SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I3P121

Volume 13 Issue 3, 266-281, March 2026

Original Article

A Hybrid Enhanced Autoencoder and DNN Model with Adaptive Swarm Optimization for Cyberattack Detection

A. Kalaivani[1], R. Pugazendi[2]

[1,2]*Department of Computer Science, Government Arts College (Autonomous), Salem, Tamilnadu, India.*

[1]*Corresponding Author : kalai24.vinod@gmail.com*

Received: 15 December 2025 Revised: 17 January 2026 Accepted: 23 February 2026 Published: 23 March 2026

Abstract - *Modern cybersecurity frameworks must include an IDS - intrusion detection system to detect and address any threats to computer networks and systems. As cyberattacks become increasingly complex, traditional intrusion detection systems often fail to perceive novel or evasive threats. To address existing IDS limitations, such as a lack of interpretability, high false-positive rates, and vulnerability, this paper presents a new hybrid IDS based on Deep Neural Networks (DNN) that can automatically learn and extract features from unprocessed network data. This study is new because it combines an Enhanced Variational Autoencoder (EVAE) with a Deep Neural Network (DNN) to perform an excellent task of representing and classifying features. The proposed EVAE–DNN architecture retains high-level latent traits while decreasing redundancy and altering model parameters using Adaptive Particle Swarm Optimization, unlike previous hybrid models. This dual-stage strategy improves learning, generalization, and investigation accuracy across attack categories. The CSE-CIC-IDS2018 dataset included user activity and network traffic patterns for training and validation. OEVAEDNN_IDS outperforms other deep learning models in F1-score, recall, precision, and accuracy. Cyber threat detection and reduction are efficient, adaptive, and effective with the proposed framework.*

Keywords - *Deep Learning, Intrusion detection system, Hyperparameter optimization, PSO, Variational autoencoder.*

1. Introduction

Nowadays, cybersecurity is riskier than ever in the linked digital world. Organizations with complicated networks are the biggest cyberattack victims. Companies use IDs to identify and regulate security events to prevent traffic and protect important data. Traditional IDS involves identifying anomalies to identify known attack patterns that depart from norms [1]. For sophisticated, confirmed, or unknown dangers, such methods may fail. IDS monitors system activity and network traffic to identify malicious attacks, illegal access, and other activities. Current digital security threats are so severe and frequent that IDS is essential for identifying and avoiding corporate network vulnerabilities. Ensuring the availability, confidentiality, and integrity of IDS information systems requires alerts for detecting malware activity and data theft attempts. IDS alerts suspicious activity online in real time. ID's early reaction and attack-detection capabilities enable bodies to act before serious damage begins. Real-time, automated guidance speeds up threat response and reduces assault damage. Data security is necessary in many sectors. Through continuous security event inspection and recording, IDS helps enterprises satisfy these compliance criteria. To identify malicious conduct, the IDS monitors network traffic and device activity. Finding patterns. Security teams get notifications to investigate or halt attacks. Before a significant breakdown or data loss, the infiltration investigation system detects first invasions and mitigates harm. These categories describe ID:

1. NIDS - Network-based IDS: Checks network activity to find tendencies that do not seem right
2. HIDS - Host-based IDS: Observers' operations on different hosts, such as file integrity, system calls, and user activity [2, 3].

DL models are helping IDS overcome these limits. Machine learning's DL branch automatically finds and predicts complex patterns in big datasets. Traditional IDS methods use human convenience extraction and rule-based systems, whereas DL models can automatically uncover complicated patterns, adapt to new risks, and analyze more. The ID is likely to play an important role in making IDS more reliable, preventing misuse, and increasing its effectiveness when moving to DL. DL models can analyze large volumes of data from sources ranging from known to unknown: system logs, network traffic, etc. Moreover, they can

continuously collect new data to adapt to changes in cybercriminal attacks [4].

DL models have distinguishing characteristics, and popular ones like CNNs (Convolutional Neural Networks) [5], RNNs (Recurrent Neural Networks) [6], Autoencoders [7], and DNNs [8] are increasingly being adopted for intrusion detection. For instance, traffic anomalies can be detected using a CNN network, since it is good at pattern recognition in the sky. At the same time, RNNs are good at handling sequential data, such as traffic flow. Deep learning (DL) has great potential but also faces challenges of interpretation, complexity, and the requirement for a large labelled images for training. But research is still going on; deep-learning-based IDS systems are poised to become a feature of current cybersecurity defense. This will make it easier to spot and respond to new threats right away. This demonstrates how the field of cybersecurity is changing due to these innovative techniques, offering organizations more robust, adaptive tools to safeguard their networks and build resilient systems that withstand an ever-growing range of cyber threats.

In an IDS, hyperparameter tuning is a challenging process for a DNN. In this process, many hyperparameter values are specified during model building and training. Network layer count, node count per layer, activation function, and learning rate are a few of the hyperparameters used [9]. Initially, physical hyperparameter tuning is labor-intensive, and a domain expert is needed to expedite the process. Thus, there is a need for an automated technique for hyper-parameter adaptation to enhance the implementation of DL models. The hyper-parameter tuning process was initially performed using grid and random searches [10], which are time-consuming. The goal is to advance an automated IDS by optimizing hyperparameters to detect cyberattacks. In DL, the possibilities are endless, including independent feature extraction, self-learning, scalability, transfer learning, and the ability to combine different DL models.

Current deep learning approaches for cyberattack detection primarily adopt either unsupervised autoencoder-based feature extraction or supervised deep neural classification independently. While autoencoders and Variational Autoencoders (VAEs) facilitate dimensionality reduction and probabilistic latent representation learning, the resulting latent embeddings often exhibit limited inter-class separability under imbalanced and noisy traffic conditions. Conversely, standalone Deep Neural Networks (DNNs) emphasize discriminative learning but may not adequately exploit structured latent manifold regularization. In addition, most frameworks employ static or heuristic hyperparameter configurations, which constrain convergence stability and generalization performance across heterogeneous datasets. The central problem addressed in this study involves the design of a hybrid deep learning framework capable of generating probabilistically regularized and highly separable latent representations while maintaining strong supervised classification performance in high-dimensional and imbalanced cyber traffic environments. The framework must also incorporate adaptive optimization mechanisms to dynamically calibrate model hyperparameters, thereby improving convergence efficiency, robustness, and detection accuracy across diverse cyberattack categories.

The proposed Enhanced Variational Autoencoder with Deep Neural Network (EVAE-DNN) framework with adaptive Particle Swarm Optimization (PSO) has a unique unified optimization-driven generative–discriminative architecture for cyberattack detection. For anomaly detection, most research uses reconstruction-based autoencoder or Variational Autoencoder (VAE) models, where classification is inferred indirectly from reconstruction loss, or fully supervised Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), or Recurrent Neural Networks (RNNs) that use raw or manually engineered features without probabilistic latent regularization. The majority of hybrid techniques handle feature extraction and classification as sequentially distinct modules and use static, heuristic, or grid-based hyperparameter tweaking.

In contrast, the proposed framework uses a tightly coupled latent representation learning mechanism in which the enhanced VAE generates structured and distribution-aware embeddings directly optimized by a supervised DNN classifier to improve inter-class separability in nonlinear and high-dimensional traffic spaces. Through iterative velocity–position updates and adaptive inertia management, adaptive PSO dynamically calibrates key architectural and learning parameters for balanced search space exploration–exploitation and improved convergence stability. This integrative, optimization-aware design improves latent manifold organization, discriminative boundary formation, and robustness under imbalanced multi-class cyberattack scenarios compared to recent deep learning–based intrusion detection systems, creating a scalable, computationally efficient framework for intelligent intrusion detection.

The following lists this study's primary contributions:

- A set of preprocessing operations, including feature removal, random sampling, replica elimination, feature selection, and normalization, is applied to the CSE-CIC-ID2018 dataset, which comprises all attack types in both the training and testing datasets.
- A hybrid architecture that combines EVAE and DNN models has been designed.
- Hyperparameters are automatically optimized using APSO techniques. This optimization technique explores the search space resourcefully.
- Several DL and ML models are contrasted with the proposed one.

This research article is prepared as follows. In Section 2, related reviews are detailed. Methodology for OEVAEDNN_IDS, including dataset preparation, the architectures of EVAE and DNN, and hyperparameter optimization, is presented in Section 3. Section 4 goes into further detail about the OEVAEDNN_IDS approach's experimental design, assessment metrics, experimental findings, and analysis. Lastly, Section 5 talks about the conclusion and improvements that could be made in the future.

2. Related Reviews

There have been many reviews of ML methods for IDS in the last few years, but few of the DL methods. Recent progress has been made in ML and DL techniques for handling large-scale data. ML and DL algorithms are employed in computer security to address problems in intrusion detection. This section includes some of the most recent research on cyberattacks using ML and DL approaches. The discussions are structured around general and uncommonly used datasets used for training and testing, and around different ML and DL approaches to IDS development.

