Journal of Theoretical and Applied Information Technology

30th September 2023. Vol.101. No 18
© 2023 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ENHANCING DATA TRANSMISSION FOR INTELLIGENT INFORMATION SYSTEMS USING SDN TECHNIQUE AND MACHINE LEARNING ALGORITHM

MOHAMED SALEH YADAM [1], HAZEM EL-BAKRY[2], SAMIR ABDELRAZEK[3]

Dept. of Information Systems, Faculty of Computer & Information Sciences, Mansoura University, EGYPT

E-mail: [1]msaleh.admin@aiu.edu.eg, [2]elbakry@mans.edu.eg , [3]samir.abdelrazek@mans.edu.eg

ABSTRACT

Intelligent information systems, which have recently undergone development and complexity, are now indispensable to the entire world. The networking strategy has unquestionably altered based on machine learning principles to be programable and dynamically configurable with the greatest flexibility and simplicity of use. The term "software-defined network" (SDN) refers to networks that are managed using software applications and SDN controllers as opposed to the more traditional network management consoles and commands, which require a lot of administrative overhead. To centralize network control and administration, SDN changed the topology of network devices to be more flexible and programable. The software-defined network's uses protocols for interacting with and managing switches is called OpenFlow (OF). With this protocol, the switches learn the routing information from the controller and then pass data packets based on this information. One of the most important components of the SDN is the controller, which is the smartest component of the network such as the Ryu controller. Including the importance of the Ryu controller in SDN. This article discussed how to enhance data traffic transmission and classification in the SDN environment. This research shows how we can track all data packets and traffics and automatically identify all data types and classify them correctly, so we can apply a security policy, bandwidth, and quota for each type. The most different thing we used is using a real SDN network environment and also connected a real physical lambda server that makes daily continuous training for all data traffic and synchs this at the same time with the SDN controller that applies this instantly on the real live traffic. Using Machine Learning (ML) and Artificial Intelligence (AI) to enhance the SDN environments and identify data traffic types automatically. The controller (using ML and AI) takes the needed action automatically according to the data types. Enhance security, Data Transmission, and Data Availability in the software-defined networking and Intelligent Systems environment.

Keywords: *Software-Defined Networking (SDN), Machine Learning, Information Systems, Artificial Intelligence, OpenFlow*

1. INTRODUCTION

Today’s modern business and all information systems have become more complicated and complex, and now we have information systems or intelligent information systems in all business, management, and industry fields. And all these information systems depend mainly on networking methodologies and the speed of networks for transmission data through these information systems, so we want to deal with these systems in easy and flexible ways of planning, building, and management. Networks have become an essential element in today’s modern business climate. Whether the network is completely on-premises, cloud-based, or a hybrid of both, networks provide the organizations’ need to run their applications, deliver vital communication links that services, and be competitive. Software-defined networking (SDN) represents a completely new way of how networks are configured, controlled, and operated dynamically and in easy flexible ways. With SDN, networking can be defined in software instead of configuring each node (router or switch) separately and repeatedly, so it is a more complex and effort-exhausting method. SDN typically utilizes a centralized SDN controller, which can make any changes in the configuration at any time in a fast, easy, and flexible way. So, we can enhance all transmitted traffics of data and configuration through all information system, and all thing will be automated and changed as we want in flexible and easy ways to configure and do that using SDN concepts. And with applying SDN concepts in Organizations, we need to apply Artificial intelligence and Machine learning concept. But also, with the development that has taken place in software-defined networking (SDN), the use of this technology has become an urgent necessity in all companies and organizations [1].

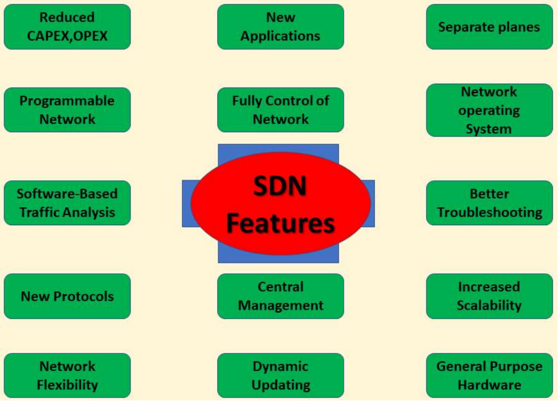


Figure 1: SDN Features

In addition to the development in the field of artificial intelligence(AI) and machine learning(ML) which has greatly impacted, SDN environments, this has become more sophisticated, flexible, and accurate, But identifying the types of data in SDN environments has become very important, as it helps to know the type of data required and sent at a high rate in the network to give it a higher priority and high bandwidth to be transmitted with no delay which speeds up and enhance the process of data transmission[2]. But the identification of traffic data types in the SDN environment is configured manually on demand, so doing this process automatically in a programable way using artificial intelligence and machine learning will make the SDN environment more and more flexible, enhanced, programmable, easy, and automated and this what we are discussing in this research [3], [4].

Most businesses are transitioning their data centers from traditional client-server designs to models that transport a lot more data (sometimes referred to as east-west traffic) between servers inside the data center. This necessitates more complex resource allocation policies and scalable network architecture[5], [6]. Furthermore, a lot of IT departments are very interested in transferring to public, private, or hybrid cloud environments. Many parts of the corporation can benefit from an SDN strategy on a high level. Because of the increased functionality of your infrastructure, you will have a competitive advantage. Because of the increased security, your company's efficiency will improve, your total cost of ownership will drop, and your risks will be reduced. You may detect and characterize endpoints and Internet of Things (IoT) devices with the aid of AI End-point Analytics, an endpoint visibility solution. With the AI Endpoint Analytics engine and the telemetry data gathered from the network from various sources, you may label endpoints. AI End-point Analytics provides profiling labels for endpoint type, hardware model, manufacturer, and operating system type. This is characterized by a multifactor classification[6], [7].

In this research, the concept and algorithms of machine learning were used with testing all this in a real live SDN environment, so we confirmed that all results were real and reflected the live activities. End-point Analytics provides your network with comprehensive insight and enforcement through the use of technologies like Trust Scores, allowing you to identify and deal with endpoints and devices that may be problematic. We can monitor and handle the issue of inconsistent and changing MAC addresses from endpoints in AI Endpoint Analytics, and you can precisely identify endpoints using a unique attribute called the DUID in place of MAC addresses. And as we see that the main technology that gives us all these features and intelligence in all information systems is SDN, but the brains of the SDN system are the SDN Controllers, which use southbound APIs to send information to switches and routers and northbound APIs to send data to applications and business logic. Moreover, in a software-defined networking (SDN) architecture, controllers are applications that govern flow control for enhanced network administration and application performance. Protocols are used by the SDN controller platform to instruct switches where to deliver packets. This platform normally operates on a server. SDN controllers minimize manual settings for individual network devices by directing traffic by forwarding policies set in place by network operators[8], [9]. So, at this point, all packets were identified and classified automatically in the SDN environment then the needed policies can be applied to each type as we need and save more time and cost with simple management.

