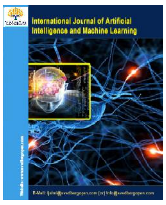
SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/



Research Paper Open Access

Intelligent Data Analytics Framework Using Ensemble-Attention Based Deep Learning Approach for Network Intrusion Detection

Manoj Kumar Prabakaran[1*], Abinaya Devi Chandrasekar[2] and M. Parvathy[3]

[1]Assistant Professor (Sr.Grade), Department of Artificial Intelligence and Data Science, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India. E-mail: manojkumarp@mepcoeng.ac.in

[2]Assistant Professor, Department of Computer Science and Engineering, Mepco Schlenk Engineering College, Virudhunagar, Sivakasi, Tamil Nadu, India.. E-mail: abinayadevi@mepcoeng.ac.in

[3]Professor, Department of Computer Science and Engineering, Sethu Institute of Technology, Madurai 115, Tamil Nadu, E-mail: parvathydurai2002@gmail.com

Article Info

Volume 6, Issue 1, January 2026
Received : 09 September 2025
Accepted : 29 December 2025
Published : 20 January 2026
doi: 10.51483/IJAIML.6.1.2026.18-36

Abstract

The exponential growth of the internet and the proliferation of internet-connected devices lead to the exhaustion of IPv4 addresses. To address this challenge IPv6 was introduced. ICMPv6 plays an important role in IPv6 and is prone to security vulnerabilities because it can be exploited for Distributed Denial of Service (DDoS) Attacks. Attackers can flood the network with the ICMPv6 messages to cause network disruptions. Henceforth, in our work, we introduce an Ensemble-Attention Hybrid Deep Learning Approach for detecting ICMPv6 messages flooding on IPv6 networks. An ensemble feature selection technique is incorporated, that combines filter and wrapper methods to identify essential features. To augment model precision, a transformer-based self-attention mechanism is employed to ascertain attention weights assigned to the selected features. By concatenating the ensemble feature selection with the self-attention mechanism, a Convolutional Neural Network (CNN) model is deployed to surpass the performance of existing methodologies. The experimentation on a benchmark dataset is carried out and the evaluation is based on metrics including False Positive Rate (FPR), detection accuracy, F-measure, recall, and precision. The proposed approach was evaluated on a benchmark dataset, achieving impressive performance metrics with a False Positive Rate (FPR) of 0.16%, detection accuracy of 99.87%, an F-measure of 99.85%, recall of 99.84%, and precision of 99.86%, demonstrating its effectiveness and reliability. Additionally, the findings indicate that the proposed approach surpasses the performance of existing methodologies.

Keywords: *ICMPv6 DDoS attacks, Machine learning, Deep Learning, Self-attention*

1. Introduction

In recent decades, the rapid creation of internet-connected devices and the widespread adoption of digital

* *Corresponding author: Manoj Kumar Prabakaran, Assistant Professor (Sr.Grade), Department of Artificial Intelligence and Data Science, Mepco Schlenk Engineering College, Sivakasi. E-mail: mano.btechme@gmail.com*

services have fundamentally transformed communication and information exchange. The surge in connected devices has led to an impending exhaustion of the available pool of IPv4 addresses (Caicedo *et al.*, 2009). To surmount these challenges and to enable the sustained growth of the digital landscape, the development and adoption of Internet Protocol version 6 (IPv6) have emerged as a pivotal solution (Radhakrishnan *et al.*, 2007). IPv6, characterized by its vastly expanded address space provides the foundation for accommodating the ever-expanding array of internet-enabled devices and services. This transition to IPv6 not only addresses the immediate scarcity of addresses but also engenders a host of benefits, including enhanced security mechanisms (Shiranzaei and Rafiqul, 2018).

The use of sophisticated machine learning algorithms to identify risks like DDoS attacks in IPv6 environments, however, presents significant ethical questions as IPv6 use increases. Techniques like data anonymization and encryption can help reduce potential threats while maintaining data security and privacy. Models must also be resistant to abuse and hostile manipulation using techniques like adversarial training. Explainable AI (XAI) techniques and regular updates are essential for preserving equity and openness, and responsible deployment is ensured by adherence to legal requirements like GDPR. These moral precautions are necessary to guarantee that the model fulfils its intended function without jeopardizing privacy or security.

IPv6 has introduced ICMPv6 as a crucial protocol within its framework. ICMPv6 plays a vital role in IPv6, as the functionality it provides is integral to the proper operation of the protocol. Without ICMPv6, IPv6 would be unable to function effectively, as ICMPv6 messages serve essential functions in enabling data routing along network paths through various nodes. Despite its importance, ICMPv6 exhibits certain limitations that expose the IPv6 protocol to various security threats. Numerous studies have demonstrated the vulnerability of ICMPv6 to various types of attacks, with one of the most severe being Denial of Service (DoS) and Distributed DoS (DDoS) attacks**.** Common ICMPV6-based attack types include Neighbor Discovery (ND) exhaustion, which overwhelms routing tables with fake solicitation messages; ICMPv6 flooding, saturating networks with Echo Request / Reply packets; Router Advertisement spoofing, misconfiguring routing tables; and reflection / amplification attacks, magnifying traffic toward a victim. DDoS attacks remain difficult to mitigate in existing systems, particularly in-home networks including various Internet of Things (IoT) devices (Cvitic *et al.*, 2021), web Enabled Computing Platforms (Singh and Gupta, 2022), Software Defined Networks (SDNs) (Mishra *et al.*, 2021), etc. The primary objective of such attacks is to overwhelm the targeted victim's machine by inundating it with a high volume of packets, consuming its resources and bandwidth, and ultimately rendering it nonfunctional. These attacks exploit IPv6's critical functionalities, posing challenges due to its vast address space, protocol complexity, and limited historical data**.**

The challenge in detecting ICMPv6-based DDoS attacks by IPv4 Intrusion Detection Systems (IDSs) is rooted in the structural disparities between these two protocols (Tiwari *et al.*, 2022). Numerous studies have delved into the intricacies of ICMPv6-DDoS attacks. Researchers have adopted two primary methods for identification: Signature-Based IDS (SIDS) and Anamoly-Based IDS (AIDS) (Elejla *et al.*, 2018; Bahashwan *et al.*, 2021; Alsadhan *et al.*, 2018; Anbar *et al.*, 2016). SIDSfor IPv6 are confined to recognizing attacks with pre-existing signatures, resulting in reduced accuracy when confronted with emerging IPv6 threats lacking established signatures. A primary concern is the insufficiency of comprehensive and diverse IPv6-specific datasets essential for training AIDSs (Tayyab *et al.*, 2020).

Confronted with these challenges, our study proposes Ensemble-Attention, a hybrid deep learning framework that employs deep learning techniques for the effective detection of ICMPv6 flooding DDoS attacks. Moving away from conventional reliance on outdated datasets, this approach leverages the capabilities of deep neural networks. This empowers Intrusion Detection Systems (IDS) to accurately identify evolving attack patterns, ensuring the uninterrupted functionality of services within IPv6 networks. The following are the key contributions of our research:

i) Our study introduces an innovative ensemble approach that integrates feature extraction methods, enhancing the ability to effectively identify ICMPv6 Distributed Denial of Service (DDoS) flooding attacks by discerning critical traffic features.

ii) Employing a transformer-based self-attention mechanism, our methodology assigns attention weights to selected features, elevating the accuracy of attack detection through the nuanced allocation of significance levels to distinct features.

iii) We orchestrate the fusion of an optimized deep learning-based classifier with an ensemble feature selection strategy. This strategy involves extracting features from eight distinct mechanisms, subjecting them to a feature intersection operation and subsequent weight calculation, resulting in a comprehensive and robust methodology for Intrusion Detection Systems (IDS).

The rest of the paper is organized as follows. Section 2 presents an in-depth review of the existing literature in the field of ICMPv6-based DDoS attack detection, providing insights into the current research landscape. Detailed explanations of the proposed methodologies are provided in this Section 3. It covers the ensemble approach, transformer-based self-attention mechanism, and the integration of a deep learning-based classifier. Section 4 outlines the experimental setup, including the conducted experiments and the subsequent analysis of results. It serves to validate the efficacy of the proposed methodologies through empirical evidence. Section 5 summarizes the key findings and implications drawn from the study. Additionally, it suggests potential avenues for future research to advance the field further.

2. Literature Review

Intrusion Detection Systems (IDS) play a critical role in network security by identifying and mitigating various types of cyberattacks, including DoS and DDoS attacks, on both IPv4 and IPv6 networks. These systems utilize a range of algorithms to achieve their goals, with Machine Learning (ML) and Deep Learning (DL) algorithms standing out as some of the most effective approaches for accurate attack detection. However, it’s important to note that there are various existing IDS, each with its strengths and limitations. Researchers in the recent past have proposed two broad categories of IDS models for ICMPV6-based DoS and DDoS detection namely Signature-Based IDS (SIDS) and Anomaly-Based IDS (AIDS) respectively (Khraisat *et al.*, 2019).

