International Journal of Advanced Trends in Engineering and Management
(ISSN 2583-7052)

Vol. V, No. 05, pp. 20-29, May

Intelligent Web Attack Classification Using a Hybrid Triple Attention–Dual Mamba Attention Network with Dandelion Optimization

[3]Ramya R R

Assistant Professor, Department of Computer Science and Engineering,
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology,
Avadi, Tamil Nadu, India.

ABSTRACT: Web attack Classification is critical for stopping attacks early, which helps prevent data loss, minimize operational downtime, and avoid paying expensive ransoms. In this paper, Hybrid Triple Attention-Dual Mamba Attention Network (HTA-DMAN) is proposed with Dandelion Optimization (DLO) for web attack classification. Data preprocessing turns messy raw data into a clean, structured format ready for analysis. Feature engineering focuses on extracting relevant features and scaling data, leading to the application of a classification model, specifically an HTA-DMAN-DLO approach. HTA-DMAN operates through coordinated functional stages to classify complex inputs like web attack and DLO optimization is used to tune hyperparameters of classifier. The performance analysis is carried out using python software and assesses the model's performance using key metrics such as precision, recall, F1-score, confusion matrix, and ROC-AUC. The proposed HTA-DMAN-DLO uses two classes, they are normal and attack. The proposed HTA-DMAN-DLO achieves high accuracy of 96% and better convergence speed. The proposed method outperforms conventional techniques, contributing a robust and reliable tool for web attack classification.

Keywords: Hybrid Triple Attention-Dual Mamba Attention Network (HTA-DMAN), Dandelion Optimization (DLO), Web attack Classification, convergence speed.

1. Introduction

As the rate of Internet usage has increased, web applications have emerged as the most popular Internet applications. By utilizing virtualization or a business platform, web apps assist firms in boosting profitability and enhancing business operations, such as in the supply chain [1]. The ability to stay connected on the Internet has revolutionized business life [2, 3]. It enables users to communicate with one another anywhere they choose. The internet is necessary for many of the everyday business operations that your organization conducts. It serves as both a data repository and the location of cloud-based digital storage. Customers' freely submitted information is stored in content management systems, shopping carts, login areas, and inquiry and submission forms. Every harmful attack on your website is unique, and since there are so many distinct kinds of attacks, it could seem hard to protect your website from them all [4,5].

One of the most dangerous and damaging illegal acts that is becoming more and more common on the internet is phishing [6]. People who use the internet to access services have been swiftly targeted by phishing attacks for the past few years [7]. Conventional web attack classification relies on static, signature-based rules and rigid taxonomies. This traditional approach struggles with modern threats because it lacks execution context, fails to adapt to polymorphic payloads, and results in high false-positive rates. Deep

learning combined with optimization is critical for web attack classification because it allows security systems to automatically learn complex, non-linear patterns from massive volumes of web traffic while minimizing false positives and latency. While deep learning models natively eliminate the need for manual feature engineering, optimization techniques ensure these networks remain fast, accurate, and lightweight enough to stop evolving web threats in real time.

Table 1: *Comparison of Conventional Methods*

Method	Convergence speed	Reliability	Response time
Transformer-Deep Neural Network (DNN) [8]	Low	High	High
Convolutional Neural Network-Long Short -Term Memory (CNN-LSTM) [9]	Low	Low	High
Galatic Swarm Optimization based Deep auto encoder (GSO-DAE)	High	Low	Low
HTA-DMAN-DLO	High	High	Low

Table 1 displays the evaluation of conventional methods such as Transformer-DNN, CNN-LSTM and GSO-DAE. These techniques have some disadvantages like low convergence speed, less reliability and High response time. To avoid these issues, an HTA-DMAN-DLO is proposed for web attack classification. The contribution of the paper is as follows,

- HTA-DMAN is proposed to achieve high-accuracy, real-time detection of complex web threats (such as SQL injection or XSS) by capturing fine-grained token sequences and long-range contextual dependencies without the heavy computational burden of traditional transformer.
- DLO is proposed to automatically tune the hyper parameters of HTA-DMAN, which improves model accuracy and generalization while avoiding poor local minimum solutions.

