

# ENHANCED ENERGY EFFICIENCY AND POWER ANALYSIS IN SOFTWARE-DEFINED NETWORKS THROUGH MALICIOUS SWITCH DETECTION

Dr. THANGARAJ E<sup>1</sup>, Dr. MOHAMED MALICK<sup>2</sup>, Dr. AROCKIA JAYADHAS S<sup>3</sup>  
Dr. BARKATHULLA<sup>4</sup>, Dr. HARIHARASUDHAN S<sup>5</sup>, R MATHU SUDHANAN<sup>6</sup>

<sup>1</sup> Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Chennai, India.

<sup>2</sup> Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Chennai, India.

<sup>3</sup> Saveetha School of Engineering, SIMATS, Chennai, India.

<sup>4</sup> Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Chennai, India.

<sup>5</sup> Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Chennai, India.

<sup>6</sup> CARE College of Engineering, Tiruchirappalli, India

E-mail: <sup>1</sup>drthangaraje@veltech.edu.in, <sup>2</sup>drmohamedmalickms@veltech.edu.in,

<sup>3</sup>arockiajayadhass.sse@saveetha.com, <sup>4</sup>barkathullakng@gmail.com, <sup>5</sup>drhariharasudhans@veltech.edu.in,

<sup>6</sup>srmdhusudhanan@gmail.com

## ABSTRACT

Software Defined Networking (SDN) has progressed by a network model to assist with different requirement of a real time traffic flow in a huge scale switch. The basic function of SDN plays a vital role in partition of data plane from the controller plane with accepting the needed changes as per the necessities. SDN provides the broad programmability of switches and its architecture concepts for the services and applications. In our proposed system, deep learning atmosphere is generated using TensorFlow and Keras atmosphere. It involves in Kaggle dataset for the trusted and the malicious flow detection. Through analysis part, power analysis is added in the controller built. SDN overall performance depends on energy consumption, load balancing and traffic management. In this proposed approach energy efficiency and badge power of the Software Defined Network is presented. We focused on the energy efficiency of the network by improving the already proposed route selection algorithm. We proposed a method for the analysis to determine the power management functionalities in various domains. The stimulation results show the power analysis feature classification helps to identify the trusted flow of switches. We propose an empirical algorithm and mathematical model of energy consumption determines controller plane to maximize the network performance and scalability. The overall performance study determines that the proposed system of drop rate, constant throughput, load balancing and energy efficiency of switches in SDN is better than the already existing algorithms. The comparative study determines the proposed algorithm is approximately saves 30% of energy.

**Keywords:** *Software Defined Networking, Kaggle, Convolutional Neural Networks, Keras, Power Analysis, load balancing, energy consumption, Tenser-flow.*

## 1. INTRODUCTION

Software Defined Networking (SDN) is a network manager who controls the flow of networks, traffic paths, services and also handling packets etc., Management services include controlling, operating, performance, managing, provisioning, accounting and security in a multi-broader environment. SDN allows a control plane for a network controller to manage the whole network by organizing routing environment for the essential switches. Data plane also called as switches are individually helps data to be forwarded to their respective tables. It is one of the major goals of Software Defined Networking

(SDN). Network management and routing computation are managed by SDN controller. Controller manages the table entries with the secured interaction protocol between controller and the switches. Lately, deep learning environment has arose and attained real accomplishments. Till now, deep learning has been used for image, face and voice recognition in computer science. But deep learning is proficient of inevitably discovering correlation in the node, switches and networks. So, it is a capable way for the next peers of intrusion recognition. Convolutional Neural Networks (CNN) is the length of the packet in Software Defined Networking (SDN). It has been collected together as

an order of deep learning environment to identify the malicious and untrusted packets. In this case, traffic control from source to destination nodes is aided by SDN switches [6]. The switch will receive routing instructions from the SDN controller for each new detected flow, allowing it to reroute to the appropriate network. Communication between the controller and the switch will be managed using OpenFlow. The switch also stores specific information related to the traffic flow and can provide this information to the controller upon request. Attacks are initiated by the user from the attacker node, causing the switch to be directly exposed. In this study, certain switches are vulnerable to effective attacks. Deep learning is useful to ably perceive attacks and vulnerable [7] in a high rate of detection. In such cases, those nodes will be considered for undermine the network. SDN controller look after the vulnerabilities in the switches.