In network traffic, many surface-level ML approaches are employed to detect and distinguish normal and malicious traffic. IDS leverages the most widely used machine learning methods, including KNN, ANN, SVM, and NB networks. In [11], a detailed survey on ML methods was reviewed and analyzed. A deep-layered RNN gathers HDLNIDS data, whereas a CNN uses convolutional layers to extract local features, thus increasing ID system accuracy and reliability. Experiments utilizing publicly available intrusion detection data evaluate the HDLNIDS system. The CICIDS-2018 dataset is accurate and realistic. Simulation findings indicate that the HDLNIDS improves accuracy and reduces data loss.

In [12], two new hybrid deep learning models were suggested to improve intrusion detection: a Transformer–Deep Neural Network and an Autoencoder–Convolutional Neural Network. To address class imbalance and improve network traffic data, an autoencoder was used. At the same time, the transformer part received contextual information from the DNN, and the CNN correctly sorted it out. To further refine the class distribution, the Edited Nearest Neighbors (ENN) method was applied.

An improved variant of SMOTE was adopted to address imbalance in multi-class scenarios, while, for binary classification tasks, to effectively mitigate class imbalance, a modified hybrid (ADASYN–SMOTE)- Adaptive Synthetic Sampling–Synthetic Minority Oversampling Technique was employed. To improve real-time detection capabilities, decrease false negatives and false positives, and successfully detect zero-day threats, the models were developed. Precision scores of 99.90% and 99.92% are obtained for binary classification when tests are performed on the CICIDS2017 dataset with the Autoencoder–CNN and Transformer–DNN models. For multi-class classification tasks, they got precision scores of 99.95% and 99.96%, respectively. The suggested hybrid models outperformed conventional techniques in addressing a variety of network attacks. The Transformer–DNN achieved 99.98% binary accuracy on the NF-BoT-IoT-v2 dataset, and the classification accuracy was around 97.90%, while the Autoencoder–CNN achieved 99.98% binary accuracy and multi-class classification accuracy around 97.95%.

The HawkPhish-DNN cybersecurity model, a new approach to detecting phishing, was developed by combining Harris Hawk Optimization (HHO) with a DNN in [13]. Duplicate URLs and domain-based features were removed during preprocessing to create a useful, unique set of features. Entropy and URL length were extracted. To improve learning and classification, the detection framework included multi-objective HHO and neural network layers like Sigmoid and ReLU activation functions. HawkPhish-DNN traded precision and recall using crowding distance, Pareto dominance, and a time-varying penalization function to reduce false positives and improve detection. Experimental investigation demonstrated that the HawkPhish-DNN cybersecurity model has low computing overhead and 99.6% accuracy on benchmark datasets with a 0.2% false-positive rate. The HawkPhish-DNN model's performance and application to real-time phishing detection provide a powerful but accessible protection against phishing without a significant false-positive rate.

The improved Intrusion Detection for Cybersecurity (IDCS) approach [14] uses ensemble training and customized Beluga Whale Optimization (IBWO) to identify network intrusions and improve cybersecurity. This approach initially normalizes feature values using min–max normalization. For feature selection, the Remora Optimization Algorithm (ROA) reduced computational complexity while keeping useful characteristics. The IDCS-ELIBWO framework used top-notch learning models, such as DBN, LSTM, and GRU architectures, to learn complex spatial and temporal features from network flows for intrusion detection. IBWO: The Improved Beluga Whale Optimization method improved detection performance by optimizing hyperparameters. After extensive examination, the IDCS-ELIBWO methodology obtained 99.77% detection accuracy, surpassing other state-of-the-art approaches. A hybrid intrusion detection strategy using GWO and PSO was presented in [15] to enhance the structure of a DNN for Sunburst attack detection. In this model, GWO optimized neuron weight parameters while PSO determined the best hidden layers and learning rate. The hybrid system was trained, validated, and assessed using open-source Sunburst attack datasets. In experiments, the hybrid DNN-based model showed resilience and reliable

identification. The hybrid PSO–GWO technique was also compared to ACO, DE, and GA hybrid optimization methods. The PSO–GWO hybrid strategy beat these optimization methods in F1-score, accuracy, recall, and precision. The authors [16] suggested an IDS utilizing SVM and Naïve Bayes classifiers, targeting a 24-feature association subset from the 42-feature NSL-KDD dataset. Data is standardized, and attributes are binaryized during preparation. The SVM classifier is 93.95% accurate. A hybrid technique using two machine learning algorithms is presented in [17] to pick and categorize data and detect dangers. This system classifies approach types using Random Forest feature selection with Classification and Regression Trees (CART), which discovers key properties. In the UNSW-NB15 dataset, the recommended technique performs better.

Using flow statistics and a five-level hybrid classification strategy improves system accuracy [18]. The first stage utilizes k-Nearest Neighbors (kNN), the next stage uses an ELM, and the next stages use a HELM. This multi-level strategy is compared to classic supervised machine learning and current methods. Excellent accuracy (84.29%) and 77.18% accuracy in detecting new assaults on the NSL-KDD benchmark dataset. Ensemble learning was used to improve intrusion detection by merging several detection techniques [19]. Seven single classifiers were compared to get the best ensemble base learners. The logistic regression, decision tree, and gradient boosting models performed well in experiments and were included in the ensemble framework. The Canadian Institute for Cybersecurity and Communications Security Establishment 2018 (CSE-CIC-IDS2018) dataset verified the methodology's efficacy. We utilized Spearman's rank correlation coefficient to detect and delete duplicate or less instructional attributes and assess feature importance.

The initial 80 characteristics were reduced to 23 relevant features. Experimental findings showed good performance with a final F1-score of 97.9%, recall of 97.1%, accuracy of 98.8%, and precision of 98.8%. In [20], a CDMDoW-AMOAFL model combines powerful metaheuristic optimization methods to identify and prevent RoW threats in federated learning. In preprocessing, the author used z-score normalization to prepare the raw data. The author used HHO (Harris Hawk Optimization) to pick the most informative dataset attributes. The authors suggested employing Gated Recurrent Units, TCNs, and CAEs in ensemble learning to identify complicated temporal and spatial cyberattack patterns. MMPA (Modified Marine Predator Algorithm) was used to alter ensemble model hyperparameters, improving classification performance. The effectiveness of the CDMDoW-AMOAFL method was demonstrated through thorough experimental studies on a Denial-of-Wallet attack detection data set, achieving a classification rate of 98.12%, which outperforms some state-of-the-art methods.

In [21], HAEMPSO (An Autoencoder with a Modified Particle Swarm Optimization) was used for efficient feature selection, and a DNN (Deep Neural Network) was also employed; the authors introduced a novel hybrid intrusion detection model for IoT environments. A refined set of DNN parameters was obtained using an improved PSO with an adaptive inertia-weighting scheme, resulting in faster training and higher detection accuracy. The experimental evaluation utilized two practical IoT-relevant datasets, UNSW-NB15 and BoT-IoT. The insignificant class obtains 99.7% of DR - detection rate and 99.9% of accuracy on the UNSW-NB15 dataset, whereas the proposed HAEMPSO framework attained 98.8% accuracy and a 99.9% DR in detecting generic attacks. In the BoT-IoT dataset, 99.22% accuracy is achieved against the DDoS HTTP attack, and 97.79% DR. In contrast, the benign class achieved 97.54% accuracy and 97.92% DR. A relative comparison with state-of-the-art ML approaches showed that the HAEMPSO-DNN framework achieved competitive performance across both detection rate and accuracy.

In [22], a two-stage hybrid technique was proposed for a DDoS prediction system. DSAE - A Deep Sparse Autoencoder with Elastic Net regularization and properly adjusted hyperparameters was used in the initial step to extract features. To categorize assaults using the retrieved feature sets, many learning models were refined and used in the second step. The model’s performance was then examined under both balanced and unbalanced data conditions. Results from experiments showed that the suggested strategy worked better than current methods. Using the CICIDS-2017 and CICDDoS-2019 datasets, the efficacy of the proposed DDoS prediction system was confirmed through evaluation, yielding accuracy values of 99.98% and 99.99%, respectively.

Recent studies show that intrusion detection research has shifted toward hybrid DL as an alternative to conventional ML techniques and metaheuristic-optimization-based models to address complex, evolving cyberattacks. While conventional ML approaches provide moderate accuracy, they face limitations with high-dimensional data, class imbalance, and zero-day attack detection. Hybrid deep learning architectures integrating CNNs, RNNs, autoencoders, transformers, and ensemble learning, combined with optimization techniques like HHO, GWO, PSO, and their variants, consistently achieve superior performance across benchmark datasets, often exceeding 99% accuracy. Advanced sampling and feature selection strategies further reduce false positives and improve detection reliability. However, most studies remain dataset-specific and offline, indicating a need for future research focused on real-time deployment, cross-dataset generalization, and computational efficiency for practical cybersecurity applications.

