Gapm-Dlnn-Based Attack Detection for Distributed Packet Switching in Local Computer Networks

Amaresan Venkatesan

v.amaresan@gmail.com

Abstract:
In local computer networks, the packet transmission mechanism provided by Ethernet has been used. Switched packets are distributed in ether and transmitted to the destination in that transmission. None of the existing works weren't focused on the buffer overflow attack type detection with mitigation. Here, Global Attention with Parametric Mish-based Deep Learning Neural Network (GAPM-DLNN)-based attack detection in packet switching is proposed in this paper. Primarily, the index is constructed using Indexed Divide and Conquer (IDC) if the packet transmission request is accepted by the main server. Next, paths are generated. The optimal path is selected using Guided Kookaburra Optimization (GKO) from the generated paths. The load is estimated and sent to the main server using the Transmission Control Protocol (TCP). Moreover, the packets are reordered in TCP. Then, by using the GAPM-DLNN approach, the attack is detected in the obtained packet. Next, by utilizing the Quantum Fuzzy Inference System (QFIS) approach, the attack is mitigated. The map-reduce function is used if the attack is stack-based overflow. If the attack is a heap overflow, then the attacked data is compared with the index table to reduce its size. If the attack is a format string, then the attacked data is blocked and sends an acknowledgment to the destination. The proposed method achieves a 97.5% Packet Delivery Ratio (PDR) as per experimental analysis.

Keywords: Global Attention with Parametric Mish based Deep Learning Neural Network (GAPM-DLNN), Guided Kookaburra Optimization (GKO), Indexed Divide and Conquer (IDC), Quantum Fuzzy Inference System (QFIS), Ethernet, Distributed Packet Switching, and Local Computer Networks.

1. INTRODUCTION

Recently, in large-scale data centers, which are based on packet-switched communication networks, more and more applications have been deployed (Huang et al., 2021). Hence, in health, business, science, or social network applications, communication networks are used (Chiesa et al., 2021). The communication network has various switch settings and displays (Adhikari et al., 2020). Data are transferred to the destination by way of a switch and display connection (Cheng et al., 2021). Data transfer is mainly handled by Ethernet. Thus, for the communication of Ethernet, accurate measurements of transfer delay are required (Turcato et al., 2020). The inaccurate measurement of delay leads to switch traffic and produces poor communication performance (Diadamo et al., 2022). In TCP, the attackers are also involved in affecting the quality of the data transmission. MultiPath TCP (MPTCP) is an innovative transport protocol currently designed and developed (Kumar & Das, 2021). If the attacker influences the single TCP, then it affects the whole communication. The Distributed Denial of Service (DDoS) attack has become a real threat recently (Aladaileh et al., 2020). Moreover, buffer overflow attacks are highly used to produce congestion, thus

causing several losses (Balarezo et al., 2020). Many methods like Machine Learning (ML) and isolation techniques are used to reduce these attacks and the congestion problem (Blöcher et al., 2021). However, the research problem is presented. Thus, a GAPM-DLNN-based attack detection with distributed packet switching in local computer networks is proposed in this paper.

1.1 Problem statement

- None of the existing research works focused on the automated buffer overflow attack and their types with mitigation for preventing data transmission in local computer networks.
- In the existing (Wei et al., 2020), path failure and waiting time were presented, which affected the data transmission process.
- In the existing (Abdelmoniem&Bensaou, 2021), traffic was increased due to the heavy network.
- Packet reordering was only presented based on the pattern of the packets in (Olmedo et al., 2020), which provided the inefficiency in data transmission.

1.2 Objectives

- The attack types are detected using GAPM-DLNN and mitigated by QFIS.
- The path failure and waiting time are reduced by the GKO algorithm.
- The server first checks the loads to reduce network complexity.
- To reorder the packets using IDC.

The remaining part is arranged as: the existing research is elucidated in Section 2, the proposed model is described in Section 3, the experimental evaluation is elucidated in Section 4, and Section 5 concludes the paper with future scope.

2. RELATED WORK

(Wei et al., 2020) developed a Shared Bottleneck-based Congestion Control (SB-CC) for congestion control as well as packet scheduling. According to the changes in window size, the scheduling scheme distributed data. As per the experimental evaluation, when analogized to the conventional model, this model provided more accurate performance.

(Abdelmoniem&Bensaou, 2021) recommended a Timely Retransmitted ACKnowledgement (T-RACK) for timely recovery from losses. The presented model was developed on the software shim layer. As per the outcome, when compared to the conventional model, the flow completion time was improved.

(Olmedo et al., 2020) suggested a TCP protocol progress by considering negative acknowledgment. the small protocol packet was supported; in addition, improvements were shown in quality of service metrics. As per the outcome, by considering diverse distances, packet error rates, bandwidths, as well as technologies, better performance was attained.

(Sahoo et al., 2020) offered a Response Time Switch Migration (RTSM) to optimize the control messages' response time during switch migration. As per the outcome, the presented research approach outperformed well than the other existing research methods. Because of not considering the load of the network layers, the traffic occurred.

(Tuan et al., 2020) established an attack mitigation system in Software Defined Network (SDN) centered networks for TCP-SYN attacks by using a machine approach. The research also incorporated the automated monitoring window time. As per the experimental result, when weighed against the conventional model, the model achieved higher performance. But, attack mitigation and malicious activity detection were more complex.

