Optimizing DDoS Traffic Classification Using Enhanced Deep Neural Network Techniques

Mohammed Fadhl Abdullah[1,2], **Ahmed Saleh AL-Hurdi**[1,3]

[1] *College of Engineering and Computing, University of Science and Technology, Aden, Yemen*

[2]*m.albadwi@ust.edu,* [3]*aalhurdi@gmail.com*

***Abstract**: This study investigates the effectiveness of an enhanced Deep Neural Network (DNN) model for classifying Distributed Denial of Service (DDoS) traffic. The proposed model integrates an Adaptive Attention Layer (AAL) technique to improve detection accuracy. Experiments were conducted using a full and a reduced feature sets with three dataset sizes, namely; 4K, 40K, and 225K samples. Results reveal a positive correlation between the number of features, dataset size, and model performance metrics. The study highlights the pivotal role of the AAL in dynamically prioritizing the most relevant features, enabling the model to more effectively detect subtle anomalies in high-dimensional traffic data. This attention mechanism significantly reduces both false positives and false negatives. Across all configurations, the proposed model consistently achieved high performance and near-perfect accuracy, precision, recall, and F1-Scores. These results underscore the model's robustness, adaptability, and potential for real-world deployment in modern DDoS detection systems.*

Keywords: DDoS Classification, Deep Neural Network, Data Normalization.

1. INTRODUCTION

Cybersecurity continues to be a paramount challenge in the digital era, driven by the escalating frequency and complexity of network-based attacks. Among these, DDoS attacks represent one of the most disruptive threats, as they strategically exploit vulnerabilities across various layers of the network protocol stack, often resulting in substantial operational disruptions and financial losses. At the network layer, common attacks include IP Spoofing, ICMP Floods, Smurf Attacks, and Routing Protocol Exploits. These methods often manipulate IP headers or exploit weaknesses in the ICMP protocol to overwhelm target systems with malicious traffic, resulting in service interruptions [1]. At the transport layer, attackers target vulnerabilities in TCP and UDP protocols. Common attacks here include TCP SYN Floods, and UDP Floods. Additionally, session hijacking and exploits targeting SSL/TLS protocols, at the session layer, further increase the risk [2].

https://doi.org/10.5281/zenodo.15771144

Traditional signature-based IDS are often inadequate in identifying novel, polymorphic, or zero-day threats. This limitation has driven a shift toward intelligent, AI-driven security systems. Artificial Intelligence (AI) and Machine Learning (ML) offer adaptive, real-time detection capabilities, making them well-suited for addressing modern cybersecurity challenges [3], [4]. AI, broadly defined as the science of building intelligent machines capable of human-like tasks, underpins the development of autonomous threat detection systems. Within this realm, ML algorithms learn from historical data, while Deep Learning (DL), a subfield of ML, employs multilayered neural networks to capture complex, non-linear patterns in high-dimensional datasets [5].

DNNs have achieved state-of-the-art results in domains such as computer vision and natural language processing. Their hierarchical learning structure enables the extraction of multi-level features, making them especially effective for detecting subtle anomalies in cybersecurity contexts. Compared to traditional Artificial Neural Networks (ANNs), DNNs provide improved predictive performance at the cost of greater computational demand which is an acceptable trade-off in many high-stakes security scenarios [6].

This study proposes an enhanced DNN-based intrusion detection model that incorporates an Adaptive Attention Layer (AAL) and data normalization techniques to improve the classification of DDoS and benign network traffic. The primary objectives of this research are to, (i) Enhance classification accuracy, (ii) Minimize detection latency, and (iii) Ensure scalability and adaptability in dynamic threat environments.

The remainder of this paper is structured as follows: Section II reviews related work on DDoS detection and deep learning-based IDS. The proposed model architecture, and methodology are described in Section III. Section IV presents experimental results and comparative evaluations. Section V concludes the study with key findings and suggestions for future research.

2. FEATURES RELATED WORK

Numerous studies have explored ML and DL methods for intrusion detection, frequently using the CIC-IDS2017 dataset as a benchmark for evaluating model performance due to its comprehensive inclusion of modern attack vectors. Ref [7] developed an intrusion detection system optimized for big data environments by integrating dimensionality reduction technique, Principal Component Analysis (PCA) with Random Forest (RF) and K-Means clustering. Their approach reduced the dataset's original 79 features to 39, selecting 45 critical features, and achieved a detection accuracy of 99.7%. Ref [8] work enhanced this dataset dimensionality while preserving specificity and sensitivity. The optimized dataset enabled faster model evaluation and maintained high classification performance across number of classifiers, supporting its applicability in real-world IDS validation. Ref [9] provided a critical analysis of the dataset, identifying structural inconsistencies that could bias IDS performance. They proposed a refined dataset version and enhancing the reliability of detection outcomes.

