Journal of Theoretical and Applied Information Technology
31st August 2022. Vol.100. No 16

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

AN EFFICIENT LINK STRENGTH CLASSIFICATION SYSTEM FOR SOFTWARE DEFINED NETWORKING USING DEEP LEARNING APPROACH

THANGARAJ ETHILU[1], ABIRAMI SATHAPPAN[2], PAUL RODRIGUES[3]

[1]Research Scholar, Department of Computer Science and Engineering, Annamalai University, Chidambaram, Tamil Nadu, India.
[2]Assistant Professor, Department of Computer Science and Engineering, Annamalai University, Chidambaram, Tamil Nadu, India.
[3]Professor, Department of Computer Science and Engineering, King Khalid University, Abha, Saudi Arabia.
E-mail: [1]ethilthangaraj@yahoo.co.in, [2]reachabisv@gmail.com, [3]drpaulprof@gmail.com.

ABSTRACT

In this paper, the link strength of each network switch in Software Defined Networking (SDN) environment system is analyzed into either good or bad using the proposed deep learning approach. This proposed method is designed with two modules as feature computation and Convolutional Neural Network (CNN) architecture. The feature computation module computes the intrinsic feature values of each network switch in SDN, and these intrinsic features are trained and classified into either good or bad using the proposed CNN architecture. The performance of the proposed SDN system is analyzed using the parameters precision, recall, accuracy, average detection rate, Packet Delivery Ratio (PDR) and latency.

Keywords: *SDN, Deep Learning, Network, Switch, Precision.*

1. INTRODUCTION

In past two decades, hardware architectures are used for security mechanism in many networking areas. This requires large number of resource elements and more expensive during the security maintenance. In order to overcome such drawbacks in hardware-based security architectures, the software architectures are now as days used for security mechanism. The implementation of these software architectures for the security enhancement in networking is called as Software Defined Networking (SDN). In SDN architectures, the control plane and the data plane are used to handle the data flow in and out the entire networking system. The specification of the reliable networking requirements is obtained from the user environment, and they are fed into control plane in SDN networking system [8-12]. The control plane in SDN networking system sends the requirement specifications to the data plane to process further. The data plane in SDN creates the virtual networking environment using Application Program Interfaces (API) modes. Network Functions Virtualization (NFV) method is used in SDN networking environment to provide the security management in networking devices in SDN system. All the SDN networking system can be operated or functioned using distributed architecture to process or handle the data. The center of the SDN networking system is centralized console which is the main responsible for the entire processing of the demands in SDN. The control plane is called as the centralized console in SDN architecture. The application plane in SDN is fully responsible for load abstraction, management of data and controls between various planes and improving the security protocols to prevent the entire SDN architecture from the various active and passive attacks [13-15].

The SDN architecture which is depicted in Fig.1, is designed with number of networking devices. These networking devices are more responsible for data and control transfer between the entire modules of SDN architecture. Due to the non-reliability and scalability of the networking devices, the malicious attacks will occur in these devices. Due to this, the networking devices become malicious networking devices which affect the performance of the entire SDN networking system. The greater number of presences of the malicious networking devices in SDN networking

architecture, the link failures occur which causes the severe data losses. In order resolve such limitations; this paper develops the novel and efficient mechanism for identifying the networking devices with bad link strength in SDN system is analyzed using the parameters precision, recall, accuracy, average detection rate, Packet Delivery Ratio (PDR) and latency which was elaborated in the literature survey by the various researchers.

The paper is structures as section 2 presents the existing security mechanisms for preventing link failures in SDN, section 3 provides the novel architecture for improving the performance of the SDN, the experimental results are discussed in section 4 and section 5 concludes the paper.

