking

Nazrul Islam
Department of Information and Communication Technology
Mawlana Bhashani Science and Technology University
Santosh, Tangail-1902
Bangladesh

Md. Fazla Rabbi
Department of Information and Communication Technology
Mawlana Bhashani Science and Technology University
Santosh, Tangail-1902
Bangladesh

S.M. Shamim
Department of Information and Communication Technology
Mawlana Bhashani Science and Technology University
Santosh, Tangail-1902
Bangladesh

Md. Saikat Islam Khan
Department of Computer Science and Engineering
Mawlana Bhashani Science and Technology University
Santosh, Tangail-1902 Bangladesh

Mohammad Abu Yousuf
Institute of Information Technology
Jahangirnagar University
Savar, Dhaka-1342
Bangladesh

ABSTRACT
Software Defined Networking (SDN) is a complete and directly programmable network model which splits the control plane to the network data plane. Link Aggregation (LAG) is the grouping of multiple links into a single aggregated logical link with a higher bandwidth of aggregated data. This research sets out the implementation of the Link Aggregation Control Protocol (LACP) on SDN using Mininet Emulator. OpenvSwitch (OVS) acts as a transfer function, while RYU acts as an OpenFlow controller. Mininet Emulator, which is installed on Ubuntu Virtual Machine (VM) for LACP implementation in SDN. The study indicates that the speed of data communication has improved using LACP. This work also addressed that LACP provides inherent automatic redundancy that dynamically redirected to flow across the remaining links while one of the multiple links used in the aggregated groups fail or disabled. Additionally, Machine Learning (ML) approaches are also used to predict bandwidth based on statistical analysis of the data set. The Internet Service Provider (ISP) can gain more advantages to forecast bandwidth and serve customers.

Keywords
Software Defined Networking (SDN), Link Aggregation Control Protocol (LACP), Open Flow, Mininet Emulator, RYU Controller

1. INTRODUCTION
The computer network is composed of computer systems associated with various computer devices connected together via communication channels to facilitate communication and resource sharing among a large number of users. It provides a quick and practical way to share and transfer information to desired individuals. The networking industries are rapidly changing the way they do business and live. Business decisions need to be made more and more quickly and the decision-maker demands immediate access to accurate information. The progress of the computer network has improved rapidly. This expansion of the computer network will have an impact on the increase in data traffic in the network. Computer networks needed to be more flexible, scalable and programmable, with greater availability. The conventional network architecture incorporates a data plan and a control plan into the identical device which is difficult to meet these requirements.As well, the traditional network is complex and difficult to sustain.

In order to overcome traditional network limitations, Software Defined Network (SDN) proposed a renewed and comprehensive network design approach. Software Defined Networks (SDN) [1-3] has provided a platform for designing computer networks, building the structure and managing all the components of the network. This is decoupled the network control plane from the data plane to optimize each of them. It appeared as a new paradigm that allowing the network to be a programmable and a pluggable component. The study [4] shows that the performance analysis of Software Defined Wireless Network (SDWN) with multiple domain for inter controller communication. It has established itself as a new paradigm in the huge cloud architecture that makes the network a programmable and plug-in component [5]. For these reasons, various organizations have been focusing heavily on the development of the SDN in recent years [6].

Moreover, the SDN is actively explored by the majority of network operators and owners. It separates the data plane from the control plane, making it possible to easily add new and powerful creative features or protocols on traditional computer networks. To further optimize, minimize operational costs and strengthen network architecture, businesses and organizations worldwide, either by deploying SDNs or by planning to deploy them on their network. The SDN will be considered the most widespread information technology in a year years [7-9]. It is estimated that US $2 billion has already been invested in discovering knowledge about the SDN [10].

Prior to the deployment of control and data plane as an

integrated system. However, the separation of the SDN design is called disintegration [11]. SDN design Application Programming Interface (API) [12] defines the communication between higher level components and a particular component of a network is granted by northbound [13] interfaces, whereas communication between the fabric switch and protocols is let on by the southbound interfaces [14]. As a result of the fertilized architecture, Southbound (API) and Northbound (API) are not physically located in the same location in SDN.

Over the past few years, improving network security, streamlining network management and improving resource efficiency have become the main goals of computer networking. Nowadays, there is more and more data traffic in the global network due to the huge upgrade of information technologies. In order to process the enormous data plus router and switch is required. Adding more switches and routers will translate into lower performance, increasing the cost. Besides, all new network devices will need to manually configure and change device trends to configure other devices connected to the network. Overcoming these limitations requires a new infrastructure design that takes into account efficient routing, access control and load balancing. Link aggregation is the most preferable solution to this problem. Link aggregation is one of the main components of elastic network design, which ensures continuity of connectivity in case of link failure. It aggregates two or more connections between devices and expands the available bandwidth. Combining two links gives double the bandwidth of a link.

To our knowledge, no work has been done to implement the LACP based on the machine learning approach. This work implements LACP on software-defined networking using machine learning to achieve outcomes of improved bandwidth in the communication channel and avoid redundancy in the event of switch failure while communicating between two channels. As well, an algorithm is proposed for the bandwidth prediction for a different area.

This document is organized along the following lines. The document review is described in Section 2 and the technical background is described in Section 3. Also, section 4 goes on to outline the tools used to implement the system. Section 5 presents the overall implementation process and describes the machine learning approach. Section 6 focuses on system outcomes and analysis. Finally, Section 7 concludes this document with upcoming work.

