ISSN: 2634 - 8853

Journal of Engineering and Applied Sciences Technology

SCIENTIFIC Research and Community

Review Article

Open Access

Network Packet Status-Aware and Ping-Integrated Attack Classification Along with Alert Generation Using Esw-Mlp and S3-Fuzzy

Amaresan Venkatesan

USA

ABSTRACT

Recently, there has been a rapid increase in attacks along with network data, thereby posing a significant threat to network security. During attack prediction, none of the traditional systems concentrated on packet status identification. Thus, by using Entropy Softsin Wrapper based Multi-Layer Perceptron (ESW-MLP) and Standard S Shaped Fuzzy (S3-Fuzzy), this paper proposes a packet status-aware attack prediction and ping-enabled Alert Generation (AG) in a network. Initially, the Canadian Institute for Cybersecurity Android Malware 2017 (CICAndMal2017) dataset is gathered and pre-processed. Then, the features are extracted, and optimal features are selected employing the Tent Chaotic-Chicken Swarm Optimization Algorithm (TC-CSOA). Next, the selected features are subjected to ESW-MLP, where the attack types are classified. Similarly, from the traffic dataset, the features are extracted, followed by feature selection. Thus, by using ESW-MLP, the packet status is identified. Similarly, the similarity between the features is estimated. Then, the AG is done based on S3-Fuzzy. Besides, via TC-CSOA-based load balancing, the network collision is diminished. Next, Ping-based PSI and Attack Classification (AC) are carried out on the switch, followed by AG. As per the experimental findings, the proposed approach had higher supremacy with 98.63% accuracy.

***Corresponding author**
Amaresan Venkatesan, USA.

Received: March 02, 2024; **Accepted:** March 09, 2024; **Published:** March 16, 2024

Keywords: Computer Network Analysis (CNA), Packet Status (PS), Packet Sniffing Identification (PSI), Standard S Shaped Fuzzy (S3-Fuzzy), Tent Chaotic-Chicken Swarm Optimization Algorithm (TC-CSOA), Entropy Softsin Wrapper based Multi-Layer Perceptron (ESW-MLP), Deep Learning (DL)

Introduction

Recently, a vital role has been played by communication technology in the life of modern organizations. Wide amounts of data are transferred by online services over communication networks in different formats, including network packets [1]. The information transmitted over a network may be vulnerable to numerous attacks [2]. Thus, network security is essential with the wide usage of networks. Thus, to detect internet traffic and anomalies, researchers conducted many studies, thus ensuring both security and quality of service [3,4].

Usually, data collection, feature extraction, and classification are encompassed in the automatic detection model. From the gathered dataset, the packet features, including length, time, and order are extracted, followed by AC [5]. To detect attacks in cloud networks, existing machine learning techniques like Random Forest (RF) and Logistic Regression (LR) were established [6,7]. Similarly, to perform network AC, DL algorithms like Deep Neural Network (DNN), Multi-Layer Perceptron (MLP), as well as Convolutional Neural Network (CNN) were introduced [8,9].

Problem statement

The drawbacks of the conventional algorithms are listed below,

- None of the traditional schemes concentrated on network pattern identification regarding the packet status.
- Raza only focused on attack-type classification during network traffic but failed to generate the alert when the network was monitored by the attacker [10].
- Hosseini & Zade had a network collision because it failed to balance the incoming and outgoing packet transmission loads [11].
- The prevailing approaches were insignificant owing to the limited number of features.

Objectives

The proposed model's major objectives are given as follows,

- The proposed ESW-MLP is established to perform packet status identification.
- A novel TC-CSOA-Ping is incorporated to identify the packet sniffing and generate the alert based on S3-Fuzzy.
- The load balancing is done by using the proposed TC-CSOA, thus reducing the network collision.
- In the proposed work, more features are extracted from both the CSV and PCAP files.

The remaining part is assembled as follows: the related models are discovered in Section 2, the proposed methodology is derived in Section 3, the performance analysis is depicted in Section 4, and the paper is concluded with future scope in Section 5.

