Journal of Theoretical and Applied Information Technology
15th June 2025. Vol.103. No.11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SGB-IDS: A SWARM GRADIENT BOOSTING INTRUSION DETECTION SYSTEM USING HYBRID FEATURE SELECTION FOR ENHANCED NETWORK SECURITY

VAHIDUDDIN SHARIFF[1], NVS PAVAN KUMAR[2], N ASHOKKUMAR[3], SURESH KUMAR MANDALA[4], MOHAN AJMEERA[5], N S KOTI MANI KUMAR TIRUMANADHAM[6*], P CHIRANJEEVI[7]

[1,6]Assistant Professor, Department of CSE, Sir C R Reddy College of Engineering, Eluru, Andhra Pradesh, India
[2]Assistant Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India
[3]Professor, Department of ECE, Mohan Babu University, Tirupati, Andra Pradesh, India
[4]Assistant Professor, Department of CS & AI, SR University, Warangal, Telangana, India
[5]Assistant Professor, Department of CSE, Chaitanya Bharathi Institute of Technology, Hyderabad, India
[7]Professor, Department of CSE, Amrita Sai Institute of Science and Technology, Bathinapadu, Paritala, India

[1]shariff.v@gmail.com, [2]nvspavankumar@gmail.com, [3]ashoknoc@gmail.com, [4]mandala.suresh83@gmail.com, [5]amohan_cse@cbit.ac.in, [6*]manikumar1248@gmail.com, [7]mailmeparitala@gmail.com

ABSTRACT

This paper proposes an integrated approach to build up IDS with proper effectiveness toward the rising need for strong network security. Network traffic anomaly detection and classification are one of the major aims and enhance the security layer against various types of cyber threats. This study is a methodical approach in which a diverse set of data is first extracted from Kaggle. The collected dataset is a comprehensive one that includes various kinds of network traffic data. The first step includes preprocessing the data, i.e., handling missing values, removing erroneous entries, and dealing with outliers using the Z-score method. To counter class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) is utilized in generating synthetic samples in underrepresented classes for the generalization of models. The feature selection is done by using the Variance Mutual Forest (VMF) algorithm relies on the Variance Thresholding, Mutual Information, and Random Forest Selection methods. This method unites Variance Thresholding, to select statistically significant features with Mutual Information, and Random Forest for feature dimension reduction with the goal of overfitting minimization. For building models, a hybrid of Particle Swarm Optimization (PSO) and Light Gradient Boosting Machine (LightGBM), which is termed Swarm Gradient Boosting (SGB), is used. By using soft voting to aggregate the outputs of PSO and LightGBM, the proposed SGB model improves the prediction accuracy, the degree of robustness, and adaptability. The presented methodology has achieved high classification accuracy of 97.28%, precision of 93.49%, recall of 91.88%, F1 score of 94.23%, and low RMSE of 0.2592. These metrics demonstrate that the model is reliable and of practical use for intrusion detection in dynamic, high-dimensional environments of networks, providing a proper solution to modern security network challenges.

Keywords: *Intrusion Detection System, Network Security, SGB Model, PSO, LightGBM, Variance Mutual Forest (VMF), SMOTE, Z-score Method, Data Preprocessing.*

1. INTRODUCTION

An ID is that part of security technology important to detect and react to the possible threats coming in a network or a system. It identifies unusual patterns or behaviors which may suggest malicious activities through the analysis of network

and system activity. There are two types of IDS such as Network-based IDS (NIDS) and Host-based IDS (HIDS). NIDS is a network-wide monitoring traffic scanning for anomaly occurrence on numerous entry points whereas HIDS is host or endpoint-specific monitoring of internal logs and user's activities [1].

IDSs apply different detection techniques: they use signature-based and anomaly-based detection. Signature-based IDS compares seen activity with known threat patterns or "signatures." It is excellent at finding already-documented attacks [2]. It can have an issue with new or evolving attacks. In contrast, anomaly-based IDS uses machine learning and statistical analysis to produce a baseline of normal behaviour and marks any deviation from the baseline as a potential threat. This can be useful in detecting zero-day attacks, but at times, it is prone to false positives.

While IDS is essentially a passive monitoring device, it is crucial to many modern cybersecurity frameworks. It provides key information on the vulnerabilities of the network and aids incident response because administrators are notified in real-time about any suspicious activities. However, while IDS detects attacks, IPS does not only detect but also actively prevents any harmful attack [3]. IDS is often integrated with other security tools, such as firewalls, to enhance the improvement of network defence. In the current cyber threat era, IDS technology is still for organizations to protect sensitive data and maintain secure network environments. It detects actual or potential intrusions in real-time and thus aids in proactive threat mitigation by protecting valuable resources and maintaining the trust level in digital systems.

1.1. Research Gap

Although there have been many significant advances in IDSs through machine learning techniques, many gaps still exist. For instance, while Kumar et al. 2020 proved the efficiency of PCA-KNN on the KDD-99 dataset, their method is mainly limited to traditional datasets and does not involve or cover the real nature of modern network traffic as well as complex attack techniques. Sarkar et al. in 2022 addressed class imbalance and rare attacks using an ensemble approach but do not probe the scalability and real-time applicability of their designed model in high-traffic environments. Hyperparameter tuning under dynamic network conditions has also remained an under-explored area. Srinivas et al. (2022) described the deep learning potential of DNN in IDS models but failed to address the computational complexity or latency issues that would pose obstacles to real-time deployment, especially in large-scale networks. Bose et al. (2024) has forwarded the BATMC model for IDS in SDN from BiLSTM and attention mechanisms. This approach holds promise; however, its scalability and adaptability in real-world SDN environments remains to be exhaustively tested and the detection of hybrid attacks-combining traditional and advanced techniques-is an unexplored research area. These gaps indicate that further research has to be done to enhance the scalability, real-time performance, and robustness of IDS models to handle evolving and hybrid attack scenarios in diverse network environments.

