NIPES - Journal of Science and Technology Research
www.nipes.org
https://journals.nipes.org/index.php/njstr

GenDBN-Ensemble: A Hybrid Framework for Intrusion Detection in Network Traffic

Sule Aishat A.[1]*, Alhassan John K.[2], Ismaila Idris [3], Alabi Isiaq O.[4], Subairu Sikiru O.[5]
1-5Department of Cyber Security Science, Federal University of Technology, Minna, Nigeria.
Corresponding author: suleaishat990@gmail.com

Article Info

Keywords: *Intrusion Detection System (IDS), Deep Belief Network, Hybrid Deep Learning, Genetic Algorithm, DDoS Attacks, Network Security.*

Received 10 October 2025
Revised 19 November 2025
Accepted 09 January 2026
Available online 10 March 2026

https://doi.org/10.37933/nipes/8.1.2026.2054

eISSN-2682-5821, pISSN-2734-2352

Abstract

Distributed Denial of Service (DDoS) and Denial of Service (DoS) attacks remain a major challenge for network security due to their impact on service availability and their evolving attack patterns. Conventional intrusion detection systems (IDS), which always depend on static rules, have trouble adapting and are especially sensitive to class imbalance, which increases the rate of false alarms. This paper presents GenDBN-Ensemble, a hybrid intrusion detection framework that uses a dynamically weighted soft-voting ensemble to integrate an Adam-optimized Deep Belief Network (DBN), Support Vector Machine (SVM), and XGBoost classifier. The Synthetic Minority Over-Sampling Technique (SMOTE), which limits oversampling by predetermined class-specific upper bounds to lower the risk of overfitting, was applied only to the training data in order to address class imbalance. A recall-sensitive Genetic Algorithm (GA) fitness function is used to guide the selection of features, while DBN training is improved through a hybrid fine-tuning loss that combines cross-entropy and reconstruction errors and the ensemble adopts a diversity-aware weighting scheme. An 80-20 stratified train-test split was used to assess the framework on the CIDDS-001 dataset, and stratified five-fold cross-validation was used during GA optimization. Both binary and multi-class classification settings were used in the experiments. The model produced macro-averaged precision, recall, and F1-scores above 0.99 in the multi-class task and 99.98% detection accuracy with a low false alarm rate in the binary task. These findings show that the proposed framework demonstrates promising performance for the detection of DoS and DDoS attacks.

Nomenclature and Abbreviations

F1-Score - Harmonic mean of Precision and Recall
ROC AUC - Receiver Operating Characteristic - Area Under Curve
L_{ce} - Cross-Entropy Loss
L_{recon} - Reconstruction Loss from Restricted Boltzmann Machines
L_{total} - Combined Hybrid Loss Function
$\alpha, \beta, \gamma, \lambda$ - Weighting coefficients in optimization and loss functions
P_{final} - Final ensemble prediction probability
KL Divergence - Kullback-Leibler Divergence

1.0. Introduction

The high rate of cyber-attacks in today's world cannot be overemphasized as the rate of internet users continue to skyrocket especially with the recent shift from onsite working to remote working by most organizations caused by the COVID-19. As a result, the scope and complexity of cyber-attacks continue to grow; placing modern network infrastructures under a great deal of strain. One of such attack is the DOS and DDoS which is known for compromising the confidentiality and integrity of networked systems as well as its ability to interfere with service availability. To address this major cyber issue, IDS have emerged as key to providing the needed security, facilitating the prompt detection of malicious activity and enhancing the general robustness of computing environments [1]. Traditional IDS usually rely on static rule sets or predefined signatures, which can be useful against known attack patterns but are really limited when it comes to detecting threats that are not previously known especially in high-traffic network environments; these conventional systems are also prone to producing high FAR and show little flexibility and scalability. In light of these difficulties, researchers have been prompted to look further into better IDS techniques that can more accurately capture the intricacy of DoS and DDoS attacks [2]. Researches have suggested the use of ML and DL as a better substitute to conventional methods since they have the capacity to detect previously unseen attacks. Although, DL is a subset of ML, it has gained more favour in the research environment especially [3] deep architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Autoencoders, and DBNs that have shown enhanced detection capabilities over time. However, their use also raises practical concerns like slower convergence in deep models, sensitivity to class imbalance, higher computational costs during training and

inference, and decreased generalization to unknown attack behaviors [4]; thus, the investigation towards hybrid and ensemble-based approaches as a way to take advantage of complementary model strengths to building an IDS with a competitive performance. Despite the developments witnessed in the world of IDS research, it is still faced with the difficulty to strike a constant balance between robustness, FAR reduction, and detection accuracy across various intrusion categories [1]. This study introduces GenDBN-Ensemble, a hybrid intrusion detection framework proposed to enhance detection performance while preserving computational efficiency on the CIDDS-001 dataset, in order to respond to these unresolved problems. The framework utilizes a dynamically weighted soft-voting ensemble to combine an Adam-optimized DBN with SVM and XGB classifiers. While the SMOTE is applied to the training data only to solve the issue of class imbalance, a recall-sensitive GA guides feature selection to minimize redundancy and enhance learning efficiency.

The suggested framework differs from many current hybrid IDS approaches by adopting a security-oriented, co-optimized design in addition to combining several learning models. To discourage needless model complexity and highlight the detection of malicious traffic, a recall-sensitive GA fitness function is utilised. Furthermore, a hybrid fine-tuning loss that incorporates reconstruction objectives and cross-entropy refines DBN training, enabling more reliable feature learning. Instead of using heuristic assignment, robustness and Kullback-Leibler divergence analyses are used to determine ensemble weights (DBN:SVM:XGBoost = 3:1:2), encouraging balanced ensemble contributions. These design decisions enable the framework to show encouraging detection performance in the evaluated dataset under binary and multi-class classification settings. This is how the rest of the paper is structured. Related intrusion detection system research is reviewed in Section 2. The suggested methodology is explained in Section 3. The results and analysis of the experiment are presented in Section 4. The discussion is presented in Section 5, and the conclusion and future research directions are presented in Section 6.

1.1. Literature Review

1.2. The Emergence of Intrusion Detection Systems

Computer systems are becoming more interconnected, which has enhanced communication but also made them more vulnerable to cyberthreats that jeopardize data confidentiality, integrity, and availability (CIA). IDS were created to monitor network or system activity, identify malicious behavior, notify administrators, or start countermeasures in order to mitigate these risks [1]. The Intrusion Detection Expert System (IDES), which employed statistical techniques for anomaly detection in 1984, came after work on threat monitoring, which is where the idea for IDS originated [6]. Later developments produced network-based tools such as the Network System Monitor (NSM) and, in 1998, SNORT, an open-source IDS that gained widespread adoption. IDS technologies developed into commercial products by the late 1990s, and they eventually became Intrusion Prevention Systems (IPS), which proactively block threats [5].