Literature Survey

Oliveira presented an intelligent cyber-AC for a network based on an intrusion detection system [12]. Here, to perform AC, techniques like MLP, LSTM, and RF were established. As per the outcome, the developed model was proved to be a reliable framework. Yet, owing to the limited number of features, this framework was inefficient [13]. deployed DL-based effective Distributed Denial of Service (DDoS) attack detection in the network using the CTU-13 dataset. To classify the DDos attack with high accuracy, the MLP was introduced. But, due to the large data, it had high computational complexity. presented hybrid evolutionary algorithms-based network attack detection [11]. To perform feature extraction and attack detection, techniques like hybrid SVM and hybrid ANN were developed. In AC, this approach had impressive outcomes. However, this framework failed to perform load balancing, which caused network collision. Rios recognized an ensemble ML-based reduction-of-quality DDoS attack detection [14]. To detect the DDoS attack significantly, techniques like fuzzy logic, MLP, as well as Euclidean distance were incorporated. However, due to the increased execution time, it had time complexity. Raza established an ML-based network attack detection using optimal class probability features [10]. Here, to classify the network attacks, class probability RF was used. Unauthorized access was proficiently prevented by this technique. Nevertheless, during a network attack, it failed to generate an alert to the user.

Proposed Methodology for Packet Status-Aware Attack Classification

By using ESW-MLP and S3-Fuzzy, this paper implements a packet status-aware and ping-integrated attack prediction and AG model. In Figure 1, the proposed work's architecture is shown.

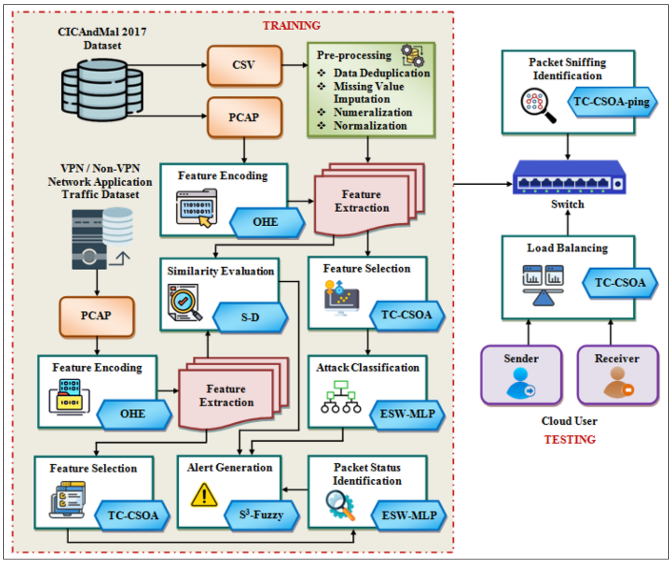


Figure 1: The diagrammatic format of the proposed approach

CICAndMal2017 dataset

Primarily, the CICAndMal2017 dataset is gathered, and from the dataset, the Comma-Separated Value (CSV) and Packet Capture (PCAP) files are extracted.

Pre-processing

Here, to enhance the data quality, the CSV data is pre-processed. Initially, by eliminating the redundant information from the dataset, the data deduplication is done. Then, by using the mean value, the missing values in the deduplicated data are replaced. Next, for representing the data in a numerical form , (Num_{data}) the numeralization is done. Lastly, to transform the feature values into 0s and 1s, the min-max normalization is applied. The normalized data (ηorm) is shown as,

$$\eta orm = \frac{Num_{data} - \min(Num_{data})}{\max(Num_{data}) - \min(Num_{data})} \tag{1}$$

Therefore, the pre-processed data is portrayed as $(\wp^\circ)$

Feature encoding

Furthermore, the PCAP data is fed into the feature encoding, where the features in the dataset are converted into numerical values based on One-Hot Encoding (OHE). The encoded features are stated as (E°).

Feature extraction

Then, $\wp^\circ$ and E° are subjected to feature extraction. Also, from the $\wp^\circ$, the CSV features like source IP, source port, and protocol are extracted. Here, from the E°, the PCAP features, such as average payload, maximum payload, and maximum-delta time are extracted. Lastly, the $d = 1$ *to* Dnumber of extracted features $(\chi_d)$is illustrated as,

$$\chi_d = (\chi_1, \chi_2, \ldots\ldots \chi_D) \tag{2}$$

Afterward, from the χ_d, the optimal features are chosen.

