Optimizing Feature Selection in Intrusion Detection Systems Using a Genetic Algorithm with Stochastic Universal Sampling

RadhaRani Akula[1], GS Naveen Kumar[2]
Research Scholar, Malla Reddy University, Hyderabad, India[1]
Associate Professor, Department of CSE (Data Science), Malla Reddy University, Hyderabad, India[2]

***Abstract*—The current study presents a hybrid framework integrating the Genetic optimization algorithm with Stochastic Universal Sampling (GA-SUS) for feature selection and Deep Q-Networks (DQN) for fine-tuning an ensemble of classifiers to enhance network intrusion detection. The proposed method enhances genetic algorithms with stochastic universal sampling (GA-SUS) combined with recursive feature elimination (RFE). An ensemble of machine learning methods which includes gradient boosting and XG boost as base learners and subsequently logistic regression as meta learner is developed. A deep Q-networks (DQN) is used to optimize the base algorithms XG boost and gradient boost. The suggested method attains an accuracy of 97.60% on the popular NSL-KDD dataset and proficiently detects several attack types, such as probe attacks and Denial of Service (DoS), while tackling the issue of class imbalance. The multi-objective optimization approach is evident in anomaly detection and enhances model generalization by diminishing susceptibility to fluctuations in training data. Nonetheless, the model's efficacy regarding infrequent attack types, such as User to Root (U2R), remains inadequate due to their sparse representation in the dataset.**

***Keywords*—GA-SUS; anomaly detection; IDS; RFE; DQN**

I. INTRODUCTION

An Intrusion Detection System (IDS) is a cybersecurity tool with the primary goal of monitoring and analysing network traffic or system activity for possible malicious behaviour and unauthorized access. IDS can identify attempts that can lead to potential intrusions, that is, whether it be network attacks, unauthorized access to systems, or any other abnormal statistics to detect that we are dealing with malware or other cyber threats [1]. IDS can identify attempts that can lead to potential intrusions, that is, whether it be network attacks, unauthorized access to systems, or any other abnormal statistics to detect that we are dealing with malware or other cyber threats [2]. Anomaly-based detection and signature-based detection are two methodologies employed by Intrusion Detection Systems to identify suspicious activities.

Integrating IDS with machine learning has remarkably improved the potency of IDS to locate cyber threads accurately [3]. However, these are insufficient to address the complex dynamic threats posed by cyber threats [4]. Machine learning mitigates these constraints by allowing Intrusion Detection Systems to learn from data, adapt to emerging threats, and enhance detection precision over time. Machine learning improves intrusion detection systems by discerning the most pertinent features for spotting intrusions. Emerging issues are seen in the increased incidence of assaults and the advancements in technologies noticed in contemporary IDS systems. Additional recommendations may be required for machine learning approaches while processing extensive data and transitioning throughout networking environments [5]. Consequently, there is a growing apprehension regarding the development of a way to extract superior high-order features when the objective is situated amongst a sea of nonstationary traffic. This necessitates the improvement of the generality and efficiency of the IDS to bolster the network's defences against novel and unidentified attacks [6].

Feature selection (dimensionality reduction) is an essential step in machine learning which entails choosing the most pertinent and informative characteristics from a dataset to enhance model performance. Feature selection minimizes model complexity, boosts generalizability, and frequently improves both accuracy and interpretability of predictions by retaining only the important features [7]. Feature selection Improves predictive accuracy by concentrating on the most pertinent features. Feature selection diminishes the likelihood of overfitting by removing noise and redundant information. It also simplifies the model, resulting in faster training and inference. In addition to that, it will minimize the storage ad memory requirements while minimizes the computational complexity. Fig. 1 showcases the importance of feature selection. There exist three kinds of feature selection strategies namely, wrapper models [9], filter methods [8] and embedded methods [10]. Filter-based approaches evaluate feature significance according to the statistical characteristics of the data. They are not related to any specific machine learning algorithm. Parallelly, wrapper methods use a specific learning algorithm to evaluate the performance of feature subsets [11]. Embedded approaches conduct feature selection during the model training phase. In this study we approach the feature selection mechanism with the aid of a wrapper method. Genetic Algorithms (GA) [12] are an optimization method derived on the concepts of genetics and natural selection. The genetic algorithms can effectively navigate extensive feature spaces and discern optimal or near-optimal feature subsets, rendering them particularly appropriate for high-dimensional datasets. The GA optimization is used for selecting most relevant features in this work.

To this end, it helps the model reduce overfitting, and thereby the model performs well when tested on unseen data. However, feature selection proved to be helpful in eliminating noisy components, resulting in an improvement in the quality of the provided dataset. In other words, when feature selection is performed properly, one is left with models that are accurate, efficient, and understandable - all qualities that are critical in the quest for insights and trustworthy predictions.

Feature selection is the core of any IDS in which the discovery of discrete features that characterize communications taking place in a network and the capability to discern between anomalous and normal is achieved. Feature selection is essential in the creation of efficient IDS by pinpointing the most pertinent aspects from network traffic data. Because of the complexity and high dimensionality of standard IDS datasets as NSL-KDD, UNSW-NB15 and CICIDS, feature selection enhances analysis by increasing detection accuracy and processing efficiency. By concentrating on the most informative attributes, the IDS can efficiently discern between regular and malicious actions. Minimizing the number of features decreases the computational load, resulting in expedited model training and real-time detection.

The ultimate aim of the current research is to propose and enhance a new approach to enhancing an NDIS by a more refined feature selection and optimization process. The primary contributions of the study are listed below.

- Enhancement of genetic algorithms with stochastic universal sampling (GA-SUS) combined with recursive feature elimination (RFE).
- An ensemble of machine learning methods which includes gradient boosting and XG boost as base learners and logistic regression as meta learner is developed.
- A deep Q-networks (DQN) is used to optimize the base learners XG boost and gradient boost.

The remainder of this paper is structured as below: Section II gives the literature review; Section III proposes the methodology; Section IV gives results and discussion and finally Section V concludes the study.