2. LITERATURE REVIEW

From the start, the bandwidth of a network link and the fault tolerance of a system are two key factors. Many efforts have been made to improve these two components over the years, and the process is ongoing. According to [15], N. Kumari *et al.* proposed the use of Long Short Term Memory for bandwidth prediction in Software Defined Data Centernetwork and also proposed to use push-based telemetry to reduce the overhead of data collection in Software Defined Data Centers. The author used ML approaches to predict bandwidth and implemented a remote measurement system with 90% accuracy.According to [16], the authors focus on the firewall on Spanning Tree Protocol (STP) on SDN and monitor the bandwidth available for network traffic. ML techniques are used to build the highest quality infrastructure.

Another study [17] describes the current fault tolerance approaches in the control layer as well as in the data layer which is easily identifiable during communication and also presented requirements and clarification to fault tolerance.It also describes that master-slaves schemes soles the requirements of fault tolerance, which provides transparent fault tolerance and fast failovers. A fault tolerance system presented as CORONET, which can recover as a result of multiple link failures by Kim et al. [18]. The system can quickly recover when a faultoccurs and easily scale large networks. Document [19] proposed a mechanism for energy efficiency in aggregating 802.3ad to traffic, which can reduce traffic by 25.4%. The mechanism has two prior components: first one is negotiation protocol for the nodes lined to 802.3ad and the second one is an algorithm for estimating the number of active links to form an aggregate link of traffic outbound.

Similarly, the author [20], stressed the ML approach for detection of DDoS attack in a network and to build a distributed denialofservice detection system (DDOS) that can show 93% precision in fault detection. In that approach on a private network dataset in anSDN environment they used AdaBoosting with decision stump as a weak classifier to train the model. Similarly, Zhang *et al.* [21] Controller synchronization issue formulated as a Markov Decision Process (MDP) with Deep Neural Network (DNN) and strengthening learning to achieve a smart synchronization policy called Multi-Armed Cooperative Synchronization (MACS). They stated MACS-based synchronization policy is 56% more efficient in abstracting latent pattern in SDN environment and rendering significant eminence because of the DNN's unprecedented ability and performs 30% better than Open Network Operating System (ONOS) and SDN controller synchronize heuristics.

Another approach [22] with combined reinforcement learning and is deep reinforcement learning and reduced long term control layer overhead by 60% and increased 14% table hit ratio in the flow entry table. This machine learning approach can efficiently minimize the SDN network control plane overhead. They stated that performance evaluations confirm that the DQN-based approach performs better than the conventional reinforcement learning approaches and results in quick concurrence.

3. TECHNICAL BACKGROUND

3.1 Link Aggregation

Link aggregation [23] is anapproach for the development of a logical link combining multiple physical lines defined in IEEE802.1AX-2008. Communication capacity and availability are augmented between devices using link aggregation where Fast Ethernet and Gigabit Ethernet technology acts as a workforce. It substantially increases the transmission speed and accessibility compared to the conventional connection using a single cable. Through the combination of parallel physical links, only one logical link is created. Figure 1 illustrates the relationship aggregation diagram.

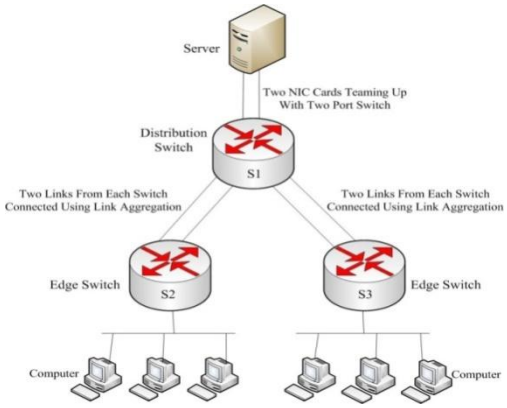

Fig 1: Link Aggregation Design Diagram [23, 24]

The distribution switch is connected to the server through linkage where two Edge switches are connected to the distribution switch. In order to make a larger link, two or more links from two or more ports on a server can connect with two or more switching ports. Upgrading the link aggregation arrows of the conventional network offers increased link capacity and improved link availability [24].

3.1.1 Higher Link Availability

The link aggregation feature increases bandwidth, provides malleable deterioration, if there is a failure and also increases availability. Prevents communication failure of interconnected devices by reducing the failure of single components. The failure of a link or channel within an aggregation minimizes the available capacity leading to uninterrupted data flow since the systemautomatically load-balances traffic across remaining links and thereby connections are maintained.

3.1.2 Link Capacity

Upgrading the conventional network link aggregation arrows provides additional link capacity and better link availability. Mostly 10 MB/s, 100 MB/s, and 1000 MB/s. Data are provided by Standard LAN technology. Aggregate links may use a combination of those speeds on one logical link. Moreover, it is possible to establish multi-gigabit links, thereby providing substantially increased bandwidth. If a capacity exceeding 1000Mb/s is required, a high-speed connection is formed with several 1000Mb/s connections.

3.1.2 Aggregating replaces upgrading

The upgrade usually happens in factors of 10 and the device cannot take advantage of the upgrade in many cases. The upgrade typically occurs in factors of 10 and the appliance cannot benefit from the upgrade in many cases. Many network administrators have already experienced upgrading network hardware, i.e. Switching from 100 Mb/s network adapters 1000 Mb/s network adapters led to a performance progress less than the 10:1 ratio implied by the hardware change or perhaps no improvement at all. Aggregation of links can be inexpensive in comparison with a native speed improvement.