2. LITERATURE REVIEW

For a better understanding of why SDN technology including the concepts of machine learning has become important, we should review some points that will declare the differences that have occurred and changed the concept of networking and information systems to be programable and automatically performed[3], [15]. Now, all organizations will have to change their current network infrastructure and use the concept of SDN including the features of machine learning which will convert all devices in organizations to be intelligent and programable. so, the first thing we will declare is the current (old) network and information system and how this concept has been changed to an intelligent and programable concept[11].

2.1 Current Network Infrastructures "Traditional Network"

Traditional networking architectures have significant limitations that must be overcome to meet modern IT requirements. Today's network must scale to accommodate increased workloads with greater agility, while also keeping costs at a minimum. But the traditional approach has substantial limitations when we describe the network complexity; we find the abundance of networking protocols and features for specific use cases has greatly increased network complexity[3]. Old technologies were often recycled as quick fixes to address new business requirements. Features tended to be vendor specific or were implemented through proprietary commands. And on the side of Inconsistent policies; the Security and quality-of-service (QoS) policies in current networks need to be manually configured or scripted across hundreds or thousands of network devices. This requirement makes policy changes extremely complicated for organizations to implement without significant investment in scripting language skills or tools that can automate configuration changes. Manual configuration is prone to error and can lead to many hours of troubleshooting to discover which line of a security policy or access control list (ACL) was entered incorrectly on a given device. In addition, when applications were removed, it was almost impossible to remove all the associated policies from all the devices, further increasing complexity[12]. Finally, when we talk about the Inability to scale; As application workloads change and demand for network bandwidth increases, the IT department either needs to be satisfied with an oversubscribed static network or needs to grow with the demands of the organization. Unfortunately, the majority of traditional networks are statically provisioned in such a way that increases the number of endpoints, services, or bandwidth requires substantial planning and redesign of the network[13].In Figure 2 we describe the architecture of the traditional network.

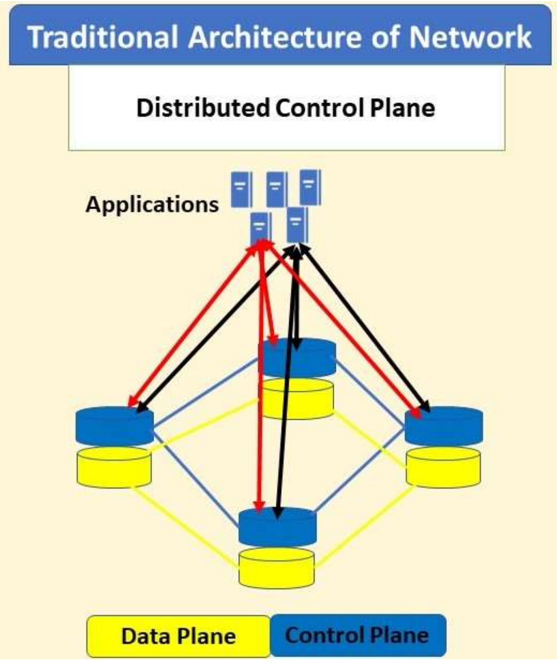

Traditional Architecture of Network
Distributed Control Plane
Applications
Data Plane
Control Plane

Figure 2: Traditional Network Architecture

2.2 SDN Architecture

SDN is a new way of looking at how networking and cloud solutions should be automated, efficient, and scalable in a new world where application services may be provided locally, by the data center, or even by the cloud[3]. This is impossible with a rigid system that's difficult to manage, maintain, and upgrade. Going forward, you need flexibility, simplicity, and the ability to quickly grow to meet changing IT and business needs. SDN most commonly means that networks are controlled by software applications and SDN controllers rather than the traditional network management consoles and commands that required a lot of administrative overhead and could be tedious to manage on a large scale. When SDN first appeared on the technology landscape, there were more rigid ideas of how SDN architectures should be designed and what defined an SDN solution. Today, customers take a broader view of what kind of SDN solution is right for them. All configuration and controlling of SDN environments are controlled from one place as shown in Figure 3. As the primary use case for SDN has evolved toward cloud automation, customers consider what they're looking for in a policy-based automation solution instead of just the specifics of the underlying SDN technology. SDN is also truly an open technology. This leads to greater interoperability, more innovation, and more flexible, cost-effective solutions. If a network is compliant with the right SDN standards, it could be controlled by multiple SDN controller applications[14]. This is better than each networking platform having its management console and commands that increase vendor lock-in and make network management even more complex. Today, multiple SDN standards are evolving in different areas, and successful SDN strategies will always be based on open, interoperable multivendor ecosystems with key open-source technologies or standardized protocols. Along with the evolution to SDN, the several technology trends affecting the architecture and design of modern data centers and enterprise networks have to be factored into SDN technology requirements[14]. In most organizations, the data center is shifting away from traditional client-server architectures to models in which significantly more data is being transferred between servers within the data center (frequently called east-west traffic), This requires more network scalability and more sophisticated policies for resource allocation. In addition, many IT departments are showing great interest in moving to public, private, or hybrid cloud environments. The top-level benefits of an SDN strategy accrue to all areas of the organization[5]. You will receive a competitive advantage because your infrastructure will do more. The speed of your business will increase, the total cost of ownership will go down and your risks will be decreased because of the greater security[15], [16].SDN Networking has many components as shown in Figure 4.