II. Literature Review

A research by Bakir et al. in study [13] explored innovative ways to enhance IDS using ML, specifically focusing on IoT networks. Using a genetic algorithm for tuning of hyperparameter along with a new hybrid feature selection, the authors propose a substantial increase of IDS effectiveness with the means of security threat identification. The authors combined several approaches looking for the most representative feature subset for detection through a hybrid feature selection methodology. Among others, Mutual Information-based Feature Selection (MIFS) is one among the several ways in which feature selection is performed by MIFS by selecting features from the original set according to their mutual information with the target value while reducing redundancy. Five (Decision Tree, XGBoost, Bagging, Extra Tree, Random Forest) ML algorithms were trained with their existing hyperparameters. The XGBoost classifier elevated the performance, reaching 99.98% F1 score and 99.98% detection accuracy. The Extra Tree algorithm had a good performance as well, detecting with an accuracy of 99.96%.

A study by Cheng et al. in study [14] developed a pioneering approach known as Detection-Rate-Emphasized Multi-objective Evolutionary Feature Selection (DR-MOFS). The selected features are important for reducing the complex data sets for better efficiency and accuracy of IDS, according to the study. The goal is to decrease the features considered, thereby simplifying the framework and increasing performance. The second main aim of the study highlights optimizing the detection rate, which must be achieved as it minimizes the number of missed attacks. Also it overcomes the limitations of the previous Feature selection approaches based on feature subset size and classification accuracy which often led to low detection rate. Experiments were conducted on well-known network intrusion detection datasets, including UNSW-NB15 and NSL-KDD, in order to validate the suggested method. The results show that DR-MOFS is better than previous methods in most of the measures of less features selected, more accuracy, and more detection rate.

A research work by Ren et al. in study [15] generated a model MAFSIDS that aims to reduce the complexity of the feature selection process by eliminating close to 80% of repeating features in comparison to the base feature set. The MAFSIDS adopts a multi-agent framework in which a large number of feature agents compete with each other. The model provides adaptability to the evolving nature of network attacks (i.e. network IDS becomes more effective against new attacks). MAFSISD improves the typical feature selection search strategy by formulating the feature selection problem as the target of MAFSIDS implemented in a multi-agent reinforcement learning framework in which the number of features selections in a general case is an exponential 2^N which it can specify those features which make up unit subsets. Here, you will find our model implementation which consists of Deep Q-Learning (a form of deep reinforcement learning). This approach allows the model to learn optimal policies for attacking the environment, through the interactions and feedback given based on the actions taken. GCNs are used to obtain deep features by MAFSIDS. As a result, this approach can significantly improve the feature selection process by allowing the model to better capture complex relationship in the data. While MAFSIDS model did very well with 96.8% accuracy rate on the dataset.

Another work by Ren et al. For example, [16] uses RFE and DT classifiers to remove 80% of all features and finds the most useful subset of features to identify all network attacks, especially unknown attacks. This article is referring on RFE which is used to assign importance the attributes in the ordinal manner of their significance related to target variable (i.e. intrusion detection in the network). Typically, the algorithm removes the least relevant features iteratively from the data. The model is refitted to the features after each iteration. The data is re-coded by way of Mini-Batch processing making the data-set relevant to the DRL model which is helpful in deriving more profound associations between features so it enhances accuracy and efficiency. Using the CSE-CIC-IDS2018 dataset for testing, the model achieved an F1-score of 94.9% and

accuracy of 96.2%. This shows that it is pretty effective at detecting network intrusions.

A study by Thajeel et al. in study [17] proposed DQN-MAFS implements a dynamic feature selection framework that continuously assesses the relevance of features in real-time and updates them accordingly. It is very important for capturing the changes in the data and eliminating irrelevant features for detection. Each feature is treated as an individual agent within the Multi-Agent System framework. Each agent acts to include/exclude a feature with some determination. Its architecture is based on reinforcement learning, which uses a deep Q-learning approach to facilitate online updates. As new labeled data becomes available, agents are rewarded to understand how much to rely on their own features and update their selection accordingly. FARD-DFS is a reward allocation sub-model within the DQN-MAFS framework.

A research work by Kavitha et al. in study [18] introduces a Deep Learning Model and Filter-based Ensemble Feature Selection for Intrusion Detection in Cloud Computing Environment. This research utilizes two publicly available datasets, NSL-KDD and KDDCup-99, for gathering the intrusion data. In the FEFS, three kinds of feature extraction process are involved, which are filter, wrapper and embedded algorithms, and it is obtained from this process that those features are extracted which will help the DLM in the training process. DLM combines RNN with a process known as Tuning Dynamic Optimization (TDO) for its optimization of weighting parameters. The proposed technique acquired a sensitivity of 0.90% and a recall of 0.93%. In relativity, the conventional methods achieved lower recall rates of 0.83% (DNN), 0.88% (RNN), 0.91% (RNN-GA) for recall, and 0.81% (DNN), 0.85% (RNN) for sensitivity.

A study by Mananayaka and Chung in study [19] proposed an innovative methodology for Network Intrusion Detection Systems (NIDS) that integrates Two-Phased Hybrid Ensemble Machine Learning with Automated Feature Selection, employing various ML classifiers to proficiently identify and shortlist the most pertinent attributes for identifying both familiar and unfamiliar attacks, thereby tackling the challenges associated with high-dimensional network data. The framework utilizes an automated feature selection engine that discerns the most pertinent elements from high-dimensional network data. Utilizing four distinct machine learning classifiers, the system may concentrate on the most pertinent information for attack detection, hence improving the accuracy and efficiency of the detection process. The suggested framework exhibited a high detection rate (0.9431) and an exceedingly low false alarm rate (0.0005) in evaluations performed on both wired and wireless networks.

A study by Yin et al. in study [20] aimed to improve the multi-classification efficiency of IDS by the judicious pertinent features selection and the reduction of feature space dimensionality. The IGRF-RFE method integrates wrapper and filter techniques to improve feature importance selection. The initial phase employs Random Forest (RF) and Information Gain (IG) to eliminate less significant features, while the subsequent phase utilizes RFE to further optimize the attribute subset by discarding features which detrimentally affect model performance. This hybrid methodology seeks to improve the precision of the MLP-based detection of intrusion model utilizing the dataset of UNSW-NB15 through the selection of a more pertinent feature collection. The feature selection procedure decreased the number of features from 42 to 23, hence eliminating redundant and less pertinent characteristics. The MLP model's accuracy increased to 84.24% from 82.25% following the use of the “IGRF-RFE” approach. The weighted F1 score improved to 82.85%, indicating enhanced overall model performance for precision and recall.