3. Proposed Methodology

The proposed procedure for identifying attacks using DNN is discussed here. Initially, the preprocessing procedures for the dataset are discussed. Next, the architecture of EVAE and DNN is discussed. In the next phase, the hyperparameters are optimized using APSO and are discussed. Finally, the discussion turns to the suggested hybrid architecture for network traffic classification. The proposed flow diagram is provided in Figure 1.

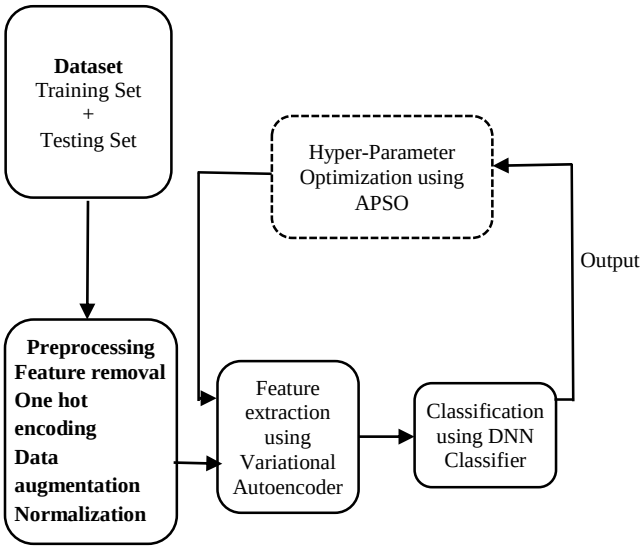


Fig. 1 Proposed architecture of the OEVAEDNN_IDS technique

3.1. Data Preprocessing Procedures

This section talks about the steps that need to be taken to clean up the dataset. To make the dataset smaller, unnecessary characteristics and duplicate data are removed first. Finally, normalization was performed. The CSE-CIC-IDS2018 dataset comprises 11 attack types and more than 3 million records. The dataset contains 80 features, including statistical features such as packet counts, byte counts, flow statistics, packet length, time interval, etc., and flow-based characteristics, including IP addresses of source and destination, duration, protocol type, service type, flags, etc. Table 1 lists all attack categories along with their types and classes.

Table 1. Description of attack categories and their classes

Category	Classes (Subtypes)
Benign	Normal traffic
Infiltration	Infiltration (generic)
Botnet	Botnet
DDoS	LOIC-HTTP, LOIC-UDP, LOIC-TCP, HOIC
DoS	SlowHTTPTest, Slowloris, GoldenEye, and Hulk
Web Attack	SQL Injection, Brute Force (HTTP), and XSS
Brute Force	FTP-BruteForce, SSH-BruteForce

The various preprocessing procedures and data imbalance techniques performed on the data set are presented below:

In the feature-removal procedure, zero- or null-valued attributes in the dataset are removed. In the random selection procedure, samples with the same number from each attack type were selected. In the duplicate elimination procedure, duplicates are removed based on attack type.

Non-numeric features are present, so the data are encoded using a one-hot encoder. The encoder converts non-numerical features into numerical features.

Data augmentation is performed using the Adaptive Synthetic Sampling (ADASYN) algorithm, which generates synthetic samples to address imbalance issues and improve the network's classification performance.

Normalization is performed to address the large gap in the dataset's dimensional features. The MinMaxScaler (Bisong 2019) was applied for data mapping into the range (0,1) as represented in eqn (1):

$$X = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Here, the maximum is represented by xmax, and the minimum values are denoted by xmin.

In the feature procedure, a set of features is selected based on the criteria. This process enables fast model creation, minimizing training duration. In this study, the best features are selected based on the highest score using the SelectKBest method.

To handle data imbalance, data-level techniques such as oversampling, undersampling, and hybrid approaches are used.

Training and testing datasets are segregated data throughout the data splitting process. This study will utilize 80% of the data in the training dataset, which is optimized with hyperparameters to make the suggested model work better.

The preprocessing step is very important for improving the quality and discriminative power of the features that the proposed EVAE - Enhanced Variational Autoencoder extracts. Preprocessing lowers data duplication and provides a consistent, well-structured input for the model in this investigation. Techniques such as data growth using duplicate elimination, one-hot encoding, normalization, and ADASYN to balance class distributions and to standardize the scales of the facility. This systematic cleaning and transformation process eliminates noise, handles missing or

non-nominal values, and reduces bias caused by unbalanced attack categories. By normalizing all features to the same range and selecting only statistically significant symptoms with SelectKBest, the EVAE Encoder can effectively capture latent representations with fixed precision and reduced reconstruction error. As a result, the symptom of the preprocessing model serves as the basis for the extraction phase, enabling the EVAE-DNN hybrid framework to achieve more stable training, faster conversion, and more stable classification of benign and malicious traffic.

3.2. Proposed OEVAEDNN_IDS Architecture

In the proposed OEVAEDNN_IDS framework, the Adaptive Particle Swarm Optimization (APSO) algorithm serves as an intelligent controller that bridges the EVAE feature extraction phase and the DNN classification phase by optimizing the hyperparameters governing both components. Initially, the EVAE is trained to extract latent feature representations from preprocessed network traffic data.

The input to the DNN is these learned latent features, which classify between attack and benign classes. During this process, APSO works as an external optimization loop at the same time, always checking how well the DNN is doing using a set fitness function, which is usually based on the F1-score and classification accuracy.

A set of potential hyperparameters for each particle in the swarm, including batch size, learning rate, number of neurons, and dropout rate, is applied to train the DNN with EVAE features. The resulting performance metrics guide APSO in updating each particle's position and velocity toward the global best solution. Once convergence is reached, the optimal hyperparameter configuration from APSO is used to finalize the DNN training. Finally, the DNN classifier helps categorize traffic as usual or an attack.

3.2.1. Enhanced Variational Auto Encoder (EVAE)

Figure 2 shows the proposed OEVAEDNN_IDS framework, the detailed architecture of the EVAE operating. An encoder network and a decoder network are present in the EVA, with two main parts connected by a latent (bottleneck) level. This composition's main objective is to preserve unaltered features by compressing the essential features of a small, inactive representation of the supplied data.

In the encoder phase, input network traffic data, with numerous flow-based and statistical features, is converted into a lower-dimensional, potentially dormant space. The encoder does not directly output the fixed facility vector; Instead, it produces two dimensions, average (μ) and (σ) denotes the standard deviation, which define the Gaussian probability distribution simultaneously for each latent variable. EVA can describe uncertainty and transformation in network data with this potential composition, which is essential for properly modeling attack behaviors.

A latent space (bottleneck layer) compresses redundant and noisy characteristics. This only preserves network traffic's most discriminating features. Reparation Tactics makes Grad-based OPTIM PTIs possible by connecting to the Gaussian distribution (z).

$$z = \mu + \sigma \odot \epsilon, \qquad \epsilon \sim \mathrm{N}(0,1) \tag{2}$$

This approach ensures that the stochastic sampling process remains differentiable, allowing effective backpropagation during training.

In the decoder stage, the sampled latent variables are used to retrieve the initial input data. The decoder serves as the encoder's mirror image and learns to reproduce the input distribution as closely as possible. The reconstruction process minimizes the loss function. It combines the Kullback–Leibler (KL) divergence loss, which guarantees that the learnt latent distribution remains around the conventional normal distribution, using the reconstruction loss, which determines how much the input and output measurements differ from one another:

$$L_{EVAE} = L_{\text{reconstruction}} + \beta\, D_{KL}(q(z|x) \| p(z)) \tag{3}$$

Where β is a regularization coefficient that balances compression and reconstruction quality.

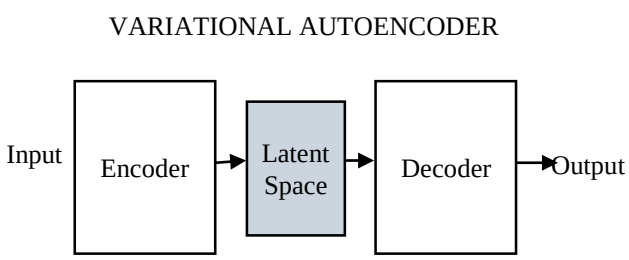


Fig. 2 Architecture of a variational autoencoder

The main improvement to the EVAE in this paper is its deeper network design, adaptive activation functions (ReLU/LeakyReLU), and fine-tuned layer configuration. These changes make it better at finding complex nonlinear features in big incursion datasets. By learning richer latent representations, the EVAE effectively separates benign and malicious traffic patterns before classification by the downstream DNN module.

3.2.2. Deep Neural Network (DNN)

DL aims to learn as many meaningful features as possible by creating multiple hidden layers to improve accuracy [23]. In a DNN structure, there is an input layer, many hidden layers, and an output layer. There are one or more neurons in each DNN layer that are completely connected in the forward direction from one layer to the next.