The classification of modern IDS can be done into three depending on the data source they monitor which are Network-based (NIDS), Host-based (HIDS), or Application-based (AIDS) [6]. Similarly, classification of IDS can also be done based on the detection techniques which include signature-based systems that recognize known attack patterns; anomaly-based systems, which model typical behavior to identify deviations; and hybrid systems, which integrate both strategies [7]. In all of these IDS techniques, hybrid IDSs are most widely used because they can lower false positives while still being flexible enough to respond to new threats.

1.3 DoS and DDoS Attacks

Among the most serious cybersecurity risk is the DoS and DDoS attacks because they compromise service availability by flooding systems with malicious traffic [8]. while DDoS attacks use numerous compromised systems, frequently arranged as botnets, making them much more challenging to counter, DoS attacks come from a single source [9]. This threat has been exacerbated by the widespread use of IoT devices, which enables hackers to access a large number of endpoints that are unprotected.

The categories of DDoS are volumetric attacks, protocol attacks and application-layer attacks. The volumetric attacks overload network bandwidth; protocol attacks takes advantage of flaws in transport or network layers; and application-layer attacks covertly overload servers by imitating valid user requests [10]. There has been a rapid increase in the rate of this attacks with some amounting to terabit levels, and the emergence of "DDoS-for-hire" services has made it easier to start these campaigns. Detecting DDoS attacks with conventional method which are frequently slow and resource-intensive, is more difficult because they are dispersed; thus, the need for intelligent, adaptable models that can effectively detect intrusions becomes necessary for effective countermeasures.

1.4 The Concept of Deep Learning (DL)

Deep learning is a subset of machine learning known for its capacity to extract hierarchical feature representations from data without the need for human feature engineering. DL architectures have been found to be useful in classification tasks that are quite intricate as compared to conventional methods that depend on manually created features [11]. The groundwork for deep learning was laid by introducing deep neural architectures as extensions of artificial neural networks with multiple hidden layers [12]. DL allows for strong generalization in a variety of fields, from cybersecurity to healthcare, by fusing low-level features into abstract higher-level representations [12].

One of DL's notable architectures, the DBN, has shown great promise for intrusion detection, outperforming traditional methods in identifying complex and hitherto unseen attacks [13]. There are a number of DL models that has been investigated in IDS research. By reconstructing typical traffic and identifying deviations, autoencoders can detect anomalies by learning compressed feature representations [14]. As DBN building blocks, RBMs offer probabilistic feature learning and have been used in network security tasks as unsupervised feature extractors [15]. DBNs, which are created by stacking RBMs with supervised fine-tuning, provide scalable and effective

intrusion detection performance by overcoming issues with local optima and lengthy training times[16], [17].

In addition to IDS, DL has been effectively used in autonomous driving, speech recognition, computer vision, healthcare, virtual assistants, fraud detection, and entertainment [18], [19]. The application of DL in several domains has show how versatile, robust and scalable it is; however, DL models are not without disadvantages which span across high computation, the need for a lot of labelled data and other problems like overfitting, poor interpretability, and reliance on data quality [20].

1.5. Related Studies

The development of a more advanced ML models is motivated by the ever-increasing threat posed by DoS and DDoS attacks. As attackers continue to devise more sophisticated means to compromise system security, researchers are also not relenting in their effort to developing a more robust models that can withstand the antics of cyber-attackers. Among the different methods for building such models, researchers have tilted towards hybrid DL models due to its ability to capture the complex spatial-temporal dynamics of network traffic. Past researches such as [21] put forward an optimized CNN-LSTM framework that recorded 99.84% accuracy on the CICDDoS2019 dataset, which implies that it can quickly adapt to unseen attacks. Another research was carried out in the SDN environment where their model [22] reported accuracies exceeding 99.8%. The proposed model was developed by modeling both short-term and long-term traffic patterns through 1D-CNN and GRU layers. Similarly, this architectural trend extends to specialized environments such as 5G networks, where the CNN-RF hybrid developed performed better than the recurrent-based models like LSTM-RF in detecting low-rate attacks [23]. The author further demonstrated that there is a very important trade-off between operational viability and raw classification accuracy. The CNN-DNN-RNN model developed in [24] shows a competitive performance in precision and recall; however, the lack of reported FAR and inference times makes practical evaluation difficult. Other models, like the GA-SVM and N-KPCA approach in [25], have trouble balancing their performance, as indicated by their lower F1-scores. Literatures that focus on computational efficiency, like [26], revealed that a CNN-LSTM-AE model can have a very low FAR of 0.04% and faster inference times of 3.86ms, even though it recorded a low overall accuracy. These traits make these models especially good for infrastructure that needs to be fast in making detection decision.

Advancements in recent times have shifted towards ensemble learning to improve resilience in diverse environments; for instance, stacking and voting classifiers have become popular, like the combination of XGB, LightGBM, and Random Forest shown in [27] is a great example of how scalable Software-Defined Cyber-Physical Systems can be. Using a HistGradientBoosting and XGB ensemble, [28] recorded 99.9% accuracy, precision, recall, and F1-score, while [29] utilized a dual-CNN ensemble that maintained a low CPU usage and faster detection time. In addition, the model that combined RNN, LSTM, HTM, and CNN, when tested on the CIC-DDoS2019 dataset, gave consistent results with an accuracy, precision, recall, and F1-score of 99.17% and a ROC AUC of 99.0% [30].

Optimization continues to be a central focus of contemporary research, especially via the application of evolutionary algorithms. In [31], it was shown that using Genetic Algorithm Wrapper Feature Selection can greatly cut down on feature redundancy, which makes detection more accurate even in very complex architectures. The DeepDefend framework in [32] has recently pushed the limits of cloud security by combining entropy forecasting through CNN-LSTM-Transformers with an AutoCNN-DT classifier. This model got 99.97% accuracy on the CIDDS-001 dataset and took 3.88 seconds to make a detection decision. Table 1 shows a summary of some of these recent contributions and how well they performed on their respective datasets.