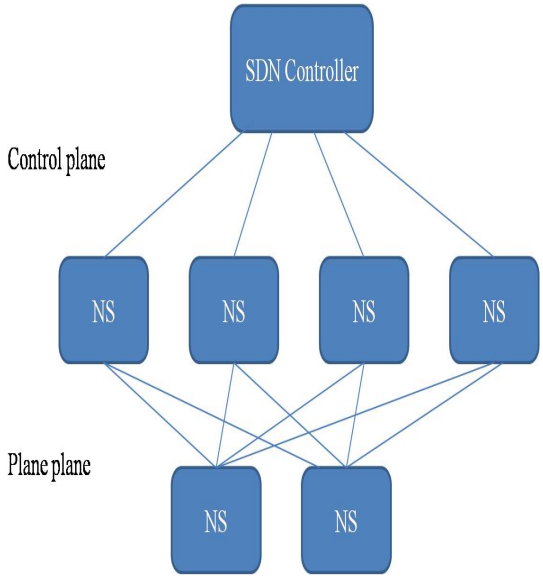

Figure. 1 SDN Architecture

2. LITERATURE SURVEY

Sugandhi Midha et al. (2021) detected and mitigated anomalies in SDN networking environment using defensive authentication approach. The authors created much intermediate protocol mechanism for improving the security authentication and the developed security authentication system detected Denial of Service attacks in SDN networking system. The authors analyzed this developed security authentication mechanism with respect to various performance metrics. Midha et al. (2020) detected and mitigated DDoS attacks in SDN networking environment using machine learning algorithm. The author used Support Vector Machine (SVM) algorithm for detecting and mitigating the DDoS attacks in SDN networks.

The PDR analysis of this work was made by the authors to implement the various routing framework protocols to test the performance of the attacks in this networking area. Saravanan et al. (2022) designed a policy enforcement system for implementing various security mechanisms in Privacy Preserving using Enhanced Shadow Honeypot technique for Data Retrieval in Cloud Computing system. The authors developed a new paradigm which was used to secure the data plane to provide reliable security authentication between networking devices and the networking environment. The authors analyzed the effectiveness of the designed Enhanced Shadow Honeypot technique to verify the security mechanism of the networking system. Woosik Lee et al. (2017) developed security constrained SDN architecture for preventing the security threats between the control and data plane. The authors framed Network Functions Virtualization (NFV) approach for improving the security authentication in SDN networking environment. The developed SDN architecture was tested and verified by different algorithms in order to verify the security mechanism. Jo et al. (2013) developed Service Overlay Networking Protocol (SONP) for providing the security mechanism between the control plane and the data plane in SDN networking system. This overlay protocol prevented un-authorized spoofing and other similar attacks from the SDN networking in this work.

3. PROPOSED METHODOLOGIES

In this work, the link strength of each network switch in SDN network environment is analyzed into either good or bad using the proposed deep learning approach. This proposed method is designed with two modules as feature computation and CNN architecture. The feature computation module computes the intrinsic feature values of each network switch in SDN, and these intrinsic features are trained and classified into either good or bad using the proposed CNN architecture.

Fig. 2 (a) shows the proposed link failure detection methodology using deep learning approach during the training of network switch links and Fig.2 (b) shows the proposed link failure detection methodology using deep learning approach during the testing of network switch link.

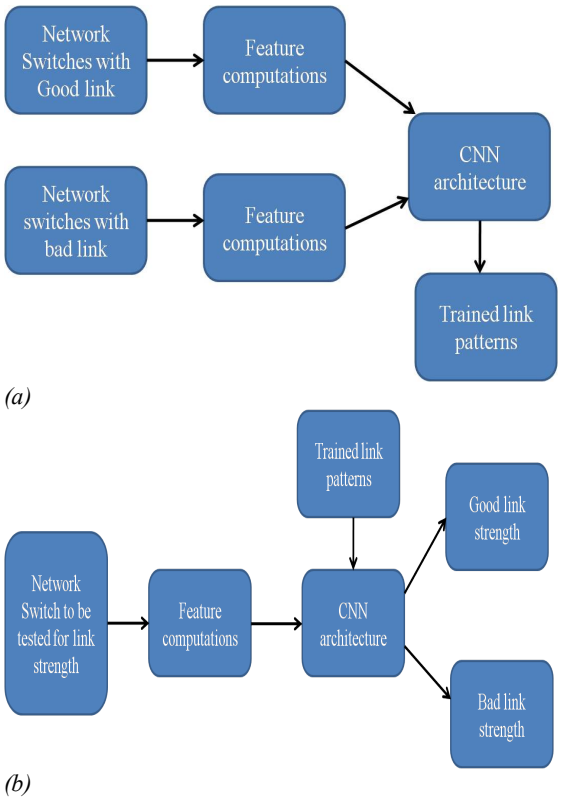

Figure 2 (a) Proposed Link failure detection methodology using deep learning approach (a) Training of network switch links (b) Testing of network switch link