Feature selection

Here, to identify the ideal features from the χ_d, the proposed TC-CSOA is established. The CSO is a bio-inspired algorithm, which stimulates the intelligent behavior of chicken swarms. By balancing exploration and exploitation, the CSO effectively provides a precise solution. But, due to the randomness of the exploration, it had pre-mature convergence issues. Thus, to upgrade the convergence rate, the tent chaotic is incorporated. Here, the extracted features (members) are assumed as the chicken swarms. Chiefly, the population of the chicken swarms is initialized as,

$$\chi_{d,e} = lw_e + rd_{d,e} \times (up_e - lw_e) \tag{3}$$

Here, e = 1 *to* E signifies the number of problem variables, lw_e and up_e specify the lower bound and upper bound, respectively, and $rd_{d,e}$ characterizes the random value. Subsequently, by considering the maximum classification accuracy (Δcc) , the fitness (Fit) is estimated.

$$Fit = \max(\Delta cc) \tag{4}$$

Next, the population of the chicken is divided into roosters, hens, and chicks. Each of the group follows their own strategies. The individual with the best fitness value is assumed as a rooster. Similarly, the worst individuals are mentioned as chickens, while the remaining members are called hens. Now, by using tent chaotic based on the rooster's foraging strategy, the position of the members is updated,

$$\Im d = \begin{cases} \upsilon \chi_{d,e}, & 0 \le \chi_{d,e} \le 1/2 \\ \upsilon(1 - \chi_{d,e}), & 1/2 \le \chi_{d,e} \le 1 \end{cases} \tag{5}$$

$$\chi_{d,e}^{R\Omega} = \chi_{d,e} * \left(1 + \Im d\left(0, \ell^2\right)\right) \tag{6}$$

$$\ell^2 = \begin{cases} 1, & Fit_a \le Fit_b \\ \exp\left(\dfrac{\left(Fit_b - Fit_a\right)}{Fit_a + \mu'}\right), & Fit_a > Fit_b \end{cases}; \qquad b \ne a \tag{7}$$

Where, $\chi_{d,e}^{R\Omega}$ signifies the new position of the roosters, $\Im d$ shows the tent chaotic matrix, ℓ^2 describes the variance, μ' signifies the normalized number, a, υ and b demonstrates the parameters. The hen's foraging process is as follows,

$$\chi_{d,e}^{H\Omega} = \chi_{d,e} + Y_1 * rd * \left(\chi_{d,e}^{rdm} - \chi_{d,e}\right) + Y_2 \cdot rd \cdot \left(\chi_{d,e}^{rdm} - \chi_{d,e}\right) \tag{8}$$

$$Y_1 = \exp\left(Fit_a - Fit_{rd}\right) \Big/ \left(abs\left(F_a\right) + \mu'\right) \tag{9}$$

$$Y_2 = \exp\left(Fit_{rd} - Fit_a\right) \tag{10}$$

Where, $\chi_{d,e}^{H\Omega}$ signifies the new position of the hen, $\chi_{d,e}^{rdm}$ specifies the randomly selected chicken, and Fit_{rd} represents the fitness value of the $\chi_{d,e}^{rdm}$. Likewise, the chicks' foraging process is given as,

$$\chi_{d,e}^{C\Omega} = \chi_{d,e} + fc * \left(\chi_{d,e}^{mother} - \chi_{d,e}\right) \tag{11}$$

Here, fc designates the follow coefficient, $\chi_{d,e}^{mother}$ portrays the chick's mother, and $\chi_{d,e}^{C\Omega}$ embodies the new position of the chick. Finally, the g=1,2...G number of selected optimal features (λ_g) is mentioned as,

$$\lambda_g = \left(\lambda_1, \lambda_2, \ldots\ldots \lambda_G\right) \tag{12}$$

The pseudo-code of the proposed TC-CSOA is given further,

```
**Input:** Extracted features $\chi_d$
**Output:** Optimal features $\lambda_g$
**Begin**
    **Initialize** $\chi_d, \lambda_g, fc$ and $\chi_{d,e}^{R\Omega}$
    **For** 1 to each $\chi_d$ do,
        **Perform** population initialization $\chi_{d,e} = lw_e + rd_{d,e} \times (up_e - lw_e)$
        **Calculate** fitness $Fit = \max(\Delta cc)$
        #roosting foraging
        **Determine** $\chi_{d,e}^{R\Omega} = \chi_{d,e} * (1 + \Im d(0, \ell^2))$
        #hen foraging
        **Evaluate**
        $\chi_{d,e}^{H\Omega} = \chi_{d,e} + Y_1 * rd * (\chi_{d,e}^{rdm} - \chi_{d,e}) + Y_2 \cdot rd \cdot (\chi_{d,e}^{rdm} - \chi_{d,e})$
        #chick foraging
        **Apply** $\chi_{d,e}^{C\Omega} = \chi_{d,e} + fc * (\chi_{d,e}^{mother} - \chi_{d,e})$
    **End For**
    **Return** $\lambda g = (\lambda_1, \lambda_2, \ldots. \lambda_G)$.
**End**
```

The AC is done as follows after feature selection.