The primary goal of the research by Saheed et al. in study [21] is to precisely detect fraudulent activity in computer networks by employing an advanced bat optimization technique in conjunction with the distinctive characteristics of the number system (residue). The work seeks to successfully diminish the complexity of the feature space by integrating the residue number system with the bat algorithm, while preserving or enhancing detection accuracy. The Bat algorithm is efficient for feature selection, although it may exhibit prolonged training and testing durations. The integration of RNS mitigates this constraint by enhancing processing speed. The study additionally utilizes PCA for feature extraction, which further enhances the chosen features. PCA facilitates the transformation of selected features into a lower-dimensional space while maximizing variance retention. The PCA + NB +Bat-RNS algorithm attained an accuracy of 97.82%. The Bat-RNS+PCA+KNN model exhibited an enhanced detection accuracy of 99.15%. The integration of the Bat method with RNS and PCA markedly improves the efficiency of the KNN classifier in intrusion detection.

A study by Francis and Sheeja in study [22] created an Intrusion Detection Model utilizing Bagging and Deep Reinforcement Learning (DRL). The model derives features from pre-processed data via the Enhanced Principal Component Optimization approach in conjunction with the Self-Optimizing Seagull Algorithm. This strategy aids in identifying pertinent features that can improve the model's efficacy. The chosen features are utilized to train the Bagging-DRL Intrusion Detection model, which integrates Convolutional Neural Networks, Multi-Layer Perceptron, Optimized Recurrent Neural Networks. The model is refined utilizing the Self-Improved Seagull Optimization Algorithm to augment detection precision. The model acquired an accuracy of 98.3% on the current dataset and 96% on the CSE-CIC-IDS2018 dataset. The framework demonstrated exceptional specificity rates of 99% for the NSL-KDD dataset and 97.6% for the CSE-CIC-IDS2018 dataset, highlighting its proficiency in accurately identifying non-intrusive cases. The sensitivity rates were robust, registering at 95% for the dataset of NSL-KDD and 98.3% for the CSE-CIC-IDS2018 dataset, indicating the model's efficacy in accurately detecting genuine intrusions.

A research paper by Rabash et al. in study [23] aims to selectively and adaptively identify pertinent characteristics in response to data alterations, tackling the issues presented by feature drift and concept drift in Intrusion Detection Systems. The suggested method employs a multi-objective optimization strategy to equilibrate several criteria, including feature relevance and feature reduction, so assuring that the chosen features enhance the classification model's performance

effectively. The research aims to increase the efficacy and precision of the IDS by the implementation of an "Enhanced Dynamic Filter-Based Feature Selection" (EDFBFS) architecture. The method utilizes a dual-mode strategy to produce optimal dynamic feature selection outcomes. The best feature set length is dictated by either the median or mean of the identified solutions in the Pareto, facilitating improved adaptation to varying circumstances. The method functions via iterative cycles encompassing initialization, crossover, and mutation processes. Throughout these cycles, objective functions are assessed according to feature relevance and feature reduction, directing the selection process. The E-DFBFS architecture proficiently tackles the issues of concept drift, facilitating enhanced adaptability in dynamic settings. Table I summarizes the contribution of previous researchers.

TABLE I. BACKGROUND WORK ANALYSIS

Study	Dataset(s)	Feature Selection Technique	Models
[13]	CICIDS2017	Mutual Information-based Feature Selection using genetic algorithm	Bagging, Random Forest XGBoost, Extra Tree and Decision Tree
[14]	NSL-KDD, UNSW-NB15	Multi-objective evolutionary algorithm	CART Decision tree, Logistic Regression, Random Forest
[15]	CSE-CIC-IDS2018, NSL-KDD	multi-agent feature selection	GCN
[16]	CSE-CIC-IDS2018	DT+RFE for feature selection	deep reinforcement learning
[17]	Four benchmark XSS datasets, which are, D3-30, D1-66, D4-30 and D2-167. T	Multi-agent feature selection and Deep Q-network	Multiple classifiers
[18]	KDDCup-99, NSL-KDD	Filter, wrapper, and embedded algorithms are classified as filter-based ensemble feature selection.	DLM is the short of RNN along with TDO
[19]	Aegean Wi-Fi Intrusion Detection Dataset	Automatic feature selection include (AFS-SVM, AFS-RF, AFS-ANN, and AFS-DT)	Two-phased Hybrid Ensemble learning
[20]	UNSW-NB15	Information gain and random forest with recursive feature elimination (RFE)	MLP
[21]	NSLKDD network data.	Bat algorithm with Residue Number System	NB, KNN
[22]	NSL-KDD and CSE-CIC-IDS2018 databases	Seagull algorithm for the enhancement of Enriched Principal Component Optimization	DRL uses MLP, CNN, while O-RNN interacts optimally with the surroundings or environment.

III. METHODOLOGY

The primary goal of this work is to develop a hybrid, ML model for network intrusion detection, in terms of feature selection, dimensionality reduction, and ensemble machine learning. The ameliorative model includes genetic algorithm (GA), recursive feature elimination (RFE), kernel linear discriminant analysis (KL), principal component analysis (PCA), deep Q-network (DQN optimization steps) and stacked ensemble learning about it. The subsequent sections define and explain each phase of the identified methodology sequentially starting from the data pre-processing phase right up to the phase dealing with the evaluation of the final model. In this part of the research, we present the architecture of the proposed model in Fig. 1.

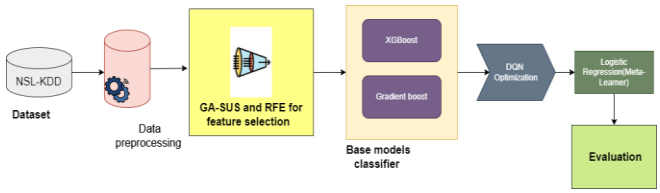


Fig. 1. Schematic architecture diagram of proposed system.

A. Preprocessing

The dataset used in this work has undergone a series of preprocessing methods to make it fit for the subsequent analysis. Firstly, the raw data is converted into a feature matrix with a corresponding vector label. The feature matrix contains a set of relative parameters that describe the network traffic, such as protocol type, packet size, and connection duration. The label vector comprises binary indicators that classify traffic into normal or incursion categories, facilitating supervised learning for ID.