### 1.1. Objectives of the Research Work

Our proposed system mainly focuses on the below objectives,

To identify malicious or untrusted switches with the help of deep learning model.

To expand the Efficiency of the Software Defined Networking (SDN) for detecting malicious and trusted switches.

Reduction of energy efficiency through energy consumption [8] mathematical model and a proposed energy saving algorithm is carried out by controller's load splitting.

The performance of SDN is enhanced by throughput, MSDR, PDR and drop rate.

### 1.2 Organization of Paper

The paper is systematized as follows. Section 2 describes the related existing systems literature survey. Section 3 describes about the methods and materials used in the analysis part, energy efficiency and badge power. The 4th section defines the complete implementation details and 5th Section defines about our experimental results and stimulation. Section 6 will lastly complete the paper with future improvements.

### 1.3 Difference from Prior Work

Traditional Software Defined Networking (SDN) architectures have primarily focused on network flexibility, programmability, and basic traffic management without emphasizing energy consumption or malicious flow detection. While previous works have proposed route optimization

algorithms and controller-based network management, they often lack an integrated approach that combines power analysis, energy efficiency improvements, and security enhancements within the SDN controller.

In contrast, the proposed system addresses these gaps by introducing a deep learning-based detection framework utilizing TensorFlow and Keras environments to accurately differentiate trusted and malicious flows using Kaggle datasets. Additionally, a dedicated power analysis feature is integrated directly into the SDN controller, allowing real-time monitoring and optimization of energy consumption.

## 2 LITERATURE SURVEY

Oliveira et al. [1] discusses about the energy consumption efficiency by reducing its frequency through SDN controller. This has been done through multicore mainframe. In this paper, Communication Network Energy Efficiency (CNEE) metric is used for energy efficiency. This metric helps to improve the energy consumption. The reduction of consumption is because of parallel implementation and lower operating frequency. This specifies that a concentration on leveraging correspondence on network requests, specifically SDN controller, can expand not only energy-efficiency but even the performance. In control plane, drop of energy consumption is a new strategy for energy related SDN. Data plane helps in overall energy consumption in the network. Power analysis of network on controller carried out for energy consumption. We focused on energy efficiency to improve the consumption which was focused on this paper. Andrzej et al. [2] focuses on the current vulnerability of the switches that are broken by attackers. Vulnerabilities can be mass-produced on determination or might result from program writer mistakes and quality less coding performance. According to our proposed system the vulnerability is broken, the switch will not be progress further. This switch is said to be sworn switch where this switch will not lose its original functionality but it can be wrought to be malicious or untrusted. This switch helps to drop packets moving through it, or redirects to the wrong node. If the widely held of switches in a network are conceded, they can plot to shut down the whole network and can make very difficult for the controller to diagnose the correct information. The entire process of reducing the risk of failure in the network, switches will be sent different hosts in the SDN controller. By this way the possibility of having a huge number of switches conceded is reduced. OpenFlow protocol is the main focus of our paper which helps to reduce the

energy consumption. Using OpenFlow, a method for detecting and monitoring malicious switches are introduced. Here, the controller gathers the statistics part to analyse the paths from source to destination of the switches. Packet dropper is one of the abnormal activities of the switches are determined. This packet drop is nothing but the switch will drop the packets intensively. To detect this packet drop, the controller uses the new algorithm for identifying malicious or untrusted flow of network. This controller can collect the report and observe the paths forwards from source to destination. Tuan et al. [3] deals with the energy efficiency and their capabilities that are exploited by combining SDN. Traffic management for trusted and untrusted flow of switches is determined in this paper. As the energy efficiency is determined by some of the optimization technique for every subcategory. Hardware related solutions for the enhancements of switches for energy efficiency are determined. Power analysis can make the energy efficient. Feature classification makes the system of flow from source to destination in the trusted flow. Mohsin et al. [4] proposed an Application Programming Interfaces (API) in management plane of SDN system. It acts as a communication between switches and SDN controller. Here, SDN system can be trusted or untrusted switch. These malicious switch in SDN allocate dummy or unrelated packets to controller for reduction of energy efficiency of networking by consumption of energy rate increased. This problem of energy degradation is addressed in this paper by identifying malicious switch to improvise the efficiency of network