2.1. Signature-Based IDS (SIDS) for ICMPV6-base DDoS and DDoS Detection

Signature-based IDS rely on predefined attack patterns or signatures to detect known attacks. While effective against well-documented attacks, they struggle with zero-day attacks and require frequent updates to maintain accuracy. However, challenges have been encountered, including addressing ICMPv6 attacks, particularly in cases involving packet fragmentation and padding beyond six octets (Atlasis, 2012).

2.2. Anomaly-Based IDS (AIDS) for ICMPV6-based DoS and DDoS Detection

Anomaly-based IDS, including AIDS, focuses on identifying deviations from normal network behavior. They build a profile of normal network activity and trigger alerts when anomalies are detected. In recent years, researchers have effectively applied machine learning and deep learning techniques to detect ICMPv6-based DoS and DDoS attacks. However, a major challenge lies in the limited availability of benchmark datasets for mitigating such attacks. To address this (Elejla *et al.*, 2018) pioneered the creation of a flow-based dataset on a real IPv6-enabled network topology. In addition to dataset creation, the selection of features is crucial for effective attack detection. Initially, researchers employed techniques such as the Information Gain Ratio (IGR) and Principal Component Analysis (PCA) to select distinctive features from input datasets (Anbar *et al.*, 2018; Anbar *et al.*, 2016). In particular (Anbar *et al.*, 2018) focused on detecting Router Advertisement (RA) based DDoS flooding attacks, utilizing the Support Vector Machine (SVM) algorithm for the predictive model. The approach, tested on an IPv6 dataset, demonstrated a high detection accuracy of 98.55% and a low false-positive rate of 3.3%. Similarly (Anbar *et al.*, 2016) devised an alternative method for detecting ICMPv6 echo request DDoS attacks, employing a Back-Propagation Neural Network (BPNN) algorithm with the selected features, achieving a detection accuracy of 98.3%.While these research efforts highlight the effectiveness of feature selection in DDoS attack detection, it is noteworthy that these studies focused on specific DDoS attack types (RA, echo request) and may not address other categories of DDoS attacks.

Recently (Elejla *et al.*, 2022) proposed an ensemble feature selection technique employing chi-square and information gain methods to identify significant features for attack detection. They utilized Long Short-Term Memory (LSTM) to evaluate these selected features. Mishra *et al.* (2022) proposed a computational intelligence and majority vote-based ensemble approach for detecting Distributed Denial of Service (DDoS) attacks, demonstrating improved detection accuracy. Conversely (El Ksimi *et al.*, 2024) conducted a study presenting an intelligent system for detecting ICMPv6-based Distributed Denial of Service (DDoS) flooding attacks, leveraging an artificial neural network. The outcomes demonstrated the framework's proficiency in identifying ICMPv6 DDoS flood assaults, achieving detection accuracy rates of 85.91% for the dataset (Elejla *et al.*, 2016).

In summary, the literature review delves into the diverse landscape of Intrusion Detection Systems (IDS) for DoS and DDoS attacks. The exploration spans SIDS solutions like Snort, Zeek, and Suricata, facing challenges in IPv6-specific attack detection. AIDS leveraging ML / DL algorithms, addresses dataset scarcity with initiatives like flow-based datasets by (Elejla *et al.*, 2018; Manickam *et al.*, 2022).

However, existing solutions for mitigating ICMPV6-based DDoS attacks suffer from several drawbacks, which include vulnerability towards dynamic DDoS attack behavior, reliance on a suboptimal feature selection process, resulting in computational overhead and poor detection outcomes, and a lack of robustness against adversarial data samples. These characteristics cause the model to produce a greater false positive rate, resulting in poor classification between genuine and attack traffic data. To address the identified research gaps, our study proposes an optimal Hybrid feature selection approach that incorporates a self-attention mechanism to effectively select and weigh intrinsic DDoS attack-based features, thereby improving the model's detection ability dynamically.

3. Methodology

The core intention of our research is to propose an intelligent detection approach that could precisely identify ICMPv6-basedDDoS attack types in a real-world environment. Our fundamental objective is to adapt advanced deep learning-based architecture that incorporates unique feature selection and feature weight calculation techniques for optimal DDoS attack detection. Hence to attain our objectives, we have proposed an Ensemble-attention approach that executes the following operations:

- Initially, the input data is collected from a benchmark dataset and the label values are converted into binary values namely 0 and 1 (0-Normal and 1-Attack).
- Then, we have adapted an ensemble feature selection mechanism that combines both filter and wrapper methods and select the features based on the threshold.

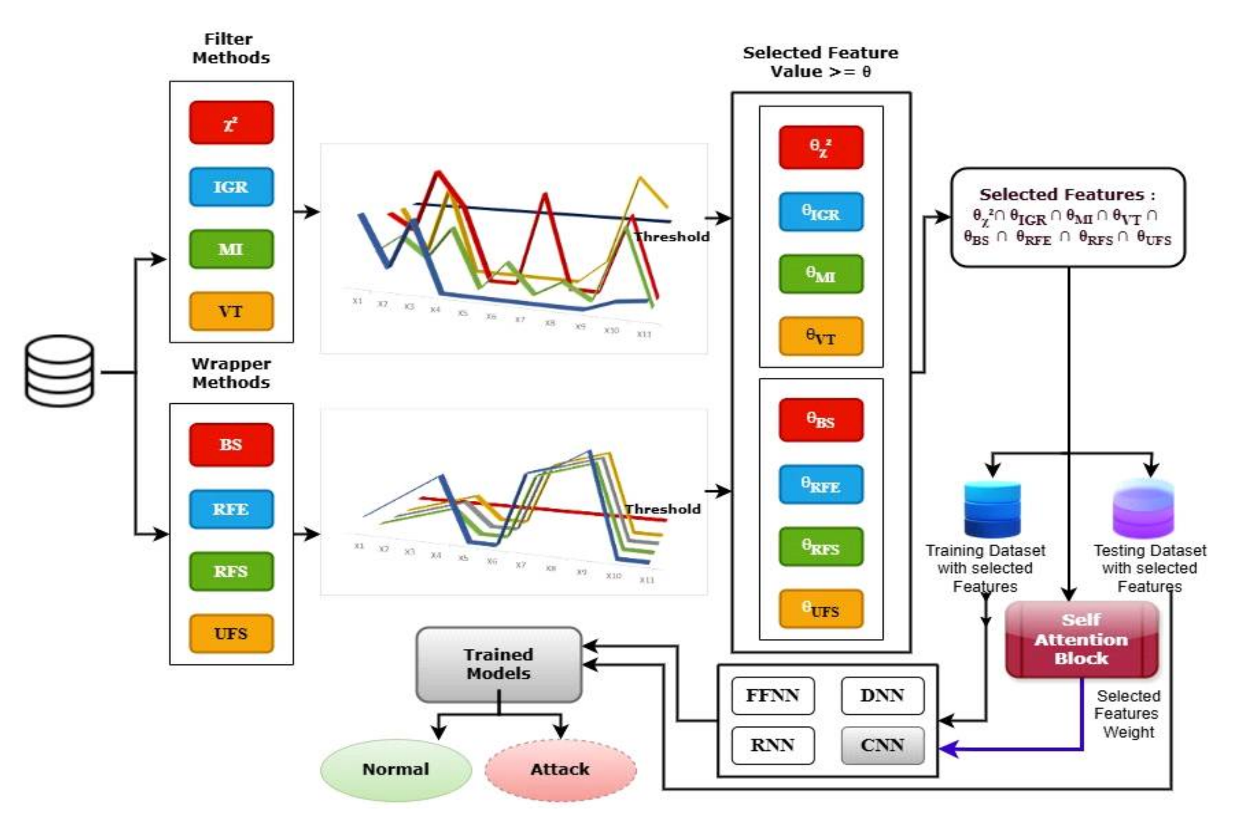

Filter Methods
χ^2
IGR
MI
VT
Wrapper Methods
BS
RFE
RFS
UFS
Threshold
X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11
Threshold
X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11
Selected Feature Value >= θ
θ_{χ^2}
θ_{IGR}
θ_{MI}
θ_{VT}
θ_{BS}
θ_{RFE}
θ_{RFS}
θ_{UFS}
Selected Features : $\theta_{\chi^2} \cap \theta_{IGR} \cap \theta_{MI} \cap \theta_{VT} \cap \theta_{BS} \cap \theta_{RFE} \cap \theta_{RFS} \cap \theta_{UFS}$
Training Dataset with selected Features
Testing Dataset with selected Features
Self Attention Block
Selected Features Weight
FFNN
DNN
RNN
CNN
Trained Models
Normal
Attack

Figure 1: Steps Involved in the Proposed Method

- The selected sets of features were fed as input to a transformer-based self-attention framework that calculates an attention score for each selected feature. Attention score determines the importance of each selected feature concerning all the other features derived.
- Both the selected features obtained through the ensemble feature selection technique and the feature weight vectors received from the self-attention module were concatenated and provided as input to the final layer for classification.
- A traditional Convolutional Neural Network (CNN) architecture is deployed at the final layer for classification. With the concatenated inputs received from the previous layers, the CNN model gets trained to make predictions on the input traffic. The steps involved in the proposed ENSEMBLE-ATTENTION framework are depicted in Figure 1.