Attack classification

Then, λ_g is subjected to the ESW-MLP, which classifies whether the packet is normal or attacked. The complex relationships are captured by the MLP, thus increasing the model's efficacy. However, it had overfitting and vanishing gradient issues. So, in the proposed approach, to improve the classifier performance, the softsin activation function and entropy-wrapper regularization are employed. In Figure 2, the architecture of the ESW-MLP is illustrated.

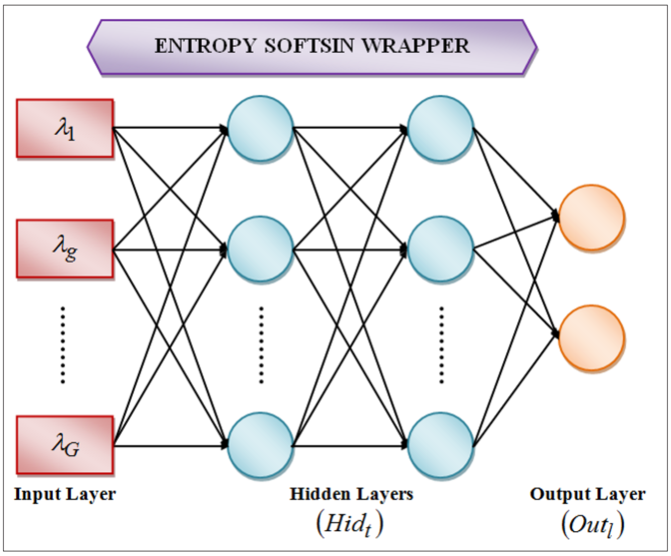


Figure 2: The structure of the proposed ESW-MLP

Initially, λ_g is captured by the input layer; then, transmits the input to the succeeding network layers. To diminish the overfitting, the proposed work introduces the entropy-wrapper regularization (Ew_{reg}).

$$Ew_{reg} = \min\left(\sum_{g=1}^{G}\left(-\log\lambda_g - \sum_{g=0}^{G}\log at \cdot \lambda_g\right)^2 + \Delta d \sum_{g=1}^{G}\log\lambda_g^{\,2}\right) \tag{13}$$

Here, at and Δd designate the constant and regularization parameters, respectively. The regularized input is exhibited as $\left(R_{\lambda_g}\right)$. To upgrade the training efficiency of the neurons, the softsin activation function is employed, which is given as,

$$\delta ft_{act} = \frac{\sin\left(\exp^{R_{\lambda_g}}\right)}{\sum_{g=1}^{G} \exp^{R_{\lambda_g}}} \tag{14}$$

The imput is processed by $t = 1 to\ T$ number of hidden layers (Hid_t) by sharing their weight (ςt) and bias (βs) to produce the solution.

$$Hid_t = \varsigma t \times \delta ft_{act}\left(R_{\lambda_g}\right) + \beta s \tag{15}$$

Lastly, the output layer ($Outl$) classifies the data into normal (no), adware (ad_{ware}), ransomware (rs_{ware}), scareware (sc_{ware}), and SMS malware (sms_{ware}) .

$$Out_l = \left| no, ad_{ware}, rs_{ware}, sc_{ware}, sms_{ware} \right| \tag{16}$$

The pseudo-code of the proposed ESW-MLP is illustrated below,

```
Input: Optimal features $\lambda_g$

Output: Classified data $Out_l$

Begin

Initialize $\lambda_g$, $\delta ft_{act}$, $Hid_t$ and $Out_l$

For 1 to each $\lambda_g$ do,

        Execute input layer
        Apply entropy wrapper regularization
        Determine softsin activation
        Process hidden layers

End For

Return $Out_l = \left| no, ad_{ware}, rs_{ware}, sc_{ware}, sms_{ware} \right|$

End
```

The proposed ESW-MLP effectively predicts the network attack.

VPN and Non-VPN network application traffic dataset

Similarly, to predict the packet status whether it is encrypted or unencrypted, the VPN along with Non-VPN network Application Traffic (VNAT) datasets are collected. Now, from the VNAT dataset, the PCAP files are extracted.

Feature encoding

Primarily, the PCAP files are inputted to the feature encoding, where the features of the dataset are converted into numerical values (PΩ).

Feature extraction

Likewise, from the PΩ, the PCAP features like average payload, maximum payload, minimum-delta time, and maximum-delta time are extracted. The extracted features are stated as (ζ_{ext}).