## 2.2 PROBLEM STATEMENT

SDN lies on separating control plane from the data plane providing flexible to re-configure controller as per the requirements. Some of the challenges are introduced through this process are power consumption, scalability and security. The Efficiency of SDN depending on load balancing, energy consumption, traffic management and power saving. Load balancing in SDN will decouple data plane forwarding to controller plane to implement centralized controller. SDN load balancer will save the running time of the network where the traffic flow will not be overloaded. Power analysis is measured for battery percentage and remaining time which are used for detected the trusted or untrusted flow of network. A new proposed algorithm is introduced for energy consumption issue. This algorithm lies on SDN controller, switches and link path for the calculation of energy efficiency. The

overall performance of energy consumption is improved by detecting the trusted or untrusted flow of network switch using routing algorithm and calculation of energy consumption mathematical model [9][10].

## 2.3 Selections of the core concern of the research

### Economic Impact:

Data centers and telecommunications networks are examples of modern network infrastructures that use a lot of energy. Increased operating costs are a direct result of high energy use. The research attempts to lower these expenses by investigating energy-efficient techniques inside SDN designs, hence increasing the economic sustainability of networks.

### Environmental Sustainability:

It is now crucial to lower the carbon footprint of ICT infrastructures as worldwide awareness of climate change grows. Energy-aware routing protocols and dynamic resource allocation are made possible by SDN's centralized control, which can result in notable drops in overall energy use. This is in line with global sustainability goals and promotes corporate social responsibility.

## 3. MATERIALS AND METHODS

### 3.1 DEEP LEARNING MODEL

Deep learning atmosphere is generated using keras and TensorFlow atmosphere. In this environment Kaggle dataset is used to detect and identify malicious and trusted switch. There are many stages of features in the deep learning model. In our proposed approach, we created a basic deep neural network through an output layer, hidden layers and the input layer. The input measurement is three and the output measurement is one. The hidden layers hold six and three neurons correspondingly. Every layer is discovered from the features corresponding to next level. We divided our work into analysis part and SDN implementation [11][12]. Through CNN and exigence importance method dataset will be analysed as feature classification and text importance. The flowchart of evaluation portion is shown in figure 1. Through this analysis feature of battery power is determined. The duration of flow indicates the flow of network as trusted or malicious code. The minimum and maximum of the features are analysed for the further implementation. This analysis part focuses on energy efficiency of the battery power of the switch. Normally, the switch will be the trusted which is predefined initially. If the flow of duration of the switch is high value means

malicious or untrusted flow. If the duration of flow is small value means trusted switch. The dataset produces various features such as packet count, byte count and duration of the flow. Along with these, battery power is added in the controller built.

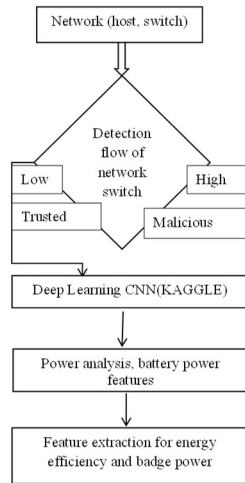


Figure 1 Flowchart of Analysis Section

### 3.2 POWER ANALYSIS

Power analysis feature is carried for the percentage of battery and the remaining time. If the battery is 100%, the switch will be reliable for 2 hours then it goes down. If the battery is plugged it will continue. Energy efficiency [12][13] is the usage of power effectiveness. Entropy value of power analysis are computed by plugged in or unplugged, percentage of battery and the remaining time left in the battery. If the battery is 100% means, for 2 hours that switch will be reliable and the battery power goes down. If it is plugged means the switch will continue. The complete step by step process of power analysis is computed through Alg 1. Switches are mentioned as S, Controller C, link state matrix as ls.

*Algorithm 1: To compute power analysis*

- (1) Start timer t and create a network S0-eth -Switch name and S1-eth1 switch name
- (eg.,) 2 host 2 switch 1 controller network are created
- (2 )Controller C sends switches to neighbors (neighb(c))
- (3) Calculate links state matrix ls=calculate\_ls()
- (4) Update ls.
- (5) Power analysis set on controller C
- (6) If battery power is high, the switch is efficient.

