Scientific Reports

https://doi.org/10.1038/s41598-026-59142-1

Article in Press

LwHM: lightweight hybrid classifier for SDN-attack detection using recursive feature elimination

Received: 30 March 2026

Accepted: 18 June 2026

Published online: 24 June 2026

Cite this article as: Kanwal K., Mujahid M., Espinosa J.C.M. *et al.* LwHM: lightweight hybrid classifier for SDN-attack detection using recursive feature elimination. *Sci Rep* (2026). https://doi.org/10.1038/s41598-026-59142-1

Khadija Kanwal, Muhammad Mujahid, Julio Cesar Martinez Espinosa, Carlos Eduardo Uc Rios, Nagwan Abdel Samee & Imran Ashraf

LwHM: Lightweight Hybrid Classifier for SDN-Attack Detection using Recursive Feature Elimination

Khadija Kanwal[1†], Muhammad Mujahid[2†], Julio Cesar Martinez Espinosa[3,4,5,6], Carlos Eduardo Uc Rios[3,4,7,8], Nagwan Abdel Samee[9], Imran Ashraf[10*]

[1]Institute of Computer Science and Information Technology, The Women University Multan, Multan, Pakistan.
[2]Artificial Intelligence and Data Analytics (AIDA) Lab, CCIS, Prince Sultan University, Riyadh, Saudi Arabia.
[3]Universidad Europea del Atlantico., Isabel Torres 21, Santander, 39011, Spain.
[4]Universidad Internacional Iberoamericana, Campeche, 24560, Mexico.
[5]Universidade Internacional do Cuanza, Cuito, Bie, Angola.
[6]Fundacion Universitaria Internacional de Colombia, Bogota, Colombia.
[7]Universidad Internacional Iberoamericana Arecibo, Puerto Rico, 00613, USA.
[8]Universidad de La Romana, La Romana, Republica Dominicana.
[9]Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia.
[10]School of Computer Science and Engineering, Yeungnam University, Gyeongsan, 38541, Republic of Korea.

*Corresponding author(s). E-mail(s): imranashraf@yu.ac.kr;
Contributing authors: khadijakanwal@wum.edu.pk; mmujahid@psu.edu.sa; julio.martinez@unini.edu.mx; carlos.uc@unini.edu.mx; nmabdelsamee@pnu.edu.sa;
†These authors contributed equally to this work.

Abstract

Internet connectivity has significantly enhanced the efficiency of daily operations, information retrieval, and global communication. However, this heightened reliance on technology has also exposed us to cybersecurity threats that are often beyond our control. Consequently, securing the data, privacy, and critical systems demands essential cybersecurity measures. This study focuses on the role of artificial intelligence in strengthening security systems to thwart network breaches. The study proposes a comprehensive three-part approach for software-defined networking (SDN) security. The first is that the concentration is on assuring data integrity and reliability for an SDN intrusion dataset. This involves critical steps such as data cleaning, preprocessing, and normalization. In the second step, six popular feature selection strategies are applied, which encompass recursive feature elimination (RFE), polynomial features, artificial neural networks, SelectKBest, least absolute shrinkage and selection operator (LASSO), and correlation-based features. These techniques help identify and incorporate significant and relevant features, thereby improving the overall model performance. The third part involves the creation of a lightweight hybrid model (LwHM) that leverages the strengths of k-nearest neighbors and decision tree models, utilizing a voting classifier. The LwHM surpasses the performance of the InSDN dataset, achieved an impressive accuracy score of 99.93% with RFE features, and enhance the SDN security efficiently.

Keywords: Network intrusion detection; network security; software defined networks; hybrid model; machine learning; feature selection

1 Introduction

Software-defined networking (SDN) is an innovative approach to network management and control that separates the control plane from the data plane in networking equipment [1]. This approach allows various networks to function independently of the core network devices, such as switches, routers, firewalls, and more, and provides a means to insulate network hardware changes from the underlying network infrastructure [2]. The separation of the control plane from the data plane in component devices enables a consistent network management technique, regardless of network complexity or size [2]. Unfortunately, while there are advantages, every new technology also brings with it novel threats. SDN technology has brought numerous benefits to network management and efficiency, but it is not immune to security challenges. SDN introduces its own set of vulnerabilities, and malicious actors can exploit these to breach security mechanisms within SDN networks [3].

Signature-based network intrusion detection systems (NIDS) have traditionally served as the predominant technique for network intrusion detection, primarily because they can quickly adapt to the evolving landscape of attacks [4]. However, their scalability has been challenged by the surge in global network traffic, as highlighted in [5]. These systems are primarily designed to identify known attacks, which can limit their effectiveness. In recent times, machine learning (ML) algorithms have gained

prominence in the development of IDS. Nevertheless, a significant challenge lies in the scarcity of suitable training datasets, as discussed in [6] and [7]. Furthermore, many of the existing datasets have faced extensive criticism for their limited capacity to accurately represent real-world scenarios, as elaborated in [7] and [8].

In recent times, the development of efficient IDS has encountered significant challenges, primarily due to the ongoing evolution of SDN. Experimentation of IDS is performed on existing datasets designed with traditional network models, while SDN-specific datasets are not available [9]. InSDN is the latest intrusion dataset that was developed to address this problem. This dataset was taken from the study [10] with a custom-built SDN model testbed and comprises attacks precisely designed for SDN attack vectors with the addition of a standard suite for exploits. Several supervised machine learning-based models were tested and trained by cross-validation of various target classifications, and performance metrics, namely recall, precision, and F1 score, were used to measure the performance. The motivation of the study [10] is to increase the foundational supervised machine learning work; refine classification techniques, highlight a significant domain-specific metric to measure performance, and focus on developing methods to identify the unseen attacks.