Figure 3 illustrates the architecture of DNN. An input vector $i = i_1, i_2, \ldots, i_n$ in, and where 'n' denotes the size, and its output vector $O(i)$, and 'o' denotes the size. The concealed layer h_j is mathematically expressed in (4) and (5).

$$h_j = \left(x_j^{l+1}\right) = f(Z_{ij} + b_j^{l+1}) \quad (4)$$

$$Z_{ij} = x_i^l w_{ij}^{l,l+1} \quad (5)$$

The complete neurons in layer j are attached to layer j. In (2) and (3), x_j^l is the neuron, $'i'$ denotes the activation function, 'l'denotes the layer, and the impact of neuron i on layer 'l' is denoted by Z_{ij} and neuron 'j' impact at layer $l + 1$. A nonlinear activation function 'f', $w_{ij}^{l,l+1}$ denotes the weight, b_j^{l+1}denotes the preference of neuron 'j'. The SoftMax layer serves as the classification activation function in this investigation. Many hidden layers of DNN are mathematically formulated in (6).

$$H_l(x) = H_l(H_{l-1}\left(H_{l-2}(\ldots H_i(x))\right)) \quad (6)$$

The hidden layers in the DNN architecture take inputs $i = i_1, i_1, i_2, \ldots, i_m$ and takes outputs as $o = o_1, o_2, \ldots, o_{c-1}$. A ReLU activation function is used on each hidden layer in a standard feed-forward system. It helps reduce fading and the dispute over fault gradients. ReLU is easy and quick to train with huge concealed layers.

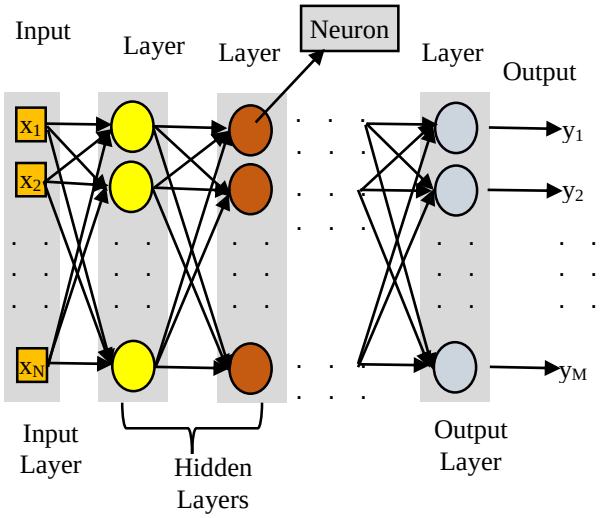


Fig. 3 Architecture of DNN

3.2.3. Hyperparameter optimization using APSO

In hyperparameters, parameters are not learnt; however, they are set once in the training model. To improve the model's performance, the hyperparameters may be adjusted. Manual tuning is a time-consuming process; hence, an effective hyperparameter optimization technique is required. This study uses an APSO to optimize the hyperparameters. PSO mimics the flocking behavior of birds. PSO is an optimization technique that is based on biology and searches the hyperparameter space in a dynamic way, making it better than random search. Here, APSO is applied to IDS to identify the balance between exploration and exploitation. This tuning process enhances classification performance. In the hyperparameter optimization procedure using APSO, the particles represent the set of hyperparameters in the DNN model. Here, S denotes the swarm; each particle pi denotes a position in the hyper-parameter space, and the velocity is updated. The $pbest_i$ and $gbest_i$ denote the private best and overall best positions derived from the fitness function calculation, which is a hyper-parameter set. The various steps involved in hyper-parameter optimization using APSO are given below:

Initialization is performed randomly for every particle's location and velocity in the hyper-parameter space.

Evaluation of every particle 'p_i' in swarm 'S' is estimated using the fitness function $f(x)$ used to train and validate the model using hyperparameters denoting its position x.

The particle's present position (pbest) and the global best position of the particle (gbest) should be updated.

Update velocity and location for every particle. The position 'x' is updated based on the updated velocity 'v'.

Terminate the process when the stopping condition is met. The best set of hyperparameters is obtained from the gbest position. Otherwise, the evaluation process continues.

Through the specified set of hyperparameters, the model is trained using the fitness function, and the outputs are checked. The aim of the fitness function is to maximize the fitness value. These APSO parameters, including the initialization range and the inertia weight, can be adjusted to find the best gbest solution. These searches are based on various hyperparameters (number of dense layers, number of neurons per layer, learning rate, batch size, number of filters in a convolutional network, kernel size, pooling factor, and epoch) to optimize.

To optimise classification accuracy, the proposed system modifies hyperparameters using the DNN APSO approach. APSO settings must be chosen to balance exploration and extraction during optimization. The swarm size (N) is 20-40 to provide population variety without computing cost. Inertia weight (w) is adaptively modified from 0.9 (global exploration in early iterations) to 0.4 (local exploitation in later iterations) to regulate the influence of previous velocities. The cognitive coefficient (c_1) and social

coefficient (c_2) are both set to 2, balancing personal and global learning. Random parameters r1 and r2 are uniformly distributed in [0,1] to add randomness, which is necessary for movement updates. The updates to velocity and position are limited to fixed ranges to prevent divergence or early convergence. With the above parameter settings, APSO dynamically adjusts the swarm behavior to converge more quickly toward the global optimal hyperparameter set, thereby improving model generalization and reducing training time, thereby benefiting intrusion detection performance practically.

Algorithm 1: Proposed OEVAEDNN_IDS Procedure

- Input:
- CSE-CIC-IDS2018 Dataset DDD containing 3 million records and 80 features
- Output:
- Optimized intrusion detection model with enhanced accuracy and reduced false positives.

Step 1: Data Preprocessing

- Input dataset D containing multiple attack and benign traffic records.
- Remove irrelevant or null features to reduce data dimensionality.
- Eliminate duplicate entries within each attack class to prevent bias.
- Perform one-hot encoding to get numerical values from category characteristics.
- Apply ADASYN (Adaptive Synthetic Sampling) to balance minority attack classes by generating synthetic samples.
- Normalize features to a uniform scale [0,1] using Limescale normalization:
- Select the best features using the *SelectKBest* method based on statistical score functions.
- Split the dataset into 20% for testing and 80% for instruction.

Step 2: Feature Extraction using Enhanced Variational Autoencoder (EVAE)

- Initialize the EVAE architecture, consisting of an *encoder*, *a latent space*, and a *decoder*.
- Encode input data into latent variables using Gaussian distributions.
- Compute reconstruction loss and KL divergence loss to train the EVAE.
- Extract optimized latent feature vectors from the trained encoder for downstream classification.

Step 3: Deep Neural Network (DNN) Classification

- Design a multi-layer feed-forward DNN comprising input, hidden, and output layers.
- Use ReLU activation in the hidden layers to mitigate vanishing gradients.
- Employ SoftMax activation to classify multi-class attacks at the output layer.
- Initialize model weights and feed latent features obtained from EVAE as inputs.

Step 4: Hyperparameter Optimization using Adaptive Particle Swarm Optimization (APSO)

1. Initialize swarm where each particle represents a hyperparameter set:
2. Assign random positions and velocities to all particles in the hyperparameter space.
3. For each iteration:
    - Evaluate the fitness function model accuracy on validation data.
    - The global best (gBest) for all particles and the personal best (pBest) for each particle are updated.
    - Update velocity and position of particles using:
    - Adaptively adjust inertia weight (w) to balance exploration and exploitation.
4. Repeat until either the convergence requirements or the maximum number of iterations are satisfied.
5. Obtain optimized hyperparameters corresponding to the final gBest.

Step 5: Model Training and Evaluation

1. Train DNN classifier with EVAE-derived features using optimized hyperparameters.
2. Evaluate the model with performance indicators on the testing dataset:
    - F1-score, recall, accuracy, and precision.
3. Compare results with baseline VAEDNN_IDS and existing IDS methods (CNN–RNN, CNN–BiLSTM, Sparse Autoencoder).
4. Select the best-performing model (OEVAEDNN_IDS) for deployment.

The feasibility of the proposed OEVAEDNN_IDS model for real-time intrusion detection was carefully examined with respect to computational complexity and inference time. The application of EVAE reduces input dimensionality and computation cost, particularly during loading, both for training and testing. The APSO-based hyperparameter optimization introduces additional training overhead, but it accelerates convergence and improves detection performance later. An average inference time of 0.83 milliseconds per sample and a training time of 178 seconds per epoch is accomplished by the final optimized EVAE–DNN model, demonstrating its applicability to real-time network monitoring. With $O(n \cdot d + m \cdot h)$ as its total computational difficulty, the number of data samples is represented by n, and the input features are denoted by d, and the DNN layers are given by m and h neurons per layer, respectively. These results confirm that the proposed approach maintains a high-throughput network system by achieving an acceptable balance between detection precision and processing performance.