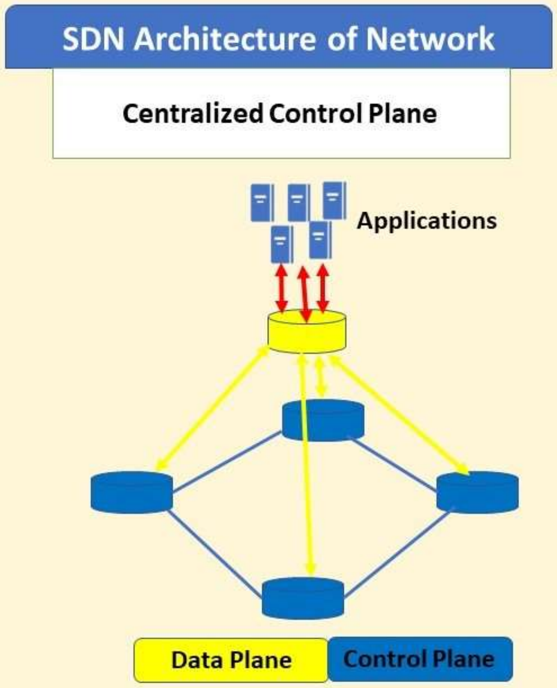


Figure 3: SDN Architecture

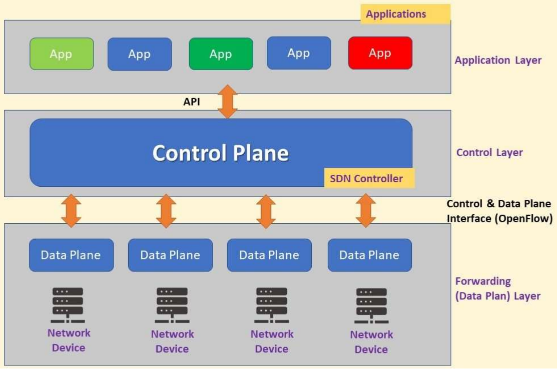


Figure 4: SDN Networking Components and Relations

2.3 OpenFlow Protocol

OpenFlow protocol as illustrated in Figure 5 is the most southbound-oriented API used in SDN, it was developed by the open networks Foundation[5], [17]. Open flow provides a layer abstraction that enables the SDN controller to safely communicate with open redirects that support flow switches and V-switches. Open flow serves as the main southward application programming interfaces (APIs) used in SDN networks, therefore, in this research, we focus mainly on the open flow-based SDN network topology. Hybrid switches reveal new possibilities by including both non-open ports and open flow. Several control messages can be sent by the controller to set up and update flow tables for a specific key.

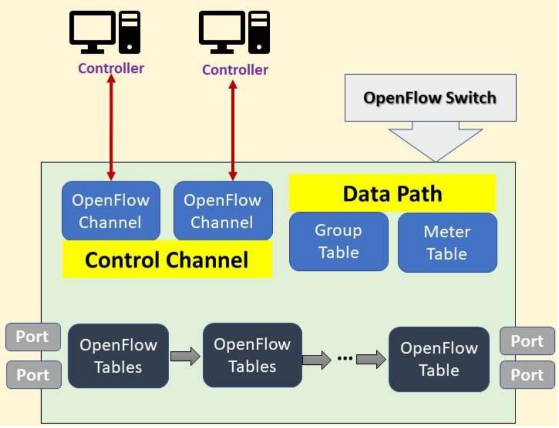


Figure 5: OpenFlow Architecture and Components

An adapter that supports open flow control and handles new incoming packets based on its flow schedule. When the new package does not match any of the entries in the flow table, there will be an error in the table. In this case, the adapter may either drop the packet or forward it to the corresponding controller using the open flow protocol, and it should be considered that the identity-based access control of the ethane project has become the first specification of open flow switches[18]. We can specify the OpenFlow protocol as described in Figure 6.

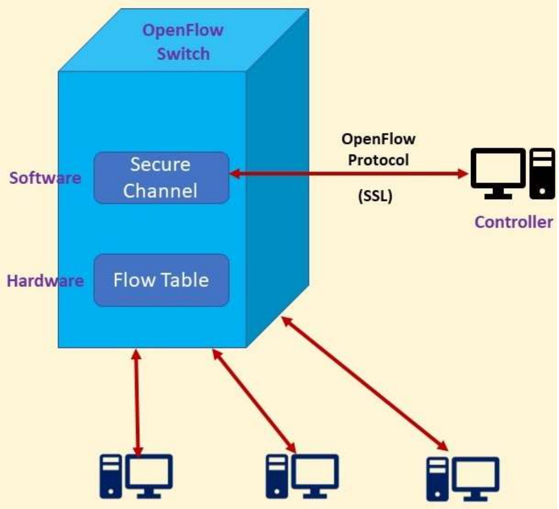


Figure 6: OpenFlow Controller Specification

Versions of the open-flow protocol have been introduced to add more reliable and flexible capabilities by including multi-flow schedules, improved matching/procedure capabilities, optical ports, group schedules, meter schedules, and synchronous schedules[19]. In addition, there are several open flow controllers available, such as POX, Beacon, Open Daylight, Floodlight, and Ryu that we will use in this proposed research.

2.4 Identification of Traffic Date Tyoes

The identification of traffic data types in today's IP networks has become an important thing with the adoption of Software Defined Networking (SDN) principles using Machine Learning (ML) techniques[9]. Traditional methodologies including identifying traffic types based on port number and payload inspection are not effective due to the continuous changes and encryption nature of the data traffic. And there were some limited attempts to utilize Supervised and Unsupervised ML algorithms to classify flows by their required bandwidth, required QoS, and their application based on various flow-level details as features[8].

In addition, ML classification and clustering of traffics can help identify network hotspots and potential bottlenecks. When using priority for quality of services to make the best use of bandwidth and QoS of flows as classifiers, Traffic Engineering (TE) was used to adjust flow paths and add virtual resources to software-defined networking environments[20]. Also, the identification of applications or web-based protocols is important for forecasting future trends and ensuring the network can meet the demand. Identifying data traffics and Network classification is therefore of great interest to ISPs, governments, and enterprise organizations. But there were many limitations to gaining the required identification of the data traffic types in network environments as follows:

- ✓ It does not identify multiple traffic flows between the same source and destination hosts. For example, if we run a ping command from host number 1 to host number X, then add voice traffic between these two hosts, it will assume the ping and voice are a unique flow [9].
- ✓ If a flow is started and stopped, the algorithm will not delete the old flow but instead update it with the latest flow statistics. This is a problem because the flow classifier considers the average packet size and average number of Bytes. If the flow is stopped for a significant amount of time, these two features will be reduced, and the resulting label of traffic will be inaccurate.

- ✓ The simulation tool (D-ITG) was unable to generate flows for gameplay or another video traffic. With a lot of internet traffic being video nowadays, this would have been helpful for real-world classification situations[21].
- ✓ Unsupervised learning using K-Means clustering performs very poorly. Perhaps this model needs to be further tuned using Hyperparameter tuning. We can also consider other models such as Density-based spatial clustering for noise-intensive applications (DBSCAN) or CNN (Convolutional Neural Networks) for future work on this project.

And to make enhancing in the above we can:

a) Create a Graphical User Interface (GUI) to visualize flow identification on different SDN controllers.
b) Use visual analytics to point out 'hot' areas of high bandwidth and QoS traffic.
c) Connect ML application to actual SAVI testbed controllers and visualize real traffic flows.
d) Use the supervised machine learning to predict categorical target variables and make the SDN controllers programable to automatically determines the traffic data types using artificial intelligence concepts and machine learning so it will give priority to the highly requested data types to use the high bandwidth through the network so it will enhance data transmission and there is no any delay in any response, and this will be done automatically using machine learning and AI concepts in SDN environments, and that what we tested in my research[9], [22].