NIDS plays a vital role in the identification of abnormal network system activity and the prevention of network attacks [11]. This study also proposed an NIDS for SDN using ML methods. The study works on a hybrid model approach and feature selection to achieve significant results. The presented technique novelties and contributions are as follows:

- The study introduces a lightweight hybrid model (LwHM) that combines the strengths of k-nearest neighbors (KNN) and decision tree (DT) models. This amalgamation is achieved through the use of a voting classifier. In this criterion, average probabilities help the proposed model achieve significantly better results.
- In the realm of intrusion detection, dealing with meaningless features can cause models to underfit, and it can also lead to architectural complexity, resulting in prolonged operational and training durations. To mitigate this challenge, the study employs six popular feature selection strategies, including recursive feature elimination (RFE), polynomial features, artificial neural networks (ANN), SelectKBest, least absolute shrinkage and selection operator (LASSO), and correlation-based features. These techniques are used to identify and incorporate significant and relevant features, which, in turn, improve the overall performance of intrusion detection models. This step contributes to more efficient and accurate security analysis.
- The study focuses on ensuring the integrity and reliability of an SDN intrusion dataset. This involves critical steps such as data cleaning, preprocessing, and normalization, which are essential for accurate and reliable analysis of network security data.
- Additionally, this research includes an extensive analysis of various aspects, including ML models, feature selection techniques, evaluation parameters, and a comprehensive comparison with existing methods. This in-depth analysis provides a holistic understanding of the strengths and weaknesses of different approaches, ultimately contributing to a more informed and well-rounded assessment of our proposed methodology.

Despite existing frameworks on intrusion detection, the proposed approach differs in several ways. First, most existing SDN rely on single classifier like SVM, CNN or RF. LwHM distinctively combines KNN and DT using voting and thereby use the complementary strengths of both algorithms. This combination improves robustness with maintaining computational efficiency. Secondly, it is lightweight ensemble which distinguishes it from computationally complex deep learning models involving the use of models like long short-term memory, transformer or generative adversarial networks. The proposed LwHM obtains good accuracy with reduce computational overhead and is more suitable for resource-constrained SDN environments. While many existing frameworks focus on classification stage exclusively, the proposed approach follows data integrity, multi-strategy feature selection and hybrid classification which makes it more suitable for SDN. It also incorporates several feature selection strategies that proves fruitful with its superior 99.93% accuracy.

This study is further divided into the following sections: Literature review is in Section 2, the proposed approach and dataset description are in Section 3, while Section 4 contains results and discussion. In the end, the conclusion is in Section 5.

2 Literature Review

ML-based classifiers have recently been applied in network intrusion detection due to their automated and scalable features. A network intrusion detection system plays a vital role in computer network security. There are numerous detection methods, but the anomaly-based automated detection technique outperforms others remarkably [12]. The experimentation is performed on KDDCUP'99 and CIC-MalMem-2022 and reported outstanding accuracy in the study [12]. Random forest (RF) is used to train network detection classifiers and adaptive synthetic sampling (ADASYN) for intrusion detection on CIC-IDS2017 in [13]. The experimentation is performed for intrusion detection on a large-scale dataset. Another research employed the long short-term memory (LSTM) model to collect temporal information, whereas a convolutional neural network (CNN) was applied to extract spatial features. In [13], the presented model enhanced the weights on the training dataset to solve the problem of class imbalance.

The study [14] employed a 1D-CNN model on time-series data using 42 features of supervised learning. A max-pooling layer is integrated with a convolutional layer to reduce computational complexity, improve output size, and increase feature count. In this article, the performance of the presented 1D-CNN is compared with RF, support vector machine (SVM), LSTM, and a hybrid system of 1D-CNN for both unbalanced and balanced training datasets. Moreover, a random over-sampling technique was used to solve the issue of unbalanced data after a complete analysis was carried out using the publicly available dataset.

Research [15] introduces a novel voting classifier comprised of a collection of machine learning models that have undergone training and improvement through the utilization of metaheuristic optimization techniques. The dipper-throated optimizer (DTO) was a refined version of the whale optimization algorithm (WOA), designed to function as a metaheuristic optimizer. This modification improves the exploration

method of the initial WOA optimizer. The voting classifier has a high level of effectiveness and reliability in classifying network attacks. The efficacy of the suggested approach for classifying binary assaults was evaluated, hence assessing the effectiveness of the proposed strategy. The efficacy, consistency, and importance of the recommended approach were evaluated through the utilization of statistical analysis and visual graphics. The obtained outcomes confirmed the superiority of the method presented for the purpose of network intrusion detection.

1D-dilated causal neural network (1D-DCNN)-based IDS for the binary categorization was used in the study [16]. To check the efficiency of the presented methods, experimentation was performed on two common publicly available databases named CSE-CIC-IDS2018 and CIC-IDS2017. Simulation results reported better performance when compared with state-of-the-art deep learning methods. In addition, CNN architecture was used for the CSE-CIC-IDS2018 to transform data into images and evaluate its efficiency using the recurrent neural network (RNN) system in the study [17]. Experimental results demonstrate that a CNN has outstanding performance using RNN on CSE-CIC-IDS2018. Another technique in [18] is suggested to improve network effectiveness for anomaly detection by combining Long Short-Term Memory with an attention mechanism (AM). The experimentation was performed on the CSE-CIC-IDS2018 dataset and reported 96.22% accuracy, 15% detection ratio, and 96% recall ratio.