Ref [10] introduced a hybrid IDS architecture integrating ML and DL. Evaluated on CIC-IDS2017, UNSW-NB15, NSL-KDD, and WSN-DS datasets, the model achieved a test accuracy of 89.26% and F1-scores up to 98.64%, effectively capturing spatial and temporal patterns. Ref [11] proposed a DNN-based model for network intrusion detection using 36 selected features and four hidden layers. Ref [12] aimed to improve AdaBoost-based IDS performance through dimensionality reduction, the system achieved an overall accuracy of 81.83%, enhancing detection efficiency.

Addressing the limitations of existing datasets, [13] introduced a custom real-traffic-based dataset comprising over 70 features and various attack types. The dataset was evaluated using Support Vector Machine, Decision Tree, and Naive Bayes classifiers. Ref [14] proposed an improved feature selection algorithm, FACO, which combines Ant Colony Optimization with a tailored fitness function and optimized pheromone update rule. Similarly, [15] combined Discrete Differential Evolution with the C4.5 algorithm to optimize feature selection, improve detection accuracy, and reduced training and testing time.

Several recent studies have explored various techniques to enhance intrusion detection performance. Ref [16] applied Information Gain to identify the most relevant features for anomaly detection, selecting 52 features and evaluating multiple classifiers, ultimately achieving a highest accuracy of 99.87%. Ref [17] investigated deep learning (DL) models for intrusion detection in IoT environments, utilizing the CIC-IDS2017 dataset. Ref [18] focused on detecting web-based attacks such as SQL Injection and Cross-Site Scripting (XSS), achieving detection accuracies of up to 99.57% when evaluated with a 78-feature dataset. Ref [19] used a RF algorithm with selective feature selection process on the KDDCup99 dataset to reduce the rates of the false alarm and improve classification accuracy. Ref [20] employed a DNN with a learning rate of 0.1 on the KDDCup99 dataset to detect attack attempts.

In comparison, the proposed model, attention-based deep learning technique, achieved an accuracy of 99.93%, surpassing several established benchmark methods [21]. Furthermore, this study demonstrates the robustness and adaptability of the proposed model by evaluating not only its accuracy but also its precision, recall, and F1-score across varying feature sets.

3. MATERIALS AND METHODS

This section outlines the design, and the evaluation of the proposed model for effective DDoS attack detection. The methodology comprises the model architecture, dataset preprocessing, feature selection, and performance evaluation.

Data Preprocessing and Normalization

This study uses Machine Learning CSV data, which come within 8 traffic monitoring sessions and 78 features which labelled either as a normal traffic (defined as Benign traffic), or anomaly traffic (referred to as Attack traffic). There are 14 types of attacks in this dataset. This study focuses on the DDOS attack. Effective preprocessing is crucial for ensuring the model's accuracy and training efficiency.

The following steps are applied: the dataset is first loaded in CSV format, followed by data cleaning to eliminate or impute non-numeric and null values. The target column, indicating benign or DDoS traffic, is then encoded into binary labels (0 or 1). Next, min-max normalization is applied to scale all features between 0 and 1, improving training stability and convergence speed. Finally, the dataset is split into training (80%) and testing (20%) sets.

Proposed Model Architecture

The traditional DNN consists of three stages, namely, the Input stage or layer, the Hidden stage which usually consists of one or more layers, then the Output stage or layer. Hidden layers include the rectified linear unit function or the activation function, such as the Sigmoid function (if the output is true or false or 0, 1), the SoftMax function (if the class label has more than two pattern shown in the last stage. In these DNNs an equal weight is applied to all input features initially, which can dilute the model’s sensitivity to attack-related patterns in high-dimensional data. The proposed architecture addresses this limitation by selectively amplifying crucial signals, particularly useful in complex intrusion detection datasets. It enhances a traditional DNN by incorporating the Adaptive Attention Layer between the input and hidden layers. This layer enables the model to dynamically assign importance to input features, thereby improving its ability to detect subtle patterns in network traffic.

The architecture comprises four main components: an input layer that receives numerical features from the dataset; the AAL, which computes attention scores to reweight input features based on their relevance; multiple hidden layers consisting of fully connected neurons with activation functions such as ReLU and Sigmoid; and an output layer that uses the Sigmoid function to perform binary classification (benign vs. DDoS attack). To explore performance trade-offs, the model is trained using different feature subsets, 78, 39, and 25 features.