3.1. Data Collection

The input data is obtained from a benchmark dataset referred to as "Labelled Flow-based Dataset of ICMPV6-based DDoS attacks" which is publicly available at *https://sites.google.com/site/flowbaseddatasets* and proposed by Elejla *et al.* (2016).

A total of 11 flow-based features relevant to ICMPv6-based DDoS attacks were constructed in the chosen dataset. Table 1 represents the description of the features available in the dataset. The core reasons behind choosing the flow-based dataset for our experimentation are as follows: a) The normal traffic data of this dataset is derived from a real-life network laboratory which makes it more reliable for evaluation as well and the attack traffic data were captured in a virtual environment using THC and SI6 toolkit. b) This dataset is widely used by researchers in the area of ICMPv6 DDoS attack detection and has proposed promising detection models (Elejla *et al.*, 2019; Elejla *et al.*, 2022; El Ksimi *et al.*, 2024). c) The feature sets were labelled based on the traffic type, normalized to non-numerical values, and perfectly balanced equating the proportions of attack and normal traffic inputs using Synthetic Minority Oversampling Technique (SMOTE).

The flow-based ICMPV6 dataset was accessible in two distinct CSV files. Table 1 provides the specifics of the proposed work, which combines the dataset and uses 70% for training and 30% for testing.

Table 1: Description of Features in the Dataset

Dataset Description	
Total Number of Rows	1, 85, 757
Categorical Data	{Normal, DDoS}
Number of Normal traffic	1, 01, 265
Number of Attack traffic	84, 492
Number of features	11
Feature list	ICMPv6Type, PacketsNumber, TransferredBytes, Length_STD, FlowLabel_STD, HopLimit_STD, TrafficClass_STD, NextHeader_STD, PayloadLength_STD, duration, Ratio

3.2. Feature Selection Using Filter and Wrapper Methods

Feature selection is a critical preprocessing step in machine learning that involves selecting a subset of relevant features from the original set of features in a dataset. It improves model performance by selecting the most informative features. By focusing on relevant features, models can better capture the underlying patterns in the data and make more accurate predictions. Removing irrelevant or redundant features can also reduce the risk of overfitting. High-dimensional datasets with a large number of features relative to the number of samples are prone to overfitting. Feature selection mitigates this risk by reducing the complexity of the model and preventing it from fitting noise in the data. By focusing on the most informative features, feature selection can reveal the underlying relationships between the input variables and the target variable, providing valuable insights into the problem domain. The research is motivated by the desire to improve the accuracy, efficiency, and interpretability of ICMPv6-based DDoS attack detection algorithms through the extraction of relevant features using ensemble filter and wrapper methods. By addressing these objectives, the research aims to contribute to the advancement of network security and the development of more effective DDoS attack detection systems.

To achieve this, we have adopted an ensemble feature selection mechanism that combines 2 broad feature selection procedures namely filter and wrapper methods. Figure 1 represents a detailed description of the adopted ensemble feature selection methodology. As can be witnessed in Figure 1, exactly 8 techniques associated with filter and wrapper methods were combined to select the final set of features. As can be inferred from Figure 1, our model incorporates a hybrid approach by combining multiple feature selection methods with intent to overcome the limitations associated with each technique and for ensuring a reliable feature selection mechanism.

In particular, Filter-based methods adopts statistical properties for assessing the relevance of features whereas Wrapper-based methods consider the performance of the model as an important criterion feature subset assessment offering a more targeted evaluation albeit at a higher computational cost.

The complementary strengths associated with both the methods leads to the rationale behind integrating them. During the initial phase, filter-based methods are deployed for feature space reduction thus eradicating irrelevant features from the input vector. This phase significantly reduces the computational complexity associated with wrapper-based methods which in turn ease the fine-tuning process of wrapper-based methods to identify optimal set of features for classification.

Our hybrid feature selection approach combining filter and wrapper methods provides a balance between detection accuracy and computational efficiency although their working principle differs among each other. This strategy eliminates overfitting, noise reduction and ensures that the selected features have a significant impact on model's classification performance. The effectiveness of this combination is supported by experimental evaluations, which demonstrate its superiority over using either method alone.

3.2.1. Filter Methods

In the case of filter methods, the features are selected based on statistical measures that were conducted to calculate the relevance of the input features without the interference of any machine learning / deep learning algorithms. Four such methods, the Chi-square test, Information gain, Mutual information, and variance threshold, are adopted to implement the filter method. Let X be the features of the input dataset represented as

$$X = \{X_1, X_2, X_3, \ldots, X_m\} \quad \text{...(1)}$$

Where m = 11 indicates the number of input features.

The chi-square test is a fundamental statistical tool to assess the association between categorical variables. Each of the filter methods calculates a statistical score for individual features. The chi-square test provides an inference on the relationship associated with feature inputs concerning the target variable and creates a statistical score for each evaluated feature. Information gain is a fundamental concept in machine learning and data analysis, particularly in the context of feature selection.

By quantifying the reduction in uncertainty about the target variable achieved by considering a particular feature, information gain helps identify the most informative features for predictive modelling. Mutual Information (MI) is a powerful concept in information theory used to quantify the amount of information shared between two random variables. A variance threshold is a technique that eradicates the features that do not meet a threshold value.

3.2.2. Wrapper Methods

The core principle behind wrapper methods is to consider the feature selection process as a search problem. It involves using a specific machine learning model to evaluate the quality of different subsets of features. Four such techniques namely backward selection, recursive feature elimination, recursive feature selection, and univariate feature selection were adopted in our work.

Table 2: Selection of Features Using Filter-Based Methods and Wrapper-Based Methods

Features Name	**X1**	**X2**	**X3**	**X4**	**X5**	**X6**	**X7**	**X8**	**X9**	**X10**	**X11**
Feature Extraction Methods	**ICMPv6Type**	**PacketsNumber**	**TransferredBytes**	**Length_STD**	**FlowLabel_STD**	**HopLimit_STD**	**TrafficClass_STD**	**NextHeader_STD**	**PayloadLength_STD**	**Duration**	**Ratio**
Chi-square	✓	✓	✓	✓					✓		✓
Information Gain	✓	✓	✓	✓					✓	✓	✓
Mutual Information	✓	✓	✓	✓		✓		✓		✓	
Variance Threshold	✓	✓	✓						✓	✓	✓
Filter Method	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓	✓
Backward Selection	✓	✓	✓	✓					✓	✓	✓
Forward Selection	✓	✓	✓	✓					✓	✓	✓
Recursive feature selection (RFS)	✓	✓	✓	✓					✓	✓	✓
Univariate feature selection (UFS)	✓	✓	✓	✓					✓	✓	✓
Wrapper Method	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓
Ensemble Feature Selection	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓

3.2.3. Ensemble Feature Selection

The basic idea is that by combining the predictions of several weak learners (models that perform slightly better than random chance), the ensemble can create a strong learner that outperforms any individual model. Table 2 provides a selection of features using filter-based methods and wrapper-based methods. To identify a common set of features as the target output, a frequency analysis technique is adopted in which the selection count of all the input features across all four filter methods is calculated. The frequency analysis of features available in a dataset concerning the filter methods. Those features with poor selection count were eliminated and a final set of features that meets our experimental threshold say (th = 2) is selected and generated as the final set of output features from filter and wrapper methods.

Ultimately, a new set of features is chosen based on the intersection of two subsets of features from the filter and wrapper methods. The dimensions of the final set of selected features are N * 7, where N represents the number of input data, and the features selected from the dataset is displayed in Table 2.

3.3. Feature Weight Calculation Using Self-Attention Mechanism

In our work, the selected features obtained through the ensemble feature selection process will be passed on to the successive layer for the feature weight calculation process as shown in Figure 2. In this phase, the selected valiant set of features will be weighed using a transformer-based mechanism to assess the importance of each feature concerning all the features in the selected set. This is done based on the concept of a self-attention mechanism (Vaswani *et al*., 2017) that enables a particular model to weigh the importance of different elements within a sequence in a context-aware manner. This mechanism is often associated with the Transformer architecture, which has revolutionized various NLP tasks. We have adopted such a mechanism to weigh the importance of each feature selected in our work. Self-attention mechanism includes 3 trainable parameters namely W^q, $W^{k,}$ and W^v which are referred to as weight matrices of query, key, and values respectively.