3. PROPOSED GAPM-DLNN-BASED DISTRIBUTED PACKET SWITCHING IN LOCAL COMPUTER NETWORKS

Here, for local computer networks, GAPM-DLNN-based distributed packet switching is proposed. The main phase of the proposed research is the buffer overflow attack detection with mitigation. In Figure 1, the proposed research work's structure is given.

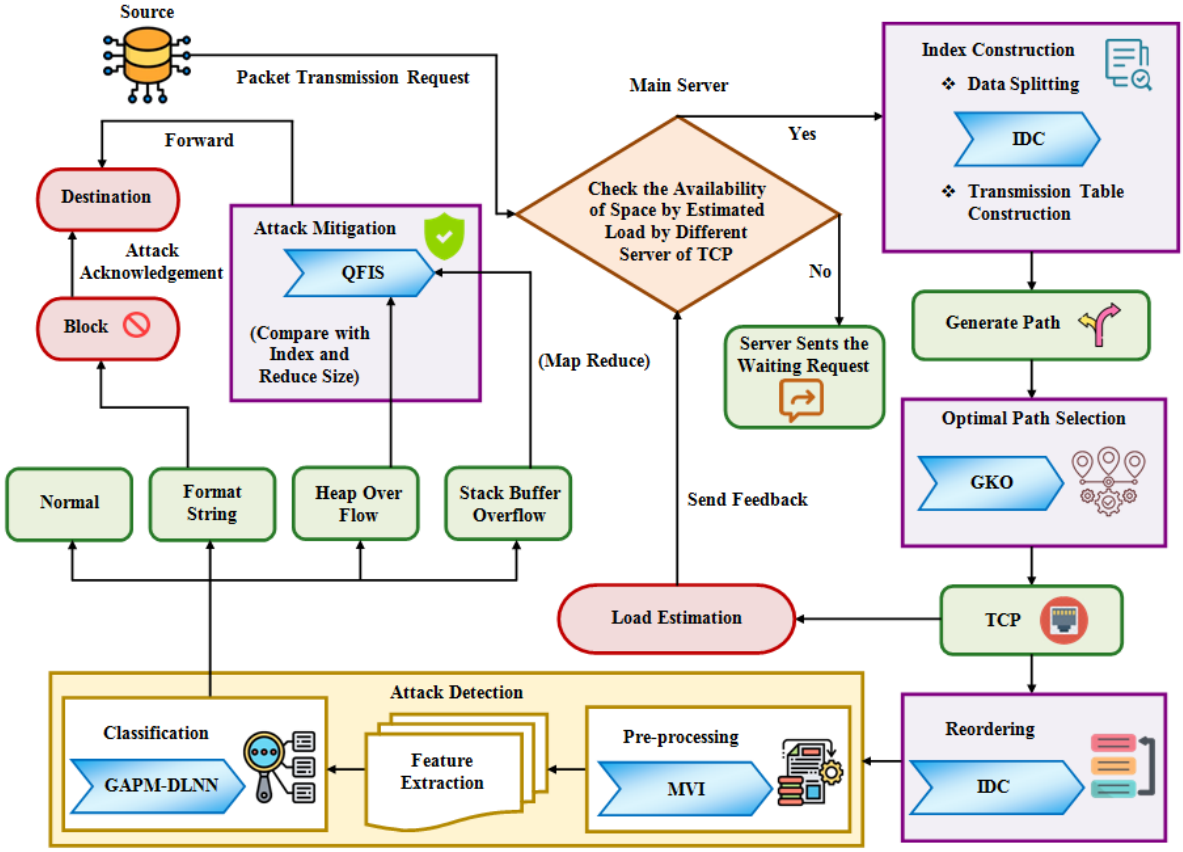


Figure 1: Structure of the proposed research work

3.1 Source

Primarily, the source device $\Re_c$ sends the packet transmission request to the main server. By estimated load ℓ_d of different TCP, the main server checks the availability of the space. The source is allowed for the data transmission by constructed index if the space is available. Else, the server sends the waiting request to the source device.

3.2 Index construction

The data is split into packets using IDC for the index construction of the allowed device for the distributed packet-switching process. Here, the data is initially split and sorted based on ascending order according to the procedure of IDC. Here, the packets $\aleph_p$ of input data $\wp_{in}$ are represented in equation (1) as,

$$\aleph_p = \{\aleph_1, \aleph_2, ..., \aleph_n\} \quad (1)$$

Here, $\aleph_n$ implies the n-number of splitted packets. Then, by considering the (A) source port number, (B) destination port number, (C) sequence number, (D) acknowledgment number, (E) data offset, (F) reserved, (G) window size, (H) checksum, (I) urgent pointer, (J) data, and so on, the transmission table is constructed.

3.3 Generate Paths

Here, an m-number of paths is generated that consists of routers and switches for the data transmission. The generated paths ψ_g are represented in equation (2) as,

$$\psi_g = \{\psi_1, \psi_2, ..., \psi_m\} \quad (2)$$

ψ_m implies the m-number of generated paths.