3.1.3 Types of Link Aggregation

Link Aggregation is commonly deployed in three connections

- Switch-to-Switch Connections

In such connections, a single aggregate link is formed by joining more than one working group. By simply aggregating multiple links, high-speed connections can be carried out without any hardware upgrades.

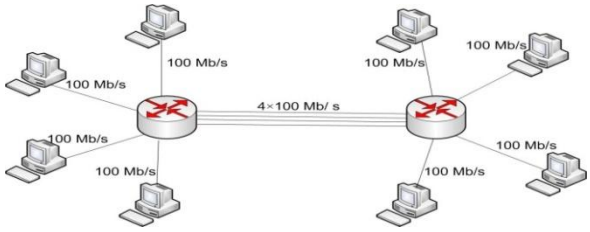

Fig 2: Switch to Switch Link Aggregation [24]

In Figure 2 above, two interconnected switches with 4 100 MB/s connections. In case of failure of one of the links, the rest of the links in the link aggregation group manages the traffic and the connection stays intact.

- Switch-to-Station (Server or Router) Connections

Nowadays, most server platforms saturate 100 MB/s link with numerous available applications. The ability to connect is a limiting factor here. On Figure-3, two switches are connected by 100Mb/s and one server is connected to each switch using four 100 MB/s links.

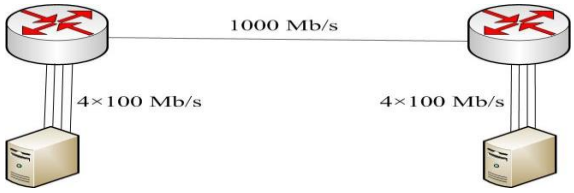

Fig 3: Continue with station link aggregation [24]

In this case, link aggregation enhances the performance of the station subject to link constraints. Performance can be enhanced without any hardware upgrade to the server or switch, but by aggregating several links.

- Station-to-Station Connections

There is no switch interference with station-to-station connections. Two servers are interconnected through an aggregate of 4 links of 1000 Mb/s in Figure 4.

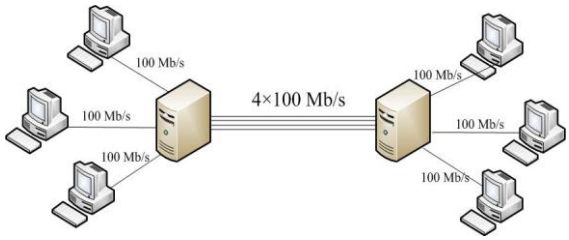

Fig4: Station to Station Link Aggregation [24]

To maintain server consistency in real time, high performance is necessary. The high-speed Station-to-Station connection can be convenient for server redundancy or multiple processing applications. This configuration is known as the backbone network.

3.2 Machine Learning in Software Defined Network

Due to technological developments the network SDN has attracted a lot of attention on the part of researchers as a modern form of network design. It is largely dependent on many network design infrastructures. Additionally, an SDN offers greater programmability in the network by separating the control plane from the data plane. However, the likelihood

of attacking the SDN also increases as the SDN, dividing it into the application layer, the data layer and the control layer [25]. In contrast, ML approaches have strong potential to address SDN problems. ML is a technology that can efficiently derive computer information and forecast exactly the future resource needs of each virtualized, software-based appliance and the future service requirements of each customer [26].

In the SDN, three types of ML algorithms, including supervised and unsupervised learning and improvement, are used for intrusion detection, bandwidth forecasting, spectrum optimization and network traffic management [27]. In supervised learning, the ML algorithm trained on the label data, wherein unsupervised learning, unlabeled data are used for training so that the ML algorithm can find the pattern on its own. Moreover, strengthening learning focuses specifically on the agent in an unpredictable and theoretically complex situation, learning to achieve a goal. Such algorithms build a stronger SDN environment.

3.2.1 Anomaly Detection

ML algorithms are integrated into the Network Intrusion Detection System (NID) in the SDN environment to achieve greater accuracy and detection speed. This approach is primarily used to deter attacks on networks and to control or follow up on malicious activities of attackers, e.g. denial of service attacks [28]. One of the key conclusions by using the NEST method of machine learning is that do not need technical experience, as in the black and white list. Numerous machine learning strategies has been used to build NEST structures, for example the closest K-NN (k-NN) [29] clustering and neighborhood algorithms, genetic and fuzzy algorithms [30] etc.

3.2.2 Rerouting Traffic

The main objective of this strategy is to divert traffic from a suspicious host to an emulated host, where it can be thoroughly investigated [31].The emulated host would respond to links triggered by the malicious host in the same way that was encountered in the preliminary anomaly that caused the malicious host to be identified.

3.2.3 Bandwidth prediction

The available bandwidth can also be predicted successfully with the help of machine learning methods. To calculate usable bandwidth, the bandwidth estimator includes prior knowledge of the network path connectivity capacity. As packets pass through a network, the dispersal that occurs provides information that may expose the associated network settings. Using a fluid flow pattern of a bottleneck connection, the detection tools measure packet dispersion to assess available bandwidth. However, problems exist if the dispersion concerning the model is skewed, e.g., non-fluid flow, packet clustering due to the disrupted coalescing, and usually unreliable time-stamping. Modelling these impacts is known to be time consuming or even insoluble. This provides the ability to use machine learning techniques to estimate bandwidth.