An intrusion detection system identifies and monitors network traffic data and activates alerts when identified threats or suspicious activities are detected; therefore, the network administrator may look at the activities and make a suitable decision [19]. The study [20] presented a hybrid intrusion detection system (HIDS) that merges four modules, such as NBFS for feature selection, a data preprocessing module, PKNN for making the decision, and outlier detection using OSVM.

Several researchers have proposed network security solutions based on the SDN model. An approach comprises a moving target defense (MTD) to design an attribute-diverse and frequently changing network system that operates using external opinion. Random deviation can consist of port number, IP address, routing path, instruction set, host identity, and so on. Some vulnerabilities have been inquired in this research on different works like SDN-GAN, flow merge, and Hydra [21].

Network intrusion detection struggles with uneven data distribution across training and testing datasets, unequal data class representation, and inadequate accuracy. The study [22] presents a two-layer soft-voting ensemble learning model using RF, light gradient boosting machine (LightGBM), and extreme gradient boosting (XGBoost) as basic classifiers to overcome the above issues. During training, the model uses adversarial validation to examine data distribution coherence and determine if a dataset re-split was needed. Experimental results show that the soft-voting ensemble learning model outperforms individual models in binary and multi-class classification problems.

Various techniques are crafted by manipulating a manner for a number of network flows as a generative adversarial network (GAN)-based approach. The SDN network has been applied in many applications, namely routing management, resource management, network security, network flow monitoring, and machine learning-based

techniques are mentioned in every application. It is a big challenge that machine learning fails to demonstrate the expected performance because of the lack of available training databases and complex network traffic patterns proposed by the SDN system. A report [23] presented the implementation of network intrusion detection with machine learning-based techniques in SDN. This report mainly focused on several ML/DL methods employed for network intrusion detection in an SDN environment. [24] presented a report on DL and ML methods to manage network traffic, specifically traffic prediction and classification in SDN.

Hybrid models combining the strengths of various machine learning and deep learning have been proposed as well for intrusion detection. For example, [25] proposed IDS that combines CNNs and recurrent neural network (RNNs). The authors utilized CNN for obtaining spatial feature while temporal patterns are recognized through RNNs. To improve critical data representation, an attention mechanism is utilized. Improved results show the superiority of the proposed hybrid approach. Along the same directions, [26] works out a hybrid approach that incorporates cryptographic optimization. In addition, blockchain technology is used for secured data sharing. Optimization is carried out using simulated annealing gradient-based optimizer with SHA3-512 for data integrity. Experiments involve the use of Ethereum blockchain with interplanetary file system. Experimental output confirms the superior performance of the hybrid approach in terms of key optimization and accuracy.

The researchers discussed the lack of availability of labeled data as an important concern, and a new hybrid learning method based on the CNN algorithm to classify the traffic flow into normal classes. A novel SD-Reg technique is applied to address the over-fitting problem and to increase the capability of network intrusion detection systems in unseen intrusion event detection [27]. The most recent InSDN dataset is used to evaluate the performance of the presented techniques. The SDN switch reports network status using the OpenFlow protocol, and the controller can instantly identify distributed denial of service (DDoS) attacks, as this dataset shows no redundancy in memory units [28]. The summary of the literature review regarding network intrusion attack detection is presented in Table 1.

3 Materials and Methods

The proposed approach was implemented on a Core i7 12th-generation machine running the Windows operating system. This system is equipped with 64 GB of RAM and a 1 TB SSD. The experiments were conducted using Jupyter Notebook with the Python programming language.

Figure 1 shows the workflow of the proposed approach. In the proposed approach, the process is initiated by acquiring the dataset from a public repository. Following this, data cleaning is carried out to eliminate missing values and employed label encoding to convert categorical values into a numeric format. Data normalization was then performed using standard scaling, a crucial step that ensures the data adheres to a normal distribution, facilitating the optimal functioning of machine learning models. After preprocessing, the dataset was split into an 80:20 ratio, with 80% of the data allocated for model training and the remaining 20% for evaluation. Further, various

Table 1: Summary of literature review regarding network intrusion attack detection.