3. MACHINE LEARNING IN SDN

When we talk about machine learning (ML) and Artificial intelligence (AI) which are mainly correlated, there are many challenges[22]. Machine learning techniques (ML)can be used.
Identify network environments defined by types in data traffic software. The identification of data types can be performed using supervised training and learning with algorithms such as neural networks and decision trees and supporting vector machines (SVMs). In supervised training, there is a need to obtain labeled training datasets, something that can be a challenge in computer networks due to the difficulty of obtaining precisely annotated network flow samples across a wide range of applications and the rate at which new applications can appear. This results in many of these methods is limited to imprecise classifications such as web hosting, secure browsing, TELNET, DNS, and VoIP. An alternative is to use unsupervised ML where the data given to the learner is unclassified. Unsupervised ML is usually used for grouping tasks, in which algorithms Group data into different groups according to similarities in feature values[23].

Our goal is to find unknown relationships in the data and to find patterns of similarity between numerous observations. Many algorithms can be used in unsupervised ML and there is a lot of literature on the application in traffic classification. Methods based on means K-Medoids self-organizing maps (SOM) and DBSCAN have been proposed for many traffic classification scenarios. They can also be used with both labeled and unlabeled data. This approach tries to overcome the difficulties in obtaining disaggregated data. It can work with a dataset where the majority of instances are Uncategorized in case there are a small number of tagged data instances that allow assignment from the groups specified in the overall dataset to the various categories[22].

3.1 AI Endpoint Analytics

AI Network Analytics uses de-identified network event data along with cutting-edge cloud learning platforms and Machine Learning (ML) algorithms to uncover important network vulnerabilities. With the help of AI Network Analytics, you can swiftly solve problems, learn about their underlying causes, spot trends, and insights, and get pertinent comparative viewpoints. With the help of the SDN controller manager's straightforward, perceptive, and potent user interface, AI Network Analytics offers this value. The first step in safeguarding an endpoint is visibility. You may detect and characterize endpoints and Internet of Things (IoT) devices with the aid of AI Endpoint Analytics, an endpoint visibility solution. You can label endpoints using the telemetry data collected from the network from multiple sources and the AI Endpoint Analytics engine[8]. Endpoint type, hardware model, manufacturer, and operating system type are the profiling labels offered by AI Endpoint Analytics. A multifactor classification describes this. With tools like Trust Scores, AI Endpoint Analytics offers your network sophisticated insight and enforcement, enabling you to spot and deal with endpoints and devices that could be dangerous[24]. In AI Endpoint Analytics, you may keep an eye on and deal with the problem of erratic and changing MAC addresses from endpoints and precisely identify endpoints using a special attribute called the DUID instead of MAC addresses. You can collect endpoint telemetry from a variety of sources with the aid of AI Endpoint Analytics[9], [20].

3.2 Key Features of AI Endpoint Analytics In SDN Environment

AI Endpoint Analytics management: You can see a detailed overview of all the endpoints that are connected to your network using the AI Endpoint Analytics dashboard. You can see how many endpoints are known, unknown, profiled, and unprofiled, as well as how many have low Trust Scores and use arbitrary MAC addresses[25]. To improve endpoint profiling and administration, the AI Proposals dash let presents intelligent profiling recommendations:

- ✓ Trust Scores to identify endpoints that might be dangerous: To make it simple for you to keep an eye on and act on potentially dangerous endpoints in your network, AI Endpoint Analytics provides Trust Scores to endpoints. A Trust Score is determined based on the quantity and frequency of behavioral anomalies that are detected and logged[20].
- ✓ Using machine learning, reduce net unknowns: Based on insights from endpoint groups, AI Endpoint Analytics offers profiling recommendations. These recommendations can help your network have fewer unidentified or unprofiled endpoints.
- ✓ Endpoint registration with AI Endpoint Analytics: Using AI Endpoint Analytics, you can onboard and profile endpoints. The endpoints are profiled using the endpoint attribute information that is gathered throughout this registration procedure[26].
- ✓ Control endpoints using the system and personalized profiling rules: To accurately profile and manage the endpoints connected to your network, use the predefined system rules and custom rules of your design.
- ✓ Endpoint registration through external sources: As we see in Figure 7 we can link several external endpoint data sources to AI Endpoint Analytics, including Configuration Management Databases (CMDB). This makes it simple for you to profile, manage, and register endpoints in your network.
- ✓ Purge endpoints following a predetermined amount of inactivity: Create an endpoint purge policy to get rid of inactive endpoints from your network after a specified amount of time. The amount of inactivity required before an endpoint must be deleted can be specified. Additionally, you can alter a purge policy such that it only affects a specific group of endpoints depending on a profiling attribute[9], [22].

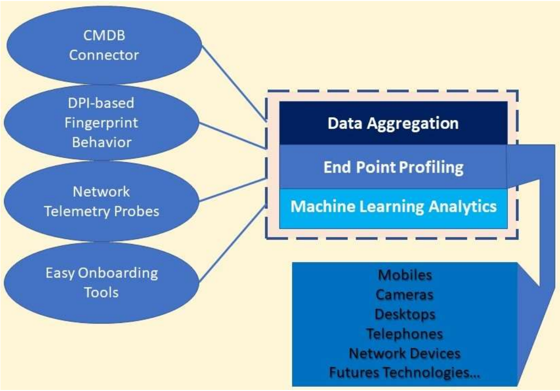


Figure 7: Configuration Management Database (CMDB)

3.3 AI Network Analytics Benefits

There are some benefits of using AI Analytics in programming networks and making all things depend on AI concepts. we mention some of these benefits below:

- ✓ Greater Visibility: The network environment is always shifting, and each network is distinct. With the help of powerful machine learning models, AI Network Analytics continuously gathers the pertinent data from local networks, correlates it with the aggregate de-identified data collection, and then establishes baselines that are pertinent to particular networks and sites. As network settings change and as the variety of devices, users, and applications increase, these baselines learn and adapt.
- ✓ More Insight: To identify the problems that could have the biggest effects on the network, AI Network Analytics employs machine learning to compare the vast amount of data coming from the network against the unique network baselines. This raises the issue's relevance. For IT to spot problems before they arise, AI Network Analytics identifies trends and patterns in network behavior.
- ✓ Directed Action: AI Network Analytics performs logical troubleshooting steps that an engineer may then execute and resolve the issue using automated workflows and machine learning techniques. This enables IT to identify problems and vulnerabilities, investigate the

underlying reasons, and take swift corrective action[20], [22].