To enhance the reliability and generalizability of the framework, the dataset was partitioned into testing and training subsets, a standard procedure for assessing model performance. The data division generally adheres to an 80:20 ratio, with 80% of the dataset designated for framework training and the residual 20% assigned for evaluating its predicted accuracy.

Prior to model training and feature selection, the data underwent supplementary preprocessing processes, encompassing demeaning and normalization of the features. Demeaning entails centering feature values around zero by subtracting the mean of each feature, whereas standardization adjusts the characteristics to achieve a standard deviation of one. These actions are essential for machine learning models, particularly when features display varying ranges or units of measurement. Standardization guarantees that all features contribute uniformly to the model, preventing those with more volatility or bigger magnitudes from overshadowing the learning process. This phase is crucial for models like as ensemble approaches and Support Vector Machines, which are sensitive to the relative scales of input features.

Standardizing the dataset before feature selection ensured a balanced representation of all features, enabling the feature selection method to discover the most pertinent qualities without bias. This thorough methodology strengthens the model's resilience, enabling it to more effectively identify trends in both legitimate and malicious network data.

B. Feature Selection using Genetic Algorithm (GA)

The process of feature selection involves reducing the number of attributes and identifying a subset of the original features. This technique is commonly utilised in data

preparation to uncover significant aspects that are often not known in advance and to eliminate superfluous or redundant features that have little bearing on classification tasks. In machine learning workflows, feature selection plays a pivotal role, particularly in enhancing the performance evaluation of classification models. The fundamental aim is to pinpoint the most crucial and informative features within the dataset, thereby improving accuracy.

Holland's genetic algorithm (GA) represents a computational optimisation methodology rooted in evolutionary biology principles. This technique operates in binary search spaces, managing a population of potential solutions. Each solution is encoded as a chromosome, comprising a finite sequence of binary digits. A fitness function assesses the viability of these solutions, with survival probability directly correlating to chromosomal fitness. The GA process commences with a randomly generated initial population, which then undergoes three primary mechanisms: selection, crossover, and mutation. The selection process identifies superior individuals for immediate progression to the next generation. Crossover involves the random exchange of chromosomal segments between two parent solutions to create offspring. Mutation introduces random alterations within individual chromosomes, contributing to genetic diversity.

This study employs Genetic Algorithms to remove inconsequential features. To achieve this objective, we designated chromosomes as a mask for attributes. For fitness evaluation, each individual in the population was assessed based on its ability to train a Random Forest classifier. If an individual selects at least one feature, the classifier is trained using these features, and its accuracy in the validation set determines the fitness score of the individual. If no features were selected, the fitness score was set to zero.

Selection was performed using stochastic universal sampling. First, the total fitness of the population was computed. The step size is then determined based on the total fitness and population size. Parents are chosen using a random start and pointers for a given size; the size is divided within the step size with the probability of high fitness being selected higher. Cross-over occurs whereby two selected parents are combined to form the offspring. A link was selected randomly and the child received some specific trait from both parents, or the first part was of one parent and the rest of the part was of other parent. Mutation is used in generating new offsprings by randomly setting bits to 0 or 1 adding new genetic feature to the population. A new population of the same size replaces the old one and this process a predefined number of generations or when some stopping criteria is fulfilled. Lastly the best from the final generation was chosen because it had the best fitness score out of all the individuals. This individual pertains to the best subset of features that are being searched sequentially by a genetic algorithm. The algorithm of GA along with mathematical formulae is given in Algorithm 1.

Algorithm 1: Genetic Algorithm for Feature Selection

Initialization:

Initialize the population $P = \{p_i \mid i = 1,2,\dots,P\}$, , where $p_i \in \{0,1\}^N$ is a binary array representing a subset of features.

Fitness Evaluation:

For each individual $p_i \in P$, compute the fitness:

Let $F(p_i)$ be the set of selected features:

$$F(p_i) = \{j \mid p_i[j] = 1, j = 1,2,\dots,N\} \quad (1)$$

If $F(p_i) \neq \emptyset$:

Then use the features of the dataset to train a random forest classifier

The accuracy acc(p_i) of the classifier is calculated.

Otherwise, $acc(p_i) = 0$

Selection (Stochastic Universal Sampling):

Calculate the total fitness:

$$total\_fitness = \sum_{i=1}^{p} \quad acc(p_i) \quad (2)$$

Determine the step size:

$$step\_size = \frac{total\_fitness}{[\frac{P}{2}]} \quad (3)$$

Select parents:

Start point:

$$start\_point = uniform(0, step\_size) \quad (4)$$

Pointers: pointers={start_point+k·step_size|k=0,1,…,⌊P2⌋−1}

The indices based on cumulative fitness are selected.

Crossover:

For each pair of parents, pi, and pj :

Random crossover point c: $c = random(0, N-1)$

Generate child:

$$c_k = (p_i[:c] \oplus p_j[c:]) \quad (5)$$

c_k inherits the first c bits from p_i and the remaining bits from p_j

Mutation:

For each child c_k:

For each bit c_k [j]:

c_k [j]=1−c_k [j]with probability μ

New Generation:

The old population was replaced with the new generation of children.

This process continues for G generations or till we meet a certain criterion is met

Output:

Identify the best individual p∗ from the final generation:

$$p* = acc(p_i) \quad (6)$$

C. Recursive Feature Elimination (RFE)

RFE is a wrapper technique for feature removal. It removes repetitive and ineffective features that minimally affect the training error, while preserving strong and independent features to enhance the framework's generalization activity. It utilizes an sequential approach for feature importance, that is a variant of "backward feature elimination". This technique first develops the model utilizing the entire set of features and then prioritizes the features according to their importance. It subsequently removes the least significant feature, reconstructs the model, and recalculates the feature importance.

Following the feature subset derived by Genetic Algorithm (GA) optimization, Recursive Feature Elimination (RFE) was used to further enhance the selection process and ascertain a more ideal collection of features. RFE functions by iteratively removing the least important features based on the amount of

contribution they make towards the improvement of the model until we arrive at the number of features we need. Feature selection is addressed by using Random Forest algorithm as a model to predict the importance of the features. Subsequent process included turning off one feature after another from the bottom, beginning from the least contributing feature and retraining of the model. This process is continued until arrive at K best features only. These features were used in the subsequent features reduction. The following sections feature reduction and estimation steps.