Ref.	Years	Methods	Purpose	Existing Limitations
[18]	2019	CNN	Achieved high accuracy using preprocessing method	Cannot perform reorganizing of convolutional layers with CNN parameters, did not employed ML models or advance DL
[14]	2020	RF, LSTM, SVM, 1D-CNN	Applicable for GRU, MLP, and ANN or a combination of them. Deep learning-based techniques showed robust performance on large-scale datasets.	SVM has not worked well when compared with other techniques and utilized 1D-CNN model.
[13]	2021	Adaptive Synthetic Sampling (ADASYN) with RF	Eight datasets using various types of network behaviors are combined in it. SMOTE, RUS, and ADASYN are combined with RF to achieve better predicted results.	Only used machine learning algorithms, the ADASYN sampling technique has shortcomings, generate unrealistic samples, and employed CICIDS 2017 dataset.
[15]	2022	SMOTE and LSH	Efficient optimization voting algorithm data scaling is used to speed up things	This technique does not work well without IOT devices
[19]	2022	LSTM	Used 10-fold cross-validation to ensure that the results were completely unbiased. cuBLSTM and cuGRUDNN are exploited on the same dataset	Hybrid model is not integrated into the NIDS, did not used multi-class classification of attacks, attained best results for binary classification.
[11]	2023	NRS-SSA, NRS, and SSA	Reduced features of datasets. Integrate the classification benefits of heterogeneous and homogeneous classifiers	The single classifier has limited classification performance, limited generalization, neighborhood rough-set and the salp swarm algorithm can be complex.
[12]	2023	SMOTE, XGboot	Most effective and efficient model for network intrusion. This system is compatible with both contemporary network topologies with the NIDS system	Did not apply the ADASYN technique. This method only solved Type-1 and Type-2 issues. The authors applied SMOTE on entire dataset for binary and multi-class, that may cause overfitting and data leakage.
[16]	2023	1D-DCNN	Achieved better precision ratios with malicious attack detection	Processing time and substantial computational resources are required
[17]	2023	CNN	Used more integrated attention mechanism Efficient detection process	Target devices lack enough computing power and storage space.

feature engineering techniques are used to extract the most valuable features, which are essential for the models to perform effectively.

Subsequently, several ML models are employed to detect attacks, including SVM, RF, DT, KNN, logistic regression (LR), and AdaBoost (ADA). Additionally, the LwHM is proposed that combines KNN and DT. Both models are computationally

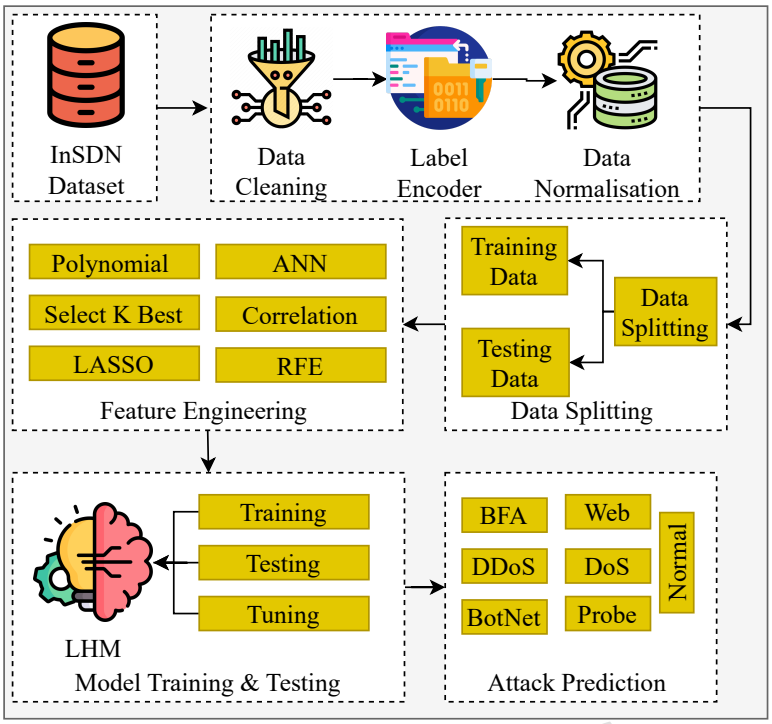

Fig. 1: Workflow of the proposed methodology.

efficient and exhibit good performance. Their diversity in performance characteristics allows us to achieve significant improvements when combining them. Finally, the performance of these models is assessed in terms of accuracy, recall, precision, and F1 score by passing the test data. This comprehensive evaluation process ensures that the models are capable of effectively detecting and classifying attacks.

3.1 Dataset Description

The InSDN dataset is employed for experimentation, which was collected by ElSayed et al.,[29]. This dataset originating from the context of SDN includes SDN-specific threats. The objective is to enhance the capability of ML models to identify SDN attacks, using the specific dataset. The InSDN dataset contains 80 descriptive attributes. It covers seven primary attack types, including brute force attacks (BFA), botnet attacks (Botnet), denial of service attacks (DoS), DDoS, probe attacks (Probe), Web-attack, and normal network traffic. In addition, the normal network traffic covers different important application services like HTTPS, HTTP, SSL, DNS, Email, FTP, SSH, etc. Also, this InSDN data was generated by utilizing four virtual machines. The class distribution of train and test data within the SDN intrusion detection dataset is outlined in Table 2.

Table 2: Attack categories and class distribution for SDN intrusion detection data.

Attacks	Training Data	Testing Data	Total Data
BFA	892	218	1110
BOTNET	137	27	164
DDoS	38656	9757	48413
DoS	42073	10398	52471
Normal	54913	13511	68424
Probe	28896	7476	36372
Web-Attack	149	43	192
Total Data	165716	41430	207146

3.2 Preprocessing

The procedure of data cleaning, encompassing the identification and elimination of missing, duplicated, or irrelevant data instances, stands as a pivotal phase in the ML workflow, as discussed in [29]. In addition to data cleaning, label encoding, as detailed in [30], is a widely adopted technique in the realm of ML. Data preprocessing, comprising data cleaning and label encoding, constitutes the foundational stage in any ML undertaking. The missing values in the dataset are either removed or replaced with appropriate values by imputation. Utilizing feature selection to choose the most suitable features for the proposed model would reduce the number of dimensions and improve the performance of the model. To represent categorical data numerically, it is necessary to employ techniques such as one-hot encoding and label encoding. Removing elements that exhibit substantial correlation or duplication and do not offer any advantages. By following these techniques, data cleaning was performed effectively, a crucial stage in the development of a precise and dependable intrusion model.