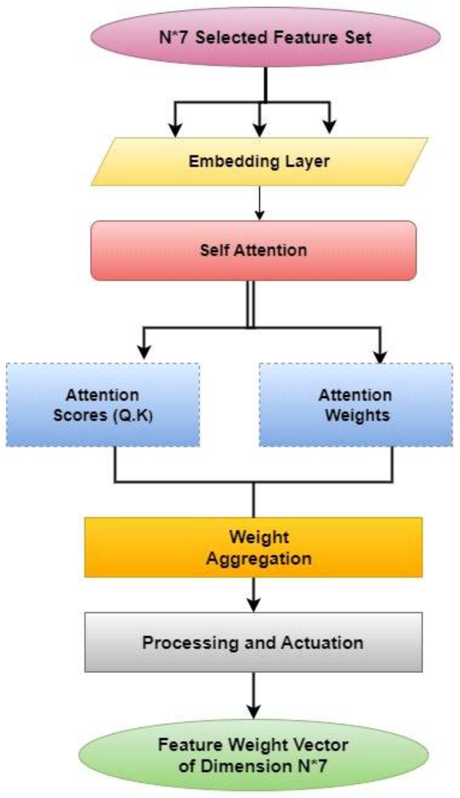


Figure 2: Feature Weight Calculation Block Using Self-Attention Mechanism

These matrices were randomly initialized and multiplied along with the input embedding (X_i's) to produce query, key, and value vectors that are considered to be the modified version of the input embedding X_i. The query vector of the current input, say Q_i, and the key vectors K_j. T of all the previous inputs are combined to create a dot product, which is then used to determine the attention weights for an input.

Query = $W_q.X_i$ ...(2)

key = $W_k.X_i$ ...(3)

Value = $W_v.X_i$ ...(4)

This is a similarity or compatibility metric, equivalent to the simplified version (multiplicative attention). Therefore, the following Equation must be used to calculate the attention weight for the nth input for a given set of inputs:

$$(Q_n, K, V) \rightarrow \sum_{i=1}^{T} \left[\frac{exp\,(Q_n \,.\, K_i^T)}{\sum_j exp\left(Q_n \,.\, K_j^T\right)} \;*\; v_i\right] \qquad ...(5)$$

The weighted aggregate of all the generated dot products is subsequently normalized using a softmax function as deduced in Equation (6) such that the attention weights add up to 1. Finally, to create the attention weight for an input embedding, the attention weight was multiplied by the value vector V_i. Therefore, the model tries to determine which key-value pair for each query it must concentrate on.

The attention weights that are computed based on the dot product of the query and key may increase exponentially, causing the softmax gradient to become insufficient. To solve this problem, a method known as scaled dot product attention is used, in which scaling is carried out to prevent the softmax from having too sharp distributions. The attention weights are determined using the following formula based on the scaled dot product attention:

$$\text{Self-Attention (Q, K, V)} \rightarrow \text{softmax}\left(\frac{Q.k^T}{\sqrt{d_k}}\right).V \qquad ...(6)$$

The resulting attention scores were divided by the square root of the input embedding dimension dk to prevent scaling of the attention weights. Attention weights for inputs concerning all other inputs must be calculated using the self-attention process. This will make it easier for a phishing detection algorithm to recognize the internal connections between the different characters in URL data.

The detailed steps involved in calculating attention weight vectors for selected DDoS based traffic features are represented in Algorithm 1.

Algorithm 1: Attention Weight Calculation Using Self-Attention Mechanism

Input: Selected feature matrix says $N \in I^{MXD}$, where N is the number of samples, I is the input dataset, M is the Number of samples and D is the dimension of the features.

Output: Feature weight vector Z of dimension MXD similar to the input vector dimension.

```
Initialize weight matrices Wq, Wk, Wv
//Compute query, key, and value matrices:
    $Q = N * Wq$
    $K = N * Wk$
    $V = N * Wv$
//Calculate raw attention scores:
    $AS = Q * K^T$
//Scale the attention scores (AS):
    $AS = AS / \sqrt{d}$  // d is the feature dimension
//Normalize attention scores using Softmax:
for i = 1 to M do
    $AS[i] = \exp(AS[i]) / \text{sum}(\exp(AS[j]) \text{ for } j = 1 \text{ to } M)$
end for
//Compute attention-weighted features:
    $FW = AS * V$
return FW as Z
```

At the initial stage of the architecture, the input feature set is duplicated into three distinct sets: query, key, and value. This tripartite separation enables the self-attention mechanism to learn relationships between features, allowing each token to serve different roles within the mechanism. Following the input stage, the feature sets-query, key, and value are each subjected to an embedding layer. This layer transforms the discrete feature values into continuous embedding vectors. These embeddings encode semantical information about the features, facilitating their further manipulation within the mechanism.

The association between each query token and the key tokens is calculated by the self-attention process. By comparing each characteristic to the whole set of important features, the mechanism determines the contextual importance of each feature through this procedure.

The technique can capture dependencies between features that contribute to the overall dataset structure by taking contextual information into account systematically.

The dot product between the query and key embeddings is used to determine the attention scores. The generated scores capture the degree of their connection by quantifying the similarity and interaction between the query and key tokens. After employing a SoftMax function to normalize the attention scores, attention weights are generated. These weights reflect the significance of each key token with the corresponding query token. The weights are guaranteed to add up to one for each query token owing to this normalization. This vector contains the computed importance weights that were assigned to each feature in the dataset, indicating their relative significance with the self-attention process. The obtained feature weight vectors are of the same dimension as that of the input feature vector.

3.4. Classification Using Convolutional Neural Network (CNN)

The feature vectors selected using the ensemble feature selection mechanism and the feature weight matrix vector obtained through the self-attention mechanism were passed on through a concatenation layer to flatten them into a single vector of dimension N * 14, where N is the total number input data. The concatenated feature vectors were passed on to the final layer of our proposed framework for classification. A traditional Convolutional Neural Network (CNN) architecture is deployed as a classifier in our framework. The CNN model is structured as shown in Figure 3.

As represented in the Figure 3, the model comprises an input layer that receives the 14 feature inputs followed by a convolutional layer 1 that applies 32 filters with a kernel size of 3 to the input data. This convolutional operation extracts local patterns and relationships within the feature space. The Rectified Linear Unit (ReLU) activation function is applied element-wise to the output of this layer, introducing non-linearity. Following the first convolutional layer, the a max pooling operation is applied with a pool size of 2. Max pooling reduces the spatial dimensions of the feature maps, helping to retain important information while reducing the computational complexity.

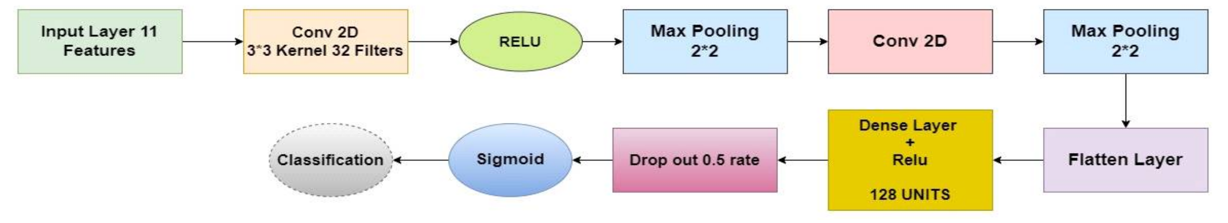


Figure 3: Structure of the Proposed Convolutional Neural Network (CNN)

The second convolutional layer applies 64 filters with the same kernel size of 3. This layer further captures more complex patterns and relationships from the already transformed features. Another max pooling operation with the same pool size of 2 follows the second convolutional layer. This step continues to down-sample the feature maps. After the convolutional and pooling layers, the flattened layer reshapes the multi-dimensional feature maps into a one-dimensional vector.

This prepares the data for the subsequent dense (fully connected) layers. The first dense layer contains 128 units. Each unit is connected to all the outputs from the previous layer. The ReLU activation function introduces non-linearity and allows the network to learn complex relationships between features. Dropout is applied after the first dense layer with a rate of 0.5. Dropout randomly deactivates a certain percentage of neurons during each training iteration, preventing overfitting by promoting the learning of more robust features.

The final dense layer has 1 unit, which represents the output for binary classification. The Sigmoid activation function squashes the output into the range [0, 1], making it interpretable as a probability. This final output indicates the model's prediction of whether the input traffic sample belongs to a DDoS attack class or not.

For training the proposed CNN model, the following hyperparameter values were fixed. The learning rate was set to 0.001 and the batch size was fixed at 64. The total number of training epochs was set to 50 and Adam optimizer was used for optimization and binary cross-entropy was used as the loss function.

4. Experimentation and Result Analysis

The proposed ENSEMBLE_ATTENTION framework is evaluated using a benchmark dataset proposed by (Elejla *et al.*, 2016). Google Colab Pro, a setup-free, cloud-based Jupyter Notebook environment, was utilized to carry out the entire experiment. The proposed model was constructed using Pytorch, an open-source machine learning library associated with Python that allows us to build neural network-based architectures. With a single precision performance of 14 TFLOPS and a memory bandwidth of 900 GB / sec, the Tesla v100 PCIe GPU accelerator is accessible in Google Colab Pro. It is additionally equipped with a 125 GB HDD and 25 GB of memory.

4.1. Experimental Design Phases and Result Analysis

Various phases of experiments were conducted in order to evaluate the aptitude of the proposed model regarding the effective detection of ICMPv6-basedDDoS attacks. Our major contributions to the proposed architecture include the following:

a. Adoption of ensemble feature selection mechanism for optimal selection of flow-drivenICMPv6-basedDDoS features.

b. Implementation of a transformer-based self-attention mechanism to precisely calculate the attention score for the individual features selected based on ensemble feature selection.

c. Deploying a Convolutional neural network model as a classifier module for predicting the nature of the input traffic.

d. Conduct a comparative analysis of the proposed model against established state-of-the-art deep learning-based Intrusion Detection System (IDS) approaches to assess the effectiveness of the architecture. To justify the novelty involved in our proposed framework, the experimentations were carried out in successive phases, with each phase set out to assess the optimality of individual stages involved in our work.