D. Dimensionality Reduction

To address the curse of dimensionality and further reduce the feature space, two dimensionality reduction techniques are employed: Two methods identified are Principal Component Analysis (PCA) and Kernel Linear Discriminant Analysis (KLDA).

KLDA was used to transform the data onto a shorter feature dimension and also minimizing the interclass distance (normal – intrusion). Based on a kernel function, KLDA can model the nonlinear relationship of features, and then establish a better feature space.

$$Z_{KLDA} = W_{KLDA}^{T} X_{top} \quad (7)$$

where WKLDA is the projection matrix obtained by maximizing the Fisher criterion.

After that, the features will be transformed by using the PCA in order to select only p principal components for comparison with the KLDA model. PCA removes projection directions determined to present high variability of the data and as such, most of the noise and redundant features.

$$Z_{PCA} = W_{PCA}^{T} Z_{KLDA} \quad (8)$$

Where WPCA is consists of eigenvectors corresponding to largest eigenvalues of the covariance matrix of Z_{KLDA}.

The final reduced dataset is denoted as Z_{final}.

E. Model Training and Stacked Ensemble Learning

1) Base learners: In order to construct a robust Intrusion Detection System (IDS), multiple base models were trained in the present study using a dataset that had been transformed into a lesser-dimensional vector space through the application of "Principal Component Analysis" (PCA). PCA, a prominent dimensionality reduction method, was utilized to identify the most critical characteristics while preserving the majority of the dataset's variation. XGBoost and Gradient Boosting Classifier were used as the main base models of the ensemble.

XGBoost is selected for handling large datasets and intricate pattern detection because of the gradient boosting framework upon which it is built. Additionally, GBC extends XGBoost, which iteratively provides better approximations to the model with fewer errors. These models complement each other to a great extent in the sense that they provide the benefit of handling numerous aspects of data complexity and drive up the predictive capability.

2) Meta classifier: A powerful binary classifier logistic regression takes the role of a meta-classifier. It is primarily deployed to merge the outcomes of the base, from which a final classification is generated. Logistic regression was again chosen because it is good at weighting the results of other models, and it calculates the best weights for each base model depending on the accuracy of the latter. The goal of this strategic integration is to increase the ability of the model to distinguish normal behaviour from non-normal or abusive behaviour.

3) Deep Q-Network (DQN) optimization

a) Q-Learning setup: Realising that the ensemble model could be enhanced, for hyperparameter tuning, we use a deep Q-network (DQN). Reinforcement learning is used in the form of a DQN, which helps in selecting the optimally-suited numerical for the hyperparameters for the best results. In this regime, the DQN influences the model in terms of the hyperparameters, and the response is a set of rewards derived from the model's evaluation results.

b) Training: When acquiring DQN, Q-values are updated when the amount of hyperparameters defined rises. The objective is to improve the reward function, which in the present case is the enhancement of the performance of the ensemble model. The same approach that is, following the above outlined feature selection scheme, benefits the DQN in a way that it is able to bring about 'fine tuning' of the hyperparameters to a level where classification differences of network activities are enhanced.

F. Proposed Model Algorithm

The combination of shortlisted features, the set of the training parameters, and performance metrics in a final model is preserved for future use. The documentation of the results comprises an evaluation of the proposed hybrid architecture for network intrusion identification. In this detailed record, the actual and the predicted markings are mentioned, which define how accuracy the model is beneficial for classifying the network threats; hence, comprehend how independent utilization of methodologies can be beneficial. The algorithm of the proposed model is given in Algorithm 2.

Algorithm 2: Proposed Machine Learning Framework for Network Intrusion Detection

Initialization

- X, y← Load data
- Hyperparameters←Set parameters for GA, RFE, KLDA, PCA , DQN, and Stacking models

Feature Selection using Genetic Algorithm (GA)

- Initialize Population:
  - Population ← Random Initialization of N chromosomes
- Evaluate Fitness:
  - For each chromosome c_i∈Population:
    - Features←Selected by c_i
    - Model←Train RandomForest on Features
    - Fitness(c_i)←Evaluate model accuracy

- Selection:
  - Selected Chromosomes←Stochastic Universal Sampling (SUS) based on Fitness
- Crossover:
  - Offspring←Apply Crossover on Selected Chromosomes
- Mutation:
  - Mutated Offspring←Apply Mutation with rate pm
- Update Population:
  - Population←Mutated Offspring
- Repeat:
  - Repeat steps for G generations or until convergence.
- Final Selection:
  - c_{best}←Chromosome with highest Fitness

Recursive Feature Elimination (RFE)

- Feature Ranking:
  - Ranked Features←RFE with RandomForest on Features selected by c_{best}
- Feature Selection:
  - Top Features←Select k best features

Dimensionality Reduction

- Apply KLDA:
  - Z_{KLDA}←KLDA on Top Features
- Apply PCA:
  - Z_{PCA}←PCA on ZKLDA reducing to p components

Model Training using Stacked Ensemble Learning

- Base Models:
  - Base Models← Train models (XGBoost, GBC) on Z_{PCA}
- Meta-Classifier:
  - Meta-Model←Train Logistic Regression on predictions of Base Models

Deep Q-Network (DQN) Optimization

- Q-Learning Setup:
  - States, Actions, Rewards, Q (s,a)←Define for DQN
- Training:
  - Q(s,a)←Train DQN to optimize hyperparameters or thresholds Q (s,a)

Model Evaluation

- Prediction:
  - $\hat{y}$←Predict using Meta-Model on test data
- Evaluation Metrics:
  - Recall, F1-Score, Accuracy, Precision, Confusion Matrix←Evaluate on $\hat{y}$

Output Results

- Save (Features, Model Parameters, Metrics)
- Visualize Performance

IV. RESULTS

This section presents the results of the current study. Basis on the results attained, it is deduced that the intelligent hybrid model of GA-SUS feature selection and stacking ensemble learning model with deep Q-learning neural network, which is proposed in the current research, is critical for using in network intrusion detection. NSL-KDD was used to benchmark the model with tests conducted to determine success rates, Precision, F1-score, recall, accuracy in differentiating between normal traffic, and anomalous traffic.