4. TOOLS USED TO IMPLEMENT THE SYSTEM

4.1 Mininet

The minuet is a network emulator that creates a virtual network using virtual switches, controllers (OpenFlow) and several hosts. These hosts can easily communicate with one another through virtual connections in the virtual network environment. The SDN environment can be easily configured at the moment to support the OpenFlow protocol. Also, Linux applications may run in the emulated network, which was created using mininet. For developmental purposes, mininet provides a network stack with the Linux kernel. Due to this, with minimal change, the network code emulated on mininet can be easily implemented with real hardware.

4.2 RYU Controller

RYU is a popular component based controller on the SDN framework. It provides software components to go forward and straightforward APIs for developers to easily build control applications and management system. As the RYU controller is open source, the source code is open on Github and it is supported by the open RYU community [32]. All source code written in Python, which is available for free with the Apache 2.0 license. The RYU controller [33] supports a variety of protocols to manage network devices, such as OpenFlow, Netconf, OF-config, etc. A brief description of OpenFlow can be found in [34]. This protocol is used when communicating between the controller and network devices. OpenFlow provides remote management of redirection tables in network switches, routers and other remote access points. RYU supports OpenFlow extensions such as 1.0, 1.2, 1.3, 1.4, 1.5 and Nicira. Some other popular SDN controller examples on the market and search for being NOX [35], Floodlight [36] and Beacon [37].

5. IMPLEMENTATION

5.1 Proposed Network Topology

A network topology was created using the Mininet API [38] and configured the required topology that appears in Figure 5. When executing the built network script, a network topology is generated. There are two relationships between host h1 and switch s1. Three other hosts h2, h3, h4 also generated and connected to the switcher. The net command enables testing of the created topology. The h1 host has been configured to execute the link aggregation feature.

More than one Network Interface Controller (NIC) can be combined into one linked interface using a method supplied by the Linux Link Driver [39]. The behavior of the linked interface depends on the mode in which the modes provide standby or load balancing services. The Linux link driver features ensure that the integrity of the links is monitored as well as.

Initially, the h1 host does link aggregation since the link driver module has been loaded into it. Two interfaces exist within the h1 host, namely h1-eth0 and h1-eth1. These two interfaces of one logical interface combine, that is to say bond0. A new bond0 logical interface was created and also assigned MAC 02:01:02: 03:04:08 addresses to bond0.

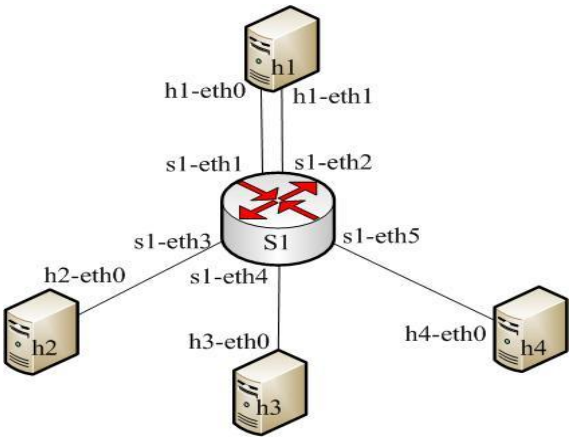

Fig 5: Proposed Network Topology

Physical interfaces have been added in h1-eth0 and h1-eth1 for creating the local interface group. MAC address has been assigned to the physical interface for easy-to-understand value. An IP address 10.0.0.1 has been assigned to the logical interfaces. At the end, make the logical interface up with appropriate commands which are performed in the *xterm* host h1.

The MAC addresses of all the interfaces are bond0, h1-eth0, and h1-eth1 are same. The logical interface bond0 is the parent and the physical interface h1-eth0 and h1-eth1 is a child. Ifconfig command shows the state of each interface. All the pre-setting for host h1 has been completed.

5.2 Proposed Machine Learning Based Design

Figure 6 provides a basic design of bandwidth prediction using machine learning approaches.

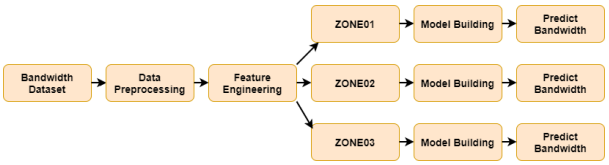

Fig 6: Proposed Machine Learning based model for predicting Bandwidth

The proposed bandwidth prediction algorithm can be presented in the following way.

Algorithm 1: Bandwidth Prediction

Input: A is the preprocessed dataset that contains dependent and independent features.

Output: αj, predicting Bandwidth for the corresponding zone.

1. Create new features including count date, and day of the week and concatenate it in the A dataset.

2. Separate the zone code feature including zone1, zone2, and zone3 respectively.

3. Perform $X_{scale} = \frac{x - x_{min}}{x_{max} - x_{min}}$ in each zone.

4. for zone [zone1, zone2, zone3]:
Return
$\alpha j = b_0 + b_1 x_1 + b_2 x_2 + \cdots b_n x_n + e$.
αj = dependent variable.
x = independent variable.
b_0 = intercept.
$b_1, b_2, \ldots b_n$ = regression coefficient.
e = error term.

5. Minimize the cost function.
for zone [zone1, zone2, zone3]:
Return *min* (||αj − *x (theta)* ||^2 + λ||*theta*||^2)
λ = *penalty term.*

6. for αj in [zone1, zone2, zone3]:
Return $SMAPE = \frac{100\%}{n} \sum_{k=1}^{n} \frac{|\alpha j - X_k|}{|\alpha j| + |X_k|}$.
αj= predicted value.
X_k= actual value.