Table 1. Summary of Literature Review

Reference	Methodology / Algorithm	Results	Strengths	Limitations
[32]	DeepDefend	99.97% accuracy, 99.97% precision, 99.97% recall, and 3.88 sec inference time. (CIDDS_001)	The model shows strong detection capabilities with high precision, recall, and an effective hybrid approach combining CNNs and DTs. GA helps improve feature selection.	The model has a relatively high inference time. It may struggle with generalizing to new attacks and can be computationally expensive.
[21]	Optimized CNN-LSTM hybrid	Accuracy: 99.84% (CICDDoS2019)	Extracts spatio-temporal features; suitable for real-time adaptation	Limited discussion on class imbalance.
[23]	CNN-RF hybrid	Accuracy: 96.52%; Recall: 99.99% on Shrew; Accuracy: 99.52% on Slow-Body	Effective for low-rate DDoS; performs well on multi-attack datasets	High detection window (120s); latency concerns.

[22]	CNN-GRU-DNN hybrid	Accuracy: 99.81% (CICDDoS2019), 99.88% (synthetic)	Handles short- and long-term traffic; robust for SDN; low-rate DDoS detection	High computational complexity
[27]	XGBoost, LightGBM, RF with LR	Accuracy: 99.89%; F1-score: 99.84% (CICIDS2017)	Strong detection accuracy	Resource consumption not reported; edge testing limited
[28]	(HistGradientBoosting + XGBoost)	Accuracy, Precision, Recall, F1: 99.90% (InSDN); Training time: 11.09s	Resource-efficient; robust; fast training	Tested only in SDN; no evaluation on extreme edge devices
[29]	Deep CNN Ensemble (2 CNNs)	Accuracy: 99.45% (CICIDS2017); Detection time: 0.061s; CPU usage: 6.02%	Scalable; real-time feasible; low latency	Limited handling of multi-vector attacks; training time 39.52s
[31]	Stacking Ensemble SVM-MLP-RF + GA Wrapper Feature Selection	Accuracy: 99.86% (train), 98.89% (unseen); 23 FP, 34 FN; Training: 2,593.38s	Reduces feature redundancy; robust; high detection precision	Very high training time; not evaluated on real-time edge-class environments

1.6. Contrasting GenDBN-Ensemble with Related Works

This section makes a distinction between the proposed model and other hybrid and ensemble-based IDS by focusing design, feature selection and decision strategy. The design of GenDBN-Ensemble framework as opposed to other hybrid models utilized a security-focused design strategy across the three modules: feature selection, representation learning, and decision fusion. When it comes to feature selection, many GA-based IDS methods optimize fitness mainly by looking at overall classification accuracy [25], [31]. These kinds of formulations work well in balanced situations, but they may make optimization favor the majority classes and make it less sensitive to attack scenarios with smaller number of instances. To address this limitation, the proposed framework employs a recall-weighted GA fitness function indicated in Equation 1 where greater emphasis is being given to attack recall ($\beta = 0.6$) relative to F1-score ($\alpha = 0.3$), while introducing a modest penalty on subset size ($\gamma = 0.1$).

$$Fitness(s) = \alpha * F1 - Score(s) + \beta * Recall(s) - \gamma * |s| \quad (1)$$

Also, during GA evolution, fitness evaluation is carried out with stratified k-fold cross-validation rather than a single hold-out validation like in some other researches [31]. This design produces more consistent and less biased feature-selection results, as outlined in Table 2.

Table 2. Comparison of GA Approaches

Aspect	Standard GA Approaches	Proposed Security-Driven GA
Primary Objective	Accuracy maximization [26, 32]	Recall-weighted optimization
Fitness Function	Fitness = Accuracy − γ * s	$Fitness(s) = \alpha * F1 - Score(s) + \beta * Recall(s) - \gamma * \|s\|$
Coefficient Values	Equal weighting or accuracy-focused	α=0.3, β=0.6, γ=0.1
Feature Penalty	Simple count-based	Complexity-aware with security feature preservation
Validation Method	Single hold-out validation	Stratified k-fold cross-validation during GA evolution

At the representation-learning stage, the framework extends conventional DBN-based intrusion detection by incorporating a hybrid fine-tuning loss that combines supervised classification loss with reconstruction error derived from pre-trained Restricted Boltzmann Machines. In contrast to previous DBN approaches that relied heavily on cross-entropy loss during fine-tuning, the proposed framework maintains generative structure while adapting the model for supervised detection tasks [15], [17]. In order to improve the way in which the model cope with noisy or changing traffic patterns within the evaluated dataset, a tunable parameter λ is used to control the balance between

feature preservation and discriminative performance. Table 3 shows the main differences between the traditional and the proposed DBN formulations.

For decision fusion, current ensemble-based IDS typically utilize equal voting, accuracy-proportional weighting, or stacking with a meta-learner [27], [28], [31], frequently without explicitly assessing model diversity. On the other hand, the proposed framework, uses a diversity-driven soft-voting strategy that uses pairwise Kullback-Leibler divergence and robustness analysis to ascertain ensemble weights instead of just accuracy.

The weighting ratio is determined through experiment where the model that has the most impact on performance is assigned the highest weight, hence, a fixed DBN:SVM:XGB weighting ratio of 3:1:2. As compared to stacking-based ensemble, this method lowers the complexity of inference time while still allowing for complementary decision-making. Table 4 gives a summary of the comparisons.

The GenDBN-Ensemble framework is a single security-driven design that brings together feature selection, hybrid deep representation learning, and diversity-informed ensemble fusion. The framework presents a more integrated implementation approach by coordinating these modules across the detection pipeline rather than treating them as an independent improvement.

Table 3. DBN Loss Function Innovation

Component	Traditional DBN [15], [16]	Proposed Hybrid DBN	Technical Advancement
Loss Function	Pure cross-entropy: Lce	Hybrid: $Ltotal=Lce+\lambda Lrecon$	Multi-objective optimization
Regularization	L1/L2 weight decay [16]	Reconstruction consistency from pre-trained RBMs	Unsupervised-supervised synergy
Generalization	Supervised only	Semi-supervised via reconstruction penalty	may support improved robustness, although this remains evaluated only on CIDDS-001
λ Optimization	Not applicable	Ablation-tuned (λ=0.5optimal)	Empirical validation of hybrid approach
Feature Learning	Discriminative only [15]	Generative + Discriminative	Enhanced feature representation

Table 4. Ensemble Weighting Strategy Comparison

Methodology	Performance-Based Weighting	Proposed Diversity-Driven Weighting	Strategic Advantage
Weight Basis	Individual model accuracy [27], [28]	KL Divergence + Robustness analysis	Complementarity maximization
Weight Assignment	Accuracy-proportional [28]	DBN:3, SVM:1, XGBoost:2 (diversity-optimized)	Error reduction through decorrelation
Diversity Metric	Not explicitly considered	Pairwise KL Divergence quantification	Explicit diversity optimization
Model Selection	Best-performing models [29]	Architecturally diverse base learners	Heterogeneous ensemble benefits
Robustness Test	Cross-validation performance	Adversarial perturbation resistance	Security-specific robustness

2.0. Methodology

This study presents a hybrid IDS framework known as the GenDBN-Ensemble, engineered to detect DoS and DDoS attacks accurately while ensuring computational efficiency. The design combines deep learning, ensemble learning, and evolutionary optimization techniques to mitigate the limitations in IDS, such as class imbalance, high FAR, feature redundancy, and limited adaptability to unseen attack patterns. Through a weighted soft-voting ensemble, the framework combines an Adam-optimized DBN with SVM and XGB classifiers. The GA is utilized for feature selection to retain important features and lower the number of dimensions, and SMOTE is used only on the training data to address class imbalance. Figure 1 shows the different stages of the entire workflow, which include

data preprocessing, feature optimization, base-learner training, and ensemble prediction.