A. Dataset

The current dataset, NSL-KDD Dataset [24] is an improved and augmented version of the old KDD Cup 99 dataset, and is more suitable for IDS assessment. This approach eliminates certain inaccuracies in the initial data, for example, the presence of multiple records, which can introduce certain biases in the evaluation of an IDS. NSL-KDD consists of several types of records and probes: normal, DoS, R2L, U2R, and probes in the network traffic records. It is widely used to compare IDS effectiveness because it provides a reasonable distribution that is close to the real traffic distribution [16].

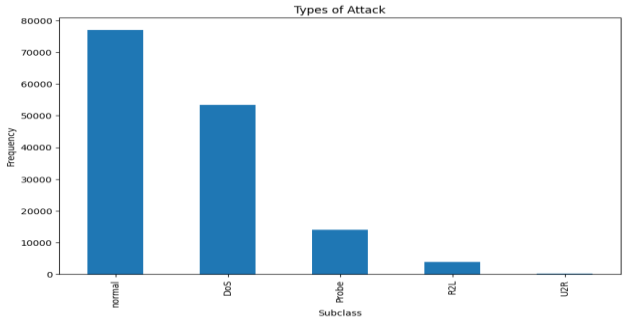


Fig. 2. Class distribution.

Fig. 2 illustrates the proportion of class labels within the dataset with the class label that appears most frequently. Such distribution forms can be skewed where some classes like 'DoS' and 'normal' are more frequent than classes like 'U2R.' Such distribution is import for model training and testing.

B. Evaluation Criteria

The assessment of the suggested model was performed using the following features: accuracy, confusion matrix, recall rate, precision rate, and F1 score. Accuracy gives a general measure of the developed model and checks correctness of the developed model. Precision, and recall measure to some extent how many of the positive instances are correctly classified and how few misclassifications in the form of false positives or false negatives are there. The F1 score is a metric that is in-between recall and precision. The confusion matrix allows estimating all the true, false, negations and positives that can be retrieved from the assessed model. It is the basis for calculating the said metrics. The formulae for the above metrics are given below.

$$Accuracy = \frac{TP1 + TN1}{TP1 + FP1 + TN1 + FN1}$$

$$Recall = \frac{TP1}{TP1 + FN1}$$

$$Precision = \frac{TP1}{TP1 + FP1}$$

$$F1 - Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

Where FN is false negative, FP is false positive, TP is true positive, TN is true negative. These outcomes are shown in various kinds of diagrams and graphs for the objective of understanding and evaluating the performances of the models.

C. Classification Performance

In Table II, the classification report of a model with GA-SUS feature selection is illustrated. The model achieves an appreciable degree of accuracy: the overall accuracy is 0. 9761. Outstanding performance for "DoS" (Denial of Service) category, shown that the model made a highly accurate detection of such kind of attacks. The "Probe" category is another category that gives a good result, but 'DoS' performance is slightly higher with good identification rate. Needless to say, weaker performance can be observed in the "R2L" category that has lower effectiveness for this kind of recognition. The "U2R" category can be said as very poor with all the parameters being nearly low. Since the presence of this category is negligible in the dataset, the detection capability shows a very poor result. As for the last "normal" group, the model correctly correlates their network activity with high performance indicators. In conclusion, the macro levels of performance at each class are low to moderate but at the same time the weighted levels indicate high competency of the model at identifying certain classes that are more dominant. Figure 3 provides confusion matrix of the model that used GA-SUS feature selection algorithm.

TABLE II. CLASSIFICATION REPORT OF MODEL USING GA-SUS FEATURE SELECTION

	precision	f1-score	recall	support
Probe	0.96	0.95	0.95	2749
DoS	0.99	0.99	0.99	10688
U2R	0.00	0.00	0.00	25
R2L	0.85	0.79	0.74	792
normal	0.98	0.98	0.98	15450
weighted average	0.97	0.98	0.98	29704
macro average	0.76	0.74	0.73	29704

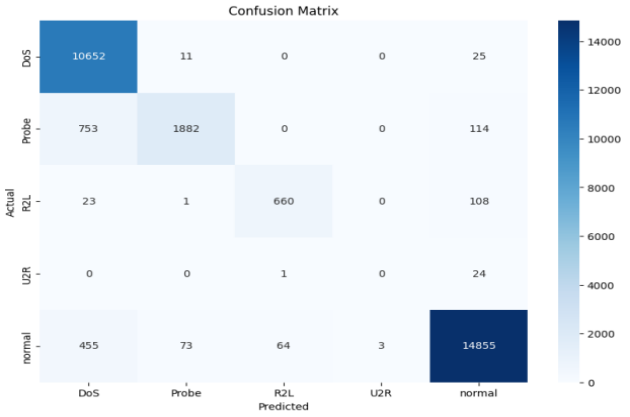


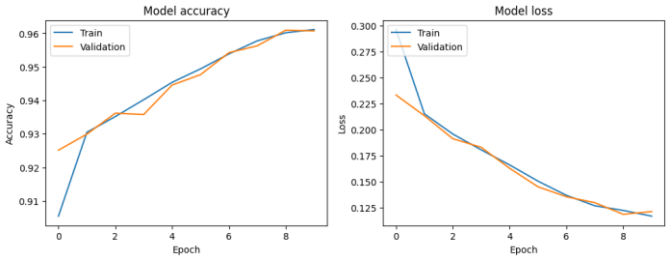
Fig. 3. Confusion Matrix of model using GA-SUS feature selection.

The suggested GA-SUS feature selection technique was contrasted with differential evolution-based algorithms that have the maturity extension feature selection proposed in [22]. When comparing the proposed GA-SUS with RFE ensemble learning approach to DE-ME, differences in performance and technique are evident.

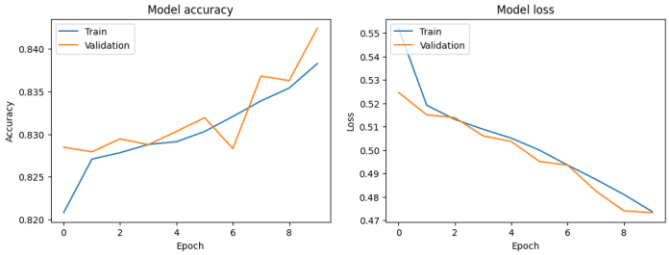
Classification Report of model using DE-ME Feature Selection is shown in Table III. Classification report shows overall high performance of the model in using feature selection from DE-ME is 94.43%. Once more, the accuracy of the model is extremely high when it comes to the detection of "DoS" and "normal" classes due to high coefficients of F1-score, recall, and precision, which equals to 0.90 and above. The macro average F1-score is calculated to be 0.72 and clearly shows the variation in the performance of the model across the classes Hence the weighted average F1-score of 0.94 reveals the complete performance of the framework; however, it somewhat biases towards the majority classes "DoS and "normal". But this means that the model is more accurate when it comes to frequent attacks but not as effective when it comes to rare attacks.