3.3 Visualization

The SDN intrusion detection dataset encompasses seven distinct attack classes: BFA, BOTNET, Probe, DDoS, DoS, Web Attack, and Normal attack. We have visualized these attacks in relation to various attributes, including flow duration, total forward packets, total backward packets, total length of forward packets, and total length of backward packets, each associated with its respective class label. Figure 2 illustrates the distribution of attributes for each attack class.

3.4 Feature Engineering Techniques

The presented model uses different feature engineering techniques that are described as follows.

- **Polynomials**: Polynomials are mathematical expressions involving coefficients and indeterminates, which incorporate operations such as addition, multiplication, subtraction, and positive-integer powers for variables. They are widely applied in the fields of science and mathematics [31]. Polynomial functions provide a means to evaluate polynomials. The polynomial creates the features up to the second degree, and advanced bias is not included. When parameter interaction-only is assigned

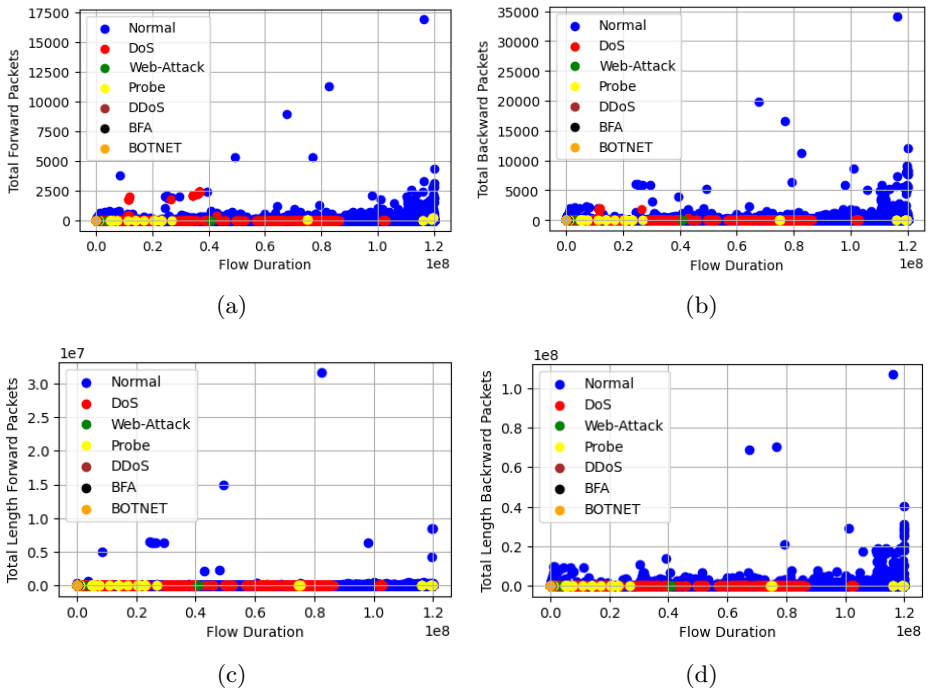

Fig. 2: Flow duration Vs. **(a)** Total forward packets, **(b)** Total backward packets, **(c)** Total length of forward packets, and **(d)** Total length of backward packets.

as True, the function automatically generates the features that are the result of multiple separate features.

- **Select K Best Algorithm**: The Select K Best algorithm is utilized to identify the k-th smallest value (k=50) within an ordered collection of values, such as numbers. This method is commonly employed to solve specific problems like finding the median, minimum, and maximum elements in the collection [32]. The chi-squared scoring function provides the best results.
- **Lasso**: Lasso is a regression analysis technique that combines regularization and variable selection to enhance prediction accuracy and interpretation in statistical approaches. It is particularly useful for improving the predictive power of regression models [33]. We set the 'l2' penalty and the lib-linear solver as a tuning parameter to solve the regression problems.
- **Artificial Neural Networks**: ANN is also utilized in network intrusion detection systems to identify attacks. It is composed of three layers: the input layer, the hidden layer, and the output layer. During the training process, the weights are optimized to reduce the quantity of incorrect classification. They can recognize complex and dynamic patterns with remarkable results, although they require large amounts of data. ANNs are also pivotal in various machine learning and deep learning applications [34].

- **Recursive Feature Elimination**: The RFE technique is employed to select features recursively, starting with the largest feature set and iteratively reducing it. Initially, an estimator is trained using the full feature set, and then features are ranked by their importance. The least important features are iteratively pruned from the current set until the desired number of features is reached [35]. In this study, the fine-tuning is done on the parameters by utilizing the top 50 features.

3.5 Machine Learning Classifiers

ML models refer to computational algorithms that possess the capability to recognize patterns or produce predictions based on datasets that have not been previously seen. The various utilized models are discussed in detail in the following subsections. The hyperparameters for ML classifiers are shown in Table 3.

Table 3: Hyperparameters for ML classifiers.