4. CONTROLLERS

SDN Controllers are the brains of the SDN environment, communicating information down to the switches and routers with southbound APIs, and up to the applications and business logic with northbound APIs. In addition, controllers are applications in a software-defined networking (SDN) architecture that manage flow control for improved network management and application performance. The SDN controller platform typically runs on a server and uses protocols to tell switches where to send packets[27]. SDN controllers direct traffic according to forwarding policies that a network operator puts in place, thereby minimizing manual configurations for individual network devices. By taking the control plane off of the network hardware and running it instead as software, the centralized controller facilitates automated network management and makes it easier to integrate and administer business applications. In effect, the SDN controller serves as a sort of operating system (OS) for the network. The controller is the core of a software-defined network. It resides between network devices at one end of the network and applications at the other end. Any communication between applications and network devices must go through the controller[28]

The controller communicates with applications -- such as firewalls or load balancers -- via northbound interfaces. The Open Networking Foundation (ONF) created a working group in 2013 focused specifically on northbound APIs and their development. The industry never settled on a standardized set, however, largely because application requirements vary so widely[10].

4.1 Ryu Controller

Ryu Controller is an open, software-defined networking (SDN) Controller designed to increase the agility of the network by making it easy to manage and adapt how traffic is handled. Ryu Controller is supported by NTT and is deployed in NTT cloud data centers as well. The Ryu Controller provides software components, with well-defined application program interfaces (APIs) that make it easy for developers to create new network management and control applications. This component approach helps organizations customize deployments to meet their specific needs; developers can quickly and easily modify the existing component[29].

The controller, via which the many network applications are designed, is the most crucial part of the SDN. In addition to other distinctions relating to the field of usage of these controllers, such as in data centers, cloud computing, and so forth, the controllers differ from one another in the programming language they support, the version of the protocol they operate in, and support for multithreading. The open-source RYU controller, which is written in Python and supports up to 1.5 of the OF protocol, is one of the most well-known of these controllers. In Figure 8 we can see how the Ryu controller fits in SDN Designs. Ryu controller can be managed to control the SDN environment with different components as shown in Figure 9.

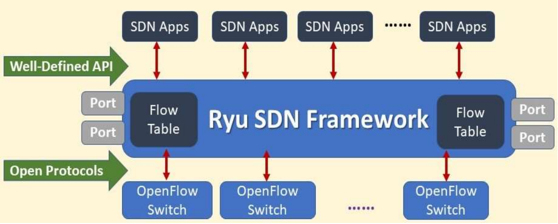


Figure 8: How Ryu Controller Fits in SDN Environment

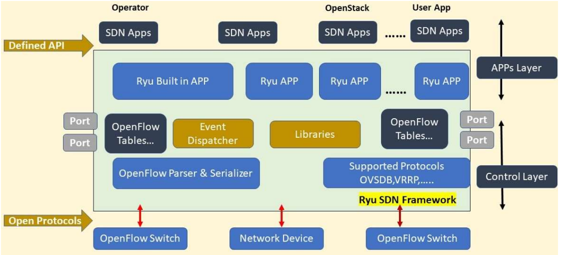


Figure 9: The Architecture of The RYU Controller

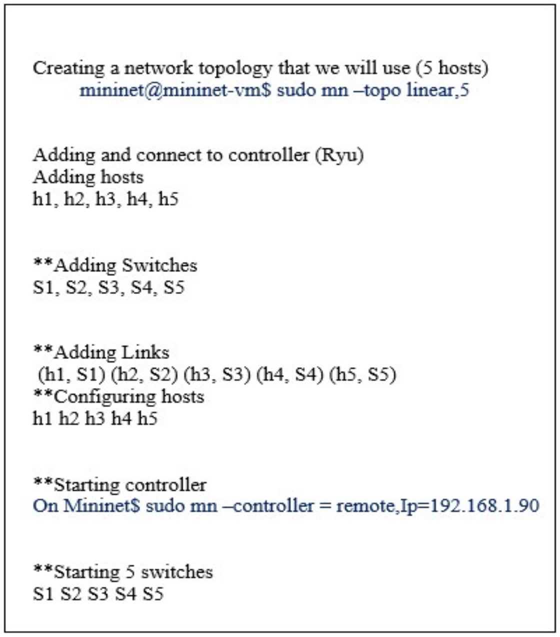
5. PROPOSED METHODS AND RESULTS

5.1 Activities

a) Configuring VMware Workstation and setup these VMs Ubuntu 20, Mininet, and Ryu controller.

b) Configuring Mininet VM and open to do bellow commands in Figure 10.

c) We can describe created topology as below in Figure 11:



Creating a network topology that we will use (5 hosts)
mininet@mininet-vm$ sudo mn –topo linear,5

Adding and connect to controller (Ryu)
Adding hosts
h1, h2, h3, h4, h5

**Adding Switches
S1, S2, S3, S4, S5

**Adding Links
(h1, S1) (h2, S2) (h3, S3) (h4, S4) (h5, S5)
**Configuring hosts
h1 h2 h3 h4 h5

**Starting controller
On Mininet$ sudo mn –controller = remote,Ip=192.168.1.90

**Starting 5 switches
S1 S2 S3 S4 S5

Figure 10: How to design a topology

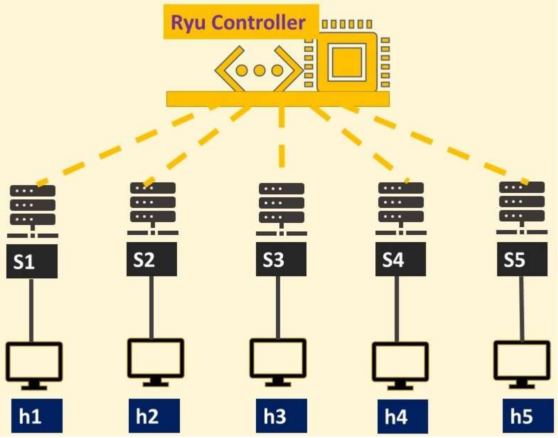


Figure 10: The topology design used in the testing lap

5.2 Results

The generated data below represents the result of 10-minute training for identifying the traffic data types. The generated files (CSV files) of data contain this bellow data defines as result attributes:

✓ Traffic Type
✓ Forward Packets
✓ Forward Bytes
✓ Delta Forward Packets
✓ Delta Forward Bytes
✓ Forward Instantaneous Packets per Second
✓ Forward Average Packets per second
✓ Forward Instantaneous Bytes per Second
✓ Forward Average Bytes per second
✓ Reverse Packets
✓ Reverse Bytes
✓ Delta Reverse Packets
✓ Delta Reverse Bytes
✓ Delta Reverse Instantaneous Packets per Second
✓ Reverse Average Packets per second
✓ Reverse Instantaneous Bytes per Second
✓ Reverse Average Bytes per second

And for the data classification and enhancing the traffic performance process we can do that through these three steps:

a) training data collection - training data collection for the selected traffic type, traffic must flow between two hosts before the script runs.
b) classification using supervised machine learning - classification of the type of traffic flowing between hosts using logistic regression
c) classification using unsupervised machine learning - classification of the type of traffic flowing between hosts using K-Means Clustering.