Feature selection

Then, by using the TC-CSOA, the optimal features are identified, which is already explained in Section 3.1.4. Therefore, the selected features are inputted to the proposed ESW-MLP.

Packet status identification

Later, by using the proposed ESW-MLP, the packet status, including encrypted (Ec) or unencrypted (Uc) is identified from δel, which is previously derived in Section 3.1.5. The identified packet status (Pkt) is exhibited as,

$$Pkt = \left(Ec, Uc\right) \tag{17}$$

Similarity evaluation

Besides, based on the Sorensen-Dice (S-D) coefficient technique, the similarity between χ_d and ζ_{ext} is evaluated.

$$Sim = \frac{2\left|\chi_d \cap \zeta_{ext}\right|}{\left|\chi_d\right| + \left|\zeta_{ext}\right|} \tag{18}$$

To show the relationship between the individual features, the similarity analysis is done, thus aiding in improving the packet analysis. The similarity score is signified as (Sim).

Alert generation

Then, based on S³-Fuzzy, Out_l, Pkt and Sim is subjected to an AG phase. The fuzzy logic provides optimal solutions to complex issues. But, due to the basic membership function, it had tuning difficulty. So, employed to improve the fuzziness results, the Standard S-shaped membership function is. Primarily, the fuzzy rules $\left(Fuz_{\nabla}\right)$ are framed as,

$$Fuz_{\nabla} = \left\{ IF \begin{pmatrix} Pkt == encrypted\ or\ unencrypted\ \&\ \&Out_l == attack name \\ \&\ \&Sim == highsimilar\ features\ and\ attack name \end{pmatrix} THEN\ alert\ users \right. \tag{19}$$

By considering the packet status, such as encrypted or unencrypted, detected attack name, and high similarity with the attack name, the alert is sent to the users. The crisp data is transformed into fuzzy data in the fuzzification unit. Then, the proposed method introduces the S³ membership function is illustrated as,

$$S3_{mem} = \begin{cases} 0, & IF(\psi \le mn) \\ 2\left(\dfrac{(\psi - mn)}{sd(j - mn)}\right)^2, & IF(mn \le \psi \le k) \\ 1 - 2\left(\dfrac{(\psi - mn)}{sd(j - mn)}\right)^2, & IF(k < \psi \le j) \\ 1, & IF(\psi \ge j) \end{cases} \tag{20}$$

$$\psi = \left(Out_l, Pkt, Sim\right) \tag{21}$$

Here, mn and sd designate the mean and standard deviation, respectively, j and k portray the parameters. Also, to perform fuzzy operations, the decision-making operator is used. Lastly, in the defuzzification unit, the fuzzy data is converted into crisp data.

Packet sniffing identification

Also, by using the TC-CSOA-ping, packet sniffing is identified in the proposed work to ensure high security. The task of predicting packet data transmitted via the network is termed Packet sniffing. The sniffing in the network is affected by ping. But, due to the increased number of load, it had time complexity. So, to minimize the latency, load balancing is coupled with the ping scheme. In Section 3.1.4, the TC-CSOA-based load balancing is derived. The above-mentioned processes depict the training of the AC and PSI models.

Cloud users

To continuously transmit the packets, the cloud users connect the network (switch) in real-time. The TC-CSOA is established to balance the transmitted packets to reduce the network collision. In 3.1.4, the derivation of the proposed TC-CSOA is already explained. By considering the fitness as minimum response time, the load balancing is done. Then, the PSI and AC are done on the switch, which alerts the users.

Results and Discussion

Here, to prove the model's trustworthiness, the performance assessment is done and the PYTHON platform is used to perform system implementation.

Dataset description

By using the two datasets, namely CICAndMal2017 and VNAT dataset, which are used to perform AC and packet status identification, the proposed work was assessed.

Performance analysis

The research methodology's performance is validated by analogizing it with traditional algorithms regarding quality factors.