7. Complete

5.3 Increase Bandwidth

The RYU application was run on the c0 *xterm* control window where the h1 host sends an LACP data unit every 30 seconds. Once the application is launched, the switch receives the LACP data unit from the h1 host and displays it in the operating log. Within the link aggregation function, an appropriate physical interface is enabled when the LACP data units are exchanged normally. When the LACP data exchange is paused, the physical interface is disabled. Stream inputs will only exist in the activated physical interfaces. For a specific physical interface, when LACP data units do not receive for a certain amount of time, stream inputs will not be saved. If an inactivated physical interface has received the LACP data unit, that interface is again activated.

The h1 host is connected to the switch using the aggregate linkage. Each time change sends an LACP data drive in response and receives the LACP data drive from the h1 host. Appropriate indicates the flow inputs into the *xterm* window of switch s1 where two flow inputs were saved. In switch, Packet-In message is sent when the LACP data unit is sent from host h1's h1-eth1 where the input port is s1-eth2 and the MAC address is 00:00:00:00:00:12 or LACP data unit is sent from h1's h1-eth0 where the input port is s1-eth1 and the corresponding MAC address is 00:00:00:00:00:11.

Packet Internet Grouper (PING) functions by transmitting an Internet Control Message Protocol (ICMP) echo request packet to the desired host and waits for a response from the desired host for the sending ICMP. There are many hosts in a single network and to attain the targeted host ping message is the way. First of all, ping the host h3 to the host h1 with the IP address 10.0.0.1. When running pings from host h3 to host h1,

the corresponding stream inputs of switch s1 are displayed with the corresponding stream control. Following the previous checkpoint, two additional stream inputs were recorded. These are the first and second entries with a small duration value compared to other previous flow entries.

Then execute pings from host h2 to host h1 with IP address 10.0.0.1. Once more, after ping, the flow inputs of switch s1 are checked. After the last entries in the feed, check point, two other entries were added. This is the first and second entries with a smaller length. As a matter of course, ping from h4 to h1 and the corresponding switch stream inputs are logged. Two additional new stream inputs are recorded in 2nd and 3rd with a small time value.

5.4 Enhanced fault tolerance

The link aggregation function significantly increases the fault tolerance of a system. While designing the topology, the host h2, h3 or h4 can communicate with the host h1 using port s1-eth1 in the switch and port h1-eth0 in the host h1. Alternatively, each host can communicate using the s1-eth2 port on the switch and the h1-eth1 port on the h1 host. Now split h1-eth0 which is an opposing s1-eth1 interface from the link aggregation group. Following separation send pings of h3 to host h1. The host h3 still communicates with the host h1 using the host's 1-eth2 and h1-eth1 ports. Further, h3 will be still able to communicate to the host h1 using port s1-eth1 in the switch and port h1-eth0 in the host h1 if h1-eth1 is removed from the aggregation. In case of failure of a link, link aggregation automatically recovers the communication using other links. The link aggregate function also performs load balancing between the different links in the link aggregate group. In order to ensure the efficient use of LACP, traffic is divided equally among the different routes. A specific link may be configured to carry a distinct class of traffic like voice. In addition, each individual link can also be set up to transport traffic from specific nodes (s) /server (s). LACP has a number of limitations as well. LACP occupies additional switching ports for connection of other nodes/systems. As well, in the event of a switch failure, it provides wiring, but no redundancy.

6. RESULTS AND ANALYSIS

6.1. Dataset

The dataset is obtained from [40] at 12:00 a.m. on October 1, 2017 at 11:00 p.m. on March 9, 2019. In total, 1048576 samples were used for the analysis. Moreover, 80% of the data was for training and 20% for testing. While there are a lot of data and found that there was a lot of noise in the dataset when analyzed the data. And using statistical analysis on the data set to remove some of the noise. The samples used for this data set are described as follows:

•HOUR_ID: Data retrieval time.

•UPDATE_TIME: Periods of data taking.

•ZONE_CODE: Region code

•BANDWIDTH_TOTAL: Total Access Bandwidth in 1 hour

MAX_USER: Maximum number of users who access concurrently within 1 hour

6.1.2 Data preprocessing

As there are only 5 functions available to predict bandwidth, few new functions are generated from the old one for a better understanding of the data. By expanding the number of characteristics and also overcoming the problem of correlations between dependent characteristics. In this work, new functionalities such as count_date and day_of_week are created to improve the performance of the model.

count_date: This will represent the first date October 1, 2017, in the training set as 0 and the last date March 9, 2019, in the training set as 524.

day_of_week: This represents the day of the week (Monday to Sunday) and is replaced by a value between 0 and 6.

6.1.3 Feature Engineering

Because the server zone and the server name are categorical features, one hot encoding technique is performed to handle such features. In this research, the ZONE CODE attribute is divided into three sections, including ZONE01, ZONE02, and ZONE03. Once ML is applied a statistical analysis to the dataset that is found that these characteristics are completely different from each other. This is chosen to create three separate models for each zoning code. After splitting the features, the Min-Max scalar is finally used to increase the data scale.

6.1.4 Model Building

After performing statistical analysis on the dataset and plot the min, max, average and median data for each day. This is found that there is a difference between the hours of the day and the days of the week. As well, the data is linear throughout the day. It is therefore based on the principle of linear regression. First, starting from the division of the ZONE_CODE characteristic into ZONE01, ZONE02 and ZONE03 respectively. Afterwards, the corresponding bandwidth for ZONE01 is taken and calculated the average bandwidth for ZONE01 by combining the count_date function. The peak regression is applied to predict the average bandwidth per day. In the peak regression formula, the count_date parameter is used because the trace has the same gradient as in the quadratic formula. (Refer to Figure 2). The same work procedure was used to forecast bandwidth in ZONE 02 and ZONE 03 respectively.