4. Results and Discussion

This part explains the assessment matrix, the outcomes analysis, and the comparative study. From the CSE-CIC-IDS 2018 dataset, the proposed OEVAEDNN_IDS model was tested, which has 11 types of network assaults and more than 3 million records of normal traffic. All experiments were run in the Python 3.7 environment on a 64-bit Windows 10 platform, including a 16GB RAM, an Intel Core i7 CPU, and a NVIDIA GTX 1080 Ti GPU. The model's implementation used the SciLearn and TensorFlow Libraries. Training was 80%, and 20% of the dataset was used for testing. ADASYN ensured the introduction of a classic class. The Adaptive Particle Swarm Optimization (APSO) algorithm automatically tunes hyperparameters, eliminating the need for manual adjustment and improving convergence.

4.1. Evaluation Metrics

The proposed model OEVAEDNN_IDS was evaluated using F1 score (f1), recall (r), accuracy (a), and precision (p). This attack is a magic (i.e., spell) in the name of True Positive (TP), a False Negative, indicating that the data are abnormal. False Positive (FP) means a typical one that is mistakenly recognized as an attack. Attack traffic falsely labeled as benign is annotated as a False Negative (FN).

The suggested IDS is evaluated using different performance indicators to measure classification accuracy, reliability, and robustness. All metrics are computed using the confusion matrix components:

TP (True Positive): The quantity of assaults that were accurately categorized as effective.

TN (True Negative): Number of typical cases that were appropriately labeled as such.

FP (False Positive): Typical occurrences are mistakenly classified as assaults.

FN (False Negative): Attacks occur that were erroneously classified as usual.

All the evaluation metrics are represented in (7), (8), (9), and (10).

Accuracy: Calculates the percentage of all samples that were correctly predicted.

$$a = \frac{TP+TN}{TP+FP+TN+FN} \tag{7}$$

It provides an overall correctness measure but can be misleading under severe class imbalance.

Precision (Positive Predictive Value): Designates how many times an attack is really an attack.

$$p = \frac{TP}{TP+FP} \tag{8}$$

High precision implies fewer false alarms and more trustworthy alerts.

Recall (Detection Rate / Sensitivity): Measures how well the model identifies every actual assault.

$$r = \frac{TP}{TP+FN} \tag{9}$$

High recall ensures that most threats are captured, minimizing missed attacks.

F1-Score: The harmonic mean of each measurement was identified, and then precision and recall were placed together.

$$f_1 = \frac{2(TP+FP)(TP+FN)}{TP} \tag{10}$$

Effective for managing unbalanced collections, as it rewards models that maintain both high detection and low false alarm rates.

4.2. Result Analysis

The productivity and performance of the OEVAEDNN_IDS model are analyzed using the training and test datasets. EVAE maintains information and extracts optimal features by reducing the dimensionality. A DNN classification model is applied alongside EVAE to categorize network traffic. Table 2 shows the hyperparameters and their values used for parameter tuning. In this study, two types of models are proposed. The first model is a hybrid architecture combining a VAE and a DNN classifier. The second model is also a hybrid architecture along with a tuning strategy. The APSO algorithm updates hyperparameters to improve classification performance.

Table 2. Hyper-parameter optimization outcome using APSO

Hyper-parameter	Value
Filter count	128
Kernel size	11
Pooling size	5
Dense layer count	3
Neuron count in the dense layer	128
Dropout rate	0.368
Learning rate	0.0008
Batch size	64
Epoch count	100

The performance of the two proposed models, VAEDNN_IDS and OEVAEDNN_IDS, on the CSE-CIC-IDS2018 dataset was compared and explained in Table 3. The output demonstrates that both strategies work well to find and

classify network risks. However, the OEVAEDNN_IDS model, which incorporates Adaptive Particle Swarm Optimization (APSO) for automatic hyperparameter tuning, exhibits slightly but consistently higher performance across all evaluation metrics. The OEVAEDNN_IDS model did better than the VAEDNN_IDS model, with 98.57% of accuracy, 98.72% of precision, 99.20% of recall, and 99.06% of F1-score. The OEVAEDNN_IDS model had an accuracy of 98.94%, a precision of 98.91%, a recall of 99.43%, and an F1-score of 99.10%.

The small but steady improvements show that adaptive optimization works well for fine-tuning model parameters, including learning rate, dropout, and neuron layout. This provides a better generalization, faster convergence, and less overfitting than the fixed-parameter training. F1-score and recall are also higher, indicating that the OEVAEDNN_IDS model is more capable of detecting malicious traffic with fewer false negatives, which is crucial for online intrusion detection systems.

Higher accuracy also reflects the model's advantage in curtailing false alarms, a key concern for operational reliability in the cyber safety domain. Taken together, these results further validate that the proposed APSO-based tuning strategy significantly enhances the investigation sensitivity and forecasting stability of the hybrid EVA-DNN framework.

Table 3. Performance comparison of the proposed approaches on CSE-CIC-ID2018

Approaches	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Proposed VAEDNN_IDS	98.57	98.72	99.2	99.06
Proposed OEVAEDNN_IDS	**98.94**	**98.91**	**99.43**	**99.1**

Figure 4 shows how well the VAEDNN_IDS and OEVAEDNN_IDS models work on the CSE-CIC-IDS2018 dataset. The VAEDNN_IDS model had an accuracy of 98.57%. The OEVAEDNN_IDS model had an accuracy of 98.94%, which implies that the OEVAEDNN_IDS model is more accurate.

The improvement demonstrates the effectiveness of Adaptive Particle Swarm Optimization (APSO) in fine-tuning hyperparameters, resulting in enhanced learning efficiency and higher intrusion detection accuracy. The accuracy values of the suggested IDS models are shown in Figure 5. 98.72% of accuracy was given by the VAEDNN_IDS model, while the OEVAEDNN_IDS model achieved a slightly higher precision of 98.91%. This improvement indicates that the OEVAEDNN_IDS model generates fewer false positives and provides more reliable identification of actual attack instances.