Model	Hyperparameters	Tuning range
RF	n_estimators=100, max_depth=50, random_state=42	n_estimators= {50 to500}, random_state = {2 to 60},max_depth = {50 to 500}
SVM	kernel='rbf', C=1.0, gamma='scale', random_state=42	Kernal = {'linear', 'poly', 'sigmoid'}, c = {1.5 to 5.0}, random_state = {2 to 60}
DT	max_depth=100, criterion='gini',class_weight="balanced" , random_state=42	random_state = {2 to 60},max_depth = {50 to 500}
NB	Default	max_iter = {3 to 20}, random_state = {2 to 60},
MLP	max_iter=1000, random_state=42	n_estimators= {50 to500}, random_state = {2 to 60},max_depth = {50 to 500}
KNN	n_neighbors=17,weights = 'uniform',leaf_size = 20	n_neighbors = {2 to10}
Light GBM	n_estimators=200, max_depth=3, learning_rate=0.2, random_state=42	n_estimators= {50 to500}, random_state = {2 to 60},max_depth = {50 to 500}
LR	C=3.0, random_state=42, solver=' sag'	random_state = {2 to 60},c = {1.5 to 5.0},solver = { sag, saga, newton-cg, liblinear}, multiclass = { multinomial , one vs rest}

3.5.1 Support Vector Machine

SVM is a versatile ML algorithm that can be applied to both regression and classification tasks [36]. In the context of our study, we have employed the SVM model on the InSDN dataset to assess its performance. This choice is motivated by SVM's widespread use in the domain of network intrusion detection, where it has been extensively studied and recognized for its performance efficiency. The primary objective of the SVM approach is to identify a hyperplane within an N-dimensional space.

This hyperplane serves as a boundary that effectively classifies data points, thereby addressing the problem of separating data in a complex space.

3.5.2 Logistic Regression

LR is employed for solving classification problems, particularly in the context of network intrusion detection [37]. This model effectively captures the relationship between one or more independent variables and specific dependent variables by estimating probabilities through a logistic function. The logistic function, often represented as a sigmoid curve, is defined in Equation 1 as follows:

$$X = \frac{j}{1 + H^{-k(x-x_0)}} \tag{1}$$

In this equation, H represents Euler's number, x_0 signifies the value for the sigmoid midpoint, j is utilized for the curve's maximal value, and k characterizes the curve's steepness.

3.5.3 Random Forest

RF is a tree-based classifier that leverages multiple weak learners to generate accurate predictions[38]. It uses the bagging technique, which involves training multiple decision trees using various bootstrap samples derived from sub-sampling the training dataset with replacement. The sample size remains the same as the training dataset. The RF model can be described using Equation 2 and Equation 3:

$$F = \text{mode } G_1(x), G_2(x), \ldots, G_s(x) \tag{2}$$

$$F = \text{mode } \sum_{m=1}^{n} G_s(x) \tag{3}$$

In these equations, F represents the final prediction based on a majority decision from the ensemble of decision trees, denoted as $G_1(x), G_2(x)$, etc., which contribute to the prediction process.

Our approach applies the RF model with three hyperparameters. We set the n_estimators parameter to 200, indicating that the RF model will consist of 200 decision trees for prediction. Additionally, the max_depth parameter is set to 50, limiting the maximum depth of the DTs to 50 levels to prevent over-complexity and overfitting.

3.5.4 Decision Tree

The DT model is a non-linear supervised classification technique known for its tree-like structure [39]. In DT, the branching points in the tree represent discriminating conditions, and the leaf nodes correspond to categorized records. One of the strengths of DT is its ease of understanding and visualization, making it suitable for a wide range of applications. However, it can be sensitive to data variations, and minor changes in the data can lead to different tree structures. In our approach, we utilize this model with two hyperparameters. One of these hyperparameters, denoted as max_depth and

set to a value of 50, constrains the DT to a maximum depth of 50 levels, which helps avoid excessive complexity.

3.5.5 Naive Bayes

NB classifier is a valuable tool for addressing classification problems, especially in the context of network security analysis [40]. This model is employed for its efficiency in rapid prediction. NB is based on a likelihood probability approach, making predictions based on probability estimates. Given a set of w vectors, $E = e_1, ..., e_w$, and a set of I classes, $G = g_1, ..., g_i$, NB classifiers are used to estimate the probabilities for each class G_d given in Equation 4:

$$P(c_d) = \frac{P(c_d|c_n)P(c_d)}{P(f_1)} \tag{4}$$

In Equation 4, $P(c_d)$ represents the probability, and the vector c_d is randomly selected as its representation, belonging to c_n as indicated in Equation 8. Estimating $c_d|c_n$ is a challenging task. Furthermore, NB operates on the assumption of independence for given values. This approach demonstrates improved performance when compared to other classifiers.

3.5.6 K-Nearest Neighbors

KNN [41] is a machine learning-based algorithm. KNN classifier supposes the similarity between the new data and available data, and also puts the new data into the class that is the same as the available classes. The KNN model is applied for regression and classification. However, it is mostly used to solve classification problems. It is a non-parametric technique, which means it does not consider assumptions on the underlying data. The KNN model at the training phase only stores the database, and when it receives new data, it categorizes that data into a class that is mostly similar to the new data.

3.5.7 Light Gradient Boosting Machine

It is a gradient-boosting model that is based on DTs to improve the effectiveness of the classifier and reduce the usage of memory [42]. Gradient-based one-side sampling(GOSS) and exclusive feature bundling (EFB) are two main techniques of LightGBM. These methods fulfill the constraints of the histogram-based technique that is first applied in all gradient-boosting decision tree (GBDT) models. Moreover, these two techniques work together to achieve efficiency.