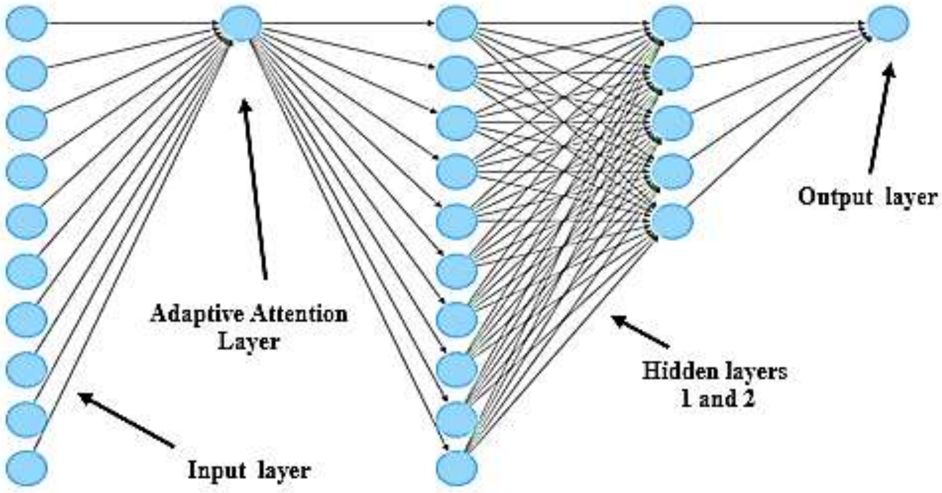

Figure 1 DNN with Adaptive Attention Layer

Adaptive Attention Layer Mechanism

The AAL operational flow includes;

- *Weight Computation*: During training, each input feature is assigned an initial weight w_i, which is updated via backpropagation.
- *Softmax Normalization*: The weights are normalized to attention scores through the softmax function.
- *Feature Reweighting*: Inputs are scaled by their attention scores to prioritize informative features.
- *Forward Propagation*: The reweighted feature vector proceeds through the network, influencing the final classification decision.

The AAL assigns importance to each feature input x_i using an attention score α_i which is computed via the softmax of learnable weights w_i. The final weighted input vector is then calculated as:

$$x_i' = \alpha_i . x_i = \frac{e^{w_i}}{\sum_{j=1}^{n} e^{w_i}} \cdot x_i \qquad \dots (1)$$

The resulting weighted inputs x_i' are forwarded to the subsequent hidden layers for further processing. This process ensures that more informative features have greater influence during forward propagation, thereby enhancing model interpretability and precision in detecting attack patterns. It also reduces overfitting by suppressing noise and irrelevant features, and significantly boosts classification metrics across various dataset sizes.

https://doi.org/10.5281/zenodo.15771144

The general architecture of the proposed model consists of five phases, which are illustrated in Figure 2 above. These phases are data collection, data processing, the proposed DNN model, training, and testing evaluation of the applied model.

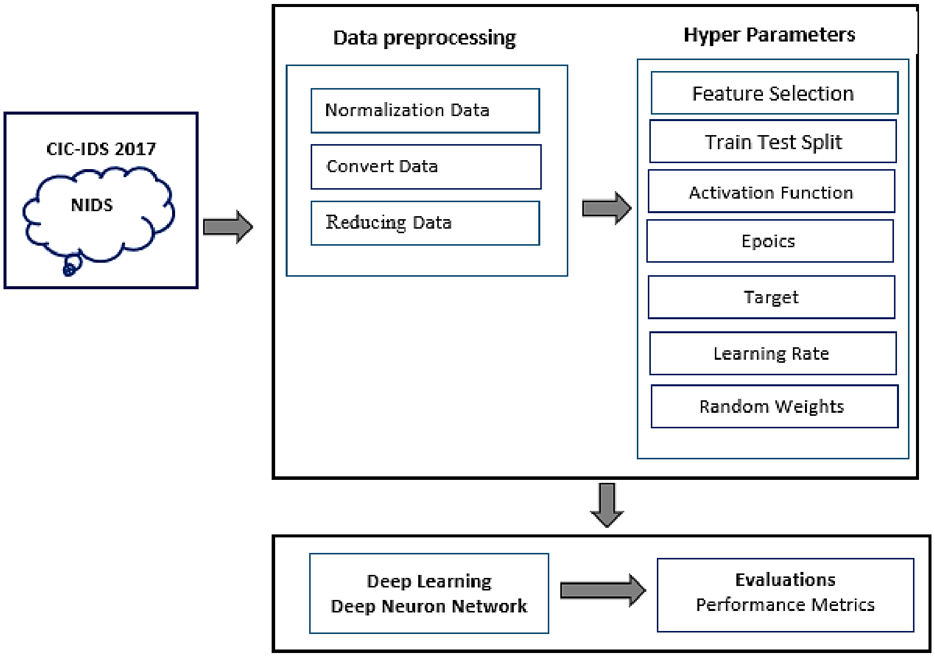

Figure 2: Proposed Model to Analysis Network Data

Training and Evaluation Process

To examine the impact of dimensionality on model performance, the following feature sets are tested (i) Full set of the dataset (78 features), (ii) Reduced set based on feature relevance (39 features), and (iii) Minimal set for lightweight computation (25 features). For each scenario, the model is trained and tested with and without the AAL and normalization layer to evaluate their effectiveness.