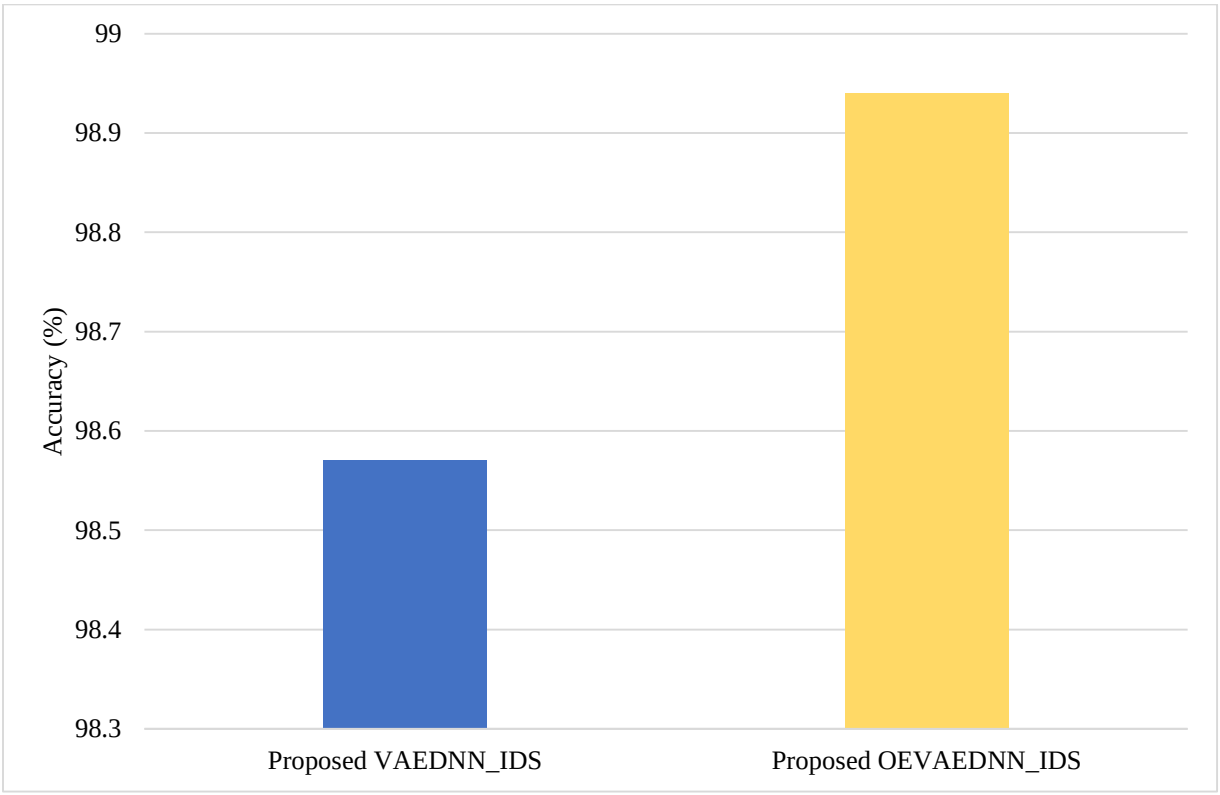


Fig. 4 Accuracy comparison

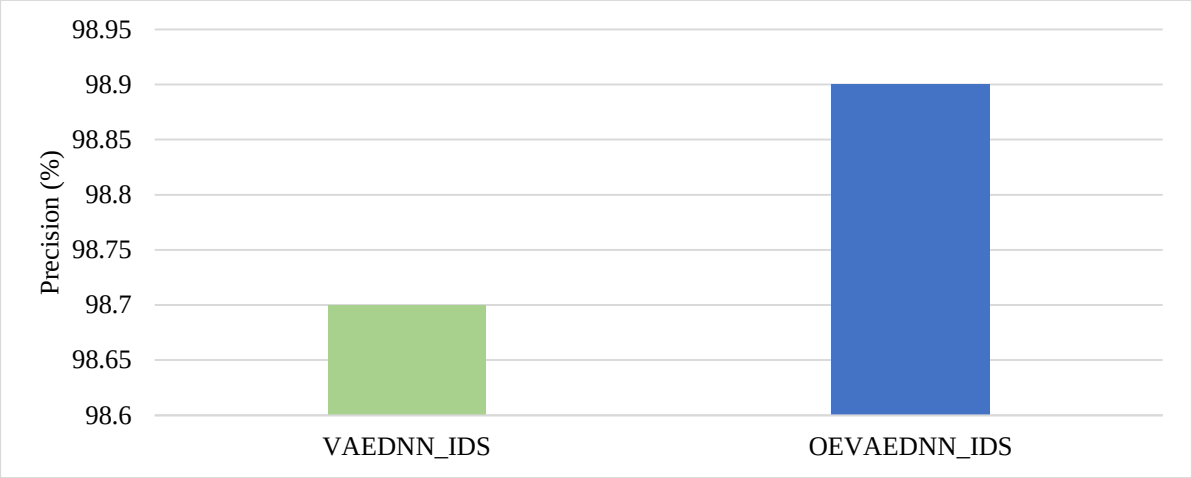


Fig. 5 Precision comparison

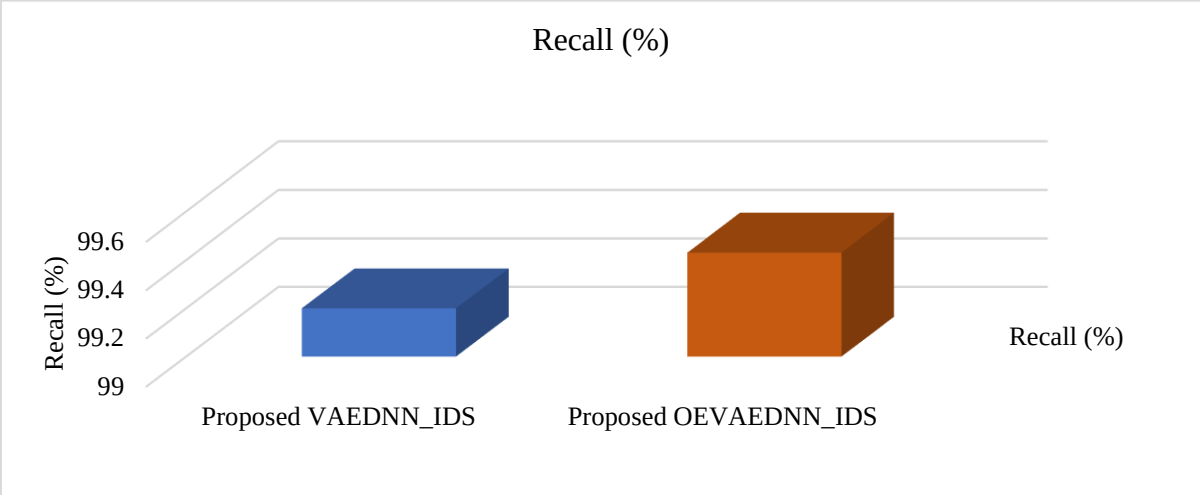


Fig. 6 Recall comparison

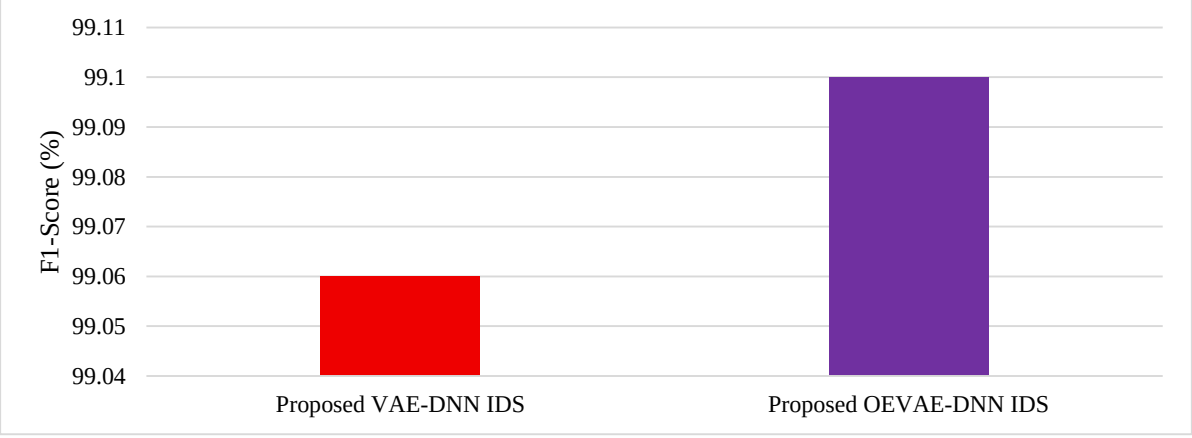


Fig. 7 F1-score comparison

Figure 6 shows the recall results of the proposed models. For the VAEDNN_IDS model, the recall is 99.20%, while the OEVAEDNN_IDS model achieves good performance with a recall of 99.43%. This enhancement shows OEVAEDNN_IDS is more efficient by achieving higher correct detection of attack instances, i.e., fewer false negatives and better threat coverage in all types of intrusions.

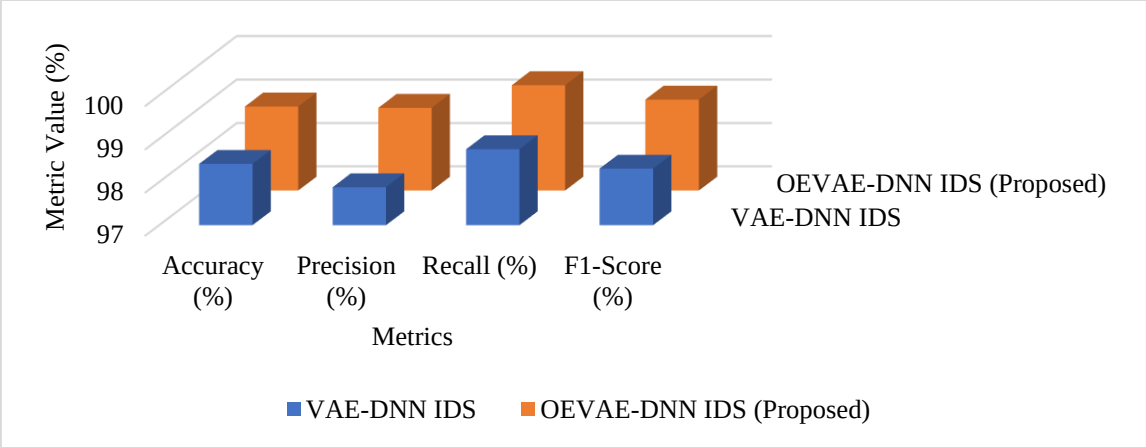


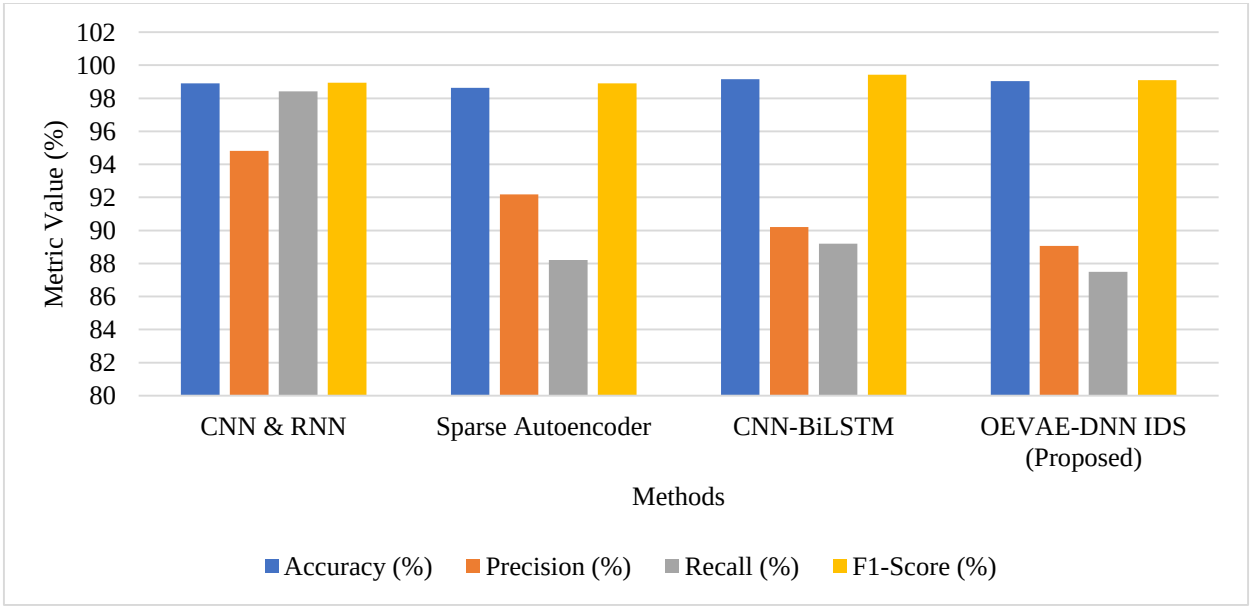
Fig. 8 Comparative analysis using various metrics on CSE-CIC-ID2018