2. Recent Works

A hybrid transformer with DNN have been presented by [8], which was utilised to identify and classification of attacks in intrusion detection systems. This method has high computational complexity, massive data hunger, and vulnerability to false alarms due to class imbalance. An IDS architecture based on CNN-LSTM have been developed in [9]. A synthetic data generation method is utilised to overcome the unnecessary statistics issues. The suggested hybrid model with the balance technique has a high computational cost, heavy memory consumption, vulnerability to distribution shifts, and difficulty with extreme class imbalances in network data, but it can categorize the circulation into normal class and attack class with realistic accuracy. [10] creates a GSO-DAE Model in the IoT Environment that focuses on identifying assaults in the IoT environment. DAE is used as a classifier model in the proposed GSO-DAE technique to accurately identify assaults in an IoT context. This technique suffers from high computational complexity, a severe tendency to overfit noisy data, and difficult optimization. To overcome these drawbacks, an HTA-DMAN-DLO is proposed in this paper for web attack classification.

3. Proposed Work Explanation

An HTA-DMAN-DLO for web attack detection's proposed block diagram is represented in figure 1. The process begins with Input Data, which features a security-themed icon suggesting the handling of sensitive or potentially compromised threat data. This raw data moves into the Data Preprocessing stage, where it undergoes Data Cleaning and Handling Missing Value protocols to ensure data integrity. Next, the pipeline enters the Feature Engineering phase, which applies Normalization to scale and standardize the features for optimal model performance. The core

of the architecture is the Classification block, which implements a specialized deep learning model called the Hybrid Triple Attention-Dual Mamba Attention Network. Inside this network, the initial Input passes through a Stem Conv Block to extract joint Spatial-Spectral Features. These features are flattened and converted via a Linear Projection of Flattening Pixels, after which Position Embedding (PE) is integrated through matrix addition. The embedded features then traverse an Attention-Bidirectional Mamba Block that concurrently processes the data through a parallel Bidirectional Mamba Block and a Self-Attention Block. The outputs of these parallel components are integrated by a Conv Fusion Module before being passed to the final Classifier.

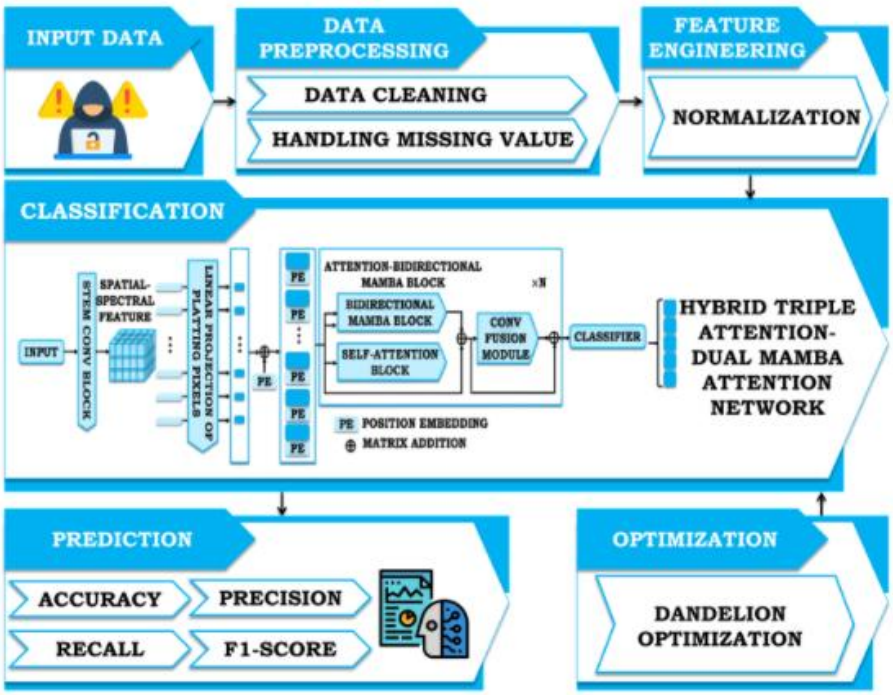

Figure 1: *Proposed block diagram of HTA-DMAN with DLO for web attack classification*

Then, the classifier's outputs are fed into the Prediction stage, where model performance is evaluated using standard metrics: Accuracy, Precision, Recall, and F1-Score. To continuously improve the system, a feedback loop connects this evaluation to an Optimization stage powered by the Dandelion Optimization algorithm, which tunes the network parameters to achieve higher accuracy.