The model training involves several stages: forward propagation, where input data passes through the attention and hidden layers to produce predictions; loss calculation, using binary cross-entropy to quantify prediction error; and backpropagation, where gradients are computed and propagated backward to update the weights via an optimizer such as Adam. The training is conducted over 50 epochs per experiment to ensure sufficient learning.

In the beginning, as the data is fed into the neural network, an exponential value of the weights (associated with each input), are calculated and then normalized to assign relative importance to each input. Each input is then multiplied by its normalized weight to determine its relative influence on the network (Equation (1)).

$$\text{Node value} = \Sigma x_i'.w_i + b \quad \dots \ (2)$$

$$Sigmoid = \frac{1}{1 + e^{-last\ node\ value}} \quad \dots (3)$$

Forward Propagation: In this step, the weighted inputs are summed along with a bias term, Equation (2). The result is then passed through the Sigmoid activation function, which ensures that the output values remain between 0 and 1, see equation (3). Once the outputs are computed, they are compared with the target values, and the error is computed as shown by Equation (4) as the difference between the actual and target values.

$$Calculated\ Error = 0.5(Target - output)^2 \ ..\ (4)$$

Backpropagation: After calculating the error, the network starts correcting it through backpropagation. First, the error gradient for the last node is computed based on the difference between the expected and actual output. Then, the error gradient for previous nodes is calculated based on their values. Finally, the weights are updated using the learning rate, adjusting the old weights in a way that progressively reduces the error.

This process is repeated over multiple epochs until the error is minimized, improving the accuracy of the neural network's predictions. Backpropagation is used to adjust the weights and minimize the error across the neural network.

Experimental Setup

The proposed technique is implemented on a system with the following specification: Intel Core i7 CPU, 8GB installed RAM, 64-bit OS. Popular Python libraries (Scikit-learn, NumPy, Pandas, and Matplotlib) are utilized for the implementation and the visualization.

Feature Selection Strategy

Effective feature selection plays a critical role in enhancing model performance, reducing computational overhead, and improving interpretability. This research evaluates the proposed DNN model using three different feature sets: the full 78-feature set, a reduced 39-feature set, and a minimal 25-feature set. By testing the model under these three feature configurations, the study evaluates: The impact of dimensionality on classification accuracy, the model's robustness in feature-constrained environments, and the importance of normalization and the attention mechanism across varying input complexities. Using a multi-level feature selection strategy oppose will help in the flexibility validation of the proposed model as well as it offers practical deployment configurations in systems with different resource limitations. The selection strategy for each is described as follows:

A. *Full Feature Set (78 Features):* The original CIC-IDS2017 dataset includes 78 features extracted from network traffic and cover a wide range of packet-level and flow-level statistics (packet sizes, time intervals between packets, protocol flags, and byte distribution).

B. *Reduced Feature Set (39 Features):* To improve processing speed while keeping detection accuracy high, a smaller set of 39 features was selected. Features with low variance or strong correlation with others were removed. The goal was to keep only the most useful features, balancing accuracy and computational efficiency.

C. *Minimal Feature Set (25 Features):* For environments with limited computing power, a minimal set of 25 features was chosen manually, based on expert analysis and previous experiments. These features, listed in Table 1, include key indicators of suspicious activity. This set is designed to keep detection effective while reducing model complexity, making it suitable for lightweight or real-time intrusion detection systems.

Table 1: demonstrates 25 feature selection.

Features Selected					
1	Bwd Packets total Length	10	Bwd IAT Std	19	Average Packet Size
2	Fwd Packet Length Min	11	Bwd IAT Min	20	Avg Fwd Segment Size
3	Bwd Packet Length Min	12	Fwd Packets/s	21	Subflow Fwd Bytes
4	Bwd Packet Length Std	13	Bwd Packets/s	22	Init_Win_bytes_forward
5	Flow IAT Mean	14	Min Packet Length	23	Init_Win_bytes_backward
6	Flow, IAT Min,	15	Packet Length Variance	24	Active Mean
7	Fwd IAT Min	16	PSH Flag Count	25	Idle Min
8	Bwd IAT Total	17	ACK Flag Count		
9	Bwd IAT Mean	18	Down/Up Ratio		