When we talk about the applications or the potential applications of the work after the discussion of the results, we can confirm that if this technique that we discussed in this research is applied in the organizations and companies, they will save more cost and more teamwork difficulty and all things will be done automatically. We can say that there will be one application or one central point of management for all the SDN networking. And this will be done by connecting a real physical machine learning training server like the Lambda server (that we already used in my research works) to the SDN controller so there will be daily training for all traffic data types and synching all this with the SDN controller at the same time, and all data traffic be classified automatically and the need policy will be applied to each data type also at the same time.

5.3 Traffic Data Types Simulation

Place The distributed traffic generator (D-ITG) application was used to generate traffic flow data used to train machine learning models. D-ITG is described as "a platform capable of producing IPv4 and IPv6 traffic by accurately duplicating the workload of existing internet applications. D-ITG can generate traffic that follows random models of packet size (PS) and overlapping departure time (IDT) that simulate application-level protocol behavior. The Integrated Technologies Group (ITG) can replicate the traffic statistical characteristics of many well-known applications (such as Telnet and VoIP-G.711 and G.723 and G.729 and the detection of voice activity, RTP-DNS compressed and network Games for this guide on the classification of traffic concepts, the following types of traffic were used: Ping, Telnet, DNS, and Voice (G.711). The choice of traffic categories is due to the limitations in the simulation tools and the problems encountered in using D-ITG as discussed in more detail in the limitations and Future Work section[24], [30], [31].

D-ITG describes the traffic used as follows:

Ping*:* Generate traffic with ping characteristics. It only works with TCP transport layer protocol. Different settings will be ignored.
Telnet*:* Generates traffic with Telnet characteristics. It works with TCP transport layer protocol. Different settings will be ignored.
DNS: Generates traffic with DNS characteristics. It works with both UDP and TCP transport layer protocols.
VoIP (voice): Generate traffic with VoIP characteristics. It only works with the UDP transport layer protocol. Different settings will be ignored.
Secure browsing: Generate traffic with secure browsing characteristics. It only works with TCP transport layer protocol. Different settings will be ignored.
Webhosting: Generate traffic with web hosting characteristics. It only works with TCP transport layer protocol. Different settings will be ignored.
Streaming: Generate traffic with streaming video characteristics. It only works with the UDP transport layer protocol. Different settings will be ignored.

5.4 Supervised Learning-Logistic Regression

The machine learning algorithm used was the supervised logistic regression model. It is used to predict categorical target variables [33-49]. Most often, the result is a binary value, but in the case of multiple goals, they choose the one that is most likely to occur. In our program, the logistic regression performed exceptionally well and with an accuracy of more than 99%. It is clear to see from the limits of accuracy, the Figure below, which consists of the first two basic components the reason for the high accuracy. The confusion matrix as described in Figure 12 can help pinpoint where the model fails to accurately identify.

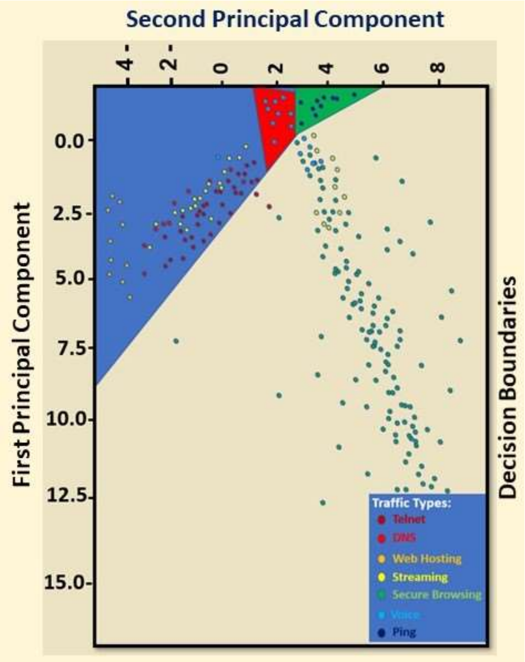
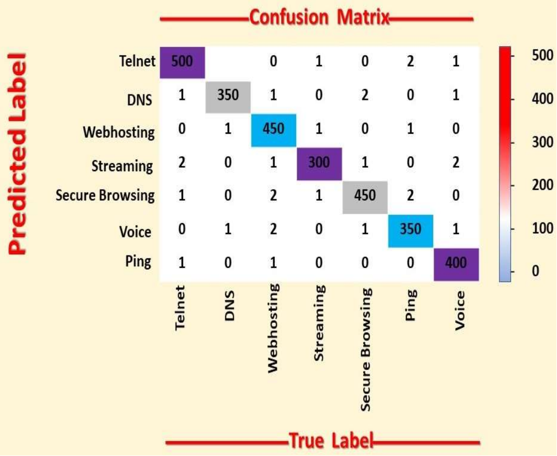


Figure 11: The confusion matrix With pinpoint

If the model tends to predict one traffic category as another more often, this will be evident in the confusion matrix. However, we see, in Figure 13 that for logistic regression, there are almost no failures[30].



	Telnet	DNS	Webhosting	Streaming	Secure Browsing	Ping	Voice
Telnet	500		0	1	0	2	1
DNS	1	350	1	0	2	0	1
Webhosting	0	1	450	1	0	1	0
Streaming	2	0	1	300	1	0	2
Secure Browsing	1	0	2	1	450	2	0
Voice	0	1	2	0	1	350	1
Ping	1	0	1	0	0	0	400

Figure 12: Confusion Matrix for Logistic Regression

6. DIFFERENCE FROM PRIOR WORK

In this research, it was proven that we can automatically identify and classify all traffic packets in SDN networking based on the data type itself by enhancing and improving the SDN controller. WEused the concept and algorithms of machine learning to train and automate this. The new provided thing we tested after doing the correct classification is that we can apply trigger policies for each traffic type such as we can apply a security policy for one and drop from another, also we can apply bandwidth quota for a type according to our needs. All these policies can be applied trigger through the SDN controller. Many Other founded researchers for example the references of papers [6] ,[22], [32], and other many papers discussed this topic but the last thing discussed is to identify the data traffic type and then edit the application on the SDN controller to can track this data type each time, and they didn't try this on a live SDN environment for a long time. But in this research, we tested and tried all that to be run automatically without editing the application each time. And the most different thing in this research is that we used a real live SDN network environment besides using the simulation tools in the first. And also, we connected a real physical Lambda server to a real SDN controller, so the Lambda server runs continuous training daily for all data and synchs all that with the SDN controller that can apply this to the real traffic instantly and with continuous trigger updates without changing and editing in the applications in SDN controller each time. So, using the real physical lambda server with a real SDN network environment for a long time makes my results to be real and factual. and for the training results, we could reach 99.8% in the simulation results and also in my real testing in the real network environment we used.