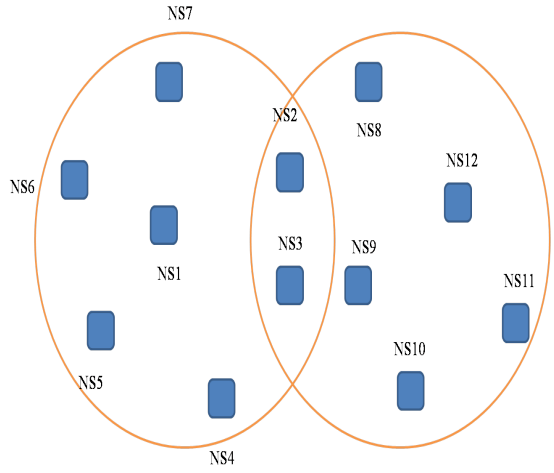

Figure. 3 Feature computations

3.1 Feature Computations

In this work, the SDN network system with two clusters is shown in Fig.3. The cluster 1 of the SDN network consists of seven numbers of network switches named as NS1, NS2…. NS7. The cluster 2 of the SDN network consists of seven numbers of network switches NS8, NS9, NS10, NS11, NS12, NS3, NS2. The network switches NS2 and NS3 are lying in both clusters in this SDN network system. The strength of the network switch is determined by its surrounding network switch links with it. In this work, the network switch NS1 is tested for its link strength with respect to its surrounding network switches.

Features are computed for each network switch based on the packet handling capability and buffering period of the packets in each network switch. In this work, the following intrinsic features are computed from each network switch. The Packet Index Weight (PIW) of each network switch is determined using the following equation.

$$PIW_i = \frac{\alpha_{ti} - \alpha_{tr}}{N}$$

Whereas α_{ti} and α_{tr} are the packet counts during transmission and during reception over the period 't' and N is the total number packets over the processing time 't1'

The PIW feature determines the packet handling capability of each network switch with respect to the center network switch.'

The Packet Buffering Time (PBT) of each network switch is determined using the following equation.

$$PBT_i = \frac{b_t * (t_t - r_t)}{t_t * r_t}$$

Whereas b_t is the buffering time of each network switch, t_t and r_t are the transmitting and receiving time period respectively.

The Loss Index Metric of each network switch is determined using the following equation.

$$LIM_i = \frac{P_l * PBT_i}{PIW_i}$$

Whereas P_l is the packet loss of the network switch during transmission and receiving the packets over the time period 't'.

3.2 Classifications

In this work, the features from the network switch with good link and the features from the network switch with bad link are computed. The computed intrinsic features from both categories of network switches are trained by the proposed deep learning structure. The feature from the network switch (to be tested) is computed and they are fed into the proposed deep learning structure using the pre trained feature patterns. For both training and classification of links of each network switch, the same CNN architecture is used.