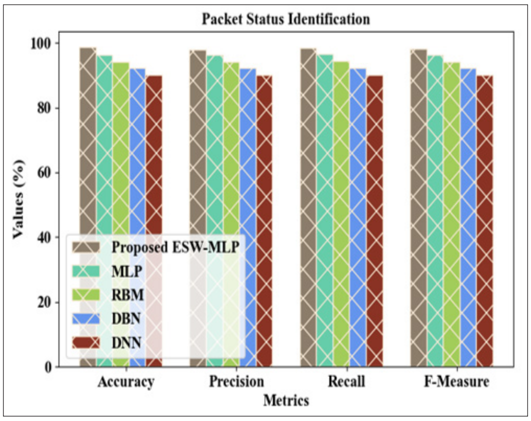


Figure 3: Performance analysis for packet status identification

Here, to prove the efficiency of the proposed ESW-MLP in packet status identification, the performance analysis is done. Regarding accuracy, precision, recall, as well as f-measure, the performance of the proposed ESW-MLP and existing algorithms like MLP, Restricted Boltzmann Machine (RBM), Deep Belief Neural network (DBN), and DNN is validated in Figure 3. For accuracy, precision, recall, and f-measure, the ESW-MLP achieved 98.63%, 98.02%, 98.32%, and 98.20%; but, the prevailing methodologies attained 93.23%, 93.27%, 93.33%, and 93.23%, respectively. Hence, due to the entropy regularization, ESW-MLP achieved higher dominance.

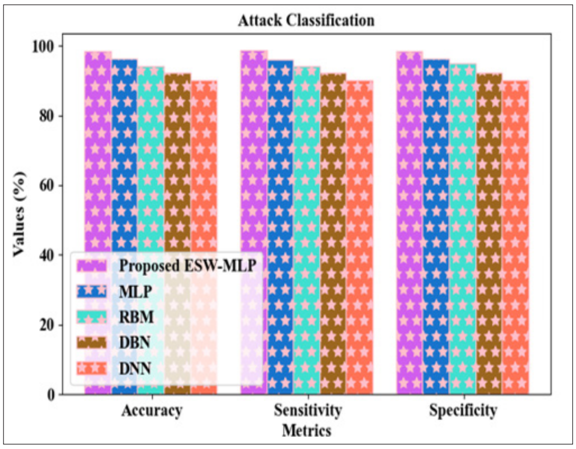


Figure 4: Performance evaluation for AC

Regarding accuracy, sensitivity, and specificity, the performance of the proposed ESW-MLP as well as prevailing methods are evaluated in Figure 4. To showcase the model's reliability in AC, a performance analysis is done. For accuracy, sensitivity, and specificity, the ESW-MLP acquired 98.36%, 98.62%, and 98.32%; but, the prevailing methods had limited outcomes due to the ineffective activation function. Due to the softsin activation function, the proposed work had better performance.

Table 1: PPV analysis

Methods	PPV (%)
Proposed ESW-MLP	98.6023
MLP	96.2547
RBM	94.1254
DBN	92.3247
DNN	90.1021

Table 2: NPV evaluation

Methods	NPV (%)
Proposed ESW-MLP	98.6235
MLP	96.3214
RBM	94.5287
DBN	92.3365
DNN	90.1247

Regarding Positive Predictive Value (PPV) together with Negative Predictive Value (NPV), the performance of the proposed ESW-MLP and traditional approaches are compared in Tables 1 and 2. For PPV and NPV, the ESW-MLP attained 98.60% and 98.62%; whilst, the existing methods obtained 93.20% and 93.32%, respectively. Thus, the proposed method had higher supremacy in AC.

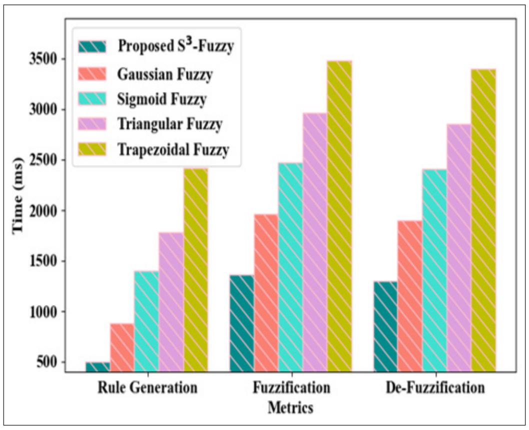
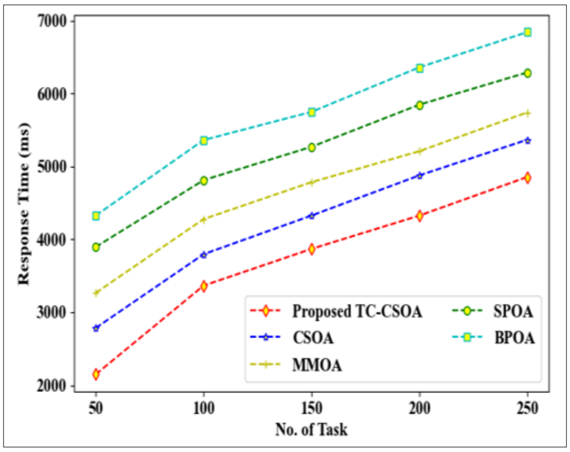
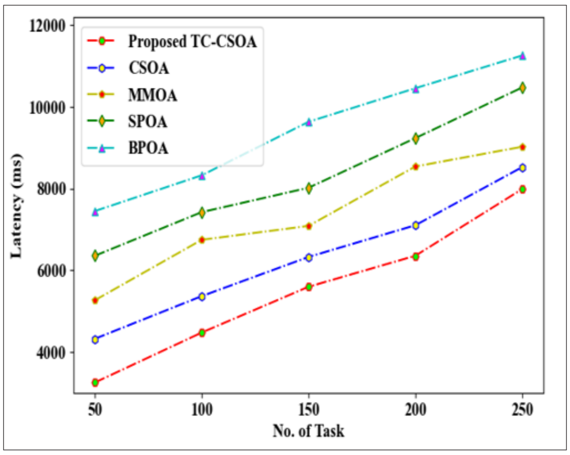