3.1 Modelling of Data preprocessing

In data preprocessing, the unstructured data is transformed into clean and standardized formats, which is suitable for DL. By enhancing data quality, reducing noise through techniques like resizing, normalization, and filtering, data preprocessing improves system performance. To improve image quality for rendering, these methods adjust pixel brightness and contrast

$$g(x) = \alpha f(x) + \beta \quad (1)$$

To individual pixel values, essential for displaying, capturing, or printing images correctly, gamma Correction performs non-linear adjustments.

$$V_{out} = AV_{in}^{\gamma} \quad (2)$$

Where V_{in} is input pixel value (0-1), V_{out} is output, and γ is the gamma factor. COA-SRP turn cleaned data into the shortest, fastest, or cheapest paths while avoiding real-world roadblocks.

3.2 Modelling of Feature engineering

By scaling the features to a common range, normalization keeps characteristics with high

values from having an undue impact on the model. There are two primary methods:

The equation for a single data point x_i is:

$$x_{normalized} = \frac{x_i - \min(X_{train})}{\max(X_{train}) - \min(X_{train})} \quad (3)$$

Crucially, the min and max values are calculated only from the training data (X_{train}). The same parameters are then used to transform the test data.

The standardization method centers the data around a mean of zero with a standard deviation of one. The equation for a single data point x_i is:

$$x_{standardized} = \frac{x_i - \mu_{train}}{\sigma_{train}} \quad (4)$$

Here, μ_{train} is the mean and σ_{train} is the standard deviation, calculated only from the training data.

3.3 Mathematical modelling of DLO

DLO is a nature-inspired meta-heuristic algorithm mimicking seed dispersal by wind. The step-by-step process involves initialization, an ascent phase (global exploration via wind velocity), a descent phase (Brownian motion localization), and a landing phase (Levy flight exploitation). Flow chart of DLO is represented in figure 2.

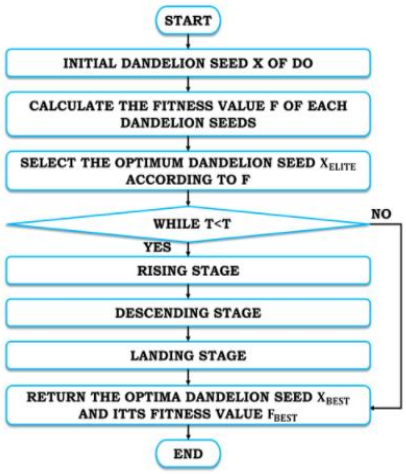

Figure 2: *Flow Chart of DLO*

3.3.1 Initialization Phase

Set population size (N), maximum iterations (T), and problem bounds $(U_B - L_B)$ and generate initial dandelion seed positions randomly within bounds:

$$D_i = rand \times (U_B - L_B) + L_B \quad (5)$$

Evaluate fitness for each individual and find the current elite/best position X_{elite}.

3.3.2 Ascent Phase (Global Exploration)

Seeds rise into the air depending on wind strength. A random wind factor determines the upward or spreading behavior. Position update is modeled using the distance to a randomly chosen seed location D_s and logarithmic multipliers:

$$D_{t+1} = D_t + \delta \times v_x \times v_y \times lnY \times (D_s - D_t) \quad (6)$$

An adaptive factor controls the search step across total iterations T.

3.3.3 Descent Phase (Local Searching)

After rising, seeds begin to descend following a random walk pattern. Brownian motion is applied to simulate the drifting descent toward promising areas:

$$D_{t+1} = D_t - \alpha \times \beta_t \times (D_{mean_t} - \alpha \times \beta_t \times D_t) \quad (7)$$

where D_{mean_t} represents the mean population position at iteration t.

3.3.4 Landing Phase (Exploitation and Convergence)

Seeds land on the ground to establish new locations, directed by Levy flight distribution to fine-tune the local optimum:

$$D_{t+1} = X_{elite} + levy(\lambda) \times \alpha \times (X_{elite} - D_t \times \sigma) \quad (8)$$

Update X_{elite} if a newly landed seed has a superior fitness value.

3.3.5 Termination

Check if the maximum iteration limit T is met. If false, return to the ascent phase; if true, output the best global solution X_{elite}.

3.4 Mathematical modelling of HTA-DMAN

An HTA-DMAN merges three distinct attention pathways (such as channel, spatial, and branch/task attention) with a dual-path Mamba (State Space Model) architecture. This design

captures fine local details, multi-directional global context, and sequence dependencies in linear time O(L), bypassing the quadratic bottleneck $O(L^2)$ of standard transformers. Figure 3 shows the architecture of HTA-DMAN for web attack classification.