3.5.8 Multilayer Perceptron

Multilayer perceptron (MLP) [43] is a feed-forward ANN classifier that maps the input data sets onto appropriate sets of outputs. MLP comprises multiple layers, and every layer is connected to the next one. MLP is used to connect to a neural network. To perform the classification tasks, SVM, NB, and MLP depend on an underlying neural network.

3.5.9 Proposed LightWeight Hybrid Model

The LwHM, a fusion of KNN and DT classifiers, is designed to harness the strengths of both algorithms to make informed predictions. Soft voting is employed, enabling the model to take advantage of class probabilities, resulting in a more robust and well-informed decision-making process as its architecture is shown in Figure 3.

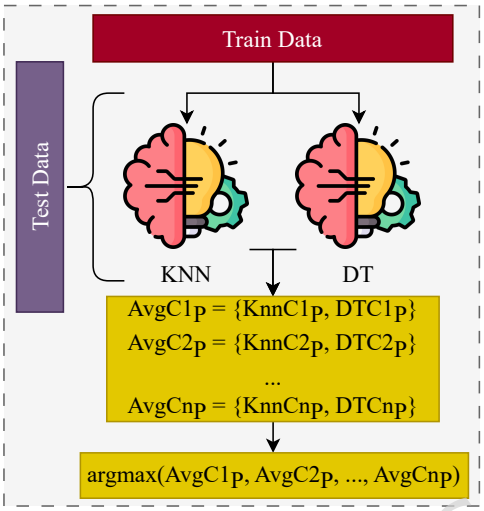

Fig. 3: Proposed LwHM architecture.

Mathematically, soft voting combines the predicted probabilities from multiple base classifiers to determine the final class label for a given input. In this case, we integrate KNN and DT classifiers, which are known for their complementary characteristics. The KNN algorithm relies on the similarity of data points and their neighbors to make predictions, while the DT is a rule-based approach that leverages hierarchical decision structures. By blending these two approaches, the model accounts for both local and global patterns in the data, enhancing its adaptability to diverse scenarios.

The LwHM is particularly useful when dealing with datasets that exhibit complex decision boundaries or a mixture of structured and unstructured data. Its ability to consider probabilities not only enhances the model's accuracy but also introduces a level of confidence in its predictions, making it a valuable tool in various classification tasks. Whether it's in image recognition, text categorization, or any domain where a fusion of local and rule-based reasoning is beneficial, the LwHM demonstrates its versatility and effectiveness.

Soft voting allows the model to consider class probabilities, which we can define mathematically as follows. The final prediction, denoted as y_f for a given input X, is determined as follows:

$$y_f(X) = \arg\max_i \left(\frac{1}{N} \sum_{j=1}^{N} P_i^j \right) \tag{5}$$

Where:

- N is the number of base classifiers (in this case, 2 for KNN and DT).
- P_i^j is the predicted probability that input X belongs to class i by the j-th base classifier.

The KNN algorithm estimates P_i^j based on the distances between the input X and its k-nearest neighbors, typically computed using the Euclidean distance formula:

$$d(X, X') = \sqrt{\sum_{p=1}^{P} (X_p - X'_p)^2} \tag{6}$$

Where:

- X_p and X'_p are the p-th features of input X and a neighbor X'.
- P is the number of features in the input data.

The DT classifier, on the other hand, uses a series of binary decision rules to partition the data. For each decision node i in the tree, the probability P_i^j can be determined based on the number of training samples that belong to class i at that node. This calculation depends on the impurity measure used, such as Gini impurity or entropy, and in our case, it is Gini impurity. By seamlessly blending these classifiers with soft voting, the Lightweight Voting Classifier provides a comprehensive and probabilistic approach to malicious traffic detection in SDN-based networks, enhancing adaptability and accuracy in the ever-changing network security landscape. Algorithm 1 shows the steps in LwHM, and the description of the algorithms is available with algorithm comments.

4 Results

To train ML models for detecting SDN attacks, the newly acquired intrusion detection attack dataset InSDN is used. Six feature engineering approaches are used to extract attack-related features, enabling the detection of various attack types. ML models, and the proposed LwHM, were applied for experiments. The evaluation criteria included accuracy, precision, recall, and F1 score, all computed using confusion matrices. In addition, cross-validation is also applied and per-class performance is also assessed.

4.1 Results Using Extracted Features from ANN and the LASSO Technique

Table 4 presents the results obtained through the utilization of extracted features from ANN and the LASSO approach. Feature selection using LASSO and ANN is widely recognized in ML domain. The performance of various ML models exhibited variability based on the employed feature selection strategy.