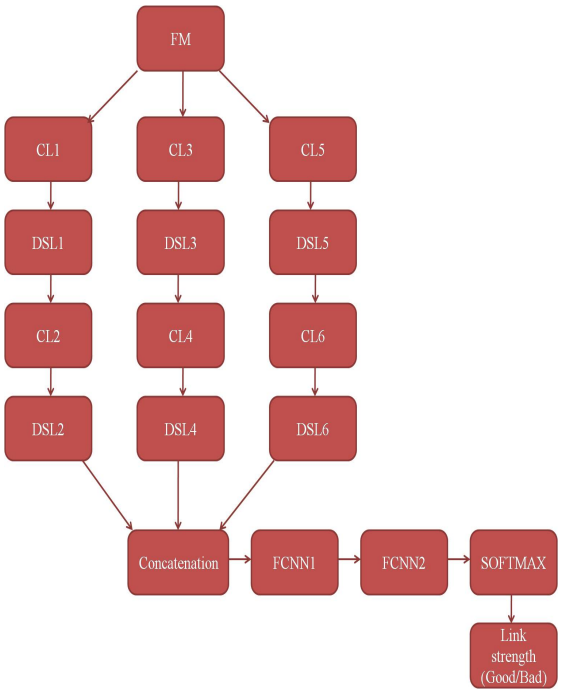

Figure. 4 Proposed CNN Architecture

The proposed CNN architecture is designed with six numbers of Convolutional Layers (CL), six numbers of Down Sampling Layers (DSL) and two numbers of Fully Connected Neural Networks (FCNN). This CNN architecture is structured into three vertical modules. Each vertical module consists of two CL layers and two DSL layers as depicted in Fig.4. The computed feature map (FM) is fed into three vertical modules.

Each CL layer in first vertical module is designed with 32 filters with a stride of 1*2. Each CL layer in second vertical module is designed with 64 filters with a stride of 1*1. Each CL layer in third vertical module is designed with 128 filters with a stride of 1*2. The CL layer in this architecture convolved the FM with its kernel and produces the Convolved Matrix (CM). The size of this CM is further reduced by passing this CM through DSL layer. The same functionality is done in other CL and DSL layers also. The final responses from DSL2, DSL4 and DSL6 are integrated using the concatenation function and the concatenated matrix is now fed into dense layers. In this work, two FCNN layers are used, in which FCNN1 is designed with 1024 fully connected biased neurons and the FCNN2 is designed with 15 fully connected biased neurons. The number of FCNN layers and its fully connected biased neuron count is chosen after several iterations in this work to produce the significant results. The final output from the FCNN 2 layer is fed into SoftMax layer, where the normalization process is done on this layer to produce the binary value. The binary value '0' is representing 'bad' strength and the binary value '1' is representing 'good' strength.

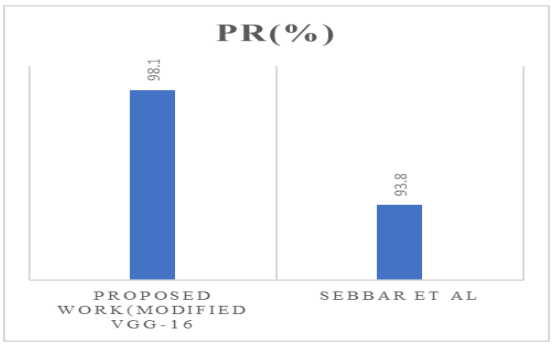

Figure. 5 Comparison of strength detection systems

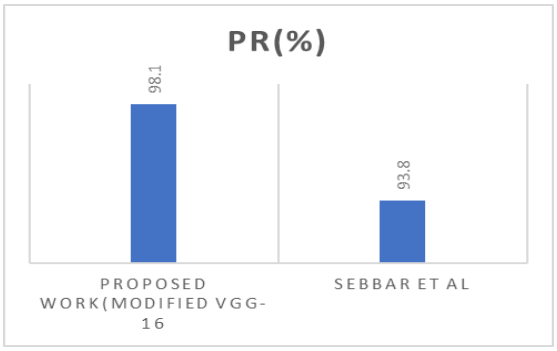

Figure. 6 Comparison of SDN systems using ALDR

4. RESULTS AND DISCUSSIONS

This proposed design methodology is simulated in Network Simulator version 2 with 100 numbers of network switches and three numbers of SDN controllers. The baud rate between each network switches is set to 120 Mb/sec and the baud rate between network switch and the SDN controller is set to 200 Mb/sec. The initial throughput of each network switch in this work is set to 15,000 b/sec and the number of low strength network switches (bad link) are set to 15 and the number of high strength network switches (good link) are set to 10. The following tables show the Margin specifications, Performance comparisons of proposed network switch bad link strength detection system, Performance comparisons of proposed SDN system using ALDR, Performance comparisons of proposed SDN system using ALDR, Performance comparisons of PDR for SDN System, Performance comparisons of Latency for

SDN System, Performance comparisons of proposed SDN system with respect to different CNN structures and Performance comparisons of proposed SDN system with respect to different state of the art methods.