4.1.1. Evaluation of Ensemble Feature Selection Mechanism (Phase I)

In this phase, the efficacy of the features selected using an ensemble mechanism regarding the classifier's prediction outcome is measured. Two unique experiments were carried out separately to validate the selected features.

Initially, the entire module corresponding to the feature selection process is eliminated from the framework design and the original 11 input features present in the input dataset are directly fed to the succeeding weight calculation and classification model for output prediction. In this design, the weight calculation model receives an N *11 feature vector input and produces feature weight vector output of the same dimension. Those feature vectors and their associated weight matrices were then concatenated and fed to the CNN model for classification. Hence the number of input units in the CNN model is restructured in accordance.

This experimental design is set to examine the impact of features selected using an ensemble mechanism in the effective detection of DDoS attacks. The restructured model is evaluated based on the following 2 metrics namely i) Accuracy curve and ii) Loss curve.

Figure 4 represents the detection accuracy curve obtained during the training and evaluation phase of the model that is designed without an ensemble feature selection module. As can be witnessed from Figure 4, the

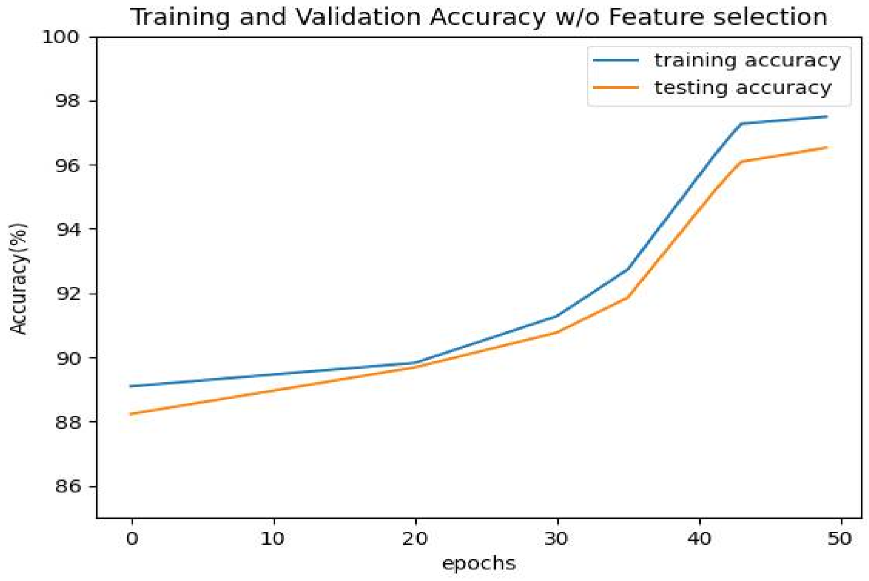


Figure 4: Detection Accuracy of the Proposed Model without Feature Selection

model exhibits a decent detection outcome of 96.58%. However, the detection accuracy of the model became stable after 40 epochs and does not seem to improve any further. Hence, we perform the validation of the proposed model concerning the features selected using the ensemble feature selection mechanism. A total of 7 features were selected as the final set of features from both the filter and wrapper methods and fed as input to the feature weight calculation and output block for classification. Figure 5 represents the detection accuracy of the proposed model with the inclusion of an ensemble feature selection module.

Based on the results obtained from Figure 5, it can be concluded that the model reaches a maximum accuracy of 99.57% during the evaluation phase which exceeds the model's performance in the absence of selected features. This has proven the fact that the adopted ensemble feature selection mechanism has produced a significant set of features that have produced a huge impact on the detection outcome of the classifier.

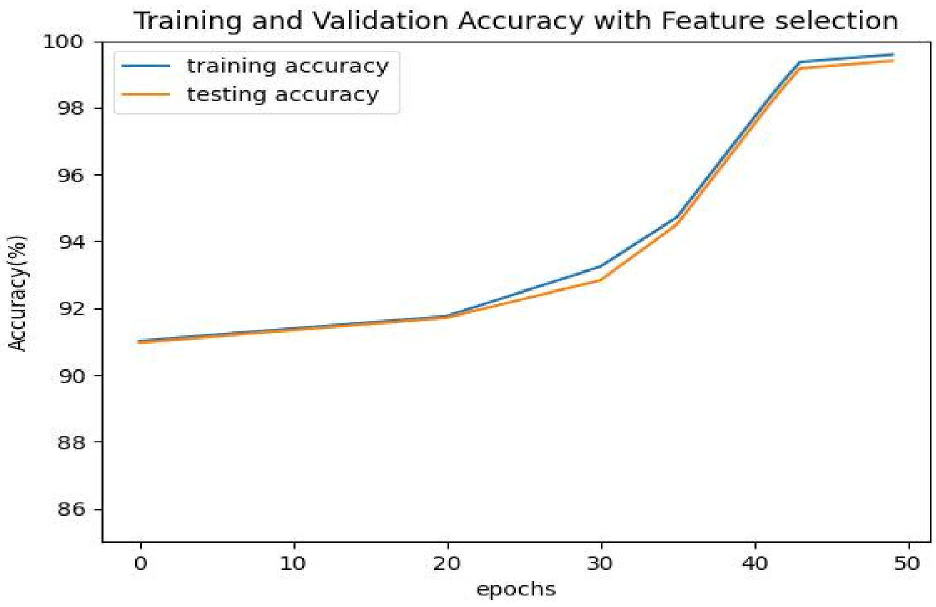


Figure 5: Detection Accuracy of the Proposed Model with Ensemble Feature Selection

4.1.2. Performance Analysis of Feature Weight Calculation Block (Phase II)

In this phase, the performance outcome of the model is assessed for the feature weight calculation block. In particular, we have significantly analyzed the attention score calculated by the self-attention mechanism for each selected feature. This experimentation intends to decide the impact of feature weight vectors in the effective detection of DDoS attacks. Similar to the experimentation conducted in phase I, two separate experiments were carried out including and excluding the feature weight calculation module. This is done in particular to assess the effectiveness of feature weight vectors' ineffective prediction outcome.

Initially, the model without attention block module is constructed and analyzed. In this design, the features selected from the ensemble module were directly fed to the final output layer for training and evaluation. The performance outcome is analyzed based on the following metrics namely Confusion matrix, TPR, TNR, FPR, and FNR. Figure 6 shows the confusion matrix obtained for the model without attention score. Out of the 92812 samples, 91067 samples were correctly identified by the model. More than 1000 attack samples were incorrectly identified as benign ones leading to a quite higher false alarm rate. To analyze the efficacy of attention scores calculated for the selected features, the following experimentation was conducted including the feature weight vector generated using a self-attention block. Hence a total of 14 features including both the selected features and its associated weight vectors were fed to the output block for classification. The confusion matrix depicted in Figure 7 represents the outcome of our proposed framework that employed a transformer-based self-attention mechanism for feature weight calculation. Based on the results inferred from the confusion matrix in Figure 7, it can be observed that only 50 odd attack samples were incorrectly identified and the model exhibits higher TPR and TNR values. Around 99% of the testing samples were correctly identified by our model incorporating the feature weight calculation module.