Evaluation metrics

Confusion matrix according to (Daniela X et, al. 2009) can be interpreted as a tool that has a function to perform analysis whether the classifier is good in recognizing the tuples of different classes. The calculation of the matrix confusion is showed in Table 2. After setting up the models, it is time to measure the performance by going through an evaluation stage. The proposed model needs to be tested to confirm its reliability based on four possible outputs, TP, FP, TN, and FN. Where:

- TP: True positives are events that are correctly identified as abnormal
- FP: False positives are legal events that are incorrectly identified as abnormal
- TN: True negatives are incidents that are correctly identified as legal activities
- FN: False negatives can be defined as possible intrusive activity that the IDS passes through as normal activity.

The models used in this work were evaluated based on accuracy, precision, recall, and F1_score.

Table 2: Confusion Matrix

Predict Label

		Intrusion	Normal
Host Label	**Intrusion**	TP	TN
	Normal	FP	FN

The following measurement metrics are used to measure the performance of a dataset:

1. **Precision:** It measures the proportion of correctly identified positive samples among all predicted positive samples. It is calculated as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives+False Positives}} \quad \ldots (5)$$

2. **Recall:** It commonly referred to as sensitivity, is the frequency at which favorable forecasts are expected to be positive. It is calculated as;

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives+False Positives}} \quad \ldots (6)$$

3. **F1-Score:** It is the weighted harmonic mean of precision and recall (the average of recall and precision values). This score accounts for false positives and negatives. Intuitively this is not an accuracy, but F1 is usually more useful than accuracy, especially if it has an uneven distribution of classes.

$$F1_{Score} = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision+Recall}} \quad \ldots (7)$$

4. **Accuracy:** It is the percentage of the correctly classified objects (see Equation 8). Accuracy is the proportion of correctly classified instances:

$$\text{Accuracy} \;=\; \frac{\text{True Positives+True Negatives}}{\text{Total Samples}} \qquad \ldots (8)$$

4. RESULTS AND ANALYSIS

Table 3 shows the DNN output with 78 features with normalization and AAL layer. Using 40K records, split data 80/20% and 50 epochs.

Table 3: Possible outputs for different thresholds.

Threshold	TP	FP	FN	TN	Precision	Recall	F1Score	Accuracy
0.1	378	57	0	365	0.869	1.000	0.930	0.928
0.2	377	22	1	400	0.945	0.997	0.970	0.973
0.3	375	2	3	420	0.995	0.992	0.993	0.993
0.4	365	1	13	421	0.997	0.966	0.981	0.988
0.5	365	1	13	421	0.997	0.966	0.981	0.988
0.6	364	1	14	421	0.997	0.963	0.980	0.987
0.7	364	1	14	421	0.997	0.963	0.980	0.987
0.8	363	1	15	421	0.997	0.960	0.978	0.986
0.9	362	1	16	421	0.997	0.958	0.977	0.985

At a threshold of 0.1, the model achieves perfect recall (1.000), detecting all DDoS attacks, but with lower precision (0.869) due to 57 false positives. While this ensures no attacks are missed, it increases false alerts, potentially overburdening network administrators. Raising the threshold to 0.3 offers the best trade-off, yielding high precision (0.995), recall (0.992), F1-score (0.993), and accuracy (0.993), making it well-suited for environments requiring both accuracy and efficiency. Beyond 0.4, precision remains high (~0.997), but recall gradually drops (to 0.958 at threshold 0.9), indicating a more conservative model that may miss some attacks. In DDoS detection, where even brief undetected incidents matter, this trade-off must be carefully managed.

Table 4: Effect of AAL and Normalization for different dataset sizes.

Dataset	TP	FP	FN	TN	Precision	Recall	F1Score	Accuracy
4K AAL+Norm	375	2	3	420	0.9947	0.9921	0.9934	99.38%
4K No AAL/Norm	373	4	5	418	0.9894	0.9868	0.9881	98.88%
40K AAL+Norm	3984	5	15	3965	0.9987	0.9962	0.9974	99.75%
40K No AAL/Norm	2695	0	1304	4000	1.0000	0.6739	0.8053	83.69%
225K AAL+Norm	25703	9	21	19410	0.9997	0.9992	0.9994	99.93%
225K No AAL/Norm	25724	15958	0	3461	0.6171	1.0000	0.7630	64.65%

Table 4 shows the DNN performance metrics with 78 features, 0.3 threshold, and normalization and AAL layer. Using different dataset sizes, split data to 80/20% and running 50 epochs.