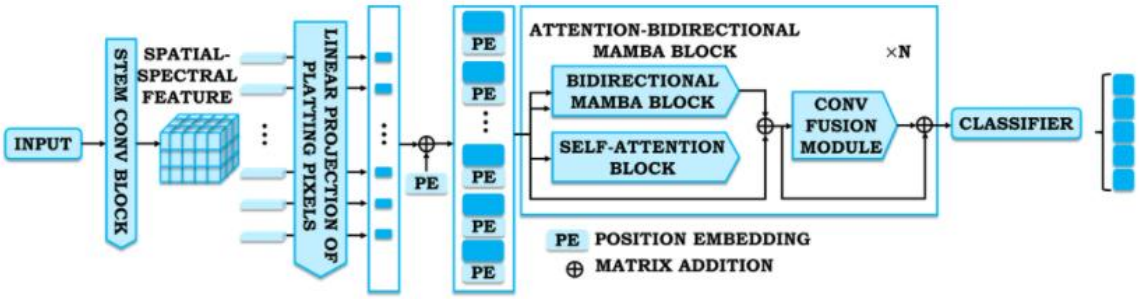

Figure 3: *Architecture of HTA-DMAN for web attack classification*

3.4.1 Input Embedding and Dual Mamba Branching

The input sequence or feature map is projected and split into dual parallel pathways to handle directional or multi-scale features. Dual Mamba blocks process the sequence via selective state-space iterations.

The continuous SSM equations are expressed as:

$$\frac{h(t)}{dt} = Ah(t) + Bx(t) \quad (9)$$

$$y(t) = Ch(t) + Dx(t) \quad (10)$$

Discretized Selective Form (with step size Δ):

$$h_t = \overline{A_t} h_{t-1} + \overline{B_t} x_t \quad (13)$$

$$y_t = C_t h_t \quad (14)$$

The dual Mamba streams output refined representations $Y_{mamba1}, Y_{mamba2} \in R^{B\times L\times D}$.

3.4.2 Dual Mamba Feature Fusion

The outputs from the dual Mamba streams are integrated using a gated or element-wise feature mixer:

$$F_{dual} = \sigma(W_1 Y_{mamba1} + W_2 Y_{mamba2}) \odot Y_{mamba1} + \big(1 - \sigma(.)\big) \odot Y_{mamba2} \quad (15)$$

Where the activation function of sigmoid is designated as σ and division wise elements is designated as $(.)$.

3.4.3 Triple Attention Mechanism

To recover any structural/contextual loss from recurrent scanning, a Triple Attention module is applied to F_{dual}.

$$Z_{triple} = A_{s2}\Big(A_{s1}\big(A_c(F_{dual})\big)\Big) \quad (16)$$

3.4.4 Final Hybrid Aggregation and Residual Mapping

The final output blends the raw input X, the dual Mamba extraction, and the triple attention refinement through residual connections to stabilize training:

$$O_{final} = Norm(X + Z_{triple}) \quad (17)$$

It provides linear-time processing efficiency, sharp local-global feature fusion, and reduced memory bottlenecks.

4. Results and Discussion

The HTA-DMAN with DLO is proposed for web attack classification, which is evaluated using python software. The name of dataset used for simulation is web network, which is taken from Kaggle.