2.1 Experimental Setup

All experiments were carried out on an ASUS workstation installed with a Windows 11, an Intel® Core™ i7 processor (3.40 GHz), a 16 GB of RAM, and an NVIDIA GeForce GTX 1660 GPU. The implementation environment was built on Python 3.9, with TensorFlow 2.13.0 used for DL parts and Scikit-learn 1.3.0 used for traditional ML tasks. NumPy 1.23.5, Pandas 2.0.3, SciPy 1.10.1, XGB 1.7.6, and imbalanced-learn 0.11.0 were some of the libraries that were utilized.

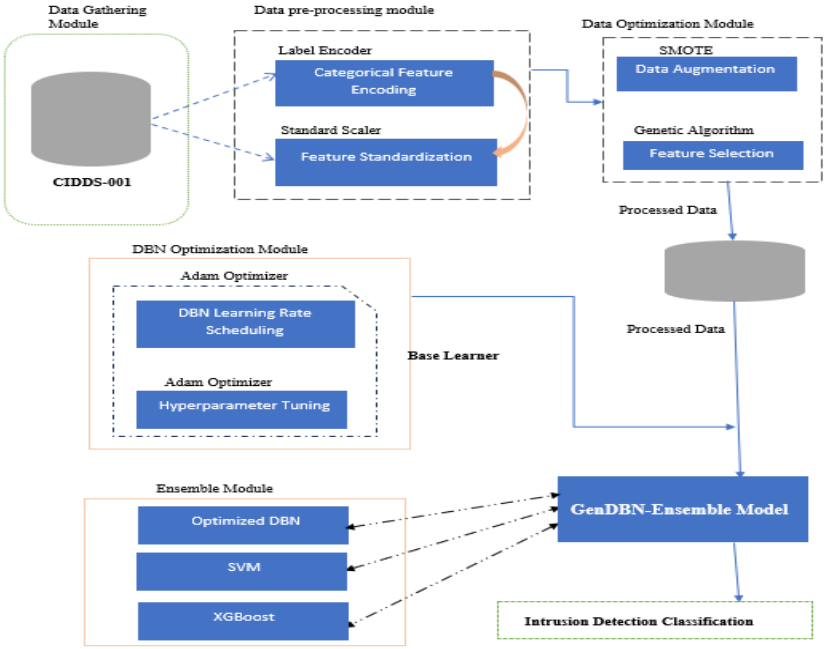

Figure 1. The Architectural Design of GenDBN-Ensemble

2.2 Dataset Acquisition and preprocessing

The dataset used for this experiment is the Coburg Intrusion Detection Dataset (CIDDS-001) which is a publicly available, flow-based dataset that is widely used to test IDSs and contains real network traffic from OpenStack environments and external servers that includes both normal and abnormal flows. Each record contains information like IP addresses, ports, protocols, timestamps, and traffic volume statistics. The dataset has some known problems, like duplicate records and a big class imbalance, however, it is still the data of choice because it closely matches how networks behave in the real world.

The first phase involved the preprocessing of the data where label encoding was utilized to encode categorical features and z-score normalization was applied to standardize numerical features. In Equation 2, μ represent the mean and σ is the standard deviation of the feature, that was derived from the training data. Two experimental settings were considered which is the binary and multi-class. For the binary classification, traffic was split into two: normal and malicious using an 80:20 stratified train-test split while in the multi-class classification task, a 70:15:15 split was utilized for training, validation, and testing, respectively. This allowed for the tuning of hyperparameters while keeping the test evaluation unbiased.

$$x' = \frac{x - \mu}{\sigma} \qquad (2)$$

In addition, feature selection was done to select feature subsets using the GA, with each chromosome serving as a binary feature mask. Equation 3 was used to assess fitness, where the number of selected features is indicated by $|s|$. In order to support security-oriented optimization, this formulation penalizes subset size while prioritizing recall. To maintain class distributions and avoid data leakage, fitness evaluation used stratified five-fold cross-validation with fixed seeds. The GA employed uniform crossover with probability 0.7, tournament selection (k = 3), a population size of 50, up to 40 generations, an adaptive bit-flip mutation that increased from 0.01 to 0.10 across generations, and elitism to retain the top two individuals. After convergence, optimization stopped, producing a final subset of ten features that were used in later tests.

$$Fitness(s) = \alpha * F1 - Score(s) + \beta * Recall(s) - \gamma * |s| \qquad (3)$$

SMOTE was only used on the training data in order to address class imbalance. Instead of requiring full class equalization, oversampling was carried out with predetermined upper bounds for minority classes, which decreased the possibility of excessive synthetic duplication. The Normal class stayed at 1,289,513 samples, while the DoS and DDoS classes grew from 28,565 to 500,000 and 39,885 to 500,000 samples, respectively. To maintain the integrity of the evaluation, validation and test sets were completely unaltered.

2.3. Model development and Training

The primary feature learner for hierarchical representation extraction was the DBN. It comprised of three hidden layers with sizes [512, 256, 128], employing ReLU activation, batch normalization, and dropout of 0.3 for hidden layers, and 0.2 for the output layer. The Adam optimizer with learning rate of 0.005, and AMSGrad enabled, minimized a hybrid fine-tuning loss that combined classification and reconstruction objectives shown in equation 4.

$$L_{total} = L_{ce} + \lambda * L_{recon} \quad (4)$$

Here, L_{ce} is the binary cross-entropy loss guiding supervised classification, and L_{recon} is the reconstruction error from the pre-trained Restricted Boltzmann Machines, acting as a regularizer to penalize anomalous feature patterns. Training used a batch size of 1024, early stopping (patience = 15), and ReduceLROnPlateau scheduling. A summary of the DBN parameter tuning and its impact is shown in Table 5.