3.4 Optimal Path Selection

Here, the optimal paths are selected for reducing the waiting time and path failure from the generated ψ_g. This research methodology uses the GKO algorithm for the path selection. By focusing in all directions, the conventional KO approach searches the prey. But, KO initially selects the prey randomly. Further preys are selected based on the selected initial prey, thus providing a poor result regarding convergence. Thus, for initial prey selection, this research methodology considers the Guided local search function. The ψ_g are considered as the kookaburra, and the position of the population is assigned by using equation (3). In equation (4), the position of the population is represented. Moreover, based on minimum distance d_t and minimum transmission time T_s, the fitness for the population is fixed, which is expressed in equation (5) as,

$$z_{j,k} = \chi^{low} + \hbar.\left(\chi^{upper} - \chi^{low}\right) \tag{3}$$

$$Z = \begin{bmatrix} Z_1 \\ \vdots \\ Z_j \\ \vdots \\ Z_N \end{bmatrix}_{N\times r} = \begin{bmatrix} z_{1,1} & \cdots & z_{1,k} & \cdots & z_{1,r} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ z_{j,1} & \cdots & z_{j,k} & \cdots & z_{j,r} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ z_{N,1} & \cdots & z_{N,k} & \cdots & z_{N,r} \end{bmatrix}_{N\times r} \tag{4}$$

$$\delta_{ss} = \min\left(d_t, T_s\right) \tag{5}$$

Where, N implies the kookaburra, r depicts the problem variables, $\hbar$ signifies the random number, χ^{upper} and χ^{low} signify the population's upper as well as lower bound, respectively, and δ_{nss} signifies the fitness function. The population position is updated in the exploration stage after evaluating the fitness function, which is given in equation (6) as,

$$z_{j,k}^{eplr} = z_{j,k} + \hbar.\left(\Im_{j,k} - \lambda_d.z_{j,k}\right) \tag{6}$$

$$\Im_{j,k} = \delta_{ss} + \eta \times \sum Y_t \times I\left(\psi_g\right) \tag{7}$$

$$z_j = \begin{cases} z_{j,k}^{epl}, & if\ \left(\delta_{ss}\right)^{epl} < \delta_{ss} \\ z_j, & else \end{cases} \tag{8}$$

Next, the position of the population updated during the exploitation stage is derived in equation (9). The exploitation stage refers to the ability of the approach to achieve better solutions near the obtained solutions and promising areas.

$$z_{j,k}^{epli} = z_{j,k} + \left(1-2\hbar\right).\frac{\left(\chi^{upper} - \chi^{low}\right)}{\varphi_{itr}} \tag{9}$$

$$z_j = \begin{cases} z_{j,k}^{epli}, & \left(\delta_{ss}\right)^{epli} < \delta_{ss} \\ z_j, & else \end{cases} \tag{10}$$

Here, $z_{j,k}^{eplr}$ and $z_{j,k}^{epli}$ specify the updated population position at the exploration and exploitation stage, correspondingly, $\hbar$ specifies the random number, $\Im_{j,k}$ depicts the initially selected prey based on guided local search that is given in equation (7), and φ_{itr} signifies the iteration count. The GKO algorithm's pseudocode is given below,

Pseudocode for GKO

Input: ψ_g

Output: $\zeta_{ss} = \{\zeta_1, \zeta_2, \zeta_3, ..., \zeta_{n_r}\}$

Begin

Initialize population, φ_{itr} and maximum iteration $\max(\varphi_{itr})$

Evaluate δ_{nss}

Set $\varphi_{itr} = 1$

While $(\varphi_{itr} \le \max(\varphi_{itr}))$ **do**

 # Updation in Exploration stage

If $((\delta_{ss})^{epl} < \delta_{ss})${

$$z_j = z_{j,k} + \hbar.(\Im_{j,k} - \lambda_d . z_{j,k})$$

} else {

z_j

} end if

 # Updation in Exploitation stage

If $((\delta_{ss})^{epli} < \delta_{ss})${

$$z_j = z_{j,k} + (1 - 2\hbar).\frac{(\chi^{upper} - \chi^{low})}{\varphi_{itr}}$$

} else {

z_j

} end if

Evaluate δ_{nss}

Set $\varphi_{itr} = \varphi_{itr} + 1$

End while

Return $\zeta_{ss} = \{\zeta_1, \zeta_2, \zeta_3, ..., \zeta_{n_r}\}$

End

In equation (11), the selected optimal path is expressed),

$$\zeta_{ss} = \{\zeta_1, \zeta_2, \zeta_3, ..., \zeta_{n_r}\} \tag{11}$$

Here, ζ_{ss} signifies the selected path set, and ζ_{n_r} expresses the n_r-number of selected paths. The input packet is forwarded to the destination via TCP with the help of the selected path.

3.5 TCP with reordering

Here, by using IDC, Transmission Control Protocol (TCP) collects the data and reorders the input data. Now, the TCP estimates the load of memory using equation (12),

$$\ell_d = T_{pd} - O_{pd} \tag{12}$$

Where, T_{pd} signifies the total space, and O_{pd} signifies the occupied space. The ℓ_d is given as feedback to the main server. the input data are recombined with the help of a constructed index (i.e., transmission table)

In the reordering process. the order is checked with the transmission table For each packet. The reordered packet is signified as γ_p.

3.6 Attack detection

Here, checking whether the attack is present or not in γ_p for avoiding the heavy traffic in the network. In the further section, the process of attack detection is described.

(a) Pre-processing

Initially, by using the missing value imputation process, the missing values of obtained γ_p are resolved. By calculating the available input values' mean value, the missing value is imputed. The pre-processed data is signified as τ_s.