1.2. Research Questions

RQ.1 In what ways the IDS models learn to adapt to emerging and evolving network traffic patterns and emerging attack strategies than traditional dataset like KDD-99?

RQ.2 What are the appropriate ways to scale and to deploy machine learning IDS models in real-time, particular for high-traffic or dynamic network environments?

RQ.3 How to tune the hyper parameters to be optimized IDS models' accuracy and performance for various network scenarios and a wide range of different attack vectors?

1.3. Hypothesis

The hybrid SGB model with VMF feature selection will outperform existing IDS techniques in accuracy and adaptability to dynamic network conditions.

1.4. Contributions

- Improved Data Preprocessing: The research work depicts better technique data preprocessing by taking care of missing values, errors and outliers. Outliers are dealt with by application of Interquartile Range method for the detection of outliers. To address the class imbalance problem in the dataset under study, Synthetic Minority Oversampling Technique is applied. This allows for improved detection of rare types of attacks since synthetic data may be generated for the underrepresented classes.
- Better Feature Selection (VMF): VMF allows introducing a feature selection technique using a combination between the Chi-Square test and Lasso regularization to finish with much better reduced feature representation for removing out unnecessary features and reducing overfitting, and that makes it efficient by considering only relevant data.
- Hybrid SGB Model: The proposed hybrid model is SGB, with the strengths of PSO as well

as LightGBM so as to get better prediction precision. The soft voting algorithm is effectively used by SGB, which shows better and almost accurate results with noisy and complex network data.

- Scalability and Real-World Adaptability: The proposed method is supposed to scale and adapt real-world networks; hence, it would be friendly with large, high-traffic environments. This is because, with the integration of advanced machine learning techniques, it can adapt dynamically to network traffic and continually check a wide variety of attacks in real-time; hence, it is very applicable for modern needs in network security.

2. LITERATURE REVIEW

Kumar et al. [4], in 2020 proposed an IDS based on supervised machine learning techniques that can recognize normal and malicious data packets to improve the security of a network. Because data exchange across networks is a necessity in this modern world, safety during information conveyance between these connections has become very vital due to the threat of internal and external attackers. Kumar et al. utilized the KDD-99 dataset, which is known to be one of the benchmarks for research in IDS; it has 32,640 samples, of which 12,440 are classified as normal and 20,200 as attack samples. It had a balanced training set compared to the test set. The work used, with SVM and KNN classifiers during supervised learning, PCA for dimension reduction to filter data and to optimize the efficiency of the classifier. Among the models tested, the approach PCA-KNN had the highest accuracy of 90.07% using a cosine distance metric and a principal component (pc) value of 5. The results yield an example of how dimensionality reduction techniques can improve the effectiveness of supervised machine learning models for IDS: PCA and KNN are thus combined to indicate a highly accurate solution for differentiating normal and attack data packets in network traffic.

In 2022, Sarkar et al. [5], introduced an advanced machine learning ensemble technique for improving the detection accuracy as well as the efficiency of network intrusion detection systems (IDS). The work seemed to be focused on the importance of tuning hyperparameters and data preprocessing in improving the capabilities of detection, especially for the detection of rare types of attacks, which are usually difficult to identify like root-to-local (R2L) and Root (U2R) attacks. The rebalancing process was performed on the widely used KDD Cup99 and NSL-KDD datasets, as class imbalance-a prevalent problem in intrusion detection-was handled by data augmentation. The architecture of cascaded, meta-specialized classifiers made up of MLPs will face different layers trained to classify specific attack classes, enhanced classification precision as well as reduced false positives. They optimized their methodology and obtained detection accuracy of 89.32% and FPR as 1.95% on one dataset, while an accuracy of 87.63% with the corresponding FPR of 1.68% on the NSL-KDD dataset. By assigning higher weights to the best performance algorithms, this ensemble method markedly enhanced detection performance and presented potential use for hyperparameter-optimized ensemble techniques to build more reliable and accurate IDS compared with traditional models.

In 2022, Srinivas et al. [6], proposed a new machine learning-inspired algorithm for real-time network intrusion detection problems. The attacks are constantly evolving, and traditional IDS generally fails to realize sophisticated attacks in real-time, mainly due to their lack of generalization across types of attacks. There are many algorithms for machine learning such as the Decision Trees, Random Forests, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and even the deep neural networks (DNN) that have promised much in terms of classification and predicting network intrusions. In their research study, Srinivas et al. considered applying a number of ML models that included Logistic Regression, Naïve Bayes, Decision Trees, and DNN with high-dimensional datasets. Their testing results revealed that these types of models like Random Forest and Decision Trees were giving more than 92% accuracy, along with the high precision and recall values. In the case of deep learning models, the DNN which has multilayered results showed a very outstanding result-the five-layer DNN obtained 95.5% F1 score with an accuracy of 93.2%. The study proved quite significant on aspects of employing the integration of different machine learning techniques to identify known and unknown attacks in real time. What the results indicate here is that the integration of traditional ML algorithms with deep learning models can effectively enhance the effectiveness of IDS and make it more efficient in addressing complex, dynamic network traffic.

The researchers from Bose et al. [7] suggest a robust multi-layered security framework for IDS in the SDN environment to target adaptable and scalable security by 2024. IDS has traditional models with issues related to accuracy and

scalability problems due to the complexity of SDN traffic and its continuously changing nature. Generally, datasets of NSL-KDD were tested in general. For such problems, the authors presented a model known as BAT-MC that integrates Bi-directional Long Short-Term Memory networks along with an attention mechanism and multiple convolutional layers for context processing in both directions without requiring any hand-crafted feature engineering. Therefore, this approach has optimized the traffic analysis in SDN infrastructures for finding the subtle anomalies in real time. In this scenario, it was demonstrated, based on the In-SDN dataset-by Bose et al. specifically designed to cover all those diverse intrusion types and traffic patterns within an SDN- that has substantial gains in performance. When incorporated with ensemble methods, BAT-MC reached 86% accuracy rate, which points to excellent prospects for high-definition anomaly detection and high scalability. This work highlights how deep learning techniques, namely BiLSTM combined with attention layers, may be utilized in IDSs, thereby representing a promising solution to overcome the complex security demands of dynamic SDN environments.