TABLE III. CLASSIFICATION REPORT OF MODEL USING DE-ME FEATURE SELECTION

	Recall	Precision	F1-score	Support
U2R	0.00	0.00	0.00	25
DoS	1.00	0.90	0.94	10688
R2L	0.83	0.91	0.87	792
Probe	0.68	0.96	0.80	2749
normal	0.96	0.98	0.97	15450
macro avg	0.70	0.75	0.72	29704
weighted avg	0.94	0.95	0.94	29704



(a). The Accuracy and loss of models using GA-SUS feature selection.



(b). The Accuracy and loss of models using DE-ME feature selection.

Fig. 4. Accuracy and loss plot.

Fig. 4 (a) and Fig. 4 (b) illustrates Accuracy and loss plot for GA-SUS and DE-ME feature selection respectively. The accuracy and loss plots compare model performance using two

feature selection methods: DE-ME and GA-SUS. For both methods, the accuracy plot shows how well the models correctly classify data over training epochs, while the loss plot tracks the error reduction. Typically, a rising accuracy and a decreasing loss indicate good model training. Comparing the two, GA-SUS likely shows better stability with smoother curves and higher final accuracy, while DE-ME may have more fluctuations, suggesting GA-SUS's feature selection yields a more consistent and accurate model. The plots help visualize the effectiveness of each feature selection approach.

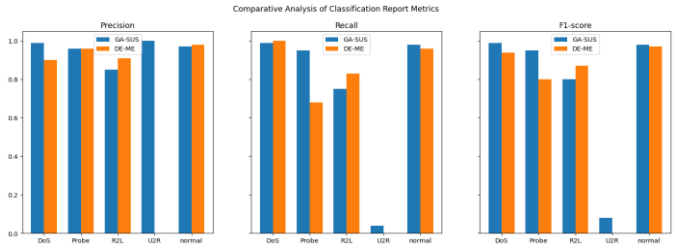


Fig. 5. Comparative performance analysis.

Fig. 5 presents the comparative performances classification algorithms. It visually compares the corresponding performance indices of two different models or features selection algorithms. This is likely to report, on the same screen, metrics such as Recall, Precision, F1-score, and even accuracy for each class, enabling a calibration. This comparison illustrates how various solutions affect the framework's ability in screening different kinds of attacks and normal traffic. In the current case and by the overlap of figure we are able to easily compare which of the GA-SUS feature selection method performs better in general and which one has a problem with certain classes. It offers information about the best and inferior aspects that can be used to strengthen the model.

D. Discussion

This study proposes a novel technique of GA-SUS with RFE for selecting the features for an IDS employing three benchmark datasets. In comparison with the existing approach, the current approach yielded results listed in Table IV.

Various studies on IDS datasets have applied different feature selection and machine learning algorithms. Our proposed model yielded decent results compared with those of other feature selection approaches in the literature.

TABLE IV. COMPARISON OF GA-SUS WITH RFE IN EXISTING STUDIES

Study	Feature Selection algorithm	Model	Accuracy achieved (%)
[25]	BukaGini(gini Importance)	Random forest classifier	99
[26]	Feature importance (RF)	RF	-
[27]	Condensed nearest neighbors (CNN)	CNN	95.54
		Radial basis function (RBF)	94.28
[20]	IGRF-RFE	MLP	-
[28]	GA in Map-Reduce	LR, SVM, RT, NB, ANN	90.45%
Proposed model	GA-SUS with RFE	Ensemble learning -DQN	97.61%

BukaGini, with a Random Forest classifier, obtained a high accuracy of about 99%. Other methods, such as Radial Basis Function (RBF) and convolutional neural network (CNN), yielded accuracies of 95.54% and 94.28%, respectively. The GA in the MapReduce approach combined with LR, RT, ANN, SVM and NB achieved 90.45% accuracy. Our model, utilising GA-SUS with Recursive Feature Elimination (RFE) and ensemble learning optimised by DQN, achieved a notable accuracy of 97.61%, displaying its robustness in intrusion detection.

Although the proposed model offers good results, certain limitations still exist. There appears to be no perfect dataset for studying invertible graphs; however, the current work employed the dataset called NSL-KDD, which has been used in most previous studies but may not portray real-life network traffic and emerging threats. Furthermore, the optimisation process used in DQN is quite efficient, but at the same time, it is costly and time consuming; hence, its applicability to large datasets or real-time data may be problematic. This study also presupposes that the selected features remain the best under various network conditions, which may not be true. Future work could consider extending the work to other types of datasets with larger and diverse groups of users, and also compare the model performance in real-time activities in dynamic network topologies.

V. CONCLUSION

The findings from this study highlight the feasibility of the proposed hybrid model of GA-SUS with RFE for feature selection and DQN for fine-tuning an ensemble learning model of classifiers for network intrusion detection. It reaches an accuracy of 97.60% on the NSL-KDD dataset and is capable of detecting different kinds of attacks, such as revival of DoS and probe attacks, as it solves the problem of class imbalance. The proposed multi-objective optimization harnessing stochastic universal sampling with a Genetic Algorithm for selection and Deep Q-Networks thus contributes to the design of new approaches for improving the generalization of the model by reducing its sensitivity to changes in the training data. As a result, the development of the study has limitations evident as follows; this kind of attack is very rare, but because it is present in the dataset very few times, the performance for such types like U2R remains below par. Future work may investigate better detection rates for these minority classes by investigating better data augmentation techniques or by using enriched deep neural networks. Furthermore, the model could be tested on other datasets as well as real-time environments, and such aspects could also be further explored. Extending this approach to address dynamic cyber threats or using it for more general and larger sets would further improve the approach to help with network security use cases.

REFERENCES

[1] A. Thakkar and R. Lohiya, "A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions," Artif. Intell. Rev., vol. 55, no. 1, pp. 453–563, 2022.