(b) Feature extraction

Here, the features, namely port ID, source IP address, destination IP address, source as well as destination of the IP address of the indicated Virtual Local Area Network (VLAN) interface, and so on are extracted from τ_s. The extracted features are represented as ω_e.

(c) Classification

Here, by using GAPM-DLNN, the buffer attack types are classified. ω_e are given as input to GAPM-DLNN for the classification process. The conventional DLNN provides efficient output for different formations of input with their hidden relationships. But, it has a problem in the extraction of all the vectors from the input data, and it also has a vanishing gradient problem. This research methodology uses a Global attention layer at the initial stage to cover all vector points of the input data to solve these research problems. Also, the Parametric Mish activation function is used for solving the vanishing gradient problem. In Figure 2, the GAPM-DLNN's structure is displayed.

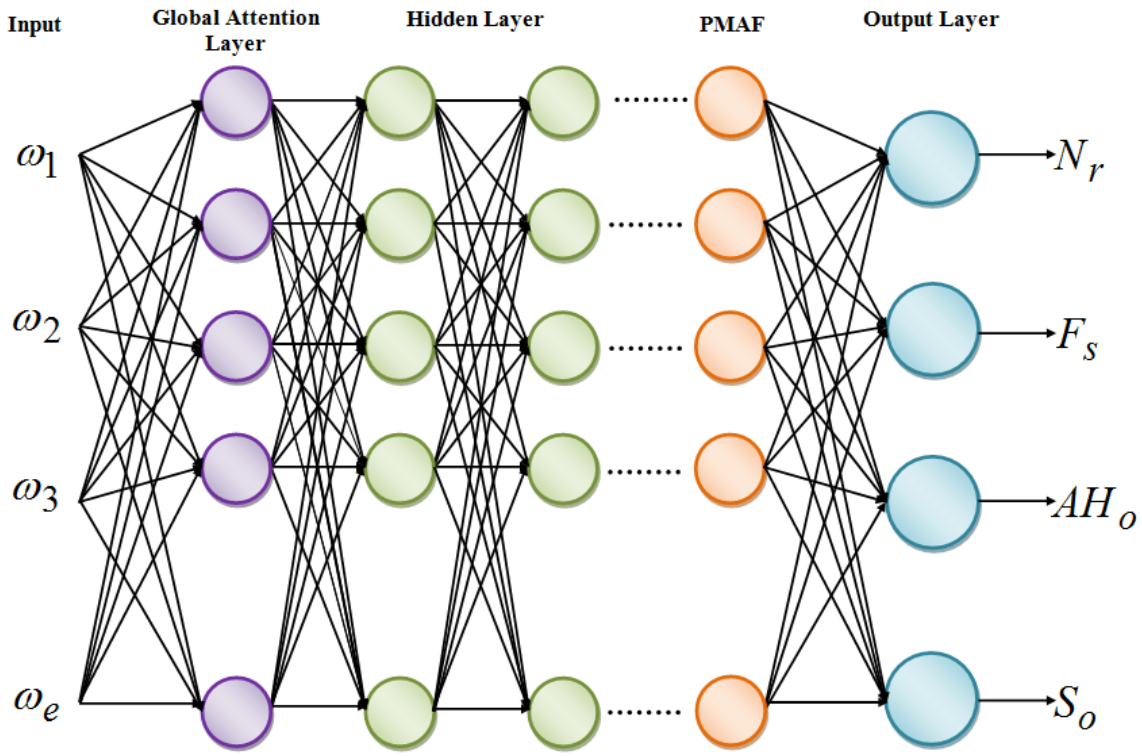


Figure 2: Structure of GAPM-DLNN

Initially, the global attention mechanism is used in the input layer, and the derivation for global attention G_l is given in equation (13) as,

$$G_l = \lambda(\omega_e) \tag{14}$$

Here, $\lambda(\ )$ outlines the calling function of global attention. Next, the hidden layer output H_l is derived in equation (15),

$$H_l = \mu_b + \sum G_l \,.\, \mu_w \tag{15}$$

Where, μ_b and μ_w signify the bias and weight values, respectively. Lastly, by using the activation function, the output layer O_l is derived, and the formulation is given in equation (16) as,

$$O_l = \begin{cases} H_l & if\ H_l \geq 0 \\ H_l \tanh(softplus(H_l)) & if\ H_l \leq 0 \end{cases} \tag{16}$$

The proposed RMLU-DLNN's pseudocode is shown below,

Psuedo code for GAPM-DLNN

Input: ω_e

Output: O_l

Begin

Initialize weight, bias, hidden and output layer

For each ω_e **do**

Construct $G_l = \lambda(\omega_e)$

Derive H_l

Calculate output function by,

if $H_l \geq 0$ **{**

$O_l = H_l$

} else {

$O_l = H_l \tanh(softplus(H_l))$

} end if

End each

Return O_l

End

The classified output is denoted as,

$$O_l = \{N_r, F_s, HA_o, S_o\} \tag{17}$$

Here, N_r, F_s, HA_o and S_o signify the normal, format string, heap overflow, and stack overflow.

4. RESULT AND DISCUSSION

Here, the proposed methodology's performance is analyzed, and in the working platform of Python, the proposed methodology is implemented.