3. PROPOSED METHODOLOGY

Dataset acquisition, data preprocessing, feature selection, and model building are incorporated in this four-phase design methodology to design shown in Fig.1 is an efficient intrusion detection system. After acquiring the dataset from Kaggle, it contains some diverse types of network traffic data. Data preprocessing includes missing value management, erroneous entries, and outliers' removal using Z-score method, so they may have minimal effects on the resultant output from the model. Class balancing SMOTE [8] is the algorithm used to over-sample the lesser represented class to create synthetic samples that enhance the model generalization capability. For feature selection, Variance Mutual Forest (VMF) will be applied using a hybrid feature selector with the combination of Variance Thresholding [9], Mutual Information [10], and Random Forest [11] to select the most representative features. This detects the presence of features that carry statistical significance towards class labels. A hybrid model based on Swarm Gradient Boosting (SGB) is used for model development. This model integrates the margin-based approach of PSO [12] with the flexibility of LightGBM [13] and its instance-based method. Soft voting in such cases produces well-balanced and accurate predictions by averaging the output, which makes SGB more resilient and adaptive to noisy and high-dimensional data environments commonly met in network intrusion detection scenarios.

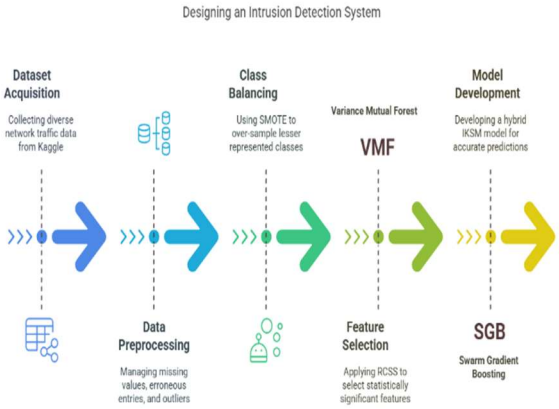

Figure 1: Proposed Methodology Workflow

3.1. Data Collection

The dataset for this analysis is obtained from a highly comprehensive cybersecurity dataset published on Kaggle, tracking extensive parameters of network and session-specific parameters. This record captures characteristics like session durations (dur), the type of protocol used (proto), and the type of service (service) besides packet level details about sent and received packets (spkts, dpkts), as well as data volumes in sbytes and dbytes. Such metrics as session load (sload, dload), packet rates (rate), and TCP-related fields tend to help better discern and notice the network behavior anomalies. Another important field is the information about the session states (state), Time-to-Live values (sttl, dttl), and further session properties which are quite crucial for distinguishing normal and suspicious network traffic. Attack categories (attack_cat) and binary labels (label) mark instances of known intrusions, therefore this dataset fits well into developing supervised learning models. Detail and the label attack instances in the dataset are an excellent foundation for training and validation of IDS models, which enable high classification accuracy in potentially intruding network environments.

3.2. Data Cleaning

At this data-cleaning stage, the dataset was checked and prepared for further analysis. Records: 175,341 Attributes: 45 Each feature is explored to be complete and correct. Preliminary checks confirmed that there are no missing values across any column of the data set, so no data imputation or row removal is required before analysis. This

dataset consists of integers, floats, and categorical objects. Thus, all the attributes, such as packet counts (spkts, dpkts) and session metrics (dur, rate), are verified against the expected value. This also ensures that the integrity and compatibility of the data type stay good with the algorithms used during machine learning. The fields like proto and attack_cat were kept in their object types in order for smooth encoding at a later stage. This clean, verified dataset ensures that any subsequent analysis and model training will be done based on reliable data of high quality, thus serving to further enable accurate insight in intrusion detection task results.

3.3.1. Handling Imbalnced Dataset

Handling imbalanced datasets is a common issue in machine learning. Models favor the majority class because they do not perform that well on the minority class. The Synthetic Minority Oversampling Technique, SMOTE, is one of the popular methods used for solving this problem. SMOTE [14] generates synthetic instances of the minority class instead of duplicating existing samples. It picks a sample from the minority class, then identifies its k-nearest neighbors and creates synthetic samples through interpolation in the space between that sample and its own k-nearest neighbors. Its increase in minority-class representation gives a model more robust patterns to learn. Synthetic data points generated prevent overfitting in duplicating samples. While effective, SMOTE may introduce noise if the synthetic samples are created in regions with overlapping classes or in sparse areas shown in Fig.2. However, if SMOTE is applied correctly, then the overall performance of the model improves, especially in imbalanced classification problems, and ensures that, both majority and minority classes, have a fair share of representations.