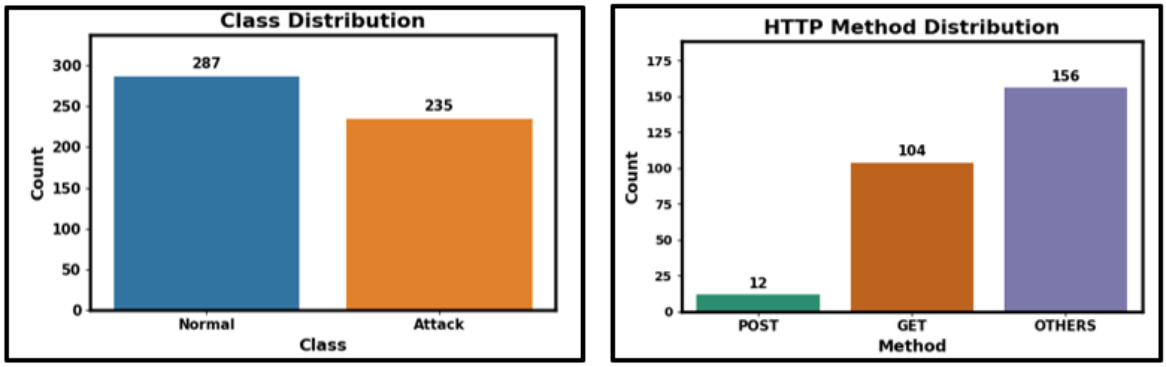

Figure 4: *Class distribution and HTTP method distribution*

Figure 4 shows the class and HTTP method distribution. In class distribution, the vertical axis measures the count of events, showing that there are 287 instances classified as normal traffic, while 235 instances are classified as attack or malicious activity. In HTTP method distribution, the value of others is the most frequent category with 156 instances, followed by get requests with 104 instances, while post requests are the least frequent, appearing only 12 times in the dataset.

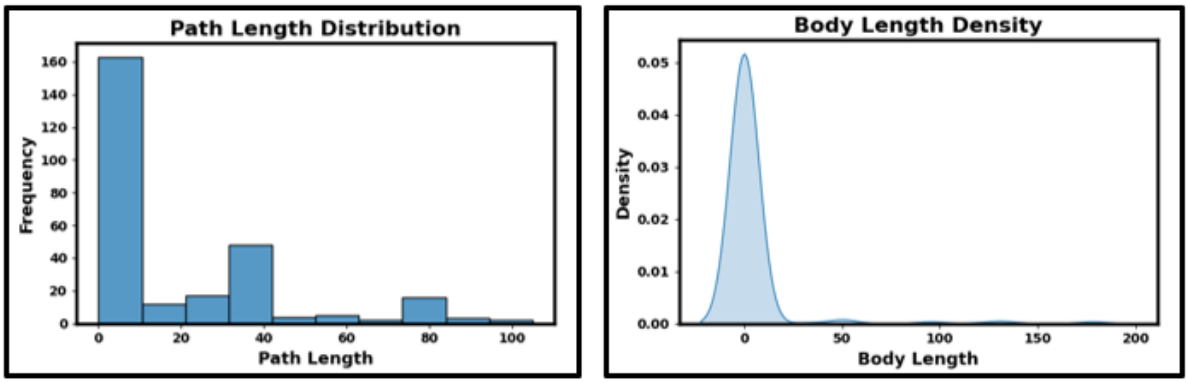

Figure 5: *Path length distribution and body length density*

Figure 5 shows the path length distribution and body length density. In path length distribution, it shows a highly right-skewed, multimodal distribution where the vast majority of observations are heavily concentrated at short path lengths between 0 and 10 units, followed by smaller, secondary clusters visible around the 40 and 80 marks. Body length density plot features a single, exceptionally sharp and narrow peak centered tightly around zero, indicating that almost all data points possess a body length near zero, with a negligible, flat tail extending toward higher values up to 200.

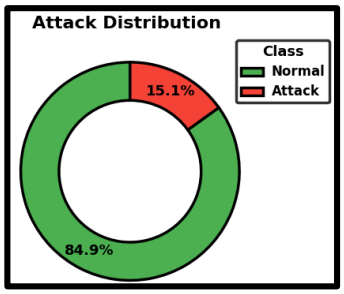

Figure 6: *Attack distribution*

Attack distribution is demonstrated in figure 6. The vast majority of the data is classified as normal activity, which is represented by the large green section of the ring and accounts for 84.9% of the total distribution. Conversely, malicious or anomalous activity is represented by the smaller red section, indicating that attacks constitute 15.1% of the observed events.