4.2 Dataset description

This research methodology uses the ASNM-CDX-2009 dataset, which is collected from publically available sources and is mentioned under the reference section, for the performance analysis. From the input data, 80% of the data is wielded for training purposes, and the remaining 20% is wielded for testing purposes.

4.3 Performance analysis

(a) Performance analysis for path selection

The proposed GKO's performance is weighed against KO, Osprey Optimization (OO), Walrus Optimization (WO), as well as Egret Swarm Optimization (ESO) algorithm.

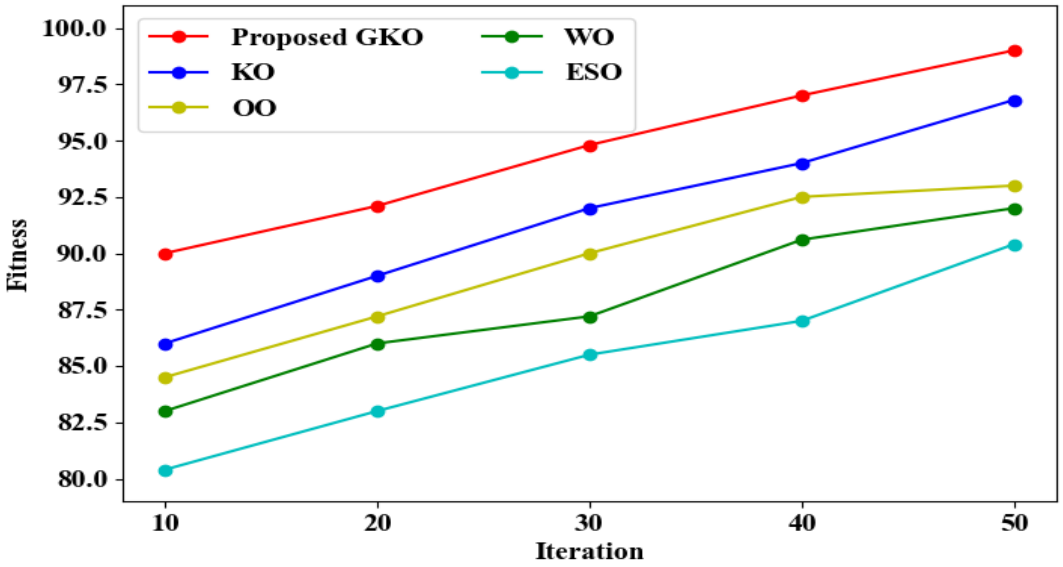


Figure 3: Fitness vs iteration analysis

The fitness vs iteration analysis is displayed in Figure 3. The improvement in fitness level is considered as %. At the 50th iteration, the GKO achieves a higher fitness of 99% because of guided local searching. But, for each iteration variation, the existing approach provided lower performance.

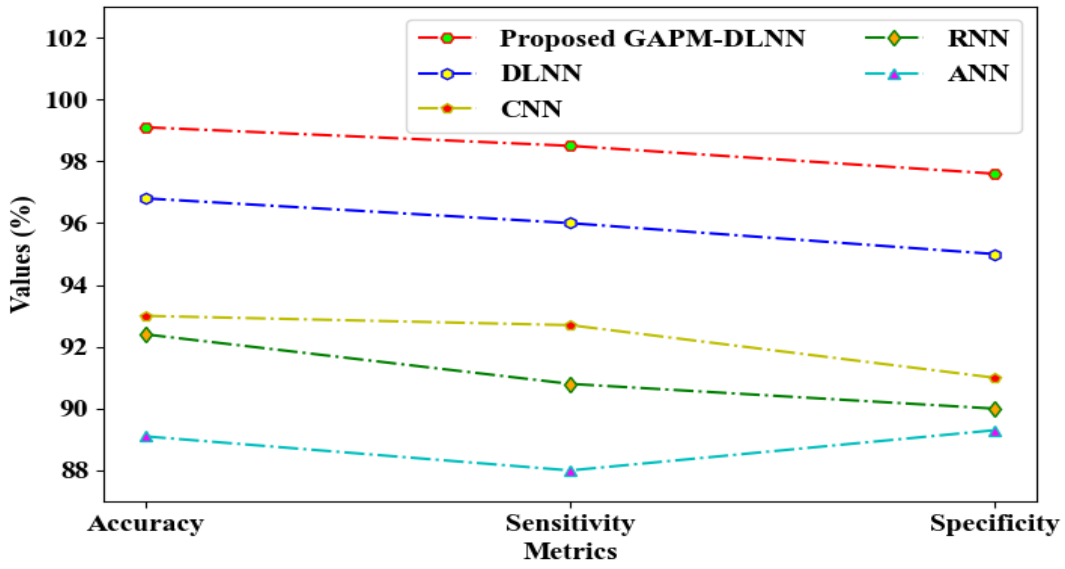
(b) Performance analysis for classification

Here, the proposed GAPM-DLNN is analyzed with the DLNN, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), as well as Artificial Neural Network (ANN).