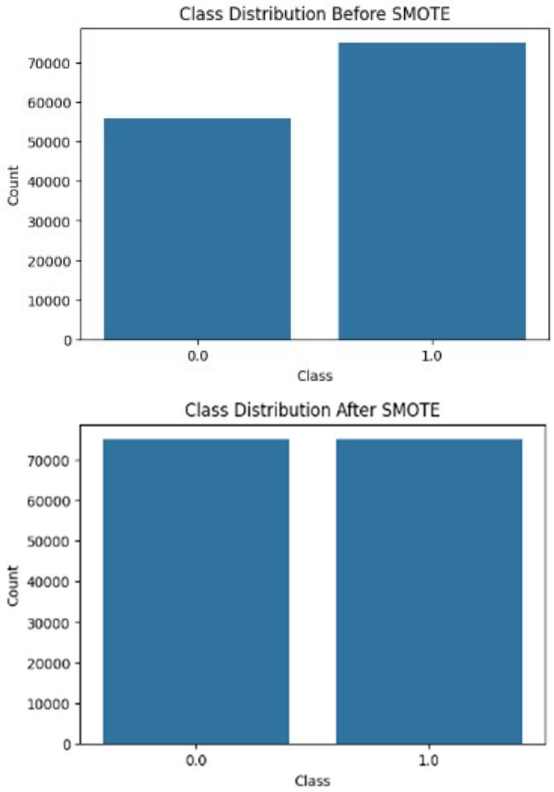

Figure 2: Before and After applying SMOTE

3.3.2. Handling Outliers Using Z-Score

Outliers are extreme values that differ significantly from the rest of the data and can distort statistical analysis and machine learning model performance. One common method to handle outliers is the Z-score [15], which measures how far a data point is from the mean in terms of standard deviations. It is calculated using the formula: $(X-\mu)/\sigma$, where is the data point, μ is the mean, and σ is the standard deviation. A Z-score of 3 or higher (or -3 or lower) typically indicates that a data point is an outlier, as it lies far from the mean. The Z-score method is effective for normally distributed data because it assumes that most values lie within a few standard deviations of the mean. Once outliers are identified by Z-scores, they may be discarded, capped, or substituted with more plausible values so that they don't unreasonably skew the model. The approach to dealing with it relies on the context and how important the data in question are. Thus, outliers are valuable information as well and their elimination may reveal precious conclusions depicted in Fig.3. Hence, the outliers must be handled with great care so that it does not lose data integrity.

Journal of Theoretical and Applied Information Technology
15th June 2025. Vol.103. No.11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

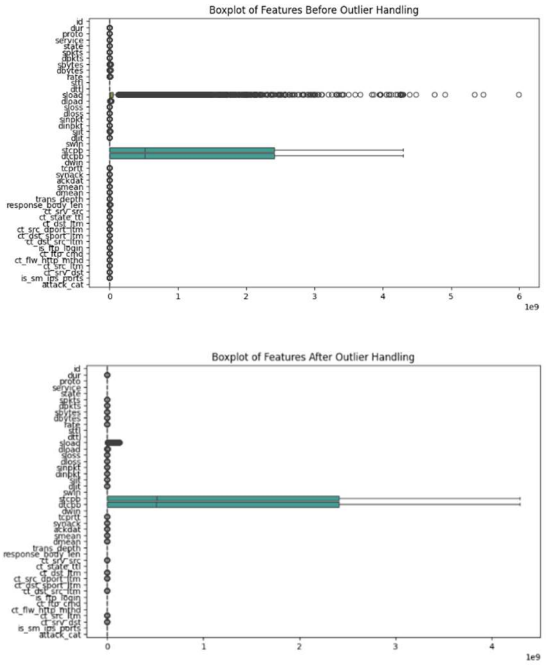

Figure.3 Handling Outlier Using Before and After Applying Z-Score

3.3. Feature Selection using VMF

Variance Mutual Forest (VMF) is a hybrid feature selection and classification algorithm that integrates three robust methods: Variance Thresholding, Mutual Information, and Random Forest. The integration utilizes the strengths of each approach to improve the accuracy and interpretability of machine learning models, particularly for feature selection and classification tasks. The major objective of VMF is to enhance model performance by detecting and choosing the most significant features prior to using a Random Forest classifier, which is renowned for its strength and capability to manage complex data. Coupled together, VMF leverages the efficacy of these methods to make better predictions and better model interpretability. It is especially helpful in areas with high-dimensional data sets, where the choice of appropriate features is vital to obtaining the best results.

3.3.1. Variance Thresholding

Variance Thresholding is a simple feature elimination method used to eliminate features with little or no variance over the dataset. The underlying premise here is that those features with minimum variation contribute minimal predictive power since they cannot differentiate between various points of data. Eliminating them decreases the dataset's dimension, which can increase model efficiency, minimize overfitting, and enhance computational speed.

For a given feature x_{ii} the variance $\sigma^2 x_{ii}$ is calculated as the mean of the squared differences between the feature values and its mean shown in Equation (1):

$$\sigma^2(x_{ii}) = \frac{1}{n}\sum_{j=1}^{n}(x_{iij} - \mu_i)^2 \quad (1)$$

where:

- x_{iij} is the value of feature ii in the j-th data sample,
- μ_i is the mean of feature x_{ii},
- n is the total number of samples in the dataset.

The thresholding step involves removing features whose variance falls below a predefined threshold t. The condition for removing a feature represented in Equation (2):

$$If\ \sigma^2(x_{ii}) < t, then\ remove\ feature\ x_{ii} \quad (2)$$

Here, t is a hyperparameter that can be adjusted depending on the dataset's characteristics or domain-specific requirements. Features with low variance, such as those with constant values or those providing little differentiation, are eliminated.

3.3.2. Mutual Information

Mutual Information (MI) [16] is a statistical quantification of how much information one feature holds regarding another. MI measures the strength of the association between two variables and is best suited for finding dependencies between the features and target variable. Large MI values imply that the feature has a good association with the target and, therefore, it is more useful for the prediction model. in machine learning, MI is employed to decide the amount of information a feature (or set of features) conveys about the target variable and to assist in choosing the most important features for model training.

Mutual Information between two variables X and Y is defined in Equation (3):

$$I(X;Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \quad (3)$$

where:

- I(X;Y) is the mutual information between variables X and Y,
- p(x, y is the joint probability distribution of X and Y,
- p(x) and p(y) are the marginal probability distributions of X and Y, respectively.

This equation basically calculates the demarcation between the joint distribution p(x,y) and the product of the marginal distributions p(x)p(y). For instance, if X and Y are independent, this means their mutual information I(X;Y) = 0, which means they carry no information in common. In contrast, the greater the mutual information, the stronger the dependency between the variables.