But from another point of view, there are some limitations and challenges in applying this current work and the technique we discussed. One of these challenges is that this technique is suitable for most large-scale organizations and companies because of the cost of purchasing a real physical machine learning server with high aspects and requirements to be connected to the SDN controller to run the continuous daily training for all data traffic. so small companies will face a problem to apply that to provide the best results. And another thing is that we should use only new versions of network switches that support AI analytics to be connected to the SDN controller and transfer the data traffic from users to the SDN controller to be processed based on the training results that are continuously received from the machine learning server.

7. CONCLUSION & FUTURE RESEARCH

Based on the aforementioned findings, we can conclude that all of these information systems rely heavily on networking techniques and network speed to transmit data. As a result, we want to deal with these systems in simple and adaptable ways during the planning, development, and management phases. The advancement of machine learning (ML) has had a significant impact on SDN systems and has made them more sophisticated, adaptable, and accurate, and provided a new approach to how networks are set up, managed, and operated in simple, flexible, and automated ways. Traffic Engineering (TE) was used to modify flow pathways and add virtual resources to software-defined networking infrastructures when utilizing priority for quality of services to make the most use of bandwidth and QoS of flows as classifiers. Identifying web-based protocols or apps is also crucial for predicting future trends and ensuring that the network can handle the demand. And we talked about the techniques for deriving behavioral patterns from network dynamics analysis to define what constitutes "normal" (baseline) behavior for that particular network. Therefore, the coming period will witness great development in the sides of applying machine learning and artificial intelligence algorithms in the field of SDN to have more intelligent control of networks and information systems and all this will dominate the future world of technology. Finally, we used a real live network environment and tracked the results for a long time with different scenarios, so we could simply provide the best packets classification way using the SDN controller and applied the suitable policy to each packet type from one management place for all network nodes. We could make enhancements in the SDN controller by connecting to a physical real training server like lambda server to automatically identify the type of all packets. Then, apply the needed policies such as bandwidth, quota, and security policies instantly and in a triggering way based on the training results that are continuously transferred from the machine learning server to update the SDN controller automatically all the time.

REFERENCES:

[1] E. G. Amoroso, "Software-Defined Networking and Network Function

Virtualization Security," in *Computer and Information Security Handbook*, 2017.

[2] Y. P. Llerena and P. R. L. Gondim, "SDN-Controller Placement for D2D Communications," *IEEE Access*, vol. 7, 2019.

[3] H. Ghalwash and C. H. Huang, "QOS for SDN-based fat-tree networks," in *Lecture Notes in Networks and Systems*, 2020.

[4] H. Hassen, S. Meherzi, and S. Belghith, "Performance Analysis of POX and OpenDayLight Controllers Based on QoS Parameters," in *Lecture Notes in Networks and Systems*, 2021.

[5] R. Wazirali, R. Ahmad, and S. Alhiyari, "Sdn-openflow topology discovery: An overview of performance issues," *Applied Sciences (Switzerland)*, vol. 11, no. 15. MDPI AG, Aug. 01, 2021.

[6] A. I. Owusu and A. Nayak, "An Intelligent Traffic Classification in SDN-IoT: A Machine Learning Approach," in *2020 IEEE International Black Sea Conference on Communications and Networking, BlackSeaCom 2020*, 2020.

[7] M. Alenezi, K. Almustafa, and K. A. Meerja, "Cloud based SDN and NFV architectures for IoT infrastructure," *Egyptian Informatics Journal*, vol. 20, no. 1, 2019.

[8] M. Peter, D. A. Aldo, F. Cheng, and S. Ong, "MATHEMATICS FOR MACHINE LEARNING." https://mml-book.com.

[9] S. Ayoubi *et al.*, "Machine Learning for Cognitive Network Management," *IEEE Communications Magazine*, vol. 56, no. 1, 2018.

[10] R. Uddin and F. Monir, "Performance Evaluation of Ryu Controller with Weighted Round Robin Load Balancer," in *Communications in Computer and Information Science*, 2021.

[11] P. Vizarreta *et al.*, "Assessing the Maturity of SDN Controllers with Software Reliability Growth Models," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, 2018.

[12] W. Zhuang, Q. Ye, F. Lyu, N. Cheng, and J. Ren, "SDN/NFV-Empowered Future IoV with Enhanced Communication, Computing, and Caching," *Proceedings of the IEEE*, vol. 108, no. 2, 2020.

[13] H. Babbar and S. Rani, "Performance evaluation of QoS metrics in software defined networking using ryu controller," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing Ltd, Jan. 2021.

[14] A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, and R. Smeliansky, "Advanced study of SDN/OpenFlow controllers," in *ACM International Conference Proceeding Series*, 2013.

[15] A. Hussein, I. H. Elhajj, A. Chehab, and A. Kayssi, "SDN security plane: An architecture for resilient security services," in *Proceedings - 2016 IEEE International Conference on Cloud Engineering Workshops, IC2EW 2016*, 2016. doi: 10.1109/IC2EW.2016.15.

[16] Y. Nakagawa *et al.*, "Dynamic virtual network configuration between containers using physical switch functions for NFV infrastructure," in *2015 IEEE Conference on Network Function Virtualization and Software Defined Network, NFV-SDN 2015*, 2016.

[17] J. V. G. De Oliveira, P. C. P. Bellotti, R. M. De Oliveira, A. B. Vieira, and L. J. Chaves, "Virtualizing Packet-Processing Network Functions over Heterogeneous OpenFlow Switches," *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, 2022.

[18] S. Wang, "Comparisons of SDN OpenFlow Controllers over EstiNet : Ryu vs . NOX," *The International Symposium on Advances in Software Defined Networks, April 19-24, 2015, Barcelona, Spain*, no. Fedora 14, 2015.

[19] B. K. Tripathy, K. S. Sahoo, A. K. Luhach, N. Z. Jhanjhi, and S. K. Jena, "A virtual execution platform for OpenFlow controller using NFV," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, 2022.

[20] A. Adamou Djergou, Y. Maleh, and S. Mounir, "Machine Learning Techniques for Intrusion Detection in SDN: A Survey," in *Lecture Notes in Networks and Systems*, 2022.