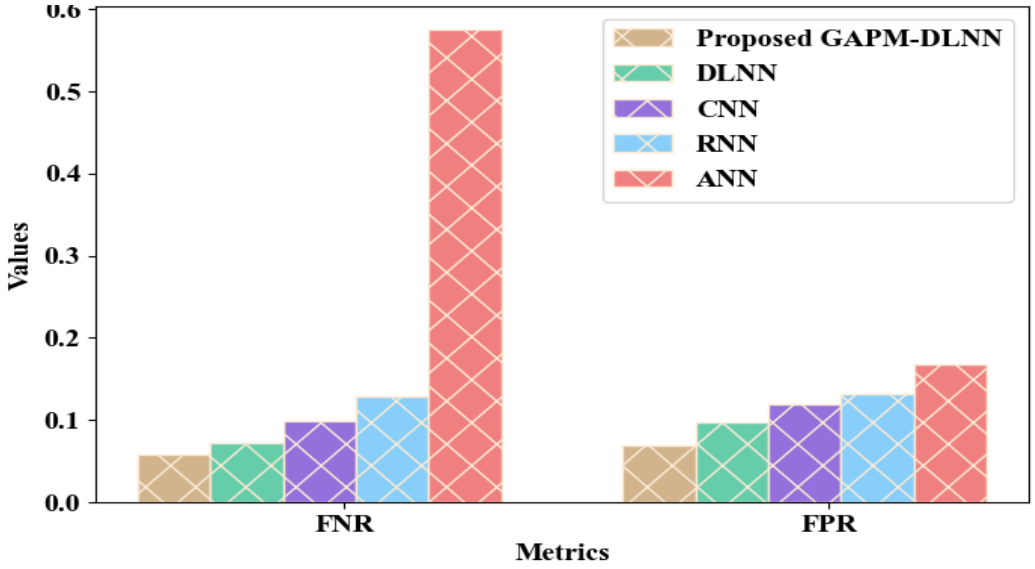
Table 1: Performance analysis of classification

Metrics	Proposed GAPM-DLNN	DLNN	CNN	RNN	ANN
Precision	98.5	96	92.5	90.8	88
Recall	97.6	95	91	90	89.3
F-measure	98.3	95.5	91.8	90.4	88.7
MCC	98.9	97	94.5	92	90.03

The classifiers' performance analysis is displayed in Table 1. For precision, recall, F-measure, as well as Mathew's Correlation Coefficient (MCC) values, GAPM-DLNN attained 98.5%, 97.6%, 98.3%, and 98.9%, respectively. The obtained values are higher than the existing methods because of the inclusion of GAPM with the DLNN.



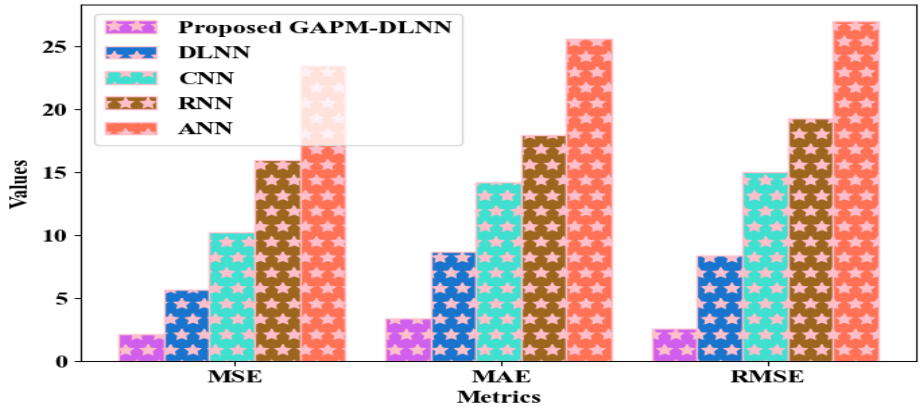
(a)



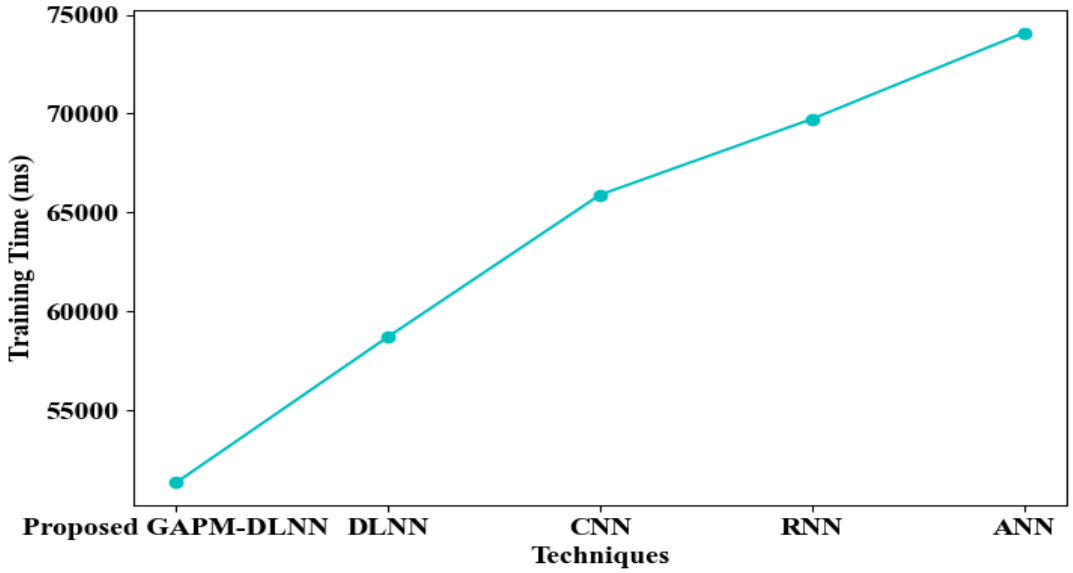
(b)

Figure 4: Graphical plot for the performance analysis of the classifier

The graphical plot for (a) accuracy, sensitivity, and specificity and (b) the False Negative Rate (FNR) together with False Positive Rate (FPR) are displayed in Figure 4. 99.1% is the proposed GAPM-DLNN's accuracy; but, the existing methods had lower performance.