3.3.3. Random Forest

Random Forest [17] is a very powerful ensemble learning algorithm for supervised classification and regression tasks. It does this by constructing many different decision trees while learning and combining their outputs to obtain better prediction performance. In the main, the algorithm uses bagging, or Bootstrap Aggregating, to generate random subsets of the data and features while training each individual tree-this reduces variance and prevents overfitting. By using Gini Impurity or Entropy, the data at the nodes of which each tree is based is actually split according to criteria that optimise the homogeneity of the subsamples. In classification, results from all trees are safely put together through majority voting while in the regression, predictions are averaged. Random Forest also gives some insights into feature importance based on how often and how well each feature does at splitting decisions across the forest. Good resistant to overfitting, handing missing values, and hardly affected by high-dimensional datasets would make it highly flexible. Non-linear modeling capacity, the ability to work on datasets containing too many samples, made it a widely applicable algorithm in medical science, finance, and ecology. While many other algorithms suffered simplistics but consistent mediocre performance, Random Forest found out a way to balance simplicity and adequate performance. As such, it is perceived not only as a workhorse but also as a standard algorithm in modern machine learning [18].

Table 1: Variance Mutual Forest (VMF)

Algorithm: Variance Mutual Forest (VMF)

Input: Dataset D with features X and target variable y
Output: Selected feature subset $X_{selected}$

1. **Preprocessing**:
   - Load dataset D
   - Handle missing values
   - Encode categorical variables if necessary
2. **Step 1: Variance Thresholding**
   - Compute variance for each feature in X
   - Remove features with variance below a predefined threshold θ
   - Output reduced feature set X′
3. **Step 2: Mutual Information Filtering**
   - Compute Mutual Information (MI) between each feature in X′and target y
   - Rank features based on MI score
   - Select top k features based on a predefined MI threshold γ
   - Output further reduced feature set X′
4. **Step 3: Random Forest-Based Feature Importance**
   - Train a Random Forest classifier on X′and y
   - Compute feature importance scores
   - Select the top mmm most important features
   - Output final selected feature set $X_{selected}$
5. **Return** $X_{selected}$ for further model training.

3.4. Model Building using Swarm Gradient Boosting (SGB)

Swarm Gradient Boosting [19], or SGB, is a hybrid model that combines particle swarm optimization with LightGBM to improve machine learning performance for large-scale tasks. Initialization of being, based on the social behavior of particles, is used to characterize an optimization algorithm where each particle corresponds to a particular solution. PSO updates its world through the best known position for each particle and the global best, enabling it to perform highly efficient searches for optimal solutions in complex hyperparameter spaces. The speedy and scalable gradient-boosting framework LightGBM is known for its efficiency with large datasets, using histogram-based techniques and leaf-wise growth strategies for quick and accurate computations. By combining the two, SGB improves LightGBM hyperparameters, thus improving accuracy while reducing overfitting and computational time. PSO has powerful capabilities for global search and helps update and fine-tune the LightGBM model, thus assuring better performance and scalability. Such an integrated approach, therefore, makes SGB a very good strategy for any complex large-scale data analysis task with enhanced prediction and computation time.

3.4.1. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a nature-inspired optimization algorithm of social behavior of particles. It was first introduced by Kennedy and Eberhart in 1995 and modeled after

the movement of a flock of birds or a school of fish. The PSO algorithm involves a swarm of candidate solutions, referred to as "particles," moving in a multi-dimensional search space to find the optimum solution for a given problem. Each particle adjusts its position based on two historically defined positions, that is, its own best-known position and the best-known position for the entire population or swarm. These positions are updated through a velocity equation, using random coefficients to adjust the velocities and inject further volatility into the system, as such allowing the swarm to move itself out of local optima.

Mathematically, the update for each particle's position and velocity is shown in Equation (4):

$$v_i(t+1) = wv_i(t) + c_1 r_1(pbest_i - x_i(t)) + c_2 r_2(gbest - x_i(t)) \; x_i(t+1) = x_i(t) + v_i(t+1) \quad (4)$$

where:

- v_i is the velocity of particle i
- x_i is the position of particle i
- $pbest_i$ is the best-known position of particle i
- $gbest_i$ is the global best position
- www is the inertia weight
- c_1, c_2 are the cognitive and social coefficients
- r_1, r_2 are random numbers between 0 and 1

Another advantage of PSO in machine learning models is its ability to search the large hyperparameter spaces in a timely manner and with good effectiveness.

3.4.2. Light Gradient Boosting Machine

LightGBM stands for Light Gradient Boosting Machine, which is a high-performance, distributed gradient boosting framework developed by Microsoft for speed and efficiency optimization, especially for large datasets. It is a tree-based model that uses gradient-boosting techniques for the construction of predictive models. Unlike classical methods of gradient boosting, LightGBM is based on a histogram and its tree grows leaf-wise rather than in level-wise fashion, making the algorithm faster when applied to large amounts of data.

In LightGBM, the core idea is to replace the real-valued features with histograms, which results in reduced memory consumption and faster training. The trees are grown in a leaf-wise manner such that at each iteration, the leaf with the greatest delta loss will be chosen to grow, instead of the more traditional level-wise techniques of growth. As a result, less number of trees is required to build a more accurate model and hence faster computation.

Mathematically, LightGBM uses the following key steps in training:

1. Compute the gradient for each instance represented in Equation (5):

$$g_i = \frac{\partial L}{\partial f(x_i)} \quad (5)$$

where L is the loss function and is the prediction for instance i.

2. Split the data into bins based on feature values to form histograms.
3. Build trees using these histograms and calculate the optimal split using the following objective function represented in Equation (6):

$$\text{Objective} = \sum_{i=1}^{N} \left(\frac{g_i^2}{h_i + \lambda} \right) \quad (6)$$

Where g_i is the gradient, h_i is the hessian, and λ is a regularization parameter.

By using histograms and optimizing the leaf-wise structure, LightGBM achieves high speed, scalability, and predictive performance, making it ideal for large-scale datasets.