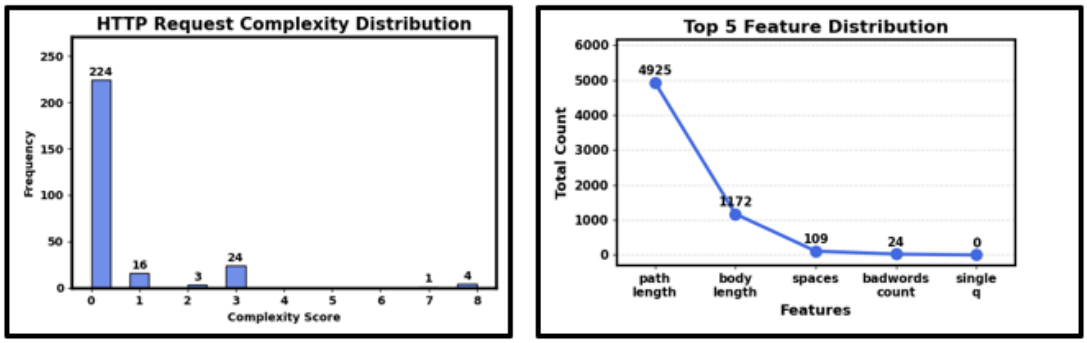

Figure 7: *HTTP request complexity distribution and Top 5 feature distribution*

Figure 7 shows the HTTP request complexity distribution and Top 5 feature distribution. HTTP request complexity distribution shows that the vast majority of requests are highly simple, with 224 instances scoring a 0, followed by a small secondary peak of 24 requests at a complexity score of 3, and only a negligible few reaching higher scores up to 8. Top 5 feature distribution shows that "path length" is overwhelmingly the most frequent feature with 4,925 counts, followed by "body length" at 1,172, while other structural markers like "spaces" (109), "badwords count" (24), and "single q" (0) appear far less often. Together, these charts indicate that the analyzed web traffic is predominantly low in complexity and primarily defined by the lengths of its request paths and bodies.

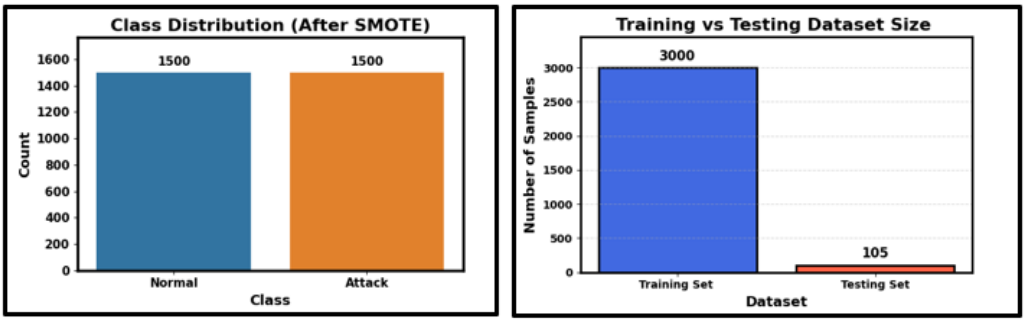

Figure 8: *Class distribution and training vs testing dataset size*

Figure 8 shows the class distribution and training vs testing dataset size. Distribution of class displays that the data has been balanced using the Synthetic Minority Over-sampling Technique (SMOTE), resulting in an equal distribution of exactly 1,500 samples each for the "Normal" and "Attack" classes. The right chart, titled "Training vs Testing Dataset Size", compares the split between the two data subsets. It illustrates a highly skewed distribution where the Training Set contains a large majority of the data with 3,000 samples, while the Testing Set is significantly smaller, consisting of only 105 samples.

Table 2: *Model Evaluation Metrics*

Accuracy	0.9619
Precision	0.9574
Recall	0.9574
F1-Score	0.9574
ROC-AUC Curve	0.9670
Log loss	0.1677
Balanced Accuracy	0.9615
Matthews CorrCoef	0.9230
Cohen Kappa Score	0.9230

Table 2 shows the model evaluation metrics. The result of accuracy is 0.9619, precision is 0.9574, recall is 0.9574, F1-Score is 0.9574, ROC-AUC curve is 0.9670, log loss is 0.1677, balanced accuracy is 0.9615, Matthews corrcoef is 0.9230 and cohen kappa score is 0.9230.

Table 3: *Classification report*

Classes	Precision	Recall	F1-Score	support
Normal	0.97	0.97	0.97	58
Attack	0.96	0.96	0.96	47
Accuracy			0.96	105
Macro avg	0.96	0.96	0.96	105
Weighted avg	0.96	0.96	0.96	105

Table 3 shows the classification report, which includes two classes, they are normal and attack. The precision of class normal is 0.97 and attack is 0.96. The recall value of class normal is 0.97 and attack is 0.96 and the F1-Score of class normal is 0.97 and attack is 0.96.