- (7) else if the battery power is low,
- (8 ) then the switch is not energy efficient
- (9) End timer t

Figure 2 shows the trusted flow of battery through power analysis. The trusted flow of switches is efficient [14] if the battery is 100% and it is reliable to connect with switch. The same way if the battery is low, then the node is not efficient and will be removed from switch immediately. Table 1 depicts the minimum and maximum value of entropy with some of the features battery power, packet count, byte count and duration. Table 2 denotes the trusted and untrusted flow of network switch through the feature classification.

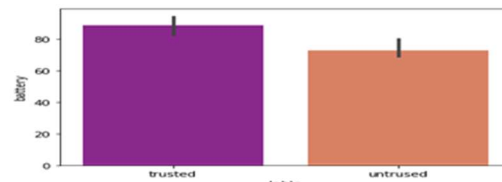


Figure 2 Battery VS Trust

Table 1 Features Min and Max

Feature	Maximum	Minimum
Battery Power	100%	75%
Packet Count	390000	45000
Byte Count	147128009	213456
Duration	800	250

Table 2 Number of trusted and untrusted features

	pckcount	bytecount	dur	battery	lable
0	50	3200	3200	100	trusted
1	5	320	320	100	trusted
2	30	1920	1920	88	trusted
3	300	19200	19200	87	trusted
4	300	19200	19200	84	trusted
5	3	192	192	82	trusted
6	23	1472	1472	69	untrusted
7	23	1472	1472	69	untrusted
8	300	19200	19200	79	trusted
9	500	32000	32000	80	untrusted

### 3.3 ENERGY EFFICIENCY MODEL

Our energy efficiency model consists of switch, controller and link path. We divide the network into nodes n, controller C, OpenFlow [14] switches and I set of links in active and sleep modes. Energy

consumption for switch is computed in the below equation 1.

$$P(S) = \alpha * P(SW) + (1 - \alpha) * P(Nsw) \text{-----} \\ \text{-----}(1)$$

P(S) is switch, P(SW) is self-active switch and P(Nsw) is energy consumption when active by neighbor. If  $\alpha = 1$ , when switch is active and plugged else 0 means switch is at sleep mode and unplugged state. Energy efficiency for self-active is computed as the below equation 2,

$$P(SW) = P(S) + \sum_{i=0}^n S(A) + f(T) * P_s \text{-----} \\ \text{---}(2)$$

Where, P(SW) is self-wake and S(A) is energy of sleep to active mode. F(T) is computed for packet processing from 0 to 1 for sleep or wake up state. Energy consumption by controllers' calculation is computed in the below equation 3.

$$P(C) = \beta * P(Ac) + (1 - \beta) * P(NC) \text{-----} \\ \text{---}(3)$$

Where P(C) is self-wake up energy consumption [15] controller, P(NC) activated by neighbor energy consumption.  $\beta = 1$ , then it is considered as active state or plugged state else  $\beta = 0$ , then it is in sleep mode or unplugged state. Active link state is calculated using the below equation 4.

$$P(link) = \min \sum_{i=0}^n p(i) \text{-----} 4$$

Where P(link) is energy consumption in active state,  $P(i) = P(X_i/C_i)$ ,  $X_i$  denotes traffic and  $C_i$  denotes capacity. However, more load is there in links means it consumes more energy. Algorithm 2 deals with the calculation of SDN for energy efficiency. After creation of networks as per the SDN architecture

*Algorithm 2: Energy Efficiency Proposed method*

- (1) creating network node
- (2) Power analysis determined in analysis part
- (3) Energy consumption calculated on switch, controller and link state
- (4) While!empty(stack) do
- (5) protocol<-Pop(stack)
- (6) for i=1 to N do
- (7) Pow<-Compute\_(protocol)
- (8) if i=0 then the power is unplugged or sleep mode or untrusted
- (9) else i=1 then the power is plugged or active mode or trusted

(10) process stack

(11) return pow

(12) end

#### 4. IMPLEMENTAION AND ARCHITECTURE OF SUGGESTED WORK

The proposed architecture of software defined networking is shown in the figure 3. From analysis part, we get the features of power analysis. When the flow starts, duration high means untrusted flow of network switch. Firstly, we fix in what all the features trusted and untrusted level switch label has to be made. Minimum and maximum level of that features are determined in analysis part. Mininet are used to create a SDN environment. Firstly, controller, number of switches and nodes are created. Here, 64 nodes are created as an example where 9 are switches, 1 controller, created data plane, management plane and controller plane.