(a)



(b)

Figure 5: Pictorial plot for (a) error analysis and (b) training time analysis

The error analysis in terms of Mean Square Error (MSE), Mean Absolute Error (MAE), together with Root Mean Square Error (RMSE) is given in Figure 5(a). The GAPM-DLNN's MSE value is 2.1, which is lower than the other methods. The training time of the classifiers is depicted in Figure 5(b). The GAPM-DLNN takes 51341ms training time, which is lower than the existing methods.

Table 2: Packet delivery ratio and throughput analysis

Methods/Metrics	Packet delivery ratio (%)	Throughput (%)
Proposed GAPM-DLNN	97.6	97
DLNN	89	91.2
CNN	85.8	88
RNN	83	84.6
ANN	79.7	81

The PDR and throughput for the classifier-based data transmission is depicted in Table 2. For PDR and throughput, the GAPM-DLNN attained 97.6% and 97%, which is higher than the existing methods.

(c) Performance analysis for attack mitigation

Here, the QFIS is compared with the existing FIS, Sigmoid Fuzzy Logic (SFC), Trapezoidal Fuzzy Logic (TFL), and Decision Rule (DR).

Table 3: Rule generation time analysis

Methods	Rule generation time (ms)
Proposed QFIS	809
FIS	1034
SFC	1287
TFL	1468
DR	1765

In Table 3, the rule generation time of the proposed and existing mitigation methods is evaluated. The rule generation time of the QFIS algorithm is 809ms, which is lower when analogized to the prevailing methods because the membership constructed in the proposed research is based on quantum probability.

4.4 Comparative analysis

Table 4: Comparative analysis of the proposed framework with the state-of-art works

Author's Name	Aim	Methods	Result	Demerits
(You et al., 2022)	Enhancement of packet routing	Reinforcement Learning (RL)	4.77ms average delivery time	However, the RL took more time.
(Yu et al., 2021)	Burst traffic prediction	Feedback-based Spiking Neural Network (SNN)	Above 90% accuracy	It was only supported for burst traffic and was not suitable for all types of traffic.
(Yusuf et al., 2023)	Congestion avoidance	Adaptive Path Selection Algorithm with Flow (APSAF)	Throughput of 35.6% as well as PDR of 31.7%	However, the framework wasn't supported for multi-domain.
(Isyaku et al., 2021)	Optimal route selection	Route Path Selection Optimization (RPSO)	Throughput of 55.73% and PDR of 12.5%	The premature convergence problem might affect the performance.
(Obaida& Salman, 2022)	Best path selection	Firefly optimization	Improves efficiency by managing services	The load wasn't balanced, which might affect the transmission performance.
Proposed	Attack detection with mitigation for packet transmission	GAPM-DLNN and QFIS	Accuracy of 99.1%, PDR of 97.6%, as well as throughput of 97%	The buffer overflow attack types are only detected.

The comparative analysis of the proposed and existing research w are depicted in Table 4. Here, since the proposed framework follows the IDC-based distributed packet switching with attack detection and mitigation process, the proposed framework achieves higher accuracy, throughput, and PDR.

5. CONCLUSION

Here, a GAPM-DLNN-based buffer overflow attack detection and mitigation process is proposed in this paper. The publically available dataset is taken for the performance analysis. As per the experimental analysis, the proposed GAPM-DLNN achieved a higher accuracy of 99.1%. Moreover, for rule generation, the QFIS took only 809ms time. Moreover, when analogized to the existing methods, the feature selection approach had higher fitness. The proposed methodology achieved higher performance centered on the other metrics also. But, the proposed framework only detected buffer overflow attacks.

Future recommendation: To improve performance, the proposed methodology will be enhanced by the detection of other types of attacks and advanced approaches in the future.