Figure 5: Performance validation of the proposed S^3-Fuzzy

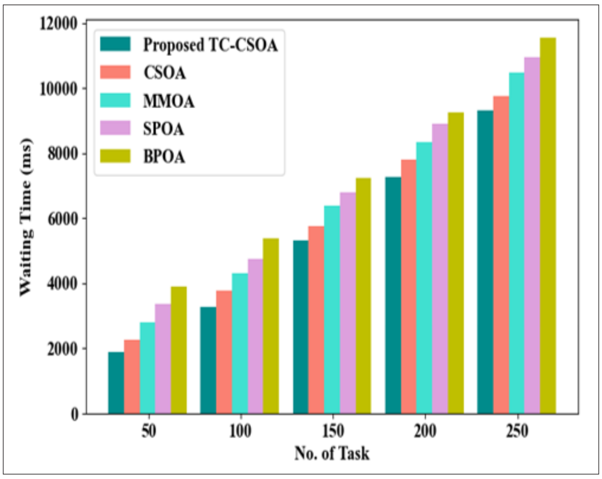
Regarding Fuzzification Time (FT), Defuzzification Time (DT), and Rule Generation Time (RGT), the performance of the proposed S^3-Fuzzy and traditional algorithms like Gaussian fuzzy, sigmoid fuzzy, triangular fuzzy, and trapezoidal fuzzy are validated in Figure 5. The presence of the S^3 membership function helps to improve the model's performance. For FT, DT, and RGT, the proposed S^3-Fuzzy obtained 1365ms, 1298ms, and 498ms; but, the existing approaches achieved 2720ms, 2641ms, and 1618ms, respectively. Thus, the proposed methodology had low time complexity.



(a)



(b)



(c)

Figure 6: Performance analysis of the proposed TC-CSOA regarding (a) response time, (b) latency, and (c) waiting time

In Figure 6, the performance of the proposed TC-CSOA and prevailing techniques like CSOA, Mosquitoes Mating Optimization Algorithm (MMOA), Social Spider Optimization Algorithm (SSOA), and Bee Pollinator Optimization Algorithm (BPOA) are validated. Here, to improve the optimization quality, the tent chaotic is employed. For response time, latency, and waiting time, TC-CSOA obtained 2148ms, 3256ms, and 1875ms; but, the existing works attained poor performance. Thus, in load balancing, the proposed work had better efficiency.

Comparative evaluation

The proposed approach's efficiency is validated by analogizing it with associated frameworks.

Table 3: Comparative analysis

Author's name	Technique	Accuracy (%)	Specificity (%)
Proposed work	ESW-MLP	98.36	98.32
Elhefnawy [15].	Hybrid Nested Genetic-Fuzzy Algorithm (HNGFA)	94.24	-
Wei et al.	Auto encoder-MLP	97.49	93.59
Lee & Singh [16].	Switch tree-RF	-	97.84
Kim [17].	CNN	95.25	91
Kushwah & Ranga [19].	Self-adaptive evolutionary extreme learning machine	93.71	96.83

The comparative analysis of the proposed work as well as related models is depicted in Table 3. The attack types are proficiently predicted by the proposed entropy softsin wrapper-based MLP. For accuracy and specificity, the ESW-MLP obtained 98.36% and 98.32%. Similarly, to perform AC, the existing models introduce techniques like HNGFA, auto encoder-MLP, RF, and CNN. But, due to poor regularization, the traditional methods were ineffective. Hence, the proposed work depicted superior performance.