##### 4.1 SWITCHES IN SDN

In this work, we normally sent the packets as trusted switch through the modified pox controller program. SDN switches onward user traffic flow from source to destination nodes in an appropriate manner. Through the features determined from analysis part, we set them for controller in built. Whenever the flow is found in the switch, routing directions from SDN controller will be received by the switch and then forwards the packets of the flow to its suitable network. Only these switches with specified condition alone the switches are allowed or else switch will be disconnected. OpenFlow protocol is the management interface which helps in communication with the controller. Here, the switch has certain information in regards of handling traffic flow and will provide data to the controller based on the request.

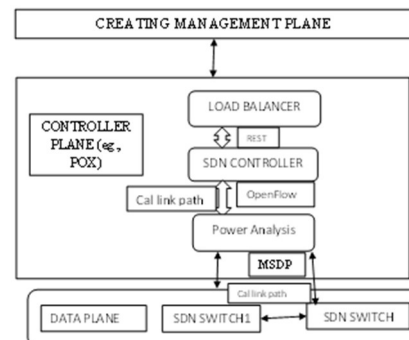




Figure 3 Proposed Architecture of SDN

## 4.2 SDN CONTROLLER

SDN Controller is the interconnected controller components which manages the switch operation and decide the routing instruction based on the power analysis and load balancer. Each and every component has whole statistics about the topology of network, which permits to respond to failures efficiently and helps in computation of finest paths for traffic. Our proposed system presents a functional block of Malicious Detection and prevention (MSDP) which is in charge for a transparent and continuous analysis of the interaction between switches and the controller through OpenFlow Protocol. It is used for the determination of malicious switches to be overserved through this interaction. The normal sent through communication reaches the destination and data will be stored as text file.

### 4.2.1 Load Balancing At Controller Plane

Load balancing on controller plane creates a network topology for load balancing the traffic flow. We compute link path capacity for minimum and maximum path allocated by the flow of network switch. The list of all possible paths [16] is selected from source nodes to destination nodes. Then, we detect many paths of minimum length of network switch. Load balancing index the minimum and maximum link path carried out through network flow.

### 4.3 Creating Network Management Plane

Through network management plane we are adding different switches, controllers and links. This management plane monitors all the network performance and setup devices. Network admins are restricted to this type of interfaces. But, some of the groups of machines are granted to access them. In our paper, management devices [17][18] are referred for all such machines. Algorithm 2 is created to detect malicious network switch. We have a file attacker for untrusted flow of network. When it is sent from host, switch will be interlinked and connected. If normal packet is sent, the flow will be trusted. But from switch if we give attacker and connected with another switch means then the other switch is considered as untrusted switch.

## 4.4 IMPLEMENTATION RESULTS

The implementation results of energy consumption, round trip time and loss are computed in the below table 3.

Table 3 Implementation Result For The Traffic Flow Of Network

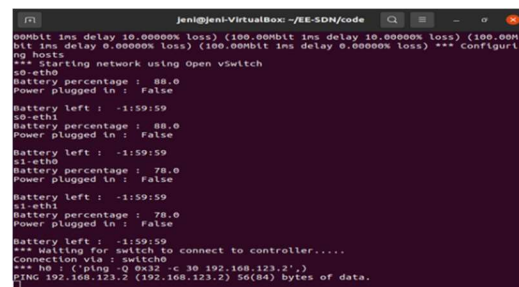
Nu mb er of pac ket s sen t	Nu mbe r of pac kets Rec eive d	Nu mb er of pac ket s dro ppe d	Los s Pcer tang e	Nu mb er of Swi tche s	RTT (rou nd trip time) (ms)	Energ y consu mmati on	Tim e(m s)
100	90	10	10	2	0.145	20.68967	1022541
500	487	13	2.6	3	2.076	10.27651	1307431
1000	704	296	29.6	5	7.081	27.45089	100600
1500	1202	298	19.886	7	7.102	27.45089	1505892

## 5. PERFORMANCE EVALUATION

### 5.1 DROP RATE

Drop rate in a host is calculated for the network flow switches with the terms of time, round trip time. Through open vSwitch [19][20] drop rate are measured in terms of mbits. The drop rate is determined by the entire number of packets that not able to reach the destination nodes.