Table 1: Margin specifications

CNN architectures	Total network switches with good links	Correctly detected network switches with good links	GLDR (%)
Modified VGG-16	85	84	98.8
Conventional VGG-16	85	79	92.9

Table 2: Performance comparisons of proposed network switch bad link strength detection system

CNN architectures	Total network switches with good links	Correctly detected network switches with good links	BLDR (%)
Modified VGG-16	15	14	93.3
Conventional VGG-16	15	12	80

Table 3: Performance comparisons of proposed SDN system using ALDR

CNN architectures	ALDR
Modified VGG-16	96.05
Conventional VGG-16	86.25

The performance of the designed system for link strength detection of the network devices are analyzed using Good Link Detection Rate (GLDR) and Bad Link Detection Rate (BLDR). The GLDR is defined as the ratio between the number of correctly detected network devices with good links and the total network devices with good links. The BLDR is defined as the ratio between the number of correctly detected network devices with bad links and the total network devices with bad links. Both GLDR and BLDR are determined in %. The percentage value of the measured parameters indicates that the system performance is high for SDN networks.

Table 4: Performance comparisons of PDR for SDN System

Presence of network switch with bad links	PDR (%)	
	Modified VGG-16	Conventional VGG-16
1	99.7	97.1
2	99.1	96.2
3	98.7	95.7
4	98.4	95.1
5	98.1	94.7
6	97.9	94.1
7	97.5	93.9
8	97.1	93.8
9	96.7	93.5
10	96.3	93.2
11	96.1	92.7
12	95.7	92.5
13	95.3	92.1
14	95.2	91.6
15	94.9	91.1

Table 5: Performance comparisons of Latency for SDN System

Presence of network switch with bad links	Latency (ms)	
	Modified VGG-16	Conventional VGG-16
1	1.2	3.87
2	1.28	3.92
3	1.56	4.98
4	1.92	5.36
5	2.03	5.93
6	2.18	6.29
7	2.93	6.57
8	3.01	6.86
9	3.28	6.95
10	3.56	7.12
11	3.98	7.46
12	4.28	7.95
13	4.76	8.46
14	4.97	8.97
15	5.29	9.89

Table 1 is the performance comparisons of proposed network switch good link strength detection system. The modified VGG-16 detects 84 network switches with good links over 85 network switches with good links. Therefore, the value of GLDR is about 98.8%. The conventional VGG-16 detects 79 network switches with good links over 85 network switches with good links. Therefore, the value of GLDR is about 92.9%.

Journal of Theoretical and Applied Information Technology
31st August 2022. Vol.100. No 16

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

Table 6: Performance comparisons of proposed SDN system with respect to different CNN structures

CNN architectures	Pr (%)	Re (%)	Acc (%)
Modified VGG-16	98.1	98.6	98.5
Conventional VGG-16	95.3	95.9	95.8

Table 7: Performance comparisons of proposed SDN system with respect to different state of the art methods

CNN architectures	Pr (%)	Re (%)	Acc (%)
Modified VGG-16	98.1	98.6	98.5
Conventional VGG-16	93.8	94.6	95.6
Ref [2]	93.1	94.8	95.1

Table 2 is the performance comparisons of proposed network switch bad link strength detection system. The modified VGG-16 detects 14 network switches with good links over 15 network switches with good links. Therefore, the value of BLDR is about 93.3%. The conventional VGG-16 detects 12 network switches with good links over 15 network switches with good links. Therefore, the value of BLDR is about 80%.

The Average Link Detection Ratio (ALDR) is defined as the average value of the GLDR and BLDR. In this work, the ALDR is about 96.05% using the modified VGG-16 architecture. The conventional VGG-16 obtains 86.25% of ALDR for the SDN system. Table 3 shows the performance comparisons of proposed SDN system using ALDR.