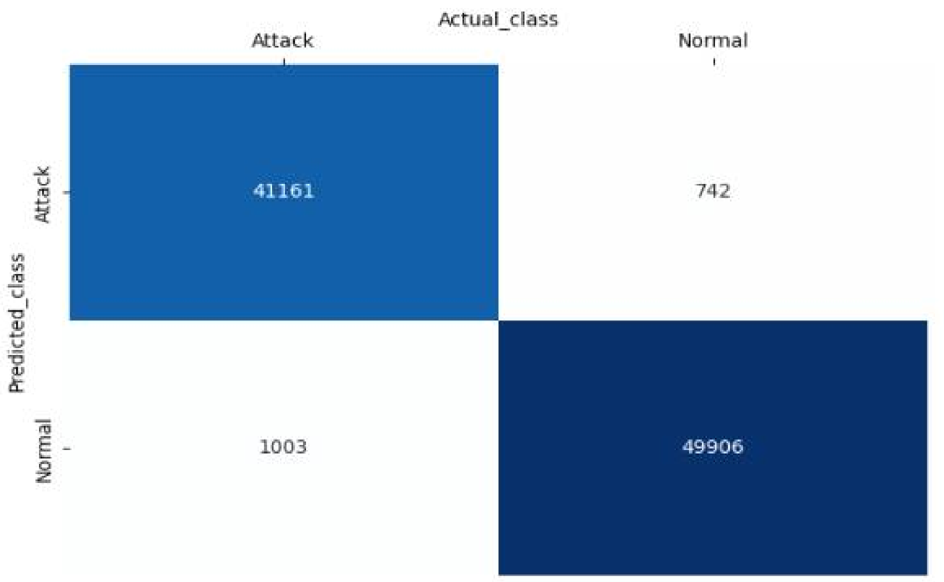


Figure 6: Confusion Matrix for the Proposed Model with Attention Score

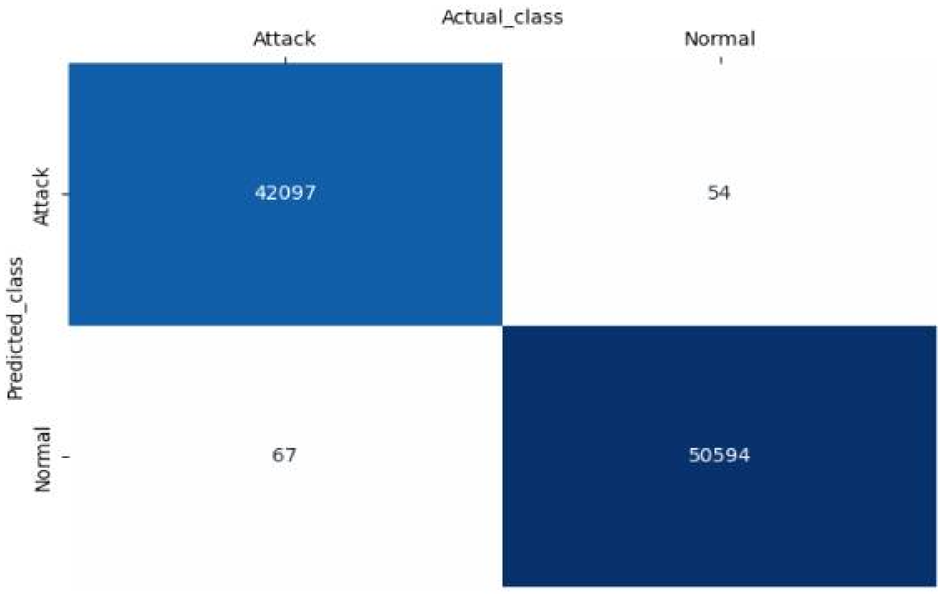


Figure 7: Confusion Matrix for the Proposed Model without Attention Score

These results have proven the fact that attention scores calculated for individual selected features concerning all the other features have made a huge impact on the detection ability of the classifier. This has not only improved the detection accuracy of the model but also significantly reduced the false alarm rate drastically.

Table 3: True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR), and False Negative Rate (FNR) Values for All the Models

	TPR	FPR	TNR	FNR
A model Without a weight calculation module	98.53%	2.38%	97.62%	1.47%
A model with a weight calculation module	99.89%	0.16%	99.84%	0.11%

Table 3 depicts the TPR, TNR, FPR, and FNR values obtained for both the experimented models in design phase II. As can be observed, the proposed model exhibited lower FPR and FNR values of 0.16 and 0.11 respectively. The experimental results obtained from design phase II suggest that the adoption of self-attention mechanism to calculate the importance of each selected feature has significantly reduced the false alarm rate and increased the detection accuracy of the classifier.

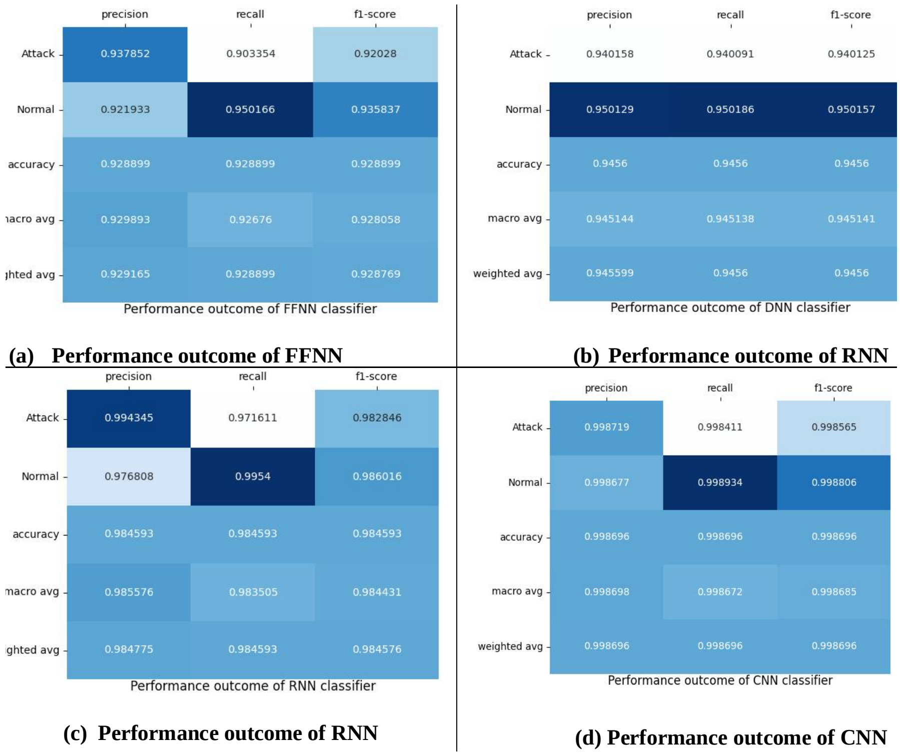
4.1.3. Evaluation of CNN as a Classifier (Phase III)

In this final phase of the experimentation, the output block comprising the classifier model is evaluated. In our work, we have incorporated Convolutional Neural Network (CNN) as the classifier module and the results obtained in the previous phases have proven that the classifier exhibited the maximum accuracy.

However, to justify the selection of the CNN model as a classifier we have further validated our proposed framework concerning on-CNN-based frameworks. Following are the various non-CNN-based neural network models considered for experimental evaluation namely Feed Forward Neural Network (FFNN), Deep Neural Network (DNN), and Recurrent Neural Network (RNN). For this experimentation, we have considered only discriminative deep learning models for fair comparison. The configurations of the experimented models were set by the proposed CNN architecture. FFNN model comprises an input layer, 2 hidden layers, and an output layer. The number of neurons in the input layer corresponds to 14 units followed by the hidden layers consisting of 128 and 64 units respectively. The sigmoid function is used as the activation function in the output layer for output prediction. In the same manner, the DNN model is structured with an additional hidden layer to enhance the depth of the intermediate layers in the model.

The number of neuron units and the activation function in the input and output layers are structured similarly to FFNN. The recurrent Neural Network model comprises intermediate Long Short-Term Module layers with 64 and returns sequences followed by an output layer with a sigmoid activation function. Following are the metrics used for evaluating the experimented classifier models namely precision, recall, F1 score, and accuracy.

In addition to these metrics, a Receiver Operating Characteristics (RoC) curve is plotted to measure the tradeoff between true positive and false positive rates obtained by all the experimented models. Figure 8 shows



	precision	recall	f1-score
Attack	0.937852	0.903354	0.92028
Normal	0.921933	0.950166	0.935837
accuracy	0.928899	0.928899	0.928899
ıacro avg	0.929893	0.92676	0.928058
ghted avg	0.929165	0.928899	0.928769

(a) Performance outcome of FFNN

	precision	recall	f1-score
Attack	0.940158	0.940091	0.940125
Normal	0.950129	0.950186	0.950157
accuracy	0.9456	0.9456	0.9456
macro avg	0.945144	0.945138	0.945141
weighted avg	0.945599	0.9456	0.9456

(b) Performance outcome of RNN

	precision	recall	f1-score
Attack	0.994345	0.971611	0.982846
Normal	0.976808	0.9954	0.986016
accuracy	0.984593	0.984593	0.984593
macro avg	0.985576	0.983505	0.984431
ghted avg	0.984775	0.984593	0.984576

(c) Performance outcome of RNN

	precision	recall	f1-score
Attack	0.998719	0.998411	0.998565
Normal	0.998677	0.998934	0.998806
accuracy	0.998696	0.998696	0.998696
macro avg	0.998698	0.998672	0.998685
weighted avg	0.998696	0.998696	0.998696

(d) Performance outcome of CNN

Figure 8(a-d): Performance Outcomes Concerning the Precision, Recall, F1 Score, and Accuracy of All the Experimented Models

the detailed performance evaluation of all the experimented models deployed as a classifier concerning precision, recall, F1 score, and accuracy. The results reveal that the adoption of CNN as the classifier entity has exhibited the maximum F1 score of 99.85%. Out of the experimented models, FFNN exhibited comparatively lower precision and recall values of around 92%. RNN model claims the second-highest F1 score of 98%, however, the recall value produced for attack inputs was not up to the mark. Hence with the results inferred from Figure 10, it shall be concluded that CNN delivers the highest F1 score making it the most suitable model for classification.

To assess the tradeoff between TPR and FPR associated with all the other experimented models a separate experimentation was conducted and the results obtained were reflected in the plotted RoC curve in Figure 9. As can be witnessed from Figure 9, CNN delivered the highest AUC (area under the curve) score of 0.9860 which is almost close to 1. Figure 10 and Figure 11 depicts the performance outcome of individual classifiers. The results suggested that the models demonstrated commendable performance with high precision and recall, contributing to effective intrusion detection in ICMPv6-based DDoS attacks. This proves the fact that CNN surpasses all the other experimented models concerning TPR, FPR, precision and recall values.