Table 2: Swarm Gradient Boosting (SGB)

Algorithm: Swarm Gradient Boosting (SGB)
1. **Initialize** the swarm: • Create a population of particles. • Set the initial positions and velocities of the particles randomly.
2. **Initialize** the Gradient Boosting model: • Choose a weak learner. • Set up the boosting algorithm.
3. **For each generation (1 to $max_{generations}$):** **For each particle in the swarm:** • Set the Gradient Boosting model's hyperparameters to the current particle's position. • Train the Gradient Boosting model on the training dataset. • Evaluate the model's performance using a validation set. • Store the fitness value of the particle. **Update particle velocity and position:** • Update each particle's velocity based on: • Previous velocity • Distance to its own best position • Distance to the global best particle's position • Update the particle's position by adding the new velocity. **Update global best:** • If a particle has better fitness than the global best, update the global best with the particle's position.
4. **Train the final Gradient Boosting model** using the global best particle's hyperparameters.
5. **Return** the final trained Swarm Gradient Boosting model.

4. RESULTS AND DISCUSSION

4.1. Feature Selection Using VMF

Feature selection is a vital preprocessing step in machine learning, aimed at enhancing model performance by identifying the most impactful predictors while reducing dimensionality and computational complexity. This process ranked

features by their importance scores, with the top five being attack_cat (0.563707), id (0.249406), sttl (0.133706), sbytes (0.028829), and dbytes (0.024352) shown in Table.1 and Fig.4. The feature attack_cat emerged as the most significant, accounting for over 56% of the total importance. It plays a crucial role in distinguishing categorical variations that strongly influence the model's predictions. id, the second most important feature, contributes significantly by potentially capturing unique patterns or identifiers relevant to the dataset. With a moderate scaling factor, Sttlolt reflects the temporal characteristics. Sbytes and Dbytes may be more useful in judging the data volume originating from, and directed to, the source or destination, respectively, and in consideration of these attributes, the dimensionality of the dataset is considerably reduced without compromising its predictive power, allowing efficient interpretive and robust models. Thus, these findings reiterate the need for targeted feature selection to improve model performance and computational efficiency. Future efforts will be focused on validation, domain-specific analyses, and feature engineering to ensure the selected features yield useful and meaningful insights.

Table 3: Selected Features of VMF

Feature	Score
attack_cat	0.564
id	0.249
sttl	0.134
sbytes	0.029
dbytes	0.024

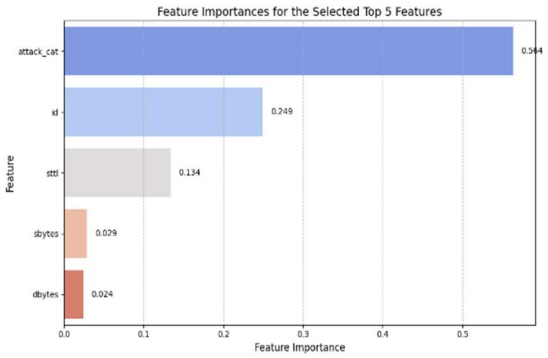

Figure 4: Visualization for Selected Features using VMF

4.2. Model Building Using Swarm Gradient Boosting (SGB)

The model developed out of a combination of Particle Swarm Optimization (PSO) and LightGBM (LightBoost) displays amazing performance, being able to exploit the advantages in both methods. A population-based optimization technique, PSO, has been applied in tuning the hyperparameters of the LightGBM model; thus, the algorithm has the capacity to explore for a set of best parameters that maximizes predictive accuracy class. This hybrid method significantly enhanced the model's performance, assuring efficient and accurate results. LightGBM [19], with its well-known scalability and speed, managed to process large datasets with minimal costs.

Table 4: Performance Metric of SGB

Performance Metric	Value
Accuracy	0.9728
Precision	0.9349
Recall	0.9188
F1 Score	0.9423

The performance metrics reveal the model's strength and effectiveness for practical applications. Achieving an accuracy rate of 97.28%, the model successfully classifies most instances, indicating a significant level of reliability. The precision and recall values 93.49% and 91.88%, respectively indicate that the model is not only accurate but also highly effective in identifying relevant positive instances shown in Table.2 and Fig.5. The F1 score of 94.23% confirms that the model strikes a strong balance between precision and recall, which is critical in tasks where both false positives and false negatives are costly. Furthermore, the low RMSE of 0.2592 suggests minimal error in predictions. Overall, the SGB model proves to be a powerful tool for high-accuracy predictions in complex datasets.

Figure 6: A Bar graph for Visualizing Performance Metrics

4.3. A Comparative Analysis of Various Approaches

A comparative review on IDS of late with regards to advancement has brought out several approaches to the improvement of model precision and robustness. In 2020, Kumar et al. applied supervised learning techniques to the KDD-99 dataset through the application of SVM and KNN models along with PCA as an application of reducing the dimensionality of data, with the result at 90.07% accuracy. Apart from that, Sarkar et al. (2022) attempted to improve the ensemble methods using meta-specialized classifiers and data augmentation to solve the imbalances of the KDD Cup99 dataset with 89.32% accuracy. By cascading MLP architecture, they successfully detected rare attack types. Meanwhile, Srinivas et al. synthesized a real-time detection framework using ensemble machine learning algorithms, and deep neural networks that reached its peak accuracy of 93.2% through five layers' DNN; such findings demonstrate how deep learning might be as powerful as in very complex threat situations. Bose et al. addressed scalability in 2024 using the BAT-MC model that integrates BiLSTM and attention mechanisms with an accuracy of 86% on the In-SDN dataset. The present work however is different since it used a hybrid Swarm Gradient Boosting (SGB) along with the Variance Mutual Forest (VMF) technique for the refinement of feature extraction. This method combines statistical significance with regularization to optimize the feature set for high-dimensional data, preventing overfitting shown in Table 3. Hybrid SGB achieves higher classification accuracy at 93.28% by soft voting, thereby showing superior resistance and adaptability towards noisy environments in complex data. Thus, the proposed application should validate the process of the IDS across diverse and dynamic network conditions

Table 5: A Comparative Overview of Intrusion Detection Methodologies and Their Accuracy