Table 4 is the performance comparisons of PDR for SDN system. The number of presences of network switches with bad links reduces the PDR rate significantly. The proposed SDN system is tested with modified VGG-16 and conventional VGG-16 CNN structures. The PDR of the proposed SDN system using modified VGG-16 is significantly higher than the PDR value of the SDN system using conventional VGG-16 structure.

The proposed SDN system using modified VGG-16 structure obtained 99.7% of PDR if the one number of network switch with bad links. The proposed SDN system using modified VGG-16 structure obtained 94.9% of PDR if the 15 number of network switch with bad links.

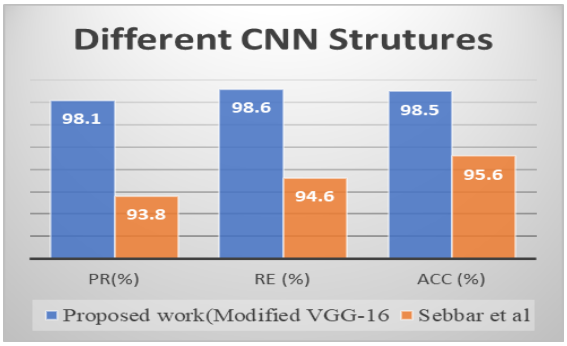

Figure. 7 Comparison of SDN systems with respect to different CNN Structures

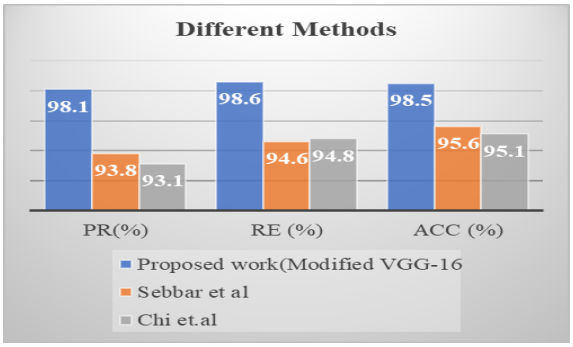

Figure. 8 Comparison of SDN systems with respect to different state of methods

Table 5 is the performance comparisons of latency for SDN system. The number of presences of network switches with bad links increases the latency significantly. The proposed SDN system is tested with modified VGG-16 and conventional VGG-16 CNN structures in terms of latency. The latency of the proposed SDN system using modified VGG-16 is significantly lower than the latency value of the SDN system using conventional VGG-16 structure.

The proposed SDN system using modified VGG-16 structure consumed 1.2 ms of latency if the one number of network switch with bad links. The proposed SDN system using modified VGG-16 structure consumed 5.29 ms of latency if the 15 number of network switch with bad links.

Further, the proposed SDN system is analyzed using sensitivity, specificity and accuracy rate using the following equations.

$$Precision\ (\mathrm{Pr}) = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

$$Recall\ (Re) = \frac{TP}{TP + FN}$$

$$Accuracy\ (Acc) = \frac{TP + TN}{TP + TN + FP + FN}$$

Whereas correctly detected network switches with good links and bad links are denoted by TP and TN respectively. The wrongly detected network switches with good links and bad links are denoted by FP and FN respectively.

The computed values of Pr, Re and Acc are in % and they are varied between 0 and 100 in values.

Table 6 shows the performance comparisons of proposed SDN system with respect to different CNN structures Modified VGG-16 and Conventional VGG-16. The proposed SDN system using modified VGG-16 structure obtains 98.1% Pr, 98.6% Re and 98.5% Acc. The conventional VGG-16 structure obtains 95.3% Pr, 95.9% Re and 95.8% Acc.

Table 7 is the performance comparisons of proposed SDN system with respect to different state of the art methods. From 7, the proposed SR system using modified VGG-16 structure obtains significant simulation results when compared with other state of the art methods.