In all dataset sizes, using AAL with normalization technique boosts precision, recall, and accuracy significantly, see Table 4. Without the propose technique there is a serious weakness where the Recall, at 40K and especially 225K, stays high but precision and accuracy collapse, the drop in accuracy from 99.93% to 64.65% (225K) when removing AAL/Norm is significant. Extremely high FP rates reduce the system's usability. The AAL and normalization layers are crucial for generalization, especially in larger datasets.

Table 5: Effect of Feature Size Reduction with AAL and 40K dataset size

Features Count	TP	FP	FN	TN	Precision	Recall	F1Score	Accuracy
78	3984	5	15	3965	0.9987	0.9962	0.9974	99.75%
39	3979	5	20	3995	0.9987	0.9950	0.9968	99.69%
25	3983	31	3	3983	0.9923	0.9992	0.9957	99.58%

Table 5 shows that all three feature sets perform excellently across all metrics. Where the reducing features to 39 has negligible impact. Even 25 features offer almost the same performance, F1 remains above 0.99. However, there is a slight increase in false positives at 25 features. So, if computational cost is a concern, reducing features to 25 may be acceptable due to its high recall and minimal F1 drop.

Table 6: With vs Without AAL for 25 Features, 40K Dataset

Method	TP	FP	FN	TN	Precision	Recall	F1Score	Accuracy
AAL + Norm	3983	31	3	3983	0.9923	0.9992	0.9957	99.57%
No AAL/Norm	3636	27	350	3987	0.9926	0.9121	0.9506	95.50%

The AAL/Norm technique ensures strong balance of high precision and near-perfect recall. See Table 6. Without it the precision still, but recall drops (many false negatives). Many attacks/events are missed. Similarly, the accuracy drops from ~99.6% to 95.5%, significant in sensitive systems like intrusion detection. The AAL and normalization improve event detection dramatically even with fewer features.

Table 7: Threshold Sweep, 78 Features, AAL/Norm, 225K

Threshold	TP	FP	FN	TN	Precision	Recall	F1	Accuracy
0.1	25709	35	15	19384	0.9986	0.9984	0.9985	99.89%
0.2	25703	9	21	19410	0.9996	0.9984	0.9990	99.93%
0.3	25702	9	22	19410	0.9996	0.9983	0.9989	99.93%
0.4	25696	8	28	19411	0.9997	0.9989	0.9993	99.92%
0.5	25695	8	29	19411	0.9997	0.9989	0.9993	99.92%

As shown in Table 7, all thresholds yield outstanding performance. F1 score remains nearly perfect across the board. The lower thresholds slightly increase FP but decrease FN. Higher thresholds reverse that. The 0.2 or 0.4 thresholds seem ideal (excellent balance of precision and recall). That is the fine-tuning of threshold between 0.2–0.4 gives the best generalization without significantly sacrificing any metric.

Table 8 Accuracy Across Datasets and AAL Configurations

Features	Dataset Size	Accuracy with AAL (%)	Accuracy without AAL (%)
78	4K	99.37	98.87
	40K	99.75	83.69
	225K	99.93	64.65
39	4K	98.37	93.12
	40K	99.68	85.37
	225K	99.90	67.12
25	4K	98.62	97.75
	40K	99.57	95.50
	225K	99.89	84.96

Table 8 highlights the strongest results achieved by the proposed model. With 39 features, the model maintained solid performance (only slightly lower than with the full set of 78 features). On the 4K dataset, it reached 98.37% accuracy, showing only a slight dip. Performance improved on the 40K dataset, achieving 99.68% accuracy, suggesting that more data helps compensate for fewer features. On the larger 225K dataset, accuracy peaked at 99.90%, confirming the model’s effectiveness even with reduced features when enough data is available.

In contrast, removing normalization and the adaptive attention layer led to noticeable drops in accuracy, 93.12% on the 4K dataset, 85.37% on 40K, and just 67.12% on 225K, indicating less stability and more errors, especially as the dataset size grew.

When the features were reduced further to 25, the model still performed well with normalization and attention in place: 98.62% accuracy on 4K, 99.57% on 40K, and 99.89% on 225K. This shows that larger datasets can offset some of the impact of feature reduction. However, removing normalization and attention in this setup caused sharper performance declines, down to 97.75%, 95.50%, and 84.96%, respectively, highlighting the critical role these techniques play in ensuring accuracy and model stability.

5. DISCUSSION

The experimental findings provide strong empirical support for the effectiveness of the AAL layer for DDoS traffic classification. Across all tested configurations, the proposed model consistently outperformed traditional DNNs and several established benchmarks.

The model exhibited exceptional generalization, particularly with the full 78-feature set, where it achieved an accuracy of 99.93% and an F1-score exceeding 0.99. And even with the reduced feature set (25 features), the model maintained high performance; accuracy of 99.58%, F1-score of 0.9957, illustrating its robustness and suitability for deployment in environments with limited computational resources.