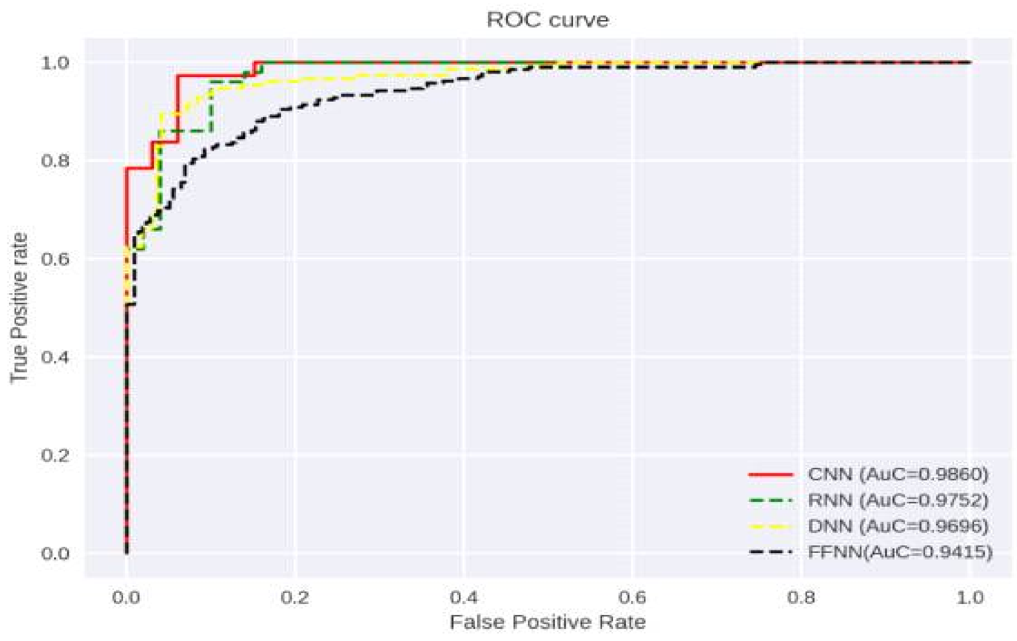


Figure 9: Receiver Operating Characteristics (RoC) Curve for Experimented Models

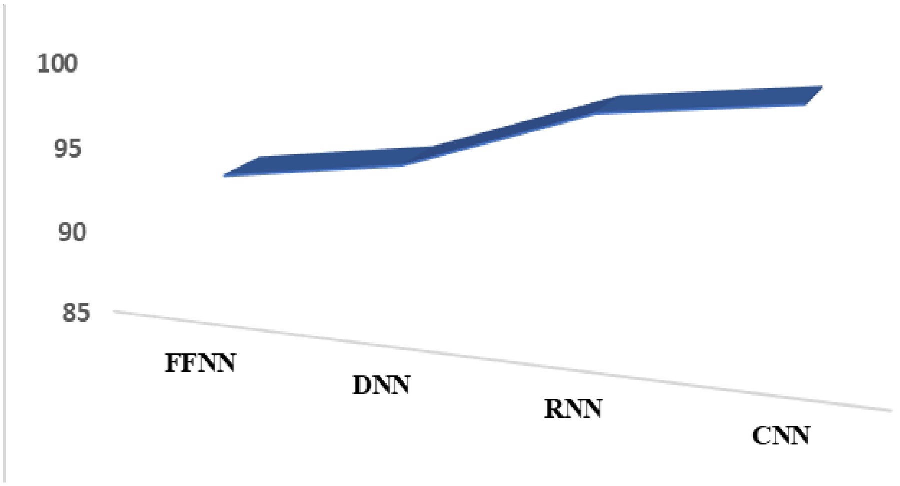


Figure 10: Precision Curve for Experimented Models

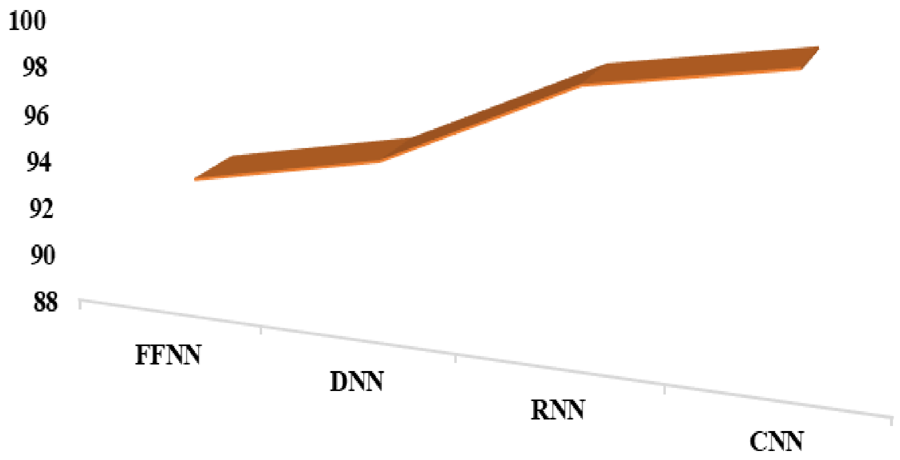


Figure 11: Recall Curve for Experimented Models

4.1.4. Comparative Analysis of the Proposed Architecture

In this section, our study focuses on assessing the effectiveness of our architecture concerning established state-of-the-art methods that have demonstrated promising results in the detection of ICMPv6-based DDoS attacks.

Table 4 shows the experimental results obtained for the proposed approach concerning the specified metrics. Results for the existing approaches (Elejla *et al.*, 2022; Elejla *et al.*, 2019; El Ksimi *et al.*, 2024) have been

Table 4: Metric-Based Comparative Analysis of Detection Models: TNR, FNR, FPR, Accuracy, and F-measure

Approaches	Models	TNR (%)	FNR (%)	FPR (%)	Accuracy (%)	F-measure (%)
Elejla *et al.* (2022)	RNN	87.1	2.3	12.8	92.3	0.92
	LSTM	**99.4**	**2.62**	**0.551**	**98.41**	**0.98**
	GRU	99.11	2.49	0.884	98.31	0.98
Elejla *et al.* (2019)	DT	-	-	0.171	85.7	0.852
	SVM	-	-	0.292	73.5	0.727
	NB	-	-	0.300	74.5	0.724
	KNN	**-**	**-**	**0.171**	**85.7**	**0.852**
	NN	-	-	0.197	83.2	0.826
El Ksimi *et al.* (2024)	RF	-	-	-	79.13	0.85
	ADB	-	-	-	85.15	0.84
	KNN	-	-	-	85.73	0.85
	DT	-	-	-	85.82	0.85
	ANN	**-**	**-**	**-**	**85.91**	0.85
Proposed Approach	FFNN	89.14	1.98	0.186	92.89	0.87
	DNN	95.56	1.35	0.444	94.56	0.91
	RNN	98.14	0.95	0.86	98.45	0.95
	CNN	**99.89**	**0.16**	**0.11**	**99.87**	**0.98**

provided as reported in their work. As can be observed, the proposed CNN model appears to outperform other models, achieving the highest TNR, lowest FNR and FPR, and the highest Accuracy and F-measure.

Experiments were performed under various attack scenarios to show the suggested model's resilience and efficacy. Consideration was given to the following circumstances. a) Low Intensity involves occasional assault traffic in primarily typical network flows. It is helpful for evaluating how sensitive the model is to small disturbances. b) Medium Intensity replicates real-world mixed traffic situations by distributing normal and attack packets equally. c) High Intensity examines the model's resilience to harsh circumstances by introducing a high concentration of attack packets. Table 5 provides the percentage of data included, whereas Table 6 provides the experimental analysis. The results presented in Table 6 demonstrate the model's efficacy in identifying ICMPv6-based DDoS attacks under a variety of circumstances, guaranteeing dependability in real-world scenarios.

Table 5: Percentage of Data for Various Attack Scenarios

Attack Intensity	No. of Attack Packets	No. of Normal Packets	Attack Packets (%)	Normal Packets (%)
Low Intensity	16,898	67,568	20%	80%
Medium Intensity	42,245	42,245	50%	50%
High Intensity	67,568	16,898	80%	20%

Table 6: Experimental Analysis of Various Attack Scenarios

Condition	Accuracy (%)	TPR (%)	FPR (%)	Precision (%)	F1 Score (%)
Low Intensity	99.85	99.80	0.20	99.83	99.81
Medium Intensity	99.87	99.85	0.16	99.86	99.85
High Intensity	99.87	99.89	0.21	99.82	99.80

In summary, the proposed Ensemble-Attention framework proves to be a formidable approach in predictive modeling, capitalizing on the amalgamation of diverse neural network architectures and attention mechanisms. This strategic integration underscores the framework's potential to enhance predictive performance. As observed in our study, the Ensemble-Attention framework exhibits a strong capability to distil valuable insights from the data. This framework not only outperforms individual models but also leverages their collective strengths, offering a promising avenue for enhanced predictive modeling in various applications.