Table 5. DBN Hyperparameter Tuning and Impact

Parameter Category	Parameter	Range/Values	Optimal Value	Impact on Model Performance
Architectural Parameters	Number of Layers	2-5 hidden layers	3	↓ 15% training time vs 4-layer
	Layer Sizes	128-1024 neurons/layer	[512, 256, 128]	↑ 2.3% F1-score
	Activation	ReLU, Sigmoid, tanh	ReLU	↓ Vanishing gradient issues
Training Parameters	Learning Rate	0.001-0.1	0.005	37% faster convergence
	Batch Size	64-2048	1024	↓ 22% GPU memory usage
	Dropout Rate	0.1-0.5	0.3 (hidden), 0.2 (output)	↓ Overfitting by 18%
Regularization	L1/L2 Mix	λ_1=1e-6 to 1e-4, λ_2=1e-5 to 1e-3	λ_1=1e-5, λ_2=1e-4	↑ 1.1% modest improvement in generalization
	Batch Norm	Position (Pre/Post) Activation	Post-ReLU	↓ 31% internal covariate shift
RBM-Specific Tuning	Gibbs Steps (k)	1-10 CD iterations	3	Balanced speed/accuracy
	Visible Units	Bernoulli/Gaussian	Bernoulli	Better for binary features
	Hidden Sparsity	0.01-0.2 target	0.05	↑ Feature diversity

For the ensemble stage, DBN, SVM with RBF kernel, and XGBoost were independently trained on the GA-selected feature subset. Weighted soft-voting was employed to combine the probabilistic outputs of the three complementary classifiers. The DBN output is a probability score $P_{DBN}(y = 1/x)$ indicating the likelihood of a given sample being malicious. The SVM known for its robust decision boundaries and generalization capability, uses a radial basis function (RBF) kernel. Platt scaling, which converts margin-based scores into calibrated probabilities $P_{SVM}(y = 1/x)$ is used to obtain probabilistic outputs from SVM. As a gradient-boosted decision tree algorithm, XGB is effective at capturing interactions and non-linear relationships between features. By applying logistic regression to the summed tree outputs, its output, $P_{XGB}(y = 1/x)$, is equally a class probability. The final prediction probability for a sample in the soft voting scheme is calculated as a weighted average of the individual model probabilities shown in Equation 5.

$$P_{Ensemble}(y = 1/x) = \frac{3 \cdot P_{DBN} + 1 \cdot P_{SVM} + 2 \cdot P_{XGB}}{6} \quad (5)$$

Pairwise Kullback-Leibler divergence and adversarial perturbation tests were used in robustness and diversity analysis to determine the weights (3:1:2), giving the DBN a higher weight due to its greater resilience. In standalone evaluation, DBN recorded highest detection accuracy hence it is being given the highest weight; XGB on the other hand, received a moderate weight; and SVM contributed as a lightweight, high-margin learner. This design maintains diversity in decision-making while making sure that each model contributes to the final decision in proportion to its reliability. The threshold-based decision rule in Equation 6 is used to determine the final class label after the ensemble probability has been calculated.

$$\hat{y} = \begin{cases} 1, & P_{Ensemble}(y = 1/x) \geq 0.5 \\ 0, & Otherwise \end{cases} \quad (6)$$

The classification goal of determining whether a network traffic instance is malicious or normal tallies with this binary decision boundary. Equation 7 computes the ensemble probability for class c in the multi-class classification, and Equation 8 uses the argmax decision rule to determine the final class label.

$$P_{Ensemble(y=c/x)} = \frac{3.P_{DBN(y=c/x)} + 1.P_{SVM(y=c/x)+2.P_{XGB(y=c/x)}}}{6} \quad (7)$$

$$\hat{y} = arg \max_{c\in\{Normal,\ DoS,\ DDoS\}} P_{Ensemble(y=c/x)} \quad (8)$$

Data splitting, model initialization, and GA operations used fixed random seeds to support reproducibility and guarantee consistent experimental conditions across runs.

2.4. Performance Metrics

The following standard metrics were adopted to evaluate the proposed model:

i. **Accuracy (AR):** proportion of correctly classified traffic instances.

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + False\ Positive + True\ Negative + False\ Negative} \quad (9)$$

ii. **Precision (PR):** proportion of true attack detections among all predicted attacks.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (10)$$

iii. **Recall (RC):** proportion of correctly detected attacks among all actual attacks.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (11)$$

iv. **False Alarm Rate (FAR):** proportion of benign traffic wrongly classified as attacks.

$$\text{False Alarm Rate} = \frac{\text{False Positive}}{\text{False Positive} + \text{True Negative}} \quad (12)$$

v. **F1 score:** This is the harmonic mean of precision and recall. It has a range of 0-1 and provides information on how many times the classifier correctly classifies and how often it misses a significant number of instances.

$$F1\ score = \frac{2}{\left(\frac{1}{Precision} + \frac{1}{Recall}\right)} \quad (13)$$

vi. **Inference Time (It):** average classification latency per sample.

$$I_t = T2 - T1 \quad (14)$$

Where True Positive represents traffics correctly classified as attacks; False Positive is normal traffic being incorrectly classified as an attack; True Negative is when normal traffic is correctly classified as normal; False Negative is attack being incorrectly classified as normal; T1 is the time when the data sample is fed into the model; and T2 is the time when the model outputs its prediction. Inference latency was computed by executing 1,000 prediction iterations on the first 100 samples of the test set following an initial warm-up run. The total elapsed time was averaged and reported in milliseconds per sample.

3.0 Results

3.1. Model performance evaluation

The proposed GenDBN-Ensemble framework was evaluated on the CIDDS-001 dataset using accuracy, precision, recall, F1-score, ROC AUC, false alarm rate, and inference time providing a holistic view of model performance.

3.2. Multi-Class Classification Performance

The aggregate and per-class metrics for multi-class classification (Normal, DoS, and DDoS) is indicated in Table 6. While DoS and DDoS attacks received high F1-scores of 0.9929 and 0.9973, respectively, normal traffic was almost perfectly detected. Weighted-average and aggregate macro metrics show balanced performance across all classes, indicating that the model is not skewed toward any specific type of traffic.

Similarly, the error analysis in Table 7 reveals that 33 samples had false positives, in which regular traffic was mistakenly identified as DoS or DDoS. 46 samples were

impacted by false negatives, which occur when attack samples are mistakenly identified as typical traffic. There were only 30 cases of misclassification between DoS and DDoS attacks, indicating very little cross-confusion.

Table 6. Multi-Class Classification Performance.