Figure 4 Drop Rate



```

jenit@jenit-VirtualBox: ~/EE-SDN/code
0.00bit ins delay 10.00000% loss) (100.00bit ins delay 10.00000% loss) (100.00
bit ins delay 0.00000% loss) (100.00bit ins delay 0.00000% loss) *** Configu
ng hosts
*** Starting network using Open vSwitch
s1-eth0
Battery percentage : 88.0
Power plugged in : false
Battery left : -1:59:59
s2-eth1
Battery percentage : 88.0
Power plugged in : false
Battery left : -1:59:59
s1-eth0
Battery percentage : 78.0
Power plugged in : false
Battery left : -1:59:59
s1-eth1
Battery percentage : 78.0
Power plugged in : false
Battery left : -1:59:59
*** Waiting for switch to connect to controller....
Connection via : switch0
*** h0 : (ping -q 0x32 -c 30 192.168.123.2')
PING 192.168.123.2 (192.168.123.2) 56(84) bytes of data.

```

### 5.2 THROUGHPUT

Throughput is computed in terms of Mbps per unit time with the below equation 5.

Throughput =  
Total time of data bits received in flows-----  
(5)

Table 4 Throughput Comparison

System to System	System to Switch	Switch to Controller	Number of Host	Number of Switches
18.8	27.6	27.8	64	10

Table 4 describes about the comparative study of our proposed system with different inputs for hosts, switch, controller etc., Figure 5 shows the constant throughput for different number of switches.

### 5.3 MSDR

Malicious Switch Detection Rate (MSDR) is the calculation of the number of malicious switches detected with the total number of malicious switches are mentioned in the below equation (6).

$$MSDR = \frac{\text{Number of correctly detected malicious switches}}{\text{Total number of malicious switches}} * 100\% \quad \text{-----(6)}$$

Table 5 displays the Malicious Switch Detection Rate (MSDR) for the trusted or untrusted flow of networks by correctly detecting the node computed with the above eqn 6.

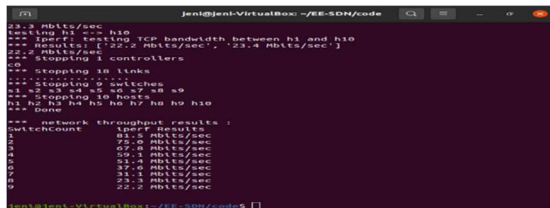


Figure 5 constant throughput with no of switches

Table 5 Computation of MSDR

No of packets sent	Received Packets	No of packet dropped	Loss (%)	No of switches	PDR
100	90	10	10	10	9
500	487	13	2.6	15	32
1000	704	296	29.6	20	35
1500	1202	298	19.886	25	60
2000	1966	34	17	30	65

### 5.5 ENERGY CONSUMPTION

Energy consumption of proposed power analysis of sample network is implemented in POX controller and selection of routing algorithm. We set Mininet VM in virtual Box. We have extensive stimulation on sample network with varying traffic flow and size. Energy consumption is measured in terms of mj from different packets sent per seconds. The variation of packets is 50/sec, 100/sec, 150/sec with varying network size. The sample network is of different sizes 10-60 hosts, 2-14 OpenFlow switches, 3-5 controllers and 30-60 network links with varying link capacity from 600 Mbps-3 Gbps. The energy consumption sample network is shown in the below figure 6.

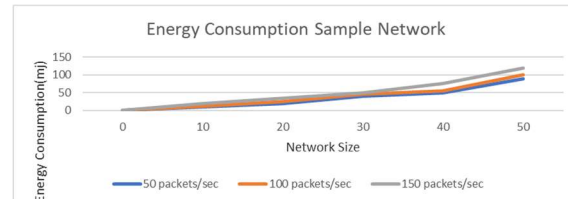


Figure 6 Energy Consumption Sample network

To check our proposed algorithm efficiency, we compared it with already proposed algorithm ERA (Efficient Routing Algorithm Selection). We applied sample network in the both the algorithm are shown in the below figure 7. Measurement of energy consumption is computed in terms of time (in

No of switches	No of Trusted switch	No of untrusted	Correctly Detected	Nodes connected	MSD R
10	7	3	2	64	67%
20	14	6	5	64	83%
30	20	10	9	50	90%

seconds) for proposed algorithm and ERA algorithm are measured in terms of energy consumption (mj). Through the experimental results, we have observed 11.2% of more energy is saved when compared with ERA's algorithm.