Threshold analysis revealed that performance remained stable across a broad range of decision thresholds, with the 0.2–0.4 interval providing the optimal balance between FPs and FNs. This tunability enhances the adaptability of the model to various operational settings.

Ablation experiments confirmed the critical contribution of the AAL and normalization. Removing these components led to significant degradation in performance, particularly on larger datasets, where precision and overall accuracy dropped dramatically (e.g., from 99.93% to 64.65% on the 225K dataset). The AAL specifically mitigates overfitting by emphasizing informative features and suppressing noise, which is especially important in high-dimensional data scenarios.

In comparison with recent studies (e.g., [7], [16], and [11]), the proposed model not only achieves superior detection rates but does so with improved efficiency and scalability. It demonstrates that a carefully integrated attention mechanism can eliminate the need for complex ensembles or excessive feature sets while still delivering competitive results.

Table 9: Comparison with Related Works

Method	Accuracy	Recall	Precision	F1-Score	Dataset
Vigneswaran 2018 [20]	93%	0.915	0.997	0.955	KDD Cup 99
Bandarupalli 2024 [11]	97.62%	0.8858	0.6529	0.6716	CIC-IDS-2017
Proposed Architecture					
Features Count 78	99.75%	0.9962	0.9987	0.9974	CIC-IDS-2017
Features Count 39	99.69%	0.9950	0.9987	0.9968	CIC-IDS-2017
Features Count 25	99.58%	0.9992	0.9923	0.9957	CIC-IDS-2017

Table 9 presents a comparative analysis between the proposed intrusion detection architecture and two related works [11], and [20]. The comparison is based on the four key performance metrics, namely, Accuracy, Recall, Precision, and F1-Score. Vigneswaran's method achieved an accuracy of 93%, a high precision of 0.997, and a lower recall (0.915) on KDD Cup 99 dataset. These values indicate some limitations in detecting all attack instances. While the other model (Ref [20]) which based on the modern CIC-IDS-2017 dataset, improved overall accuracy to 97.62% but suffered from low precision (0.6529) and F1-score (0.6716), These values indicate higher false positives and suboptimal balance between detection and error rates.

In contrast, our proposed model, which based on the same modern CIC-IDS-2017 dataset, achieved superior results across all the four metrics, regardless of the feature count used. With 78 features, the model attained an accuracy of 99.75%, recall of 0.9962, precision of 0.9987, and an F1-score of 0.9974. When the feature set was reduced to 39 and then to just 25, the model retained high accuracy (99.69% and 99.58%, respectively) with only marginal decreases in other metrics.

6. CONCLUSION

This study introduced an enhanced DNN-based intrusion detection model for classifying DDoS and benign network traffic. By leveraging diverse dataset configurations, the proposed model consistently achieved high performance, reaching up to 99.93% accuracy alongside near-perfect precision, recall, and F1-scores.

The integration of a dynamically reweighting input features technique enhances the model's sensitivity to attack-relevant patterns and maintains an outstanding performance even when feature dimensionality was reduced to 25. The consistent performance of our model highlights the model's robustness, scalability, and efficiency, confirming its suitability for practical deployment in resource-constrained cybersecurity environments without compromising detection accuracy. Future research will focus on deploying the model in live network environments and extending its functionality to accommodate a wider spectrum of attack types and evolving threat dynamics.

References

[1]. A. Mughal, “Cyber Attacks on OSI Layers: Understanding the Threat Landscape”, Journal of Humanities & Applied Science Research, vol. 3, no. 1, pp.1-18, 2022.

[2]. H. S. Obaid, & E. H. Abeed, “DoS and DDoS attacks at OSI layers”, International Journal of Multidisciplinary Research & Pubs, vol. 2, no. 8, pp.1-9, 2020.

[3]. O. A. Ajala, et al, “Review of AI and machine learning applications to predict and thwart cyber-attacks in real-time, “Magna Scientia Advanced Research and Reviews, vol 10, no 1, pp. 312-320, 2024.

[4]. M. Alqashbari, N. A. Munassar, M. F. Abdullah M. F, & A. ALHurdi, “Enhancing Performance Through Dynamic AES Round Keys and Adaptive Data Shifting Techniques “, Technological Applied and Humanitarian Academic Journal, vol 2, no 1, pp. 18-37, 2025.

[5]. K. Sharifani, & M. Amini, “Machine learning and deep learning: A review of methods and applications”, World Information Technology and Engineering Journal, vol. 10, no. 7, pp. 3897-3904, 2023.