(a) Per-Class Performance

Class	Precision	Recall	F1-Score	ROC AUC	Accuracy (%)
Normal	0.9998	0.9999	0.9998	0.99999	99.99
DoS	0.9947	0.9912	0.9929	0.99987	99.12
DDoS	0.9962	0.9984	0.9973	0.99994	99.84

(b) Overall Multi-Class Performance

Metric Type	Precision	Recall	F1	AUC
Macro Avg	0.9969	0.9965	0.9967	0.99993
Weighted Avg	0.9996	0.9996	0.9996	0.99997

Table 7. Confusion Matrices.

(a) Normalized Confusion Matrix.

Actual \ Predicted	Normal	DoS	DDoS
Normal	0.9999	0.0000	0.0001
DoS	0.0006	0.9912	0.0082
DDoS	0.0001	0.0015	0.9984

(b) Absolute Count Confusion Matrix.

Actual \ Predicted	Normal	DoS	DDoS	Total
Normal	275,772	11	22	275,805
DoS	37	6,068	17	6,122
DDoS	9	13	8,530	8,552

3.3 Binary Classification Performance

The performance of the baseline DBN, optimized DBN, and proposed ensemble for binary classification is presented in Table 8 and Figure 2. With an accuracy of 99.98%, the ensemble outperformed both the optimized DBN (99.12%) and the baseline DBN (98.34%). Similar patterns were observed in precision, recall, and F1-score, with the ensemble attaining 99.97%, 99.99%, and 99.98%, respectively.

Table 8. Performance of the Three Model Variants

Metric	DBN Alone (%)	Optimized DBN (%)	Our Ensemble Model (%)
Accuracy	98.34	99.12	**99.98**
Precision	98.41	99.08	99.97
Recall	98.12	99.23	99.99
F1-Score	98.26	99.15	99.98
ROC AUC	99.87	99.93	99.999
False Alarm Rate	1.55	0.44	**0.02**

The ensemble recorded 99.999% on ROC AUC and the FAR dropped to 0.02%, which is a 98.7% decrease from the baseline DBN. These findings show that the ensemble can simultaneously reduce false positives and improve accuracy.

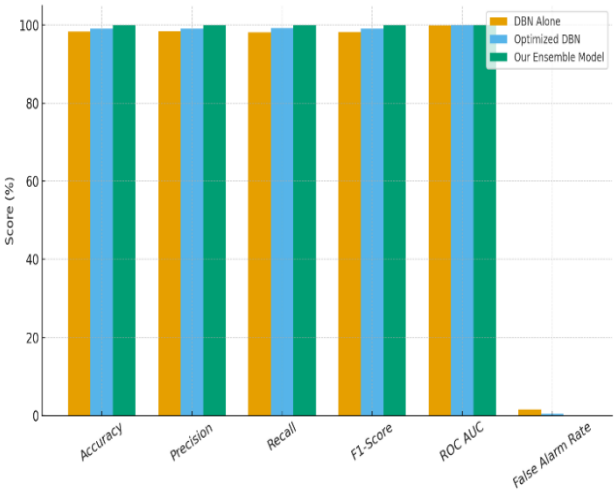

Figure 2. Performance of the Model Variants

The outcomes of the proposed model are also contrasted with current models in Table 9 where the suggested framework demonstrated competitive performance in attack detection. This comparison is a qualitative reference rather than a direct performance ranking due to the fact that different datasets, feature engineering, and evaluation procedures are utilized by different studies.

3.4 Impact of Model-Level Optimizations: Hybrid Loss and GA Feature Selection

The internal improvements of the DBN were the main focus of the evaluation's initial phase. A standard cross-entropy baseline was used as a benchmark for the hybrid loss function, which combines standard cross-entropy with RBM's reconstruction error. The hybrid loss function, specifically at an optimal regularization value of λ=0.5, achieved a 99.12% accuracy, as Table 10 illustrates.

Table 9. Comparative overview of GenDBN-Ensemble and previous approaches

Reference	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FAR (%)	Inference Time
[32]	DeepDefend (CNN+DT)	99.97	99.97	99.97	99.97	-	3.88sec
[24]	CNN_DNN_RNN	99.05	98.66	99.68	99.17	1.74	-
[25]	N-KPCA+GA_SVM	98.87	94.1	92.0	89.8	0.02	-
[30]	RNN-LSTM-HTM-CNN	99.17	99.17	99.17	99.17	-	-
[28]	Ensemble Classifier (EC)	99.90	100	99.00	99.00	0.00	11.09sec
[26]	CNN + LSTM + AE	80.75	-	71.42	-	0.04	3.86ms
[31]	Stacking Ensemble Model	99.86	99.82	99.71	99.71	-	-
[23]	CNN + RF	96.52	95.46	86.39	95.60	11.03	268.37 sec
[22]	1D CNN + GRU + DNN Hybrid	99.81	99.96	99.90	99.93	-	-
Our Model	**GenDBN-Ensemble**	**99.98**	**99.97**	**99.99**	**99.98**	**0.02**	**1.8ms**

This is a 0.78% improvement over the baseline and a huge 71.6% drop in the FAR. This shows that the hybrid approach greatly improves feature retention and classification boundaries.

Table 10. DBN Loss function comparison

Loss Type	λ Value	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC AUC (%)	FAR (%)
Cross-Entropy	-	98.34	98.41	98.12	98.26	99.87	1.55
Hybrid Loss	0.1	98.89	98.76	98.95	98.85	99.91	0.89
Hybrid Loss	**0.5**	**99.12**	**99.08**	**99.23**	**99.15**	**99.93**	**0.44**
Hybrid Loss	1.0	98.95	98.82	99.01	98.91	99.90	0.67

At the same time, the GA was used to reduce the feature space from the complete 25-feature CIDDS-001 set to a 10-feature subset that was optimized for recall. This reduction not only kept the detection rates high, but it also made computations more efficient, cutting down inference time by 37% and training time by 59%. The performance metrics in Table 11 shows that the features chosen by the GA let the model focus on the factors that are most useful for finding network threats.

Table 11. Feature selection impact

# Features	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC AUC (%)	FAR (%)
25	98.67	98.54	98.73	98.63	99.89	0.92
10	**99.12**	**99.08**	**99.23**	**99.15**	**99.93**	**0.44**

3.5 Ensemble Strategy and Diversity-Driven Weighting

The evaluation's final phase looked at how the DBN, SVM, and XGB models were combined. Kullback-Leibler (KL) divergence was used in a diversity study to measure the base learners' complementarity. The different KL divergence values and high complementarity scores between 0.72 and 0.78, as shown in Table 12, validate that the chosen models offer non-redundant viewpoints on the dataset.