### 5.4 PDR

Packet Delivery Ratio (PDR) is the calculation of the ratio of number of received packets with the total number of sent packets denoted by eqn 7. PDR will find the errors or issues which lead to the good throughput. The comparison of Packet Delivery Ratio between the packets is shown in the below table 6.

$$PDR = \frac{\text{No of Packets Received}}{\text{No of switches}} \quad \text{-----(7)}$$

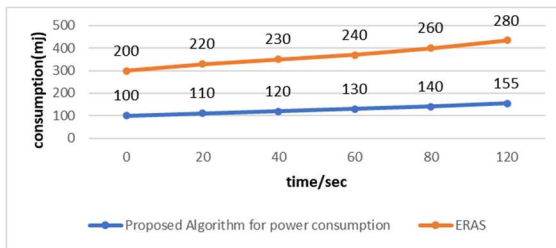


Figure 7 Comparison of proposed power consumption algorithm with the already existing ERA algorithm.

## 5.6 LIMITATIONS OF WORK

Determining optimal locations for SDN controllers to minimize latency and maximize efficiency is an ongoing research challenge.

Table. 7. SDN energy-efficiency related works

	Energy Consumption Strategy	OpenFlow Complaint	Multi-controller	Controller Consume Energy
[1]	Parallel processing	Yes	Yes	High
[2]	Switch Migration	Yes	No	High
[4]	DEA Algorithm	Yes	No	Medium
Proposed Method	Energy Efficiency and badge power of the SDN	Yes	No	Low

## 6 NOVEL CONTRIBUTIONS:

### Use of Deep Learning for Malicious Flow Detection:

- A deep learning environment is created using TensorFlow and Keras.
- A Kaggle dataset is used to detect trusted and malicious flows in the SDN.

### Integration of Power Analysis into SDN Controller:

- Power analysis is newly added inside the controller, which is not typically done in basic SDN setups.

### Energy Efficiency Focus:

- Improved an existing route selection algorithm specifically to enhance energy efficiency.
- Badge power (minimum or standby power) and energy consumption are analyzed carefully.

### Proposed a New Algorithm and Mathematical Model:

- A new empirical algorithm and mathematical model are proposed to manage energy consumption efficiently at the controller plane.

### Performance Improvement:

- 30% energy savings compared to existing algorithms.
- Better results for drop rate, constant throughput, load balancing, and energy efficiency.

## 7 PROBLEMS AND RESEARCH ISSUES

Although the proposed system achieves significant improvements in energy efficiency, malicious flow detection, and overall SDN performance, several challenges and open research issues remain:

### Scalability to Extremely Large Networks:

While the current system improves performance in a large-scale switch environment, further research is needed to validate scalability across very large SDN deployments involving millions of flows and hundreds of switches.

### Real-Time Power Monitoring Overhead:

Integrating power analysis into the SDN controller introduces additional processing and monitoring overhead, which may affect controller response times under heavy traffic loads. Optimizing this overhead without compromising accuracy remains an open challenge.

### Dynamic Adaptation to Evolving Attack Patterns:

The deep learning model is trained on a Kaggle dataset, but real-world attack patterns may evolve. Thus, there is a need for continuous learning mechanisms that



allow the system to adapt dynamically to new and unseen malicious behaviors.

- **Energy Consumption Modeling Accuracy:**

The empirical algorithm and mathematical model developed for energy consumption provide good results, but they could be further refined to account for more dynamic network conditions, such as bursty traffic or heterogeneous hardware characteristics.

- **Integration with Emerging Technologies:**

The current framework primarily targets traditional SDN architectures. Integration and validation with edge computing, 5G/6G networks, and IoT-based SDN systems represent important future research directions.

## 8 CONCLUSION AND FUTURE ENHANCEMENTS

In this paper, we identify the problem of untrusted flow of network switch in SDN using power analysis in the OpenFlow protocol. This protocol identifies malicious behaviour of SDN switches. Through mathematical model of energy consumption and energy consumption routing algorithm we have sleep and active nodes of power are analysed using Software Defined Networking (SDN). Performance metrics of constant throughput, MSDR and PDR are analysed for the computation of energy saving. This energy saving algorithm is used for real time and large scale SDN application which is approximately 30% higher energy saving when compared to other existing algorithms. The sample networks of proposed energy consumption algorithm show much less energy compared with another existing algorithm (ERA's). Here, the power analysis is compared to produce trusted flow of network switches. The future enhancements can be focused on other features of dataset of SDN which can transmit the network switch for the more energy efficiency and can focus on accuracy of proposed energy consumption algorithms.