The F1 scores of the suggested models are shown in Figure 7. While the OEVAEDNN_IDS significantly improved to 99.10%, the VAEDNN_IDS obtained an F1-score of 99.06%. This suggests that the optimized model balances precision and recall, resulting in more reliable and consistent intrusion detection performance.

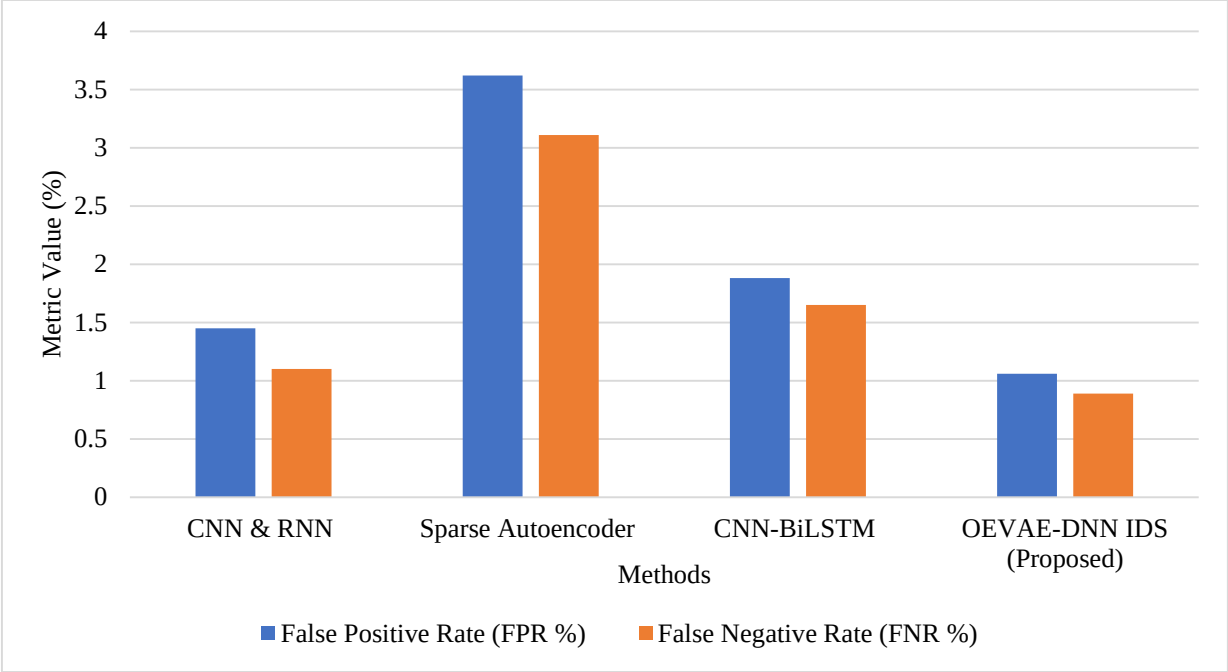
Figure 8 presents the two best proposed IDS models (i.e, VAEDNN_IDS, OEVAEDNN_IDS) performed against the CSE-CIC-IDS2018 dataset. This research found that the OEVAEDNN_IDS model significantly outperformed the baseline spatially-transformed VAE (VAE DNN) IDS, achieving 98.94% accuracy, 98.94% recall, 98.91% precision, and 99.10% F1-score. The enhancement suggests that introducing APSO improves the accuracy and efficiency of learning through hyperparameter optimization in the model. This makes it more reliable for detecting cyberattacks, with higher precision and recall.

Table 4 presents the relative performance of different IDS (Intrusion Detection System) architectures: CNN–RNN, Sparse Autoencoder, CNN–BiLSTM, and the proposed OEVAEDNN_IDS model. The evaluation metrics include the F1-score, recall, accuracy) and precision, the FPR- false positive rate, and FNR- negative rate (FNR).

The performance of the suggested OEVAEDNN_IDS model performs best in terms of accuracy with 98.94%, precision with 98.91%, recall with 99.43%, and an F1-score of 99.10%, while maintaining a lower value for both FPR (1.06%) and FNR (0.89%) compared to all other methods used above. Experimental results show that the OEVAEDNN_IDS model effectively strikes a better trade-off between the accuracy rate of abnormal detection and the number of false alarms, demonstrating robustness against various attacks.



(a)



(b)

Fig. 9 (a), and (b) Comparison of existing methods with the suggested OEVAEDNN_IDS model

The OEVAEDNN_IDS model achieves the highest accuracy (98.94%), as shown in Figure 9, demonstrating its remarkable ability to categorize both regular and attack traffic. The suggested model's precision (98.91%) and recall (99.43%) figures further show that it can reduce false alarms and accurately find different sorts of attacks. The F1-Score (99.10%), balanced categorization, and performance are assessed using the harmonic mean of recall and accuracy. In contrast, earlier models such as the Sparse Autoencoder and CNN–BiLSTM achieve lower accuracy and precision due to their limited feature learning and optimization capabilities. Moreover, the OEVAEDNN_IDS model demonstrates the lowest FPR (1.06%) and FNR (0.89%), signifying its robustness in avoiding false alerts and undetected attacks. The reduced false rates are a direct result of the adaptive hyperparameter optimization performed by APSO, which efficiently tunes the batch size, learning rate, and neuron count to get the best possible convergence.

Table 4. Comparison of present techniques with the suggested OEVAEDNN_IDS model

Reference	Approach	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FPR (%)	FNR (%)
Qazi et al. (2023)	CNN & RNN	98.90	98.64	99.15	99.03	1.45	1.10
Aminanto & Kim (2018)	Sparse Autoencoder	94.81	92.18	–	89.06	3.62	3.11
Said et al. (2023)	CNN-BiLSTM	98.42	–	–	–	1.88	1.65
Proposed (OEVAE-DNN IDS)	**EVAE-DNN**	**98.94**	**98.91**	**99.43**	**99.10**	**1.06**	**0.89**

The suggested model's detection rate is shown in Figure 10. Regarding the attack types of botnet, infiltration, DoS, DDoS, brute-force, online assault, and benign, the Proposed OEVAEDNN_IDS model achieved the most significant detection rate, 98%. The OEVAEDNN_IDS model had the highest detection rate of 98.82% for the benign category.

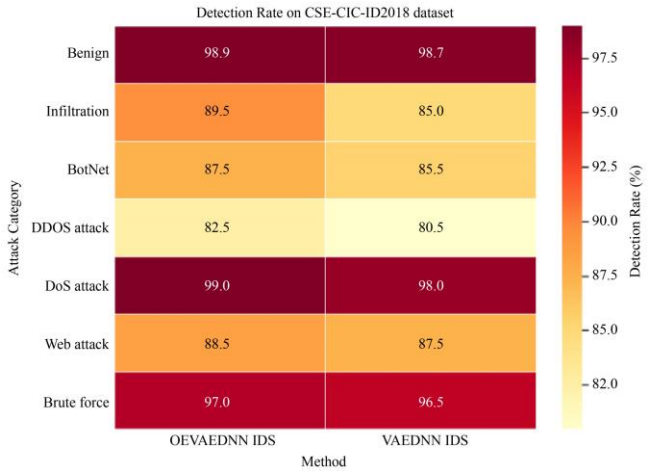


Fig. 10 Detection rate of different categories on the CSE-CIC-ID2018 dataset

4.3. Study Limitations and Future Research Directions

Despite the excellent detection accuracy of the suggested OEVAEDNN_IDS framework, its assessment is imperfect on the CSE-CIC-IDS2018 dataset, which raises doubts about how useful it is because it does not adequately show how complicated and varied real-world network traffic is. Although APSO-based hyperparameter optimization improves performance, the model's reliance on offline batch training may limit its scalability in dynamic or real-time settings.