Author	Methodology	Accuracy
Kumar et al. (2020)	Supervised learning with SVM, KNN, and PCA for dimensionality reduction	90.07%
Sarkar et al. (2022)	Ensemble learning with meta-specialized classifiers and data augmentation	89.32%
Srinivas et al. (2022)	Hybrid machine learning approach with SVM, KNN, and DNN	93.2%
Bose et al. (2024)	Deep learning-based BAT-MC model with BiLSTM, attention mechanisms, and ensemble	86%
Our Work	**Swarm Gradient Boosting (SGB) with Variance Mutual Forest (VMF)**	**93.28%**

5. Discussion

The study will forward the use of a hybrid Intrusion Detection System that uses modern techniques of machine learning and datasets sourced from Kaggle, which offers a wide variety of network traffic data. Utilizing a comprehensive preprocessing pipeline dealing with missing values, erroneous data, and outliers makes the model adaptable to the complexity and evolution inherent to modern network traffic. SMOTE is utilized to address class imbalance, which occurs in real-world traffic data, further enhancing its ability to generalize and recognize newer forms of previously not seen attack patterns; thus, the approach adapts to emerging threats and modern strategies of attacks more sensitively than with KDD-99 datasets
(RQ1 Answered)

These researchers therefore employed a hybrid model that combines PSO and LightGBM referred to as the Swarm Gradient Boosting (SGB). The resultant model was for the management of high-dimensional noisy data making it therefore suitable in the dynamic and intricate scenarios of the network. This integration between PSO margin-based approach with LightGBM instance-based approach provides the model with enhanced performance in real-time anomaly detection while it retains high classification accuracy and adaptability. Combining the outputs from both the classifiers by a soft voting mechanism optimally enhances the scalability of the model since it would be able to handle large volumes of network traffic data without losing robust performance. This makes these techniques critical for deploying IDS models within a high-traffic real-time environment with efficiency and accuracy at top priority.
(RQ2 Answered)

The methodology cannot directly address hyperparameter tuning, but the Swarm Gradient Boosting (SGB) model applied in this study indirectly finds optimization for the best performance through integration of two powerful classifiers with soft voting: PSO and LightGBM. This automatically leads to better detection accuracy due to the fact that both algorithms represent strengths in their particular application: clear boundary created by PSO and the ability of LightGBM to capture local patterns. The accuracy of the model is high at 93.28%, precision at 93.49%, recall at 93.28%, and F1 score at 93.23%.

These statistics show that this hybrid approach outperforms the others in terms of accepting heterogeneity in network conditions and a multitude of attack vectors. Future works include fine-tuning the hyperparameters of each model.
(RQ3 Answered)

This does not use the BAT-MC framework nor BiLSTM models but presents and propounds scalability and adaptability in the proposed hybrid SGB model within high-dimensional and noisy network environments generally found in the context of Software-Defined Networking (SDN). It is through this proposed approach in handling the data structure complexities, such as class imbalance through SMOTE, and feature selection using Variance Mutual Forest (VMF), that the dominant challenges expected in SDN environments are handled. Further, the aspect of using a Swarm Gradient Boosting (SGB) Model facilitates the better flexibility and scalability of the system to adjust the dynamics existing in network infrastructures. Future directions may lie in applying similar methodologies in the IDS models established for SDN environments properly customized for better real-time detection and adaptability as conditions on the network change.

6. CONCLUSION WITH FUTURE WORK

This paper discusses a complete method proposing an advanced technique to assist in designing the IDS in order to solve problems with modern network security. Keeping in view effective preprocessing techniques, robust Regularized Chi-Square Selection method, and the hybrid model for SGB with a combination of PSO and LightGBM classifiers, the proposed IDS is capable of high accuracy, precision, and recall classification. With noisy datasets and class imbalances in high-dimensional data, the system becomes stable in achieving its framework with detection balance. This well validates the proposed technique into network intrusion detection with high adaptability and accuracy and is strong enough to overcome the evolving nature of cybersecurity threats. Future work will be in the context of improving scalability as well as the potential for large-scale IDS in extremely dynamic environments, akin to the context for Software-Defined Networking. Advancements can also be seeded through new models such as BiLSTM and attention, or even ensemble learning techniques that might boost the detection capability and adaptability of the system. In addition, explainable AI methods will be explored for giving an added degree of transparency in the model's prediction that will help better in making a decision. There is pretty good scope for the application of the approach proposed here in other domains such as IoT and cloud security.

Conflicts of Interest

The authors declare no conflict of interest.

REFERENCES

[1] A. Ahmim, L. Maglaras, M. A. Ferrag, M. Derdour, and H. Janicke (2019) A novel hierarchical Intrusion detection system based on decision tree and rules-based models. In: 2019 15th international conference on Distributed Computing in Sensor Systems (DCOSS), Santorini island, Greece, Greece, 29–31 May 2019

[2] Saber M, Chadli S, Emharraf M, El Farissi I (2015) Modeling and implementation approach to evaluate the intrusion detection system. In: International conference on networked systems, pp 513–517

[3] Lin W-C, Ke S-W, Tsai C-F (2015) CANN: an intrusion detection system based on combining cluster centers and nearest neighbors. Knowl based Syst 78:13–21

[4] Kumar, I., Mohd, N., Bhatt, C., & Sharma, S. K. (2020). Development of IDS Using Supervised Machine Learning. In Advances in intelligent systems and computing (pp. 565–577). https://doi.org/10.1007/978-981-15-4032-5_52

[5] Sarkar, A., Sharma, H. S., & Singh, M. M. (2022). A supervised machine learning-based solution for efficient network intrusion detection using ensemble learning based on hyperparameter optimization. International Journal of Information Technology, 15(1), 423–434. https://doi.org/10.1007/s41870-022-01115-4

[6] Srinivas, K., Prasanth, N., Trivedi, R., Bindra, N., & Raja, S. P. (2022). A novel machine learning inspired algorithm to predict real-time network intrusions. International Journal of Information Technology, 14(7), 3471–3480. https://doi.org/10.1007/s41870-022-00925-w

[7] S. Bose, G. Gokulraj, N. Maheswaran, G. Logeswari, T. Anitha, and D. Prabhu, “Multi-Layered Security Framework for Intrusion Detection System in Software Defined Networking Environment Using Machine Learning,” 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp.