6.2 Performance Evaluation

The metrics used to evaluate the performance of the proposed model is following:

$$SMAPE = \frac{100\%}{n}\sum_{k=1}^{n}\frac{|G_k - X_k|}{|G_k| + |X_k|}$$

Where SMAPE is the Symmetric mean absolute percentage error, G_k, is the predicted value and X_k, is the real value. The model shows SMAPE=5.227 (SMAPE<10% is excellent) for the dataset. The model also shows an average of 81.61 percent accuracy for ZONE01, ZONE02, and ZONE03 respectively. The prediction of the bandwidth for each zone is presented in Fig 7. (a)Figure 7 shows the projected bandwidth of each zone using peak regression and the associated bandwidth for the count_date function. All other features built into the data set are built into the count_date function to calculate the total bandwidth. Finally, peak regression is applied to predict bandwidth across each area, including ZONE01, ZONE02 and ZONE03, respectively. Based on Figure 7, it is clear that peak Regression predicts bandwidth nearly perfectly. However, there are some fluctuations in the

curve as a result of the noise present in the data set. Nevertheless, the overall performance is outstanding. As a result, the proposed approach would be a remarkable candidate in bandwidth forecasting.

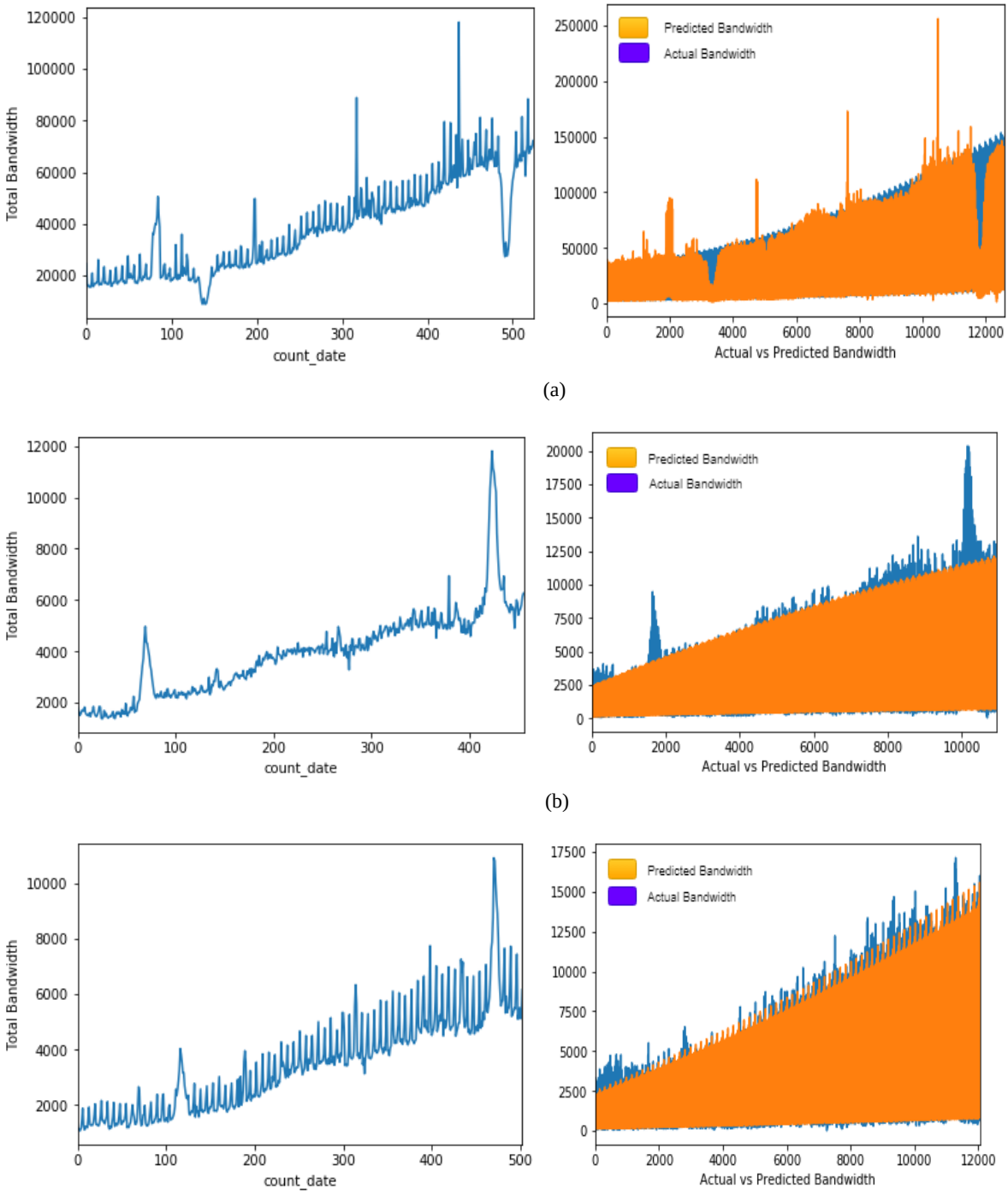

Fig 7: a. ZONEO1 Bandwidth (grouped by count_date) and Total Bandwidth vs Predicted Bandwidth
b.ZONEO2 Bandwidth (grouped by count_date) and Total Bandwidth vs Predicted Bandwidth
c.ZONEO3 Bandwidth (grouped by count_date) and Total Bandwidth vs Predicted Bandwidth

7. CONCLUSIONS

This study presents a set of results based on simulation and analysis of the data sets using the Machine Learning approaches. LACP provides increased bandwidth capability and an integrated security system that is essential for each network administrator. As well, the use of existing LACP hardware can be implemented in the SDN architecture. It will minimize operating costs to improve the efficiency and flexibility of a system.If a link fails, the LACP automatic configuration protocol provides a dynamic transition into standby mode. In addition, the proposed ML approach would be an effective approach to bandwidth forecasting and could

be used by ISPs to enhance customer service. This is an on-going research that will be implemented in the real world and will also improve the performance of the network, including Quality of Service (QoS).