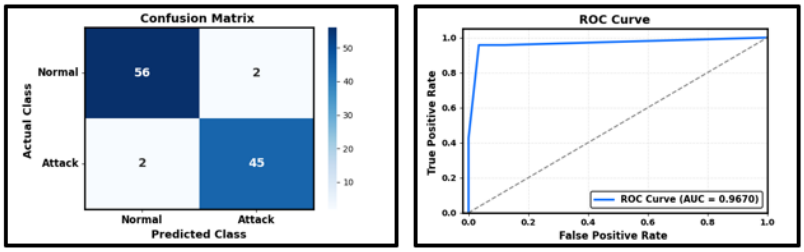

Figure 9: *Confusion matrix and ROC Curve*

Figure 9 demonstrates the confusion matrix and ROC curve. In confusion matrix, the model correctly identified 56 "Normal" instances and 45 "Attack" instances. There were 4 total errors: 2 attacks were missed (predicted as normal), and 2 normal instances were falsely identified as attacks. In ROC curve, the dashed diagonal line characterizes the starting point effectiveness of a entirely random classifier, which has an AUC of 0.5. In contrast, the solid blue line indicates that the model accomplishes a very high True Positive Rate while maintaining an exceptionally low False Positive Rate. The overall quality of the model is quantified by the AUC value of 0.9670 shown in the legend, where a value close to 1.0 signifies excellent discriminative ability and highly accurate classification.

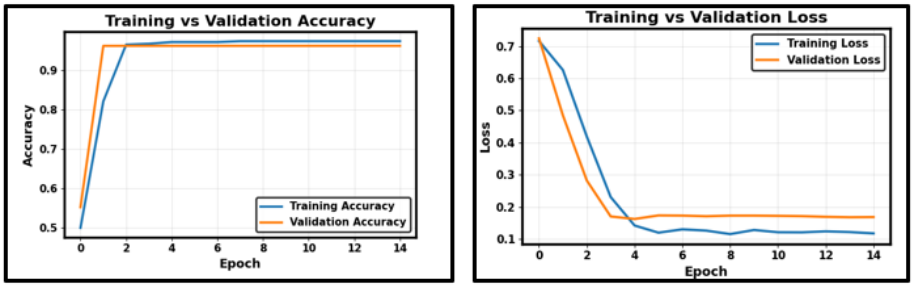

Figure 10: *Training vs testing accuracy and loss*

Figure 10 demonstrates the accuracy and loss of proposed HTA-DMAN-DLO. The training and testing accuracy of proposed HTA-DMAN-DLO is 96% and training vs testing loss of proposed HTA-DMAN-DLO is 10%.

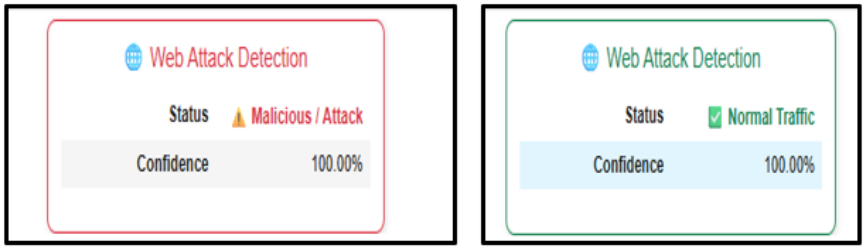

Figure 11: *Web attack detection*

Figure 11 shows the web attack detection results. First image appears to be a notification from a security tool indicating that a malicious attack has been detected. The system indicates a high confidence level of 100.00% regarding this threat. The second image shows a status indicator from a cybersecurity tool indicating "Normal Traffic". The 100% confidence level suggests the security system has high certainty that no malicious activity is currently detected.

Table 4: *Comparison of accuracy and response time*

Method	Accuracy	Response time
Transformer-DNN [8]	92%	0.5s
CNN-LSTM [9]	93%	0.4s
GSO-DAE [10]	94%	0.3s
Proposed HTA-DMAN-DLO	96%	0.2s

Table 4 shows the comparison of accuracy and response time. The accuracy of proposed HTA-DMAN-DLO is 96% and the response time of proposed HTA-DMAN-DLO is 0.2s. This method maximizes detection accuracy for complex web threats like SQL injection and cross-site scripting while maintaining linear computational scaling.

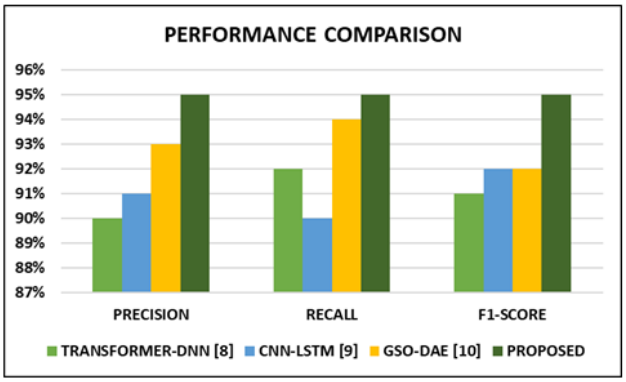

Figure. 12: *Performance evaluation*

Performance evaluation is represented in figure 12. The precision, recall and F1-Score of proposed HTA-DMAN-DLO is 95%. HTA-DMAN-DLO It minimizes false positives, ensuring high reliability in critical decisions where false alarms carry high operational or safety cost.