Taking advantage of this variability, equal weighting and majority vote were contrasted with the proposed weighting scheme (3:1:2). With a high accuracy of 99.98% and a low FAR of 0.02%, Table 13's data show the effectiveness of the diversity-driven method.

Table 12. Pairwise KL Divergence

Model Pair	KL Divergence	Complementarity Score
DBN vs SVM	0.342	0.78
DBN vs XGBoost	0.287	0.72
SVM vs XGBoost	0.315	0.75

Table 13. Ensemble Weighting Comparison

Ensemble Method	Weights	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC AUC (%)	FAR (%)
DBN Alone	-	99.12	99.08	99.23	99.15	99.93	0.44
SVM Alone	-	96.45	95.89	96.78	96.33	98.45	2.87
XGBoost Alone	-	97.82	97.45	97.96	97.70	99.25	1.65
Equal Weights	1:1:1	99.67	99.65	99.72	99.68	99.97	0.21
Majority Voting	Binary	99.58	99.52	99.65	99.58	99.96	0.28
Proposed Weights	**3:1:2**	**99.98**	**99.97**	**99.99**	**99.98**	**99.999**	**0.02**

With a final framework accuracy of 99.98% and an inference time of 1.8ms, the ablation study's results validate a cumulative performance boost of 1.64% over the 98.34% baseline. Table 14 shows the statistical p-values which support all major performance differences, demonstrating that the integration of hybrid loss, GA-based feature selection, and diversity-driven weighting presents a dependable and statistically significant improvement in IDSs.

Table 14. Statistical Significance (p-values)

Comparison	Accuracy	F1-Score	FAR
Hybrid vs Standard Loss	0.0032*	0.0028*	0.0015*
GA vs All Features	0.0087*	0.0074*	0.0042*
Proposed vs Equal Weights	0.0125*	0.0098*	0.0063*

**Statistically significant ($p < 0.05$)*

The GenDBN-Ensemble framework in particular, shows promise in combating DoS and DDoS attacks based on the findings, which further validates that combining many approaches into a hybrid model can improve detection efficiency and accuracy over traditional approaches.

The performance of the model was improved by using an Adam optimizer to fine-tune the DBN in addition with a GA for feature selection. This combination lead to a reduction in the FAR, a frequent problem in security systems, while also improving overall accuracy. Similarly, SMOTE was used on the training data to address class imbalance thereby improving performance across minority attack classes. Compared to the baseline model's FAR of 1.55%, the final ensemble model achieved a FAR of 0.02% showing a significant improvement. The framework's ability to identify previously unseen attacks was further improved by its hybrid loss function, which integrated several error measuring techniques. As stated in Section 3.3, fixed random seeds were used at every stage of the experiment to guarantee reproducibility, from data splitting to algorithm execution.

The DBN, SVM, and XGB models' predictions were successfully fused by the ensemble's decision-making approach, which weighted each model based on its own dependability. This method helps create a solid and reliable detecting mechanism. With an average inference time of 1.80ms per sample, the system's efficiency underscores its potential appropriateness for real-time applications. Overall, the GenDBN-Ensemble framework performed better than the other models assessed in this study, implying that it could potentially be used in real-world network environments to detect DoS and DDoS attacks accurately.

4.0. Conclusion

This research presents the GenDBN-Ensemble framework, a hybrid model for identifying DoS and DDoS attacks. The framework utilized a DBN that was fine-tuned with Adam optimization algorithm, a recall-weighted GA-based feature selection, and an ensemble of DBN, SVM, and XGB. The model's performance was improved by a hybrid loss function which made the system even better at finding attacks that it had never seen before. These components work together to help the framework find threats accurately and in a lesser amount of time, about 1.8ms per sample. One key result of this study is that the FAR was reduced to 0.02%, from the initial 1.55% for the baseline DBN which helped avoid false alarms. Feature selection also reduced the input space, which made training and inference faster without lowering the accuracy of detection. Similarly, the combination of multiple classifiers in the ensemble provided better results compared to the individual models. Future work could look further into adapting the framework to detecting additional types of attacks, such as Remote-to-Local (R2L), Probe,and User-to-Root (U2R), and extending it to additional datasets like NSL-KDD or CIC-IDS2017.

Funding
This research received no external funding.

Conflict of Interest
The authors declare no conflict of interest.

Ethics Approval

No human participants or personal data were involved. The CIDDS-001 dataset is publicly available and anonymized.

Author Contributions (CRediT)

Content development, Implementation, investigation, methodology, writing original draft, review & editing: S.A.A; Conceptualization, quality assessment, review & editing, supervision: J.K.A; Quality assessment, review & editing, supervision: I.I; Resources, quality assessment, supervision, review & editing: I.O.A; Resources, quality assessment, supervision, revsiew & editing: S.S.O.

References

[1] Z. K. Maseer, R. Yusof, S. A. Mostafa, N. Bahaman, O. Musa, et B. A. S. Al-Rimy, « DeepIoT.IDS: Hybrid deep learning for enhancing IoT network intrusion detection », *Computers, Materials and Continua*, vol. 69, nº 3, 2021, doi: 10.32604/cmc.2021.016074.

[2] J. P. Anderson, « Computer security threat monitoring and surveillance », *Technical Report James P Anderson Co Fort Washington Pa*, 1980, doi: citeulike-article-id:592588.

[3] N. Aslam, S. Srivastava, et M. M. Gore, « A Comprehensive Analysis of Machine Learning- and Deep Learning-Based Solutions for DDoS Attack Detection in SDN », *Arab. J. Sci. Eng.*, vol. 49, nº 3, p. 3533-3573, mars 2024, doi: 10.1007/s13369-023-08075-2.

[4] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, et K. J. Kim, « A survey of deep learning-based network anomaly detection », *Cluster Comput.*, vol. 22, 2019, doi: 10.1007/s10586-017-1117-8.

[5] R. Martin, « Snort-Lightweight Intrusion Detection for Networks », 1999.

[6] M. Macas et C. Wu, « Review: Deep Learning Methods for Cybersecurity and Intrusion Detection Systems », dans *Proceedings - 2020 IEEE Latin-American Conference on Communications, LATINCOM 2020*, Institute of Electrical and Electronics Engineers Inc., nov. 2020. doi: 10.1109/LATINCOM50620.2020.9282324.

[7] R. K. Mishra, G. Y. S. Reddy, et H. Pathak, « The Understanding of Deep Learning: A Comprehensive Review », 2021, *Hindawi Limited*. doi: 10.1155/2021/5548884.

[8] A. Prakash, M. Satish, T. Sri Sai Bhargav, et N. Bhalaji, « Detection and Mitigation of Denial of Service Attacks Using Stratified Architecture », dans *Procedia Computer Science*, Elsevier B.V., 2016, p. 275-280. doi: 10.1016/j.procs.2016.05.161.