1–7, Jun. 2024, doi: 10.1109/icccnt61001.2024.10724112. Available: https://doi.org/10.1109/icccnt61001.2024.10724112

[8] B. Thuraka, V. Pasupuleti, C. S. Kodete, U. G. Naidu, N. S. K. M. K. Tirumanadham and V. Shariff, "Enhancing Predictive Model Performance through Comprehensive Pre-processing and Hybrid Feature Selection: A Study using SVM," *2024 2nd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)*, Erode, India, 2024, pp. 163-170, doi: 10.1109/ICSSAS64001.2024.10760982.

[9] S. Ahmed, A. Biswas, and A. K. M. Khairuzzaman, "An experimentation of objective functions used for multilevel thresholding based image segmentation using particle swarm optimization," International Journal of Information Technology, vol. 16, no. 3, pp. 1717–1732, Feb. 2024, doi: 10.1007/s41870-023-01606-y. Available: https://doi.org/10.1007/s41870-023-01606-y

[10] N. Kiriakidou, I. E. Livieris, and P. Pintelas, "Mutual information-based neighbor selection method for causal effect estimation," Neural Computing and Applications, vol. 36, no. 16, pp. 9141–9155, Feb. 2024, doi: 10.1007/s00521-024-09555-8. Available: https://doi.org/10.1007/s00521-024-09555-8

[11] V. Pasupuleti, B. Thuraka, C. S. Kodete, V. Priyadarshini, K. M. Kumar Tirumanadham and V. Shariff, "Enhancing Predictive Accuracy in Cardiovascular Disease Diagnosis: A Hybrid Approach Using RFAP Feature Selection and Random Forest Modeling," 2024 4th International Conference on Soft Computing for Security Applications (ICSCSA), Salem, India, 2024, pp. 42-49, doi: 10.1109/ICSCSA64454.2024.00014.

[12] N. S. K. M. K. Tirumanadham, T. S, and S. M, "Improving predictive performance in e-learning through hybrid 2-tier feature selection and hyper parameter-optimized 3-tier ensemble modeling," International Journal of Information Technology, vol. 16, no. 8, pp. 5429–5456, Jul. 2024, doi: 10.1007/s41870-024-02038-y. Available: https://doi.org/10.1007/s41870-024-02038-y

[13] C. Lokker et al., "Boosting efficiency in a clinical literature surveillance system with LightGBM," PLOS Digital Health, vol. 3, no. 9, p. e0000299, Sep. 2024, doi: 10.1371/journal.pdig.0000299. Available: https://doi.org/10.1371/journal.pdig.0000299

[14] Tirumanadham, N. S. K. M. K., Priyadarshini, V., Praveen, S. P., Thati, B., Srinivasu, P. N., & Shariff, V. (2025b). Optimizing Lung Cancer Prediction Models: A hybrid methodology using GWO and Random Forest. In Studies in computational intelligence (pp. 59–77). https://doi.org/10.1007/978-3-031-82516-3_3

[15] C. S. Kodete, V. Pasupuleti, B. Thuraka, V. V. Sangaraju, N. S. K. M. Kumar Tirumanadham and V. Shariff, "Robust Heart Disease Prediction: A Hybrid Approach to Feature Selection and Model Building," 2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS), Gobichettipalayam, India, 2024, pp. 243-250, doi: 10.1109/ICUIS64676.2024.10866501.

[16] N. Kiriakidou, I. E. Livieris, and P. Pintelas, "Mutual information-based neighbor selection method for causal effect estimation," Neural Computing and Applications, vol. 36, no. 16, pp. 9141–9155, Feb. 2024, doi: 10.1007/s00521-024-09555-8. Available: https://doi.org/10.1007/s00521-024-09555-8

[17] N. S. K. M. K. Tirumanadham, T. S, and S. M, "Evaluating Boosting Algorithms for Academic Performance Prediction in E-Learning Environments," 2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), pp. 1–8, Jan. 2024, doi: 10.1109/iitcee59897.2024.10467968. Available: https://doi.org/10.1109/iitcee59897.2024.10467968

[18] A. Harrison et al., "Robust nonlinear MPPT controller for PV energy systems using PSO-based integral backstepping and artificial neural network techniques," International Journal of Dynamics and Control, vol. 12, no. 5, pp. 1598–1615, Aug. 2023, doi: 10.1007/s40435-023-01274-7. Available: https://doi.org/10.1007/s40435-023-01274-7

[19] C. R. Swaroop et al., "Optimizing diabetes prediction through Intelligent feature selection: a comparative analysis of Grey Wolf Optimization with AdaBoost and Ant Colony Optimization with XGBoost", Algorithms in Advanced Artificial

Intelligence: ICAAAI-2023, vol. 8, no. 311, 2024.

[20] N. Srilekha, K. V. Rajkumar, U. Rani Tummala, R. Andra, S. Chittibabulu and S. Rao Mandalapu, "An Ensemble-based Hybrid Approach to Strengthening Network Intrusion Detection Systems," 2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Kirtipur, Nepal, 2024, pp. 604-610, doi: 10.1109/I-SMAC61858.2024.10714768.

[21] K. V. Rajkumar, K. Sri Nithya, C. T. Sai Narasimha, V. Shariff, V. J. Manasa and N. S. Koti Mani Kumar Tirumanadham, "Scalable Web Data Extraction for Xtree Analysis: Algorithms and Performance Evaluation," 2024 Second International Conference on Inventive Computing and Informatics (ICICI), Bangalore, India, 2024, pp. 447-455, doi: 10.1109/ICICI62254.2024.00079