[21] H. A. Alamri, V. Thayananthan, and J. Yazdani, "Machine Learning for Securing SDN based 5G Network," *Int J Comput Appl*, vol. 174, no. 14, 2021.

[22] P. Amaral, J. Dinis, P. Pinto, L. Bernardo, J. Tavares, and H. S. Mamede, "Machine learning in software defined networks: Data collection and traffic classification," in

Proceedings - International Conference on Network Protocols, ICNP, IEEE Computer Society, Dec. 2016, pp. 91–95.

[23] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, “Survey on SDN based network intrusion detection system using machine learning approaches,” *Peer Peer Netw Appl*, vol. 12, no. 2, 2019.

[24] K. M. Abdullah, B. N. Adday, R. A. Jaleel, I. M. Burhan, M. A. Salih, and M. M. A. Zahra, “Integrating of Promising Computer Network Technology with Intelligent Supervised Machine Learning for Better Performance,” *Webology*, vol. 19, no. 1, 2022.

[25] I. Ahmad *et al.*, “Machine Learning Meets Communication Networks: Current Trends and Future Challenges,” *IEEE Access*, vol. 8, 2020.

[26] K. Shinan, K. Alsubhi, A. Alzahrani, and M. U. Ashraf, “Machine learning-based botnet detection in software-defined network: A systematic review,” *Symmetry (Basel)*, vol. 13, no. 5, 2021.

[27] L.-A. J. O. C. (lajc) Vol, V. ; D. Haro-Mendoza, L. Tello-Oquendo, and L. A. Marrone, “A comparative evaluation of the performance of open-source SDN controllers,” 2020.

[28] S. Ahmad and A. H. Mir, “Scalability, Consistency, Reliability and Security in SDN Controllers: A Survey of Diverse SDN Controllers,” *Journal of Network and Systems Management*, vol. 29, no. 1, 2021.

[29] R. K. Chouhan, M. Atulkar, and N. K. Nagwani, “Performance Comparison of Ryu and Floodlight Controllers in Different SDN Topologies,” in *1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing and Communication Engineering, ICATIECE 2019*, 2019.

[30] S. Liang, W. Jiang, F. Zhao, and F. Zhao, “Load Balancing Algorithm of Controller Based on SDN Architecture Under Machine Learning,” *Journal of Systems Science and Information*, vol. 8, no. 6, 2020.

[31] M. Janat and N. Sudha, “a Survey on Security Threats and Solutions for Sdn Using Machine Learning Approach,” *International Journal of Emerging Technology and Innovative Engineering*, vol. 5, no. 8, 2019.

[32] M. H. H. Khairi *et al.*, “Detection and Classification of Conflict Flows in SDN Using Machine Learning Algorithms,” *IEEE Access*, vol. 9, 2021.

[33] Hazem M. El-Bakry, "An Efficient Algorithm for Pattern Detection using Combined Classifiers and Data Fusion," *Information Fusion Journal*, vol. 11, issue 2, April 2010, pp. 133-148.

[34] Hazem M. El-Bakry, and Nikos Mastorakis "New Fast Normalized Neural Networks for Pattern Detection," Image and Vision Computing Journal, vol. 25, issue 11, 2007, pp. 1767-1784.

[35] Hazem M. El-Bakry, and Qiangfu Zhao, "Speeding-up Normalized Neural Networks For Face/Object Detection," Machine Graphics & Vision Journal (MG&V), vol. 14, No.1, 2005, pp. 29-59.

[36] Menna_Elkhateeb, Abdulaziz_Shehab, and Hazem_El-bakry, “Mobile Learning System for Egyptian Higher Education Using Agile-Based Approach,” Education Research International, Volume 2019, Article ID 7531980, 13 pages.

[37] Hazem M. El-Bakry, and Nikos Mastorakis, "A New Fast Forecasting Technique using High Speed Neural Networks," *WSEAS Transactions on Signal Processing*, vol. 4, Issue 10, Oct. 2008, pp. 573-595.

[38] Hazem M. El-Bakry, and Qiangfu Zhao, "Fast Normalized Neural Processors For Pattern Detection Based on Cross Correlation Implemented in the Frequency Domain," Journal of Research and Practice in Information Technology, Vol. 38, No.2, May 2006, pp. 151-170.

[39] Hazem M. El-Bakry, "Fast Virus Detection by using High Speed Time Delay Neural Networks," Journal of Computer Virology, vol.6, no.2, 2010, pp.115-122.

[40] Hazem M. El-Bakry, and Nikos Mastorakis, "Realization of E-University for Distance Learning," *WSEAS Transactions on Computers*, vol. 8, issue 1, Jan. 2009, pp. 48-62.

[41] Hazem El-Bakry: “Comments on Using MLP and FFT for Fast Object/Face Detection,” Proc. of IEEE IJCNN’03, Portland, Oregon, pp. 1284-1288, July, 20-24, 2003.

[42] Hazem El-Bakry, “Face Detection Using Neural Networks and Image Decomposition,” Proc. of INNS-IEEE

International Joint Conference on Neural Networks, 12-17 May, 2002, Honolulu, Hawaii, USA.

[43] Hazem El-Bakry, "Fast Face Detection Using Neural Networks and Image Decomposition," Proc. of the 6th International Computer Science Conference, AMT 2001, Hong Kong, China, December 18-20, 2001, pp.205-215.

[44] Hazem M. El-Bakry and Mohamed Hamada, "A New Implementation for High Speed Neural Networks in Frequency Space," Lecture Notes in Artificial Intelligence, Springer, KES 2008, Part I, LNAI 5177, pp. 33-40.

[45] Hazem M. El-Bakry, and Qiangfu Zhao, "Fast Time Delay Neural Networks," International Journal of Neural Systems, vol. 15, no.6, December 2005, pp.445-455.

[46] Hazem M. El-Bakry, "A Novel High Speed Neural Model for Fast Pattern Recognition," Soft Computing Journal, vol. 14, no. 6, 2010, pp. 647-666.

[47] Hazem M. El-Bakry, "New Fast Time Delay Neural Networks Using Cross Correlation Performed in the Frequency Domain," Neurocomputing Journal, vol. 69, October 2006, pp. 2360-2363.

[48] Hazem M. El-Bakry “Fast Iris Detection for Personal Verification Using Modular Neural Networks,” Proc. of the 7th Fuzzy Days International Conference, Dortmund, Germany, October 1-3, 2001, pp. 269-283.

[49] Hazem M. El-Bakry, M. A. Abo-elsoud, and M. S. Kamel, “Fast Modular Neural Networks for Human Face Detection,” Proc. of IEEE-INNS-ENNS International Joint Conference on Neural Networks, Como, Italy, Vol. III, pp. 320-324, 24-27 July, 2000.