8. REFERENCES

[1] Z. Latif, K. Sharif, F. Li, M. M. Karim, S. Biswas, and Y.Wang, "A comprehensive survey of interface protocols for Software defined networks," J. Netw. Comput.Appl., vol.156, no. July 2019, p. 102563, 2020, doi: 10.1016/j.jnca.2020.102563.

[2] Islam, N., Rahman, M.H., Nasir, M.K., "A Comprehensive Analysis of QoS in Wired and Wireless SDN Based on Mobile IP", *International Journal of Computer Network and Information Security (IJCNIS)*, Vol.13,No.5, pp.18-28, 2021. DOI: 10.5815/ijcnis.2021.05.02

[3] Y. Zhao, Y. Li, X. Zhang, G. Geng, W. Zhang, and Y. Sun, "A Survey of Networking Applications Applying the Software Defined Networking Concept Based on Machine Learning," *IEEE Access*, vol. 7, pp. 95397–95417, 2019, doi: 10.1109/ACCESS.2019.2928564.

[4] Rahman M.H., Islam N., Swapna A.I., Habib M.A. (2020) Analysis of Software Defined Wireless Network with IP Mobility in Multiple Controllers Domain. *In:Cyber Security and Computer Science. ICONCS 2020. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering,*vol 325. Springer, Cham. https://doi.org/10.1007/978-3-030-52856-0_42

[5] L. Nkenyereye, L. Nkenyereye, S. M. Riazul Islam, Y. H. Choi, M. Bilal, and J. W. Jang, "Software-defined network-based vehicular networks: A position paper on their modeling and implementation" *arXiv*, pp. 1–14, 2019.

[6] G. E. Vaillant, "Defense Mechanisms" *Encycl. Hum. Behav. Second Ed.*, vol. 52, no. 2, pp. 659–666, 2012, doi: 10.1016/B978-0-12-375000-6.00124-5.

[7] D. Dobrev and D. Avresky, *"Comparison of SDN Controllers for Constructing Security Functions"* 2019 IEEE *18th Int. Symp. Netw. Comput. Appl. NCA* 2019, pp. 1–5, 2019, doi: 10.1109/NCA.2019.8935053.

[8] "Network world, gartner: 10 critical it trends for the next five years" [Online Available] http://www.networkworld.com/news/2012/102212gartner-trends-263594.html. Accessed: 2 July, 2021

[9] M.t. review, 10 emerging technologies: Tr10: software-defined networking." [Online Available]http://www2.technologyreview.com/article/412194/tr10software-defined-networkingAccessed: 20 September, 2021

[10] "Enterprise networking, IDC: SDN a 2 billion market by 2016" [Online Available] http://www.enterprisenetworkingplanet.com/datacenter/-idcsdn-a-2-billion-market-by-2016.html. Accessed: 21 September, 2021

[11] "Inside SDN architecture." [Online Available] https://www.sdxcentral.com/resources-/sdn/inside-sdn-architecture/. Accessed: 17 July, 2021

[12] "SDN networking: Sdx, sdn nfv apis and sdks" [Online Available]https://www.sdxcentr-al.com/comprehensive-list-of-sdn-apis/. Accessed: 20 July, 2021

[13] "What are sdn northbound api is?" [Online Available] https://www.sdxcentral.com-/resources/sdn/north-bound-interfaces-api/. Accessed: 23 July, 2021

[14] "What are sdn southbound api is?" [Online Available] https://www.sdxcentral.com/-resources/sdn/southboundinterface-api/. Accessed: 27July, 2021

[15] N. Kumari, P. M H, S. Kumar Gangarapu, and K. Subramaniam, "Deep Recurrent Neural Network for Bandwidth Prediction in Software Defined Data Center Networks" Proc. CONECCT 2020 - 6th IEEE Int. Conf. Electron. Comput. Commun. Technol., 2020, doi: 10.1109/CONECCT50063.2020.9198338.

[16] Islam, N., Shamim, S. M., Rabbi, M. F., Khan, M. S. I., andYousuf, M. A. (2021). Building Machine Learning Based Firewall on Spanning Tree Protocol over Software Defined Networking.In Proceedings of International Conference on Trends in Computational and Cognitive Engineering (pp. 557-568).Springer, Singapore.

[17] L. Seidlitz and C. Perner, "Fault tolerance in SDN," no. April, pp. 3–7, 2020, doi: 10.2313/NET-2020-04-1.