[6]. O. A. Montesinos López, & J. Crossa. Multivariate statistical machine learning methods for genomic prediction. Springer Nature. 2022. DOI: 10.1007/978-3-030-89010-0

[7]. V. K. Swarnkar, A. Ambhaikar, & S. Swarnkar, “Big Data Security Enhancement Based Intrusion Detection System Using K-Mean Clustering of Decomposed Features”, INFORMATION TECHNOLOGY IN INDUSTRY, vol. 9, no. 1, pp. 387-394, 2021.

[8]. K. Boukhamla, J. Coronel, “CICIDS2017 Dataset: Performance Improvements and Validation as a Robust Intrusion Detection System Testbed”, International Journal of Information and Computer Security, vol 16 no. 2, pp. 20–32, 2021. DOI: 10.1504/IJICS.2021.10039325

[9]. R. Panigrahi, and S. Borah, “A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems”, Int. J. Eng. Technol., vol. 7, no. 24, pp. 479_482, 2018.

[10]. M. Sajid, et. al, “Enhancing intrusion detection: a hybrid machine and deep learning approach”, Journal of Cloud Computing, vol. 13, no. 1, 2024.

[11]. G. Bandarupalli, “Efficient Deep Neural Network for Intrusion Detection Using CIC-IDS-2017 Dataset”, 2024. DOI: https://doi.org/10.21203/rs.3.rs-5424062/v1

[12]. A. Yulianto, P. Sukarno, & N. Suwastika, “Improving adaboost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset”, Paper presented at the Journal of Physics: Conference Series, 2019.

[13]. S. A. Siyyal, F. Y. Khuawar, E. Saba, A. L. Memon, & M. R. Shaikh, “Analyzing ml-based IDs over real traffic”, International Journal of Innovations in Science & Technology, vol. 4, no. 3, pp. 621-640, 2022.

[14]. Peng et al, “An improved feature selection algorithm based on ant colony optimization”, IEEE Access, vol. 6, pp. 69203-69209, 2018.

[15]. E. Popoola, and A. Adewumi, “An Efficient feature selection technique for network intrusion detection system using discrete differential evolution and decision tree”, Int. J. Network Security, vol. 19, no. 5, pp. 660_669, 2017.

[16]. D. Stiawan, M. Idris, A. M. Bamhdi, & R. Budiarto, “CICIDS-2017 dataset feature analysis with information gain for anomaly detection”, IEEE Access, vol. 8, pp. 132911-132921, 2020.

[17]. J. Jose, & D. Jose, “Deep learning algorithms for intrusion detection systems in the Internet of Things using CIC-IDS 2017 dataset”, International Journal of Electrical and Computer Engineering (IJECE), vol. 13, no. 1, pp. 1134-1141, 2023.

[18]. A. Halimaa, & K. Sundarakantham, “Machine learning-based intrusion detection system”, *Paper presented at the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. 2019.

[19]. Singh et al. “Optimization of FAR in intrusion detection system by using random forest algorithm'', SSRN Electron. J., vol. 5, pp. 3-6, 2019.

[20]. R. Vigneswaran, K. Vinayakumar, P. Poornachandran, “Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security,” *in (2018) 9th International Conference on Computing, Communication and Networking Technologies*, ICCCNT 2018, Institute of Electrical and Electronics Engineers Inc., Oct. 2018. 10.1109/ICCCNT.2018.8494096

[21]. A. S. Al-Hurdi, M. F. Abdullah, “Learning to Detect: An Enhanced DNN Model with Adaptive Attention for Classifying DDoS Traffic,” Engineering and Technology Journal, vol. 5, no. 6, pp. 5425-5431, 2025. DOI: 10.47191/etj/v10i06.14, I.F. – 8.482.

BIOGRAPHIES OF AUTHORS

Professor. Mohammed Fadhl Abdullah is currently a professor of computer engineering in the Faculty of Engineering at Aden University in Yemen. He received his Master's and Ph.D. degrees in computer engineering from the Indian Institute of Technology, Delhi, India, in 1993, and 1998. He was the editor-in-chief of Aden University Journal of Information Technology (AUJIT). He is a founding member of the International Center for Scientific Research and Studies (ICSRS). His main research interests are in the fields of machine learning, parallel algorithms, and cybersecurity. He can be contacted at email: m.albadwi@ust.edu, or al_badwi@hotmail.com.

Ahmed Saleh Khaled was born in May 1981 in Lahj Republic of Yemen completed secondary education in Taiz 2000 and earned his bachelor’s degree in computer science from the University Science of Technology- in 2004- Taiz and completed Master from Arab Academy 2023. He is currently pursuing a Ph.D. in Information Technology at the University of Science & Technology, Aden. He is contacted at: *aalhurdi@gmail.com*,

https://doi.org/10.5281/zenodo.15771144