5. CONCLUSIONS

In this paper, the SDN system is designed using modified VGG-16 architectures. The proposed SDN system using modified VGG-16 structure obtained 99.7% of PDR if the one number of network switch with bad links. The proposed SDN system using modified VGG-16 structure obtained 94.9% of PDR if the 15 number of network switch with bad links. The proposed SDN system using modified VGG-16 structure consumed 1.2 ms of latency if the one number of network switch with bad links. The proposed SDN system using modified VGG-16 structure consumed 5.29 ms of latency if the 15 number of network switch with bad links. The proposed SDN system using modified VGG-16 structure obtains 98.1% Pr, 98.6% Re and 98.5% Acc. The proposed SR system using modified VGG-16 structure obtains significant simulation results when compared with other state of the art methods. The future work of the efficient link strength classification system for software defined networking using deep learning approach can be investigated by using unsupervised learning methods.

REFERENCES:

[1] Sebbar, A., ZKIK, K., Baddi, Y. MitM detection and defense mechanism CBNA-RF based on machine learning for large-scale SDN context. J Ambient Intell Human Comput 11, 5875–5894 (2020).

[2] P. -W. Chi, M. -H. Wang and Y. Zheng, "SandboxNet: An Online Malicious SDN Application Detection Framework for SDN Networking," 2020 International Computer Symposium (ICS), 2020, pp. 397-402.

[3] Midha S., Kaur G. (2020), "SVM Implementation for DDoS Attacks in Software Defined Networks", 'International Journal of Innovative Technology and Exploring Engineering (IJITEE)', ISSN: 2278–3075, Volume-10 Issue-1, November 2020, Page No. 205-212.

[4] Saravanan, T., & Thillaiarasu, N. (2021, March). Optimal Grouping and Belief based CH selection in mobile ad-hoc network using Chunk Reliable Routing Protocol. In 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) (pp. 933-940). IEEE.

[5] Sugandhi Midha, Khushbu Tripathi and M.K.Sharma, "Novel Approach to detect Anomalies using Defensive Algorithm in SDN flows", Turkish Journal of Computer and Mathematics Education, Vol.12 No.2 (2021), 536- 542.

[6] Jo, J.Y.; Lee, S.Y.; Kong, J.U.; Kim, J.W. A centralized network policy controller for SDN-based service overlay networking. J. Korea Inf. Commun. Soc. 2013, 38, 266–278.

[7] Woosik Lee andN amgi Kim, "Security Policy Scheme for an Efficient Security Architecture in Software-Defined Networking", Information 2017, 8(2), 65.

[8] Saravanan, T., & Sasikumar, P. Assessment and Analysis of Action Degeneracy Due to Blackhole Attacks in Wireless Sensor Networks. In Proceedings of 6th International Conference on Recent Trends in Computing: ICRTC 2020 (p. 345). Springer Nature.

[9] Satasiya, D.; Raviya, R.; Kumar, H. Enhanced SDN security using firewall in a distributed scenario. In Proceedings of the Advanced Communication Control and Computing Technologies, Ramanathapuram, India, 25–27 May 2016; pp. 25–27.

[10] Husssein, A.; Elhajj, I.H.; Chehab, A.; Kayssi, A. SDN security plane: An architecture for resilient security services. In Proceedings of the IEEE International Conference on Cloud Engineering Workshop, Luxembourg, Luxembourg, 12–15 December 2016; pp. 4–8.

[11] Saravanan, T., & Nithya, N. S. (2019). Modeling displacement and direction aware ad hoc on-demand distance vector routing standard

for mobile ad hoc networks. Mobile Networks and Applications, 24(6), 1804-1813.

[12] Gonzaiez, C.; Charfadine, S.M.; Flauzac, O.; Nolot, F. SDN-based security framework for the IoT in distributed grid. In Proceedings of the International Multidisciplinary Conference on Computer and Energy Science, Split, Croatia, 13–15 July 2016; pp. 13–15.

[13] Rawat, D.B.; Reddy, S.R. Software defined networking architecture, security and energy efficiency: A survey. IEEE Commun. Surv. Tutor. 2017, 19, 325–346.