[2] S. Hajj, R. El Sibai, J. Bou Abdo, J. Demerjian, A. Makhoul, and C. Guyeux, "Anomaly - based intrusion detection systems: The requirements, methods, measurements, and datasets," Trans. Emerg. Telecommun. Technol., vol. 32, no. 4, p. e4240, 2021.

[3] F. Sharif, "The Role of Ensemble Learning in Strengthening Intrusion Detection Systems: A Machine Learning Perspective," 2024.

[4] S. Ali, S. U. Rehman, A. Imran, G. Adeem, Z. Iqbal, and K.-I. Kim, "Comparative evaluation of ai-based techniques for zero-day attacks detection," Electronics, vol. 11, no. 23, p. 3934, 2022.

[5] M. Di Mauro, G. Galatro, G. Fortino, and A. Liotta, "Supervised feature selection techniques in network intrusion detection: A critical review," Eng. Appl. Artif. Intell., vol. 101, p. 104216, 2021.

[6] S. Das et al., "Network intrusion detection and comparative analysis using ensemble machine learning and feature selection," IEEE Trans. Netw. Serv. Manag., vol. 19, no. 4, pp. 4821–4833, 2021.

[7] H. Liu et al., "Evolving feature selection," IEEE Intell. Syst., vol. 20, no. 6, pp. 64–76, 2005.

[8] N. Sánchez-Maroño, A. Alonso-Betanzos, and M. Tombilla-Sanromán, "Filter methods for feature selection–a comparative study," in International Conference on Intelligent Data Engineering and Automated Learning, 2007, pp. 178–187.

[9] N. El Aboudi and L. Benhlima, "Review on wrapper feature selection approaches," in 2016 international conference on engineering & MIS (ICEMIS), 2016, pp. 1–5.

[10] H. Liu, M. Zhou, and Q. Liu, "An embedded feature selection method for imbalanced data classification," IEEE/CAA J. Autom. Sin., vol. 6, no. 3, pp. 703–715, 2019.

[11] Y. B. Wah, N. Ibrahim, H. A. Hamid, S. Abdul-Rahman, and S. Fong, "Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy.," Pertanika J. Sci. Technol., vol. 26, no. 1, 2018.

[12] M. Zivkovic et al., "Hybrid genetic algorithm and machine learning method for covid-19 cases prediction," in Proceedings of international conference on sustainable expert systems: ICSES 2020, 2021, pp. 169–184.

[13] H. Bakır and Ö. Ceviz, "Empirical enhancement of intrusion detection systems: a comprehensive approach with genetic algorithm-based hyperparameter tuning and hybrid feature selection," Arab. J. Sci. Eng., pp. 1–19, 2024.

[14] Z.-H. Cheng, H. Shang, and C. Qian, "Detection-Rate-Emphasized Multi-objective Evolutionary Feature Selection for Network Intrusion Detection," arXiv Prepr. arXiv2406.09180, 2024.

[15] K. Ren, Y. Zeng, Y. Zhong, B. Sheng, and Y. Zhang, "MAFSIDS: a reinforcement learning-based intrusion detection model for multi-agent feature selection networks," J. Big Data, vol. 10, no. 1, p. 137, 2023.

[16] K. Ren, Y. Zeng, Z. Cao, and Y. Zhang, "ID-RDRL: a deep reinforcement learning-based feature selection intrusion detection model," Sci. Rep., vol. 12, no. 1, p. 15370, 2022.

[17] I. K. Thajeel, K. Samsudin, S. J. Hashim, and F. Hashim, "Dynamic feature selection model for adaptive cross site scripting attack detection using developed multi-agent deep Q learning model," J. King Saud Univ. Inf. Sci., vol. 35, no. 6, p. 101490, 2023.

[18] C. Kavitha, T. R. Gadekallu, N. K, B. P. Kavin, and W.-C. Lai, "Filter-based ensemble feature selection and deep learning model for intrusion detection in cloud computing," Electronics, vol. 12, no. 3, p. 556, 2023.

[19] A. K. Mananayaka and S. S. Chung, "Network intrusion detection with two-phased hybrid ensemble learning and automatic feature selection," IEEE Access, vol. 11, pp. 45154–45167, 2023.

[20] Y. Yin et al., "IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," J. Big Data, vol. 10, no. 1, p. 15, 2023.

[21] Y. K. Saheed, T. O. Kehinde, M. Ayobami Raji, and U. A. Baba, "Feature selection in intrusion detection systems: a new hybrid fusion of Bat algorithm and Residue Number System," J. Inf. Telecommun., vol. 8, no. 2, pp. 189–207, 2024.

[22] E. Geo Francis and S. Sheeja, "Enhanced intrusion detection in wireless sensor networks using deep reinforcement learning with improved feature extraction and selection."

[23] A. J. Rabash, M. Z. A. Nazri, A. Shapii, and M. K. Hasan, "Non-Dominated Sorting Genetic Algorithm based Dynamic Feature Selection for Intrusion Detection System," IEEE Access, 2023.

[24] S. Mohanty and M. Agarwal, "Recursive Feature Selection and Intrusion Classification in NSL-KDD Dataset Using Multiple Machine Learning Methods," in 2nd International Conference on Computing, Communication, and Learning, CoCoLe 2023, 2024, pp. 3–14.

[25] M. A. Bouke, A. Abdullah, K. Cengiz, and S. Akleylek, "Application of BukaGini algorithm for enhanced feature interaction analysis in intrusion detection systems," PeerJ Comput. Sci., vol. 10, p. e2043, 2024. DOI:10.7717/peerj-cs.2043

[26] N. M. Khan, N. Madhav C, A. Negi, and I. S. Thaseen, "Analysis on improving the performance of machine learning models using feature selection technique," in Intelligent Systems Design and Applications: 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018) held in Vellore, India, December 6-8, 2018, Volume 2, 2020, pp. 69–77.

[27] F. Z. Belgrana, N. Benamrane, M. A. Hamaida, A. M. Chaabani, and A. Taleb-Ahmed, "Network intrusion detection system using neural network and condensed nearest neighbors with selection of NSL-KDD influencing features," in 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), 2021, pp. 23–29.

[28] D. Mehanović, D. Kečo, J. Kevrić, S. Jukić, A. Miljković, and Z. Mašetić, "Feature selection using cloud-based parallel genetic algorithm for intrusion detection data classification," Neural Comput. Appl., vol. 33, pp. 11861–11873, 2021