[18] H. Kim, M. Schlansker, J. Renato Santos, J. Tourrilhes, Y. Turner and N. Feamster. CORONET: Fault tolerance for Software Defined Networks. 2012 20th IEEE International Conference on Network Protocols (ICNP), doi: 10.1109/ICNP.2012.6459938

[19] H. Imaizumi, T. Nagata, G. Kunito, K. Yamazaki, and H. Morikawa, "Power saving mechanism based on simple moving average for 802.3ad link aggregation," 2009 IEEE Globecom Work. Gc Work., 2009, doi: 10.1109/GLOCOMW.2009.5360735.

[20] Sen, S., Gupta, K. D., and Ahsan, M. M. "Leveraging machine learning approach to setup software-defined network (SDN) controller rules during DDoS attack". In Proceedings of International Joint Conference on Computational Intelligence (pp. 49-60). Springer, Singapore.

[21] Z. Zhang, L. Ma, K. Poularakis, K. K. Leung, J. Tucker, and A. Swami, "MACS: Deep reinforcement learning based SDN controller synchronization policy design," Proc. - Int. Conf. Netw. Protoc. ICNP, vol. 2019-Octob, no. i, pp. 1–11, 2019, doi: 10.1109/ICNP.2019.8888034.

[22] T. Y. Mu, A. Al-Fuqaha, K. Shuaib, F. M. Sallabi, and J. Qadir, "SDN flow entry management using reinforcement learning," ACM Trans. Auton. Adapt. Syst., vol. 13, no. 2, 2018, doi: 10.1145/3281032.

[23] Irawati, L. D., Hariyani, Y. S., & Hadiyoso, S. Link Aggregation Control Protocol on Software Defined Network. International Journal of Electrical and Computer Engineering (IJECE), 7(5), 2706, 2017. https://doi.org/10.11591/ijece.v7i5.pp2706-2712

[24] Link aggregation according to IEEE standard 802.3ad https://www.juniper.net/documentation/en_US/junose 15.1/top ics/task/configuration/802.3ad-link-aggregation-configuring.html, Accessed: 28 July,2021

[25] W. Zhijun, X. Qing, W. Jingjie, Y. Meng, and L. Liang, "Low-Rate DDoS Attack Detection Based on Factorization Machine in Software Defined Network,"

IEEE Access, vol. 8, pp. 17404–17418, 2020, doi: 10.1109/ACCESS.2020.2967478.

[26] D. S. L. Wei, K. Xue, R. Bruschi, and S. Schmid, "Guest Editorial Leveraging Machine Learning in SDN/NFV-Based Networks," IEEE J. Sel. Areas Commun., vol. 38, no. 2, pp. 245–247, 2020, doi: 10.1109/JSAC.2019.2959197.

[27] M. M. Raikar, S. M. Meena, M. M. Mulla, N. S. Shetti, and M. Karanandi, "Data Traffic Classification in Software Defined Networks (SDN) using supervised-learning," Procedia Comput. Sci., vol. 171, no. 2019, pp. 2750–2759, 2020, doi: 10.1016/j.procs.2020.04.299.

[28] Amma, N. G. B., Selvakumar, S., & Velusamy, R. L. (2020). A Statistical Approach for Detection of Denial of Service Attacks in Computer Networks. IEEE Transactions on Network and Service Management, 4537(c), 1–12. https://doi.org/10.1109/TNSM.2020.3022799

[29] Kar, P., Banerjee, S., Mondal, K. C., & Chattopadhyay, S. (n.d.). for Hierarchical Filtration of Anomalies. Springer, Singapore. https://doi.org/10.1007/978-981-13-1742

[30] Elhag, S., Fernández, A., Alshomrani, S., and Herrera, F. (2019). Evolutionary fuzzy systems: A case study of Intrusion detection systems. In Studies in Computational Intelligence (Vol. 779). Springer International Publishing.https://doi.org/10.1007/978-3-319-91341-4_9

[31] N. Provos et al., "A virtual honeypot framework" in USENIX Security Symposium, vol. 173, 2004, pp. 1–14.

[32] "osrg/ryu, github." [Online Available] https://github.com/osrg/ryu. Accessed: 22 July, 2021

[33] Islam, M.T., Islam, N. andRefat, M.A. Node to Node Performance Evaluation through RYU SDN Controller. Wireless PersCommun 112, 555–570 (2020). https://doi.org/10.1007/s11277-020-07060-4

[34] J. Xiao, S. Chen, and M. Sui, "The strategy of path determination and traffic scheduling in private campus networks based on SDN," Peer-to-Peer Netw. Appl., vol. 12, no. 2, pp. 430–439, 2019, doi: 10.1007/s12083-017-0623-z.

[35] "NOXRepo." [Online Available]http://noxrepo.org/wp/ Accessed: 12 June, 2021

[36] "Floodlight OpenFlow Controller-Floodlight Project." [Online Available] http://www.projectfloodlight.org/floodlight/. Accessed: 15 June, 2021

[37] "Home-Beacon-Confluence" [Online Available] https://openflow.stanford.edu/display/Beacon/Home/. Accessed: 17 June, 2021

[38] F. Keti and S. Askar, "Emulation of Software Defined Networks Using Mininet in Different Simulation Environments,"Proc. - Int. Conf. Intell. Syst. Model. Simulation, ISMS, vol. 2015-Octob, pp. 205–210, 2015, doi: 10.1109/ISMS.2015.46.

[39] Bonding [Online Available] http://www.linuxfoundation.org/collaborate/ workgroups/-networking/bonding Accessed: 20 June, 2021

[40] Data set [Online Available] https://drive.google.com/drive/folders/1abvPEs6Lbu UXxQAnLy_5H9yz2nSwRUkQ?usp=sharing, Accessed: 20 June, 2021