Conclusion

Here, by using ESW-MLP and S3-Fuzzy, this paper proposed a packet status-aware attack prediction and ping-enabled AG in a network. The AC and AG are significantly performed by the proposed ESW-MLP and S3-Fuzzy. Moreover, to identify the packet sniffing, the TC-CSOA-ping was employed, thus elevating the model's consistency. As per the experimental findings, in AC and packet status identification, the proposed ESW-MLP achieved an accuracy of 98.36% and 98.63%, respectively. Similarly, the proposed S3-Fuzzy obtained an RGT of 498ms, showing the low time complexity. Lastly, the prevailing works were surpassed by the proposed methodology in CNA. But, the proposed work only focused on PSI.

Future scope

The types of packet sniffing will be identified and prevention measures will be recommended in the future.

References

1. Sikos LF (2020) Packet analysis for network forensics: A comprehensive survey. Forensic Science International: Digital Investigation 32: 1-12.
2. Thakkar A, Lohiya R (2020) Attack classification using feature selection techniques: a comparative study. Journal of Ambient Intelligence and Humanized Computing 12: 1-18.
3. Salman O, Elhajj IH, Kayssi A, Chehab A (2020) A review on machine learning–based approaches for Internet traffic classification. Annals of Telecommunications 75: 1-38.
4. Song W, Beshley M, Przystupa K, Beshley H, Kochan O, et al., (2020) A software deep packet inspection system for network traffic analysis and anomaly detection. Sensors (Switzerland) 20: 1-41.
5. Shen M, Liu Y, Zhu L, Xu K, Du X, et al., (2020) Optimizing feature selection for efficient encrypted traffic classification: A systematic approach. IEEE Network 34: 20-27.
6. Alduailij M, Khan QW, Tahir M, Sardaraz M, Alduailij M, et al., (2022) Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method. Symmetry 14:1-15.
7. Dhanya KA, Vajipayajula S, Srinivasan K, Tibrewal A, Kumar TS, et al., (2023) Detection of Network Attacks using Machine Learning and Deep Learning Models. Procedia Computer Science 218: 57-66.
8. Krupski J, Graniszewski W, Iwanowski M (2021) Data transformation schemes for cnn-based network traffic analysis: A survey. Electronics (Switzerland) 10:1-35.
9. Azab A, Khasawneh M, Alrabaee S, Choo KKR, Sarsour M (2022) Network traffic classification: Techniques, datasets, and challenges. Digital Communications and Networks 1-17.
10. Raza A, Munir K, Almutairi MS, Sehar R (2023) Novel Class Probability Features for Optimizing Network Attack Detection With Machine Learning. IEEE Access 11:98685-98694.
11. Hosseini S, Zade BMH (2020) New hybrid method for attack detection using combination of evolutionary algorithms, SVM, and ANN. Computer Networks 173: 1-40.
12. Oliveira N, Praca I, Maia E, Sousa O (2021) Intelligent cyber attack detection and classification for network-based intrusion detection systems. Applied Sciences (Switzerland), 11: 1-21.
13. Ahmed S, Khan ZA, Mohsin SM, Latif S, Aslam S, et al., (2023) Effective and Efficient DDoS Attack Detection Using Deep Learning Algorithm, Multi-Layer Perceptron. Future Internet 15:1-24.
14. Rios Vde M, Inacio PRM, Magoni D, Freire MM (2021) Detection of reduction-of-quality DDoS attacks using Fuzzy Logic and machine learning algorithms. Computer Networks 186:1-18.
15. Elhefnawy R, Abounaser H, Badr A (2020) A hybrid nested genetic-fuzzy algorithm framework for intrusion detection and attacks. IEEE Access 8: 98218-98233.
16. Lee JH, Singh K (2020) SwitchTree: in-network computing and traffic analyses with Random Forests. Neural Computing and Applications 2:1-12.
17. Kim J, Kim J, Kim H, Shim M, Choi E (2020) CNN-based network intrusion detection against denial-of-service attacks. Electronics (Switzerland) 9:1-21.
18. Kushwah GS, Ranga V (2021) Optimized extreme learning machine for detecting DDoS attacks in cloud computing. Computers and Security 105:1-21.
19. Wei Y, Jang-Jaccard J, Sabrina F, Singh A, Xu W, et al., (2021) AE-MLP: A Hybrid Deep Learning Approach for DDoS Detection and Classification. IEEE Access 9:146810-146821.