REFERENCES

Dataset: https://www.fit.vut.cz/person/ihomoliak/public/asnm/ASNM-CDX-2009.html

1. Abdelmoniem, A. M., &Bensaou, B. (2021). T-RACKs: A Faster Recovery Mechanism for TCP in Data Center Networks. *IEEE/ACM Transactions on Networking*, *29*(3), 1074–1087. https://doi.org/10.1109/TNET.2021.3059913
2. Adhikari, N., Logeshwaran, J., &Kiruthiga, T. (2022). The Artificially Intelligent Switching Framework for Terminal Access Provides Smart Routing in Modern Computer Networks. BOHR International Journal of Smart Computing and Information Technology, 3(1), 45-50.
3. Aladaileh, M. A., Anbar, M., Hasbullah, I. H., Chong, Y. W., &Sanjalawe, Y. K. (2020). Detection Techniques of Distributed Denial of Service Attacks on Software-Defined Networking Controller-A Review. *IEEE Access*, *8*, 143985–143995. https://doi.org/10.1109/ACCESS.2020.3013998
4. Balarezo, J. F., Wang, S., Chavez, K. G., Al-Hourani, A., Fu, J., &Sithamparanathan, K. (2020). Low-rate TCP DDoS Attack Model in the Southbound Channel of Software Defined Networks. *2020 14th International Conference on Signal Processing and Communication Systems, ICSPCS 2020 - Proceedings*, 1–10. https://doi.org/10.1109/ICSPCS50536.2020.9310040
5. Blöcher, M., Wang, L., Eugster, P., & Schmidt, M. (2021). Switches for HIRE: Resource scheduling for data center in-network computing. *International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS*, 268–285. https://doi.org/10.1145/3445814.3446760
6. Cheng, L., Wang, Y., Liu, Q., Epema, D. H. J., Liu, C., Mao, Y., & Murphy, J. (2021). Network-aware locality scheduling for distributed data operators in data centers. *IEEE Transactions on Parallel and Distributed Systems*, *32*(6), 1494–1510. https://doi.org/10.1109/TPDS.2021.3053241
7. Chiesa, M., Kamisinski, A., Rak, J., Retvari, G., &Schmid, S. (2021). A Survey of Fast-Recovery Mechanisms in Packet-Switched Networks.*IEEE Communications Surveys and Tutorials*, *23*(2), 1253–1301. https://doi.org/10.1109/COMST.2021.3063980
8. Diadamo, S., Qi, B., Miller, G., Kompella, R., &Shabani, A. (2022). Packet switching in quantum networks: A path to the quantum Internet. *Physical Review Research*, *4*(4), 1–17. https://doi.org/10.1103/PhysRevResearch.4.043064
9. Huang, J., Lyu, W., Li, W., Wang, J., & He, T. (2021).Mitigating Packet Reordering for Random Packet Spraying in Data Center Networks.*IEEE/ACM Transactions on Networking*, *29*(3), 1183–1196. https://doi.org/10.1109/TNET.2021.3056601
10. Isyaku, B., Bakar, K. A., Zahid, M. S. M., Alkhammash, E. H., Saeed, F., &Ghaleb, F. A. (2021). Route path selection optimization scheme based link quality estimation and critical switch awareness for software defined networks. *Applied Sciences (Switzerland)*, *11*(19), 1–17. https://doi.org/10.3390/app11199100
11. Kumar, V. A., & Das, D. (2021). Data sequence signal manipulation in multipath TCP (MPTCP): The vulnerability, attack and its detection. *Computers and Security*, *103*, 1–28. https://doi.org/10.1016/j.cose.2021.102180
12. Obaida, T. H., & Salman, H. A. (2022).A novel method to find the best path in SDN using firefly algorithm.*Journal of Intelligent Systems*, *31*(1), 902–914. https://doi.org/10.1515/jisys-2022-0063
13. Olmedo, G., Lara-Cueva, R., Martínez, D., & de Almeida, C. (2020). Performance analysis of a novel TCP protocol algorithm adapted to wireless networks. *Future Internet*, *12*(6), 1–17. https://doi.org/10.3390/fi12060101
14. Sahoo, K. S., Tiwary, M., Sahoo, B., Mishra, B. K., RamaSubbaReddy, S., &Luhach, A. K. (2020). RTSM: Response time optimisation during switch migration in software-defined wide area network. *IET Wireless Sensor Systems*, *10*(3), 105–111. https://doi.org/10.1049/iet-wss.2019.0125
15. Tuan, N. N., Hung, P. H., Nghia, N. D., Van Tho, N., Van Phan, T., &Thanh, N. H. (2020). A DDoS

attack mitigation scheme in ISP networks using machine learning based on SDN. *Electronics (Switzerland)*, *9*(3), 1–19. https://doi.org/10.3390/electronics9030413

16. Turcato, A. C., Dias, A. L., Sestito, G. S., Flauzino, R., Brandao, D., Sisinni, E., & Ferrari, P. (2020). Introducing a cloud based architecture for the distributed analysis of Real-Time Ethernet traffic. *2020 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2020 - Proceedings*, 235–240. https://doi.org/10.1109/MetroInd4.0IoT48571.2020.9138288
17. Wei, W., Xue, K., Han, J., Wei, D. S. L., & Hong, P. (2020).Shared Bottleneck-Based Congestion Control and Packet Scheduling for Multipath TCP.*IEEE/ACM Transactions on Networking*, *28*(2), 653–666. https://doi.org/10.1109/TNET.2020.2970032
18. You, X., Li, X., Xu, Y., Feng, H., Zhao, J., & Yan, H. (2022). Toward Packet Routing With Fully Distributed Multiagent Deep Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *52*(2), 855–868. https://doi.org/10.1109/TSMC.2020.3012832
19. Yu, A., Yang, H., Nguyen, K. K., Zhang, J., &Cheriet, M. (2021). Burst Traffic Scheduling for Hybrid E/O Switching DCN: An Error Feedback Spiking Neural Network Approach. *IEEE Transactions on Network and Service Management*, *18*(1), 882–893. https://doi.org/10.1109/TNSM.2020.3040907
20. Yusuf, M. N., Bakar, K. bin A., Isyaku, B., Osman, A. H., Nasser, M., &Elhaj, F. A. (2023). Adaptive Path Selection Algorithm with Flow Classification for Software-Defined Networks.*Mathematics*, *11*(6), 1–24. https://doi.org/10.3390/math11061404