[14] Zargar, S.T.; Joshi, J.; Tipper, D. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Commun. Surv. Tutor. 2013, 15, 2046–2069.

[15] Yan, Q., Yu, F. R., Gong, Q., & Li, J. (2015). Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. IEEE Communications Surveys & Tutorials, 18(1), 602-622.

[16] Saravanan, T., Saravanakumar, S., Rathinam, G., Narayanan, M., Poongothai, T., Patra, P. S. K., & Sengan, S. (2022). Malicious attack alleviation using improved time-based dimensional traffic pattern generation in UWSN. Journal of Theoretical and Applied Information Technology, 100(3).

[17] Prathibha, S., Bino, J., Ahammed, T., Das, C., Oion, S. R., Ghosh, S., & Afroj, M. (2022, January). Detection Methods for Software Defined Networking Intrusions (SDN). In 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI) (pp. 1-6). IEEE.

[18] Chen, J., Wang, Y., Huang, X., Xie, X., Zhang, H., & Lu, X. (2022). ALBLP: adaptive load-balancing architecture based on link-state prediction in software-defined networking. Wireless Communications and Mobile Computing, 2022.

[19] Saravanan, T., & Nithya, N. S. (2020, December). Mitigation of attack patterns based on routing reliance approach in MANETs. In 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN) (pp. 387-392). IEEE.

[20] Sharadqeh, A. (2022). A novel algorithm for software defined networks model to enhance the quality of services and scalability in wireless network. International Journal of Electrical & Computer Engineering (2088-8708), 12(2).

[21] Kao, M. T., Sung, D. Y., Kao, S. J., & Chang, F. M. (2022). A Novel Two-Stage Deep Learning Structure for Network Flow Anomaly Detection. Electronics, 11(10), 1531.

[22] Abid, M. A., Afaqui, N., Khan, M. A., Akhtar, M. W., Malik, A. W., Munir, A., ... & Shabir, B. (2022). Evolution towards smart and software-defined internet of things. AI, 3(1), 100-123.

[23] Saravanan, T., & Saravanakumar, S. (2021, December). Privacy Preserving using Enhanced Shadow Honeypot technique for Data Retrieval in Cloud Computing. In 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N) (pp. 1151-1154). IEEE.

[24] Kafetzis, D., Vassilaras, S., Vardoulias, G., & Koutsopoulos, I. (2022). Software-Defined Networking meets Software-Defined Radio in Mobile Ad hoc Networks: State of the Art and Future Directions. IEEE Access.

[25] Khamaiseh, S., Al-Alaj, A., Adnan, M., & Alomari, H. W. (2022). The Robustness of Detecting Known and Unknown DDoS Saturation Attacks in SDN via the Integration of Supervised and Semi-Supervised Classifiers. Future Internet, 14(6), 164.

[26] Othman, S. B., Almalki, F. A., & Sakli, H. (2022). Internet of things in the healthcare applications: overview of security and privacy issues. Intelligent Healthcare, 195-213.

[27] Zhu, Y., Ahmad, A., Ahmad, W., Vatin, N. I., Mohamed, A. M., & Fathi, D. (2022). Predicting the Splitting Tensile Strength of Recycled Aggregate Concrete Using Individual and Ensemble Machine Learning Approaches. Crystals, 12(5), 569.

[28] Sarothi, S. Z., Ahmed, K. S., Khan, N. I., Ahmed, A., & Nehdi, M. L. (2022). Machine Learning-Based Failure Mode Identification of Double Shear Bolted Connections in Structural Steel. Engineering Failure Analysis, 106471.

[29] Gao, Y., González, V. A., Yiu, T. W., Cabrera-Guerrero, G., & Deng, R. (2022). Predicting Construction Workers' Intentions to Engage in Unsafe Behaviours Using Machine Learning Algorithms and Taxonomy of Personality. Buildings, 12(6), 841.

[30] Javadpour, A., Pinto, P., Ja'fari, F., & Zhang, W. (2022). DMAIDPS: a distributed multi-agent intrusion detection and prevention system for cloud IoT environments. Cluster Computing, 1-18.