[9] J. J. Kponyo, J. O. Agyemang, G. S. Klogo, et J. O. Boateng, « Lightweight and host-based denial of service (DoS) detection and defense mechanism for resource-constrained IoT devices », *Internet of Things (Netherlands)*, vol. 12, déc. 2020, doi: 10.1016/j.iot.2020.100319.

[10] M. A. Al-Shareeda, S. Manickam, et M. A. Saare, « DDoS attacks detection using machine learning and deep learning techniques: analysis and comparison », *Bulletin of Electrical Engineering and Informatics*, vol. 12, nº 2, p. 930-939, avr. 2023, doi: 10.11591/eei.v12i2.4466.

[11] Z. Wang, Y. Zeng, Y. Liu, et D. Li, « Deep Belief Network Integrating Improved Kernel-Based Extreme Learning Machine for Network Intrusion Detection », *IEEE Access*, vol. 9, p. 16062-16091, 2021, doi: 10.1109/ACCESS.2021.3051074.

[12] Y. Lecun, Y. Bengio, et G. Hinton, « Deep learning », *Nature*, vol. 521, nº 7553, p. 436-444, 2015, doi: 10.1038/nature14539.

[13] G. E. Hinton, « Learning multiple layers of representation », octobre 2007. doi: 10.1016/j.tics.2007.09.004.

[14] R. A. R. Ashfaq, X. Z. Wang, J. Z. Huang, H. Abbas, et Y. L. He, « Fuzziness based semi-supervised learning approach for intrusion detection system », *Inf. Sci. (N Y).*, vol. 378, p. 484-497, févr. 2017, doi: 10.1016/j.ins.2016.04.019.

[15] X. Cai, S. Hu, et X. Lin, *Feature Extraction Using Restricted Boltzmann Machine for Stock Price Prediction*. Institute of Electrical and Electronics Engineers., 2012.

[16] A.-R. Mohamed, D. Yu, et L. Deng, « Investigation of Full-Sequence Training of Deep Belief Networks for Speech Recognition », 2010.

[17] Y. Fu, F. Lou, F. Meng, Z. Tian, H. Zhang, et F. Jiang, « An intelligent network attack detection method based on RNN », *Proceedings - 2018 IEEE 3rd International Conference on Data Science in Cyberspace, DSC 2018*, p. 483-489, juill. 2018, doi: 10.1109/DSC.2018.00078.

[18] L. Deng *et al.*, *2013 IEEE International Conference on Acoustics, Speech, and Signal Processing : proceedings : May 26-31, 2013, Vancouver Convention Center, Vancouver, British Columbia, Canada*. USA: IEEE, 2013.

[19] M. Bojarski *et al.*, « Explaining How a Deep Neural Network Trained with End-to-End Learning Steers a Car », avr. 2017, [En ligne]. Disponible à: http://arxiv.org/abs/1704.07911

[20] L. Ashiku et C. Dagli, « Network Intrusion Detection System using Deep Learning », dans *Procedia Computer Science*, Elsevier B.V., 2021, p. 239-247. doi: 10.1016/j.procs.2021.05.025.

[21] D. S. Rajput et A. K. Upadhyay, « Enhanced Network Defense: Optimized Multi-Layer Ensemble for DDoS Attack Detection », *International Journal of Experimental Research and Review*, vol. 46, p. 253-272, 2024, doi: 10.52756/ijerr.2024.v46.020.

[22] H. Elubeyd et D. Yiltas-Kaplan, « Hybrid Deep Learning Approach for Automatic DoS/DDoS Attacks Detection in Software-Defined Networks », *Applied Sciences (Switzerland)*, vol. 13, nº 6, mars 2023, doi: 10.3390/APP13063828.

[23] H. Kumar, Aoudni Yassine, Ortiz Geovanny Genaro Reivan, Jindal Latika, Miah Shahajan, et Tripathi Rohit, « Light Weighted CNN Model to Detect DDoS Attack over Distributed Scenario », *Security and Communication Networks*, vol. 2022, 2022, doi: 10.1155/2022/7585457.

[24] U. Mbasuva et G. A. L. Zodi, « Designing Ensemble Deep Learning Intrusion Detection System for DDoS attacks in Software Defined Networks », dans *Proceedings of the 2022 16th International Conference on Ubiquitous Information Management and Communication, IMCOM 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/IMCOM53663.2022.9721785.

[25] Alluraiah K. et Chetty Manna, « Ensemble Learning Method for DDoS Attack Mitigation in Web Based Networks », 2024.

[26] A. Ahmim, F. Maazouzi, M. Ahmim, S. Namane, et I. Ben Dhaou, « Distributed Denial of Service Attack Detection for the Internet of Things Using Hybrid Deep Learning Model », *IEEE Access*, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3327620.

[27] R. Mall, K. Abhishek, S. Manimurugan, A. Shankar, et A. Kumar, « Stacking ensemble approach for DDoS attack detection in software-defined cyber–physical systems », *Computers and Electrical Engineering*, vol. 107, p. 108635, avr. 2023, doi: 10.1016/J.COMPELECENG.2023.108635.

[28] A. V. Kachavimath et D. G. Narayan, « An Efficient DDoS Attack Detection in SDN using Multi-Feature Selection and Ensemble Learning », *Procedia Comput. Sci.*, vol. 252, p. 241-250, janv. 2025, doi: 10.1016/J.PROCS.2024.12.026.

[29] S. Haider *et al.*, « A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks », *IEEE Access*, vol. 8, p. 53972-53983, 2020, doi: 10.1109/ACCESS.2020.2976908.

[30] Jummah N. S et Ashkafaki A. T, « Hybrid Ensemble Deep Learning Framework for Efficient DDoS Attack Detection in Software-Defined Networks », 2024.

[31] T. E. Ali, Y. W. Chong, S. Manickam, M. N. Yusoff, K. L. A. Yau, et A. D. Zoltan, « A Stacking Ensemble Model with Enhanced Feature Selection for Distributed Denial-of-Service Detection in Software-Defined Networks », *Engineering, Technology and Applied Science Research*, vol. 15, nº 1, p. 19232-19245, févr. 2025, doi: 10.48084/etasr.8976.

[32] M. Ouhssini, K. Afdel, E. Agherrabi, M. Akouhar, et A. Abarda, « DeepDefend: A comprehensive framework for DDoS attack detection and prevention in cloud computing », *Journal of King Saud University - Computer and Information Sciences*, vol. 36, nº 2, févr. 2024, doi: 10.1016/j.jksuci.2024.101938

.