Additionally, the system relies solely on flow-level characteristics, which may not be sufficient to identify complex attacks such as encrypted or stealthy threats, and its deep learning-based architecture offers limited interpretability, which could hinder its widespread implementation. Lastly, the framework is verified in a centralized, GPU-based setting, without accounting for deployment limitations associated with distributed networks, edge environments, or the Internet of Things.

Future study may focus on enhancing the robustness and generalizability of the proposed OEVAEDNN_IDS architecture by validating it across various benchmark datasets and real-world network traffic, as well as exploring cross-dataset evaluation and transfer learning methodologies to tackle emerging attack patterns.

Adding online and incremental learning to the model would enable it to respond in real time to new and evolving cyber threats. Using Explainable Artificial Intelligence (XAI) methodologies can clarify decision-making and build confidence among analysts by providing more insight into how models make decisions. To improve detection accuracy, speed up calculations, and save energy all at the same time, it is required to think about adopting hybrid or multi-objective optimization methods. Lastly, making the framework function in cloud-edge and IoT settings by employing lightweight architectures and adding temporal modeling techniques such as Transformers or graph neural networks could make it even better at finding coordinated, multi-stage hacks.

The improved performance of the proposed EVAE-DNN framework with adaptive Particle Swarm Optimization (PSO) is quantitatively reflected in its superior evaluation metrics compared with existing state-of-the-art approaches. Specifically, the proposed model achieves an accuracy of 98.94%, surpassing CNN-RNN (98.90%) and CNN-BiLSTM (98.42%) architectures, while significantly outperforming Sparse Autoencoder-based models (94.81%).

Precision improves to 98.91%, compared to 98.64% reported for CNN-RNN and 92.18% for Sparse Autoencoder models, indicating enhanced reliability in attack prediction. Recall reaches 99.43%, exceeding CNN-RNN (99.15%), demonstrating improved sensitivity to malicious traffic instances.

The F1-score of 99.10% further confirms balanced performance across precision and recall metrics. Importantly, the False Positive Rate (FPR) is reduced to 1.06%, compared to 1.45% (CNN-RNN) and 3.62% (Sparse Autoencoder), while the False Negative Rate (FNR) decreases to 0.89%, lower than 1.10% (CNN-RNN), 1.65% (CNN-BiLSTM), and 3.11% (Sparse Autoencoder). These measurable improvements result from the probabilistically regularized latent space generated by the enhanced VAE, which improves inter-class separability, combined with supervised DNN-based discriminative refinement and adaptive PSO-driven hyperparameter optimization.

5. Conclusion

In the cybersecurity domain, this study provides some contributions using DL techniques. This study proposed VAEDNN_IDS and OEVAEDNN_IDS models for the classification of attacks on the CSE-CIC-IDS-2018 dataset. Feature elimination and one-hot encoding are the Data preprocessing techniques, which are included in the suggested approach, and data augmentation and normalization are performed. Moreover, a hybrid architecture combining a DNN classifier for classification and a VAE for feature extraction is introduced.

The significance of fine-tuning the hyperparameters using the APSO method is discussed. For assessment, Performance indicators like f1-score, recall, accuracy, and precision are utilized. The outcomes show that the OEVAEDNN_IDS model effectively detects attacks by categorizing network traffic into attack and regular classes. The accuracy obtained by the OEVAEDNN_IDS model was 98.94%. The IDS paradigm may be cloud-based in the future. Hyperparameters may be tuned using hybrid bio-inspired optimization techniques.

References

[1] E. S. M. El-Alfy, and K. A. Al-Utaibi, "Learning mechanisms for anomaly-based intrusion detection: Updated review," In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1273-1281), 2017. [CrossRef] [Google Scholar] [Publisher Link]

[2] W. Haider, G. Creech, Y. Xie, and J. Hu, "Windows-based data sets for evaluation of robustness of host-based intrusion detection systems (IDS) to zero-day and stealth attacks," *Future Internet*, vol. 8, no. 3, pp. 29, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[3] M. Arun and G. Gopan, "Effects of natural light on improving the lighting and energy efficiency of buildings: toward low energy consumption and CO2 emission," *International Journal of Low-Carbon Technologies*, vol. 20, pp. 1047-1056, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[4] M. M. Najafabadi, F. Villanustre, T.M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, pp. 1, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[5] W. Jo, S. Kim, C. Lee, and T. Shon, "Packet preprocessing in CNN-based network intrusion detection system," *Electronics*, vol. 9, no. 7, pp. 1151, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[6] M. Arun, D. Barik, and S. S. Chandran, "Exploration of material recovery framework from waste–A revolutionary move towards clean environment," *Chemical Engineering Journal Advances*, vol. 18, p. 100589, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] C. Ieracitano, A. Adeel, F. C. Morabito, and A. Hussain, "A novel statistical analysis and autoencoder-driven intelligent intrusion detection approach," *Neurocomputing*, vol. 387, pp. 51-62, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] P. Devan, and N. Khare, "An efficient XGBoost–DNN-based classification model for network intrusion detection system," *Neural Computing and Applications*, vol. 32, no. 16, pp. 12499-12514, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] M. Mohaimenuzzaman, Z. S. Abdallah, J. Kamruzzaman, and B. Srinivasan, "Effect of hyper-parameter optimization on the deep learning model proposed for distributed attack detection in Internet of Things environment," arXiv preprint arXiv:1806.07057, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[10] J. Gu and S. Lu, "An effective intrusion detection approach using SVM with naïve Bayes feature embedding," *Computers & Security*, vol. 103, pp. 102158, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] E. U. H. Qazi, M. H. Faheem, and T. Zia, "HDLNIDS: hybrid deep-learning-based network intrusion detection system," *Applied Sciences*, vol. 13, no.8, pp. 4921, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] H. Kamal, and M. Mashaly, "Enhanced Hybrid Deep Learning Models-Based Anomaly Detection Method for Two-Stage Binary and Multi-Class Classification of Attacks in Intrusion Detection Systems," *Algorithms*, vol. 18, no. 2, p. 69, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[13] S. A. A. A. Alsaidi, H. J. Mohammed, R. R. N. Al Ogaili, Z. A. Dashoor, A. H. Alsaeedi, D. Al-Shammary, and A. Ibaida, "HawkPhish-DNN cybersecurity model: adaptive hybrid optimization and deep learning for enhanced multi-objective phishing URL detection," *International Journal of Information Technology*, vol. 17, pp. 1-17, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[14] F. Alhayan, N. Alruwais, M. Alamgeer, A. M. Alashjaee, M. Abdullah, A. O. Khadidos, F. S. Alallah, and A. Alshareef, "Design of advanced intrusion detection in cybersecurity using an ensemble of deep learning models with an improved beluga whale optimization algorithm," *Alexandria Engineering Journal*, vol. 121, pp. 90-102, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[15] M. Almseidin, A. Gawanmeh, M. Alzubi, J. Al-Sawwa, A. S. Mashaleh, and M. Alkasassbeh, "Hybrid deep neural network optimization with particle swarm and grey wolf algorithms for sunburst attack detection," *Computers*, vol. 14, no. 3, p. 107, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[16] K. K. Gulla, P. Viswanath, S.B. Veluru, and R.R. Kumar, "Machine learning based intrusion detection techniques," *In Handbook of Computer Networks and Cyber Security: Principles and Paradigms*, pp. 873-888, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Z. Chkirbene, S. Eltanbouly, M. Bashendy, N. AlNaimi, and A. Erbad, "Hybrid machine learning for network anomaly intrusion detection," In 2020 *IEEE International Conference on Informatics, IoT, and enabling technologies (ICIoT)*, pp. 163-170, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] M. Latah, and L. Toker, "An efficient flow-based multi-level hybrid intrusion detection system for software-defined networks," *CCF Transactions on Networking*, vol. 3, no. 3, pp. 261-271, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Q. R. S. Fitni, and K. Ramli, "Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems," In 2020 *IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pp. 118-124, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[20] E. Akhmetshin, D. Hudayberganov, R. Shichiyakh, S. Yellisetti, L. K. Pappala, R. D. Shukla, and S. Chandra, "An intelligent federated learning boosted cyberattack detection system for Denial-Of-Wallet attack using advanced heuristic search with multimodal approaches," *Scientific Reports*, vol. 15, no. 1, pp. 14265, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[21] Y. K. Saheed, A. A. Usman, F. D. Sukat, and M. Abdulrahman, "A novel hybrid autoencoder and modified particle swarm optimization feature selection for intrusion detection in the internet of things network," *Frontiers in Computer Science*, vol. 5, pp. 997159, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] R. K. Batchu, and H. Seetha, "A hybrid detection system for DDoS attacks based on deep sparse autoencoder and light gradient boost machine," *Journal of Information & Knowledge Management*, vol. 22, no. 01, pp. 2250071, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] F. Feng, X. Liu, B. Yong, R. Zhou, and Q. Zhou, "Anomaly detection in ad-hoc networks based on deep learning model: A plug and play device," *Ad hoc networks*, vol. 84, pp. 82-89, 2019. [CrossRef] [Google Scholar] [Publisher Link]