5. Discussion

In particular, the proposed framework adopts ensemble feature selection that combines filter and wrapper methods for selecting optimal subset of features from the input dataset by adopting advanced techniques such as correlation analysis, mutual information etc., that highlights the significance of each feature with respect to the target variable. Combining the features extracted using filter and wrapper methods strengthen the optimality of feature selection process since common set of extracted features were finally selected for evaluation. Although, various studies have been conducted in the recent past incorporating advanced feature selection techniques, there were no particular attempts carried out implementing an ensemble approach combining primal feature selection techniques for ICMPV6 based DDoS attack detection. The experimental outcome as well signifies the effectiveness of our proposed ensemble feature selection mechanism in which separate experimentations were carried out by including and excluding the Ensemble feature selection mechanism. Also, our work focused on building a transformer based self-attention mechanism to validate the importance of each features selected using ensemble feature selection mechanism. Adoption of self-attention mechanism for feature weight calculation strengthens the classifier of our framework with additional information about each features selected in regard to its inherent relationship with all the features in the optimal subset. In order to validate the effect of feature weights obtained using self-attention mechanism, we carried out unique experimentations by assessing the attention score obtained for each feature with respect to classification result. And the results obtained ensure the importance of attention weights associated with each features selected showcasing the dominance of self-attention mechanism in DDoS attack detection. To the best of our knowledge, this represents the first attempt at utilizing self-attention mechanisms to compute feature weight vectors for DDoS traffic features.

6. Conclusion

In the face of an evolving digital landscape characterized by the exponential proliferation of internet-enabled devices and services, the transition from Internet Protocol version 4 (IPv4) to IPv6 has become an imperative step. IPv6's expanded address space provides a solution to the impending scarcity of addresses, offering enhanced security, streamlined routing, and compatibility with modern networking technologies. This transition, however, brings forth new challenges, notably the vulnerabilities inherent in Internet Control Message Protocol version 6 (ICMPv6) messages. ICMPv6 messages set the stage for Distributed Denial of Service (DDoS) attacks that exploit these vulnerabilities. The consequences of such attacks are dire, incapacitating servers and disrupting digital operations, impacting users and organizations alike. In response to these challenges, this research has embarked on a pioneering journey, leveraging the power of deep learning techniques to fortify Intrusion Detection Systems (IDS) against evolving attack patterns, particularly ICMPv6 flooding DDoS attacks. Traditional IDS, often reliant on historical datasets, struggle to identify emerging threats. To bridge this gap, we have proposed an Ensemble-Attention Framework that produces novel contributions. Our proposed framework exhibits remarkable accuracy, achieving 99.89% while maintaining a low false positive rate of 0.16%. However, it's important to acknowledge the complexities inherent in adopting a hybrid deep-learning-based ensemble mechanism. Hence in our future work, we have planned to adapt Bio-Inspired algorithms for the feature selection process which would ease the process by dynamically selecting an optimal subset of features leading to an effective detection framework suitable for real-time deployment.

References

Alsadhan Abeer Abdullah, Abir Hussain and Mohammed M. Alani. (2018). Detecting NDP Distributed Denial of Service Attacks Using Machine Learning Algorithm Based on Flow-Based Representation. in *2018 11th International Conference on Developments in eSystems Engineering (DeSE)*, 134-140, IEEE.

Anbar Mohammed, Rosni Abdullah, Bassam Naji Al-Tamimi and Amir Hussain. (2018). A Machine Learning Approach to Detect Router Advertisement Flooding Attacks in Next-Generation IPv6 Networks. *Cognitive Computation,* 10, 201-214.

Anbar Mohammed, Rosni Abdullah, Redhwan M.A. Saad, Esraa Alomari and Samer Alsaleem. (2016). Review of Security Vulnerabilities in the IPv6 Neighbor Discovery Protocol. in *Information Science and Applications (ICISA),* 603-612, Springer, Singapore.

Atlasis, Antonios. (2012). Security Impacts of Abusing IPv6 Extension Headers. in *Black Hat Security Conference,* 1-10.

Bahashwan Abdullah Ahmed, Mohammed Anbar, IznanHusainy Hasbullah, Ziyad R. Alashhab and Ali Bin-Salem. (2021). Flow-Based Approach to Detect Abnormal Behavior in Neighbor Discovery Protocol (NDP). *IEEE Access,* 9, 45512-45526.

Caicedo Carlos E., James B.D. Joshi and Summit R. Tuladhar. (2009). IPv6 Security Challenges. *Computer,* 42(2), 36-42.

Cvitic, I., Perakovic, D., Gupta, B.B. and Choo, K.K.R. (2021). Boosting-Based DDoS Detection in Internet of Things Systems. *IEEE Internet of Things Journal*, 9(3), 2109-2123.

El Ksimi, A., Leghris, C., Lafraxo, S. and Verma, V.K. (2024). Icmpv6-Based DDoS Flooding-Attack Detection Using Machine and Deep Learning Techniques. *IETE Journal of Research*, 70(4), 3753-3762.

Elejla, O.E., Belaton, B., Anbar, M., Alabsi, B. and Al-Ani, A.K. (2019). Comparison of Classification Algorithms on ICMPv6-Based DDoS Attacks Detection. in *Computational Science and Technology: 5th ICCST 2018*, 29-30 August, 347-357, Springer, Kota Kinabalu, Malaysia, Singapore.

Elejla, O.E., Anbar, M. and Belaton. B. (2016). Flow-Based Datasets. https://sites.google.com/site/flowbaseddatasets

Elejla Omar E., Mohammed Anbar, Bahari Belaton and Basem O. Alijla. (2018). Flow-Based IDS for ICMPv6-Based DDoS Attacks Detection. *Arabian Journal for Science and Engineering,* 43(12), 7757-7775.

Elejla Omar E., Mohammed Anbar, Bahari Belaton and Shady Hamouda. (2019). Labeled Flow-Based Dataset of ICMPv6-Based DDoS Attacks. *Neural Computing and Applications,* 31, 3629-3646.

Elejla Omar E., Mohammed Anbar, Shady Hamouda, Serri Faisal, Abdullah Ahmed Bahashwan and Iznan H. Hasbullah. (2022). Deep-Learning-Based Approach to Detect ICMPv6 Flooding DDoS Attacks on IPv6 Networks. *Applied Sciences,* 12(12), 6150.

Khraisat, A., Gondal, I., Vamplew, P. and Kamruzzaman, J. (2019). Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges. *Cybersecurity*, 2(1), 1-22.

Manickam Selvakumar, Adnan Hasan Bdair Alghuraibawi, Rosni Abdullah, Zaid Abdi Alkareem Alyasseri, Karrar Hameed Abdulkareem, Mazin Abed Mohammed and Ayman Alani. (2022). Labelled Dataset on Distributed Denial-of-Service (DDoS) Attacks Based on Internet Control Message Protocol Version 6 (icmpv6). *Wireless Communications and Mobile Computing*.

Mishra, A., Gupta, N. and Gupta, B.B. (2021). Defense Mechanisms against DDoS Attack Based on Entropy in SDN-Cloud Using POX Controller. *Telecommunication Systems*, 77(1), 47-62.

Mishra, A., Joshi, B.K., Arya, V., Gupta, A.K. and Chui, K.T. (2022). Detection of Distributed Denial of Service (DDoS) Attacks Using Computational Intelligence and Majority Vote-Based Ensemble Approach. *International Journal of Software Science and Computational Intelligence (IJSSCI)*, 14(1), 1-10.

Radhakrishnan, R., Majid Jamil, Shabana Mehfuz and Moinuddin Moinuddin. (2007). Security Issues in IPv6. in *International Conference on Networking and Services (ICNS'07)*, 110-110, IEEE.

Shiranzaei Atena, and Rafiqul Zaman Khan. (2018). IPv6 Security Issues—A Systematic Review. *Next-Generation Networks: Proceedings of CSI-2015*, 41-49.

Singh, A. and Gupta, B.B. (2022). Distributed Denial-of-Service (DDoS) Attacks and Defense Mechanisms in Various Web-Enabled Computing Platforms: Issues, Challenges, and Future Research Directions. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 18(1), 1-43.

Tayyab Mohammad, Bahari Belaton and Mohammed Anbar. (2020). ICMPv6-Based DoS and DDoS Attacks Detection Using Machine Learning Techniques, Open Challenges, and Blockchain Applicability: A Review. *IEEE Access,* 8, 170529-170547.

Tiwari Asheesh, Shivam Saraswat, Utkarsh Dixit and Sagar Pandey. (2022). Refinements in Zeek Intrusion Detection System. in *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 1, 974-979, IEEE.

Vaswani Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin. (2017). Attention is All You Need. *Advances in Neural Information Processing Systems,* 30.

Cite this article as: Manoj Kumar Prabakaran, Abinaya Devi Chandrasekar and M. Parvathy (2026). Intelligent Data Analytics Framework Using Ensemble-Attention Based Deep Learning Approach for Network Intrusion Detection. *International Journal of Artificial Intelligence and Machine Learning,* 6(1),18-36. doi: 10.51483/IJAIML.6.1.2026.18-36.