## REFERENCES:

- [1] Oliveira, Tadeu F., Samuel Xavier-de-Souza, and Luiz F. Silveira. "Improving energy efficiency on SDN control-plane using multi-core controllers." *Energies* 14.11 (2021): 3161.
- [2] Kamisiński, A., & Fung, C. (2019, October). Flowmon: Detecting malicious switches in software-defined networks. In *Proceedings of the 2019 Workshop on Automated Decision Making for Active Cyber Defense* (pp. 39-45).
- [3] Tang, Tuan A., et al. "Deep learning approach for network intrusion detection in software defined networking." 2016 international conference on wireless networks and mobile communications (WINCOM). IEEE, 2016.
- [4] Masood, Mohsin, et al. "Energy efficient software defined networking algorithm for wireless sensor networks." *Transportation Research Procedia* 40 (2019): 1481-1488.
- [5] Priyadarsini, Madhukrishna, Padmalochan Bera, and Mohammad Ashiqur Rahman. "A new approach for energy efficiency in software defined network." 2018 Fifth International Conference on Software Defined Systems (SDS). IEEE, 2018.
- [6] Phan, Truong Khoa. Design and management of networks with low power consumption. Diss. Université Nice Sophia Antipolis, 2019.
- [7] Chai, Rong, et al. "Energy consumption optimization-based joint route selection and flow allocation algorithm for software-defined networking." *Science China Information Sciences* 60 (2019): 1-14.
- [8] de la Cruz, Adrian Flores, et al. "Optimization of power consumption in SDN networks." *The ninth international conference on emerging networks and systems intelligence*. 2017..
- [9] Nencioni, Gianfranco, Bjarne E. Helvik, and Poul E. Heegaard. "Including failure correlation in availability modeling of a software-defined backbone network." *IEEE Transactions on Network and Service Management* 14.4 (2017): 1032-1045.
- [10] Dias de Assunção, Marcos, et al. "On designing SDN services for energy-aware traffic engineering." *Testbeds and Research Infrastructures for the Development of Networks and Communities: 11th International Conference, TRIDENTCOM 2016, Hangzhou, China, June 14-15, 2016, Revised Selected Papers*. Springer International Publishing, 2017.
- [11] Wei, Yunkai, et al. "Energy-aware traffic engineering in hybrid SDN/IP backbone networks." *Journal of Communications and Networks* 18.4 (2016): 559-566.
- [12] Rodrigues, Bruno B., et al. "GreenSDN: Bringing energy efficiency to an SDN emulation environment." 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). IEEE, 2015.

- [13] Huertas Celdrán, Alberto, et al. "Policy-based management for green mobile networks through software-defined networking." *Mobile Networks and Applications* 24 (2019): 657-666.
- [14] Rawat, Danda B., and Chandra Bajracharya. "Software defined networking for reducing energy consumption and carbon emission." *SoutheastCon 2016*. IEEE, 2016.
- [15] Berestizshevsky, Konstantin, et al. "SDNoC: Software defined network on a chip." *Microprocessors and Microsystems* 50 (2017): 138-153.
- [16] Kumar, H., and P. Gupta. "Sdn security issue and resolution." *Indian J. Appl. Res* 7.2 (2017).
- [17] Gonzalez, Andres J., et al. "A fault-tolerant and consistent SDN controller." *2016 IEEE global communications conference (GLOBECOM)*. IEEE, 2016.
- [18] Lee, Seungsoo, Changhoon Yoon, and Seungwon Shin. "The smaller, the shrewder: A simple malicious application can kill an entire sdn environment." *Proceedings of the 2016 ACM international workshop on security in software defined networks & network function virtualization*. 2016.
- [19] Scott-Hayward, Sandra, Sriram Natarajan, and Sakir Sezer. "A survey of security in software defined networks." *IEEE Communications Surveys & Tutorials* 18.1 (2016): 623-654.
- [20] Akhunzada, Adnan, et al. "Securing software defined networks: taxonomy, requirements, and open issues." *IEEE Communications Magazine* 53.4 (2017): 36-44.