5. Conclusion

In this paper, a hybrid HTA-DMAN-DLO has been proposed for web attack classification. The performance analysis is carried out using python software. HTA-DMAN-DLO are effective at identifying, classifying, and mitigating web attack, achieving 96% accuracy. Utilizing optimized feature sets allows for efficient, real-time detection without requiring massive, computationally expensive models. The developed technique is associated with conventional techniques like Transformer-DNN, CNN-LSTM and GSO-DAE in terms of precision, recall and F1-Score. The DL models, specially HTA-DMAN-DLO offer high accuracy of 96%, and less computational time of 0.2s in detecting and classifying web attacks. Future work must address issues with imbalanced datasets, the need for increased resilience against adversarial attacks, and the refinement of real-time, proactive prevention.

References

1. Umara Urooj; Bander Ali Saleh Al-Rimy; Anazida Binti Zainal; Faisal Saeed; Abdelzahir Abdelmaboud; Wamda Nagmeldin, Year: 2024, “Addressing Behavioral Drift in Ransomware Early Detection Through Weighted Generative Adversarial Networks”, IEEE Access, Vol: 12, pp. 3910-3925.
2. Udayakumar R; Joshi A; Boomiga S.S; Sugumar, R, Year: 2023, “Deep fraud Net: A deep learning approach for cyber security and financial fraud detection and classification”, Journal of Internet Services and Information Security, Vol: 13, No: 3, pp.138-157.
3. Olivia Jullian; Beatriz Otero; Eva Rodriguez; Norma Gutierrez; Héctor Antona; Ramon Canal, Year: 2023, “Deep-learning based detection for

cyber-attacks in iot networks: A distributed attack detection framework", Journal of Network and Systems Management, Vol: 31, No: 2.

4. Donghyun Min; Yungwoo Ko; Ryan Walker; Junghee Lee; Youngjae Kim, Year: 2022, "A Content-Based Ransomware Detection and Backup Solid-State Drive for Ransomware Defense", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol: 41, No: 7, pp. 2038 – 2051.
5. Ahsan Ahmed; Md Aktarujjaman; Mohammad Moniruzzaman; Md Shahab Uddin; Arifa Akter Eva; M. F. Mridha; Kyoungyee Kim; Jungpil Shin, Year: 2024, "A Hybrid Deep Learning and Quantum Optimization Framework for Ransomware Response in Healthcare", IEEE Open Journal of the Computer Society, Vol: 7, pp. 154 – 165.
6. Horacio Rodriguez-Bazan; Grigori Sidorov; Ponciano Jorge Escamilla Ambrosio, Year: 2023, "Android Ransomware Analysis Using Convolutional Neural Network and Fuzzy Hashing Features", IEEE Access, Vol. 11, pp. 121724 – 121738.
7. Abdullah IA Alzahrani; Manel Ayadi; Mashael M Asiri; Amal Al-Rasheed; Amel Ksibi, Year: 2022, "Detecting the presence of malware and identifying the type of cyber-attack using deep learning and VGG-16 techniques", Electronics, Vol: 11, No: 22, pp. 3665
8. Hesham Kamal; Maggie Mashaly, Year: 2025, "Enhanced Hybrid Deep Learning Models-Based Anomaly Detection Method for Two-Stage Binary and Multi-Class Classification of Attacks in Intrusion Detection Systems", Algorithms, Vol: 18, No: 2.
9. Rasha Almarshdi; Laila Nassef; Etimad Fadel; Nahed Alowid, Year: 2022, "Hybrid Deep Learning Based Attack Detection for Imbalanced Data Classification", Intelligent Automation & Soft Computing, Vol: 35, No: 1, pp. 297-320
10. Jaya Dipti Lal; Shahnawaz Ayoub; Prashant D Hakim; S. Prabagar; Vijay Kumar Dwivedi; Mohit Tiwari, Year: 2023, "Hybrid Deep Learning based Attack Detection and Classification Model on IoT Environment", IEEE, pp. 1-5.