Algorithm 1 Algorithm for LwHM

Input:
Training dataset with n features and m samples: $(X_{\text{train}}, Y_{\text{train}})$
Testing dataset with n features: X_{test}
Number of neighbors for KNN: k
Decision Tree model
Output:
Predicted class labels for samples in X_{test}
while not converged **do** ▷ Convergence condition
    Select a sample x_i from the testing dataset ▷ Step 1
    Calculate Euclidean distance to the training samples:
        $D_E = \text{EuclideanDistance}(x_i, X_{\text{train}})$ ▷ Step 2
    Calculate the Manhattan distance to the training samples:
        $D_M = \text{ManhattanDistance}(x_i, X_{\text{train}})$ ▷ Step 2
    Find the k nearest neighbors with KNN using Euclidean distance:
        $N_{E1}, N_{E2}, ..., N_{Ek}$ ▷ Step 3
    Predict class labels using the Decision Tree model:
        $Y_{\text{DT}} = \text{DecisionTreePredict}(x_i)$ ▷ Step 3
    Calculate Class Probabilities using KNN for Euclidean neighbors: ▷ Step 4
        $P_E = \frac{1}{k} \sum_{i=1}^{k} \text{ClassProbability}(N_{Ei})$
    Calculate Class Probabilities using Decision Tree: ▷ Step 4
        $P_{\text{DT}} = \text{ClassProbability}(Y_{\text{DT}})$
    Calculate the final predicted class probabilities using soft voting: ▷ Step 5
        $P_{\text{final}} = \frac{1}{2}(P_E + P_{\text{DT}})$ ▷ Two models: KNN and Decision Tree
    Assign the class label corresponding to the highest probability as the final prediction for x_i: ▷ Step 6
        $\hat{y}_i = \arg\max_c P_{\text{final}}[c]$
end while
Output:
Return the predicted class labels for samples in X_{test}

RF achieved an accuracy of 97% using ANN features and 98% with LASSO features. The superior performance of RF when using LASSO features is attributed to the effectiveness of LASSO in selecting relevant features for this specific dataset. LASSO's ability to perform feature selection and reduce dimensionality likely led to a more focused set of features, improving RF's classification accuracy. SVM also demonstrated improved performance with the LASSO approach compared to ANN, highlighting the suitability of LASSO-selected features for this algorithm. LR, DT, and KNN exhibited consistent levels of accuracy with both feature selection methods.

However, NB showed significantly lower accuracy with an accuracy rate of only 35% using ANN features and 41% with LASSO features. MLP achieved a classification accuracy of 96% with ANN features, surpassing the 90% accuracy obtained with LASSO data. The ANN features outperformed the LASSO method by a 6% increase in accuracy. Notably, except for KNN and MLP, all models exhibited unsatisfactory performance in detecting BFA and BOTNET attacks.

KNN exhibited a remarkable performance, achieving an accuracy of 99.3% with both ANN and LASSO features. KNN's strength lies in its ability to identify patterns and make predictions based on the similarity of data points. The selected features, whether from ANN or LASSO, likely encapsulated key information that made KNN particularly effective in this context.

Table 4: Results using extracted features from ANN and the LASSO techniques.

	ANN Features					LASSO Features			
Model	Accuracy	Class	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
RF	0.978	BFA	0.00	0.00	0.00	0.985	0.00	0.00	0.00
		BOTNET	0.00	0.00	0.00		0.00	0.00	0.00
		DDoS	1.00	1.00	1.00		1.00	1.00	1.00
		DoS	0.97	0.99	0.98		0.98	0.98	0.98
		Normal	0.99	0.96	0.98		0.98	1.00	0.99
		Probe	0.94	0.99	0.96		0.99	0.99	0.99
		Web-Attack	0.00	0.00	0.00		0.00	0.00	0.00
		macro avg	0.56	0.56	0.56		0.56	0.57	0.57
SVM	0.594	BFA	0.00	0.00	0.00	0.589	0.00	0.00	0.00
		BOTNET	0.00	0.00	0.00		0.00	0.00	0.00
		DDoS	0.99	0.98	0.98		0.96	0.97	0.98
		DoS	0.38	1.00	0.55		0.39	0.99	0.52
		Normal	0.98	0.35	0.52		0.97	0.35	0.52
		Probe	1.00	0.00	0.00		0.99	0.00	0.00
		Web-Attack	0.00	0.00	0.00		0.00	0.00	0.00
		macro avg	0.48	0.33	0.29		0.47	0.35	0.25
LR	0.574	BFA	0.00	0.00	0.00	0.577	0.00	0.00	0.00
		BOTNET	0.00	0.00	0.00		0.00	0.00	0.00
		DDoS	0.76	1.00	0.86		0.76	1.00	0.87

Continued on next page

Table 4 – continued from previous page

	ANN Features					LASSO Features			
Model	Accuracy	Class	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
		DoS	0.81	0.07	0.13		0.21	0.10	0.14
		Normal	0.48	0.98	0.64		0.56	0.97	0.71
		Probe	0.92	0.00	0.00		0.07	0.00	0.01
		Web-Attack	0.00	0.00	0.00		0.00	0.00	0.00
		macro avg	0.36	0.29	0.24		0.23	0.30	0.25
KNN	0.993	BFA	0.83	0.89	0.86	0.993	0.81	0.67	0.73
		BOTNET	1.00	1.00	1.00		0.93	1.00	0.96
		DDoS	1.00	1.00	1.00		1.00	1.00	1.00
		DoS	1.00	0.99	1.00		1.00	0.99	1.00
		Normal	1.00	1.00	1.00		1.00	1.00	1.00
		Probe	1.00	1.00	1.00		0.99	1.00	1.00
		Web-Attack	0.52	0.28	0.36		0.56	0.12	0.19
		macro avg	0.91	0.88	0.89		0.90	0.82	0.84
DT	0.326	BFA	0.00	0.00	0.00	0.326	0.00	0.00	0.00
		BOTNET	0.00	0.00	0.00		0.00	0.00	0.00
		DDoS	0.00	0.00	0.00		0.00	0.00	0.00
		DoS	0.00	0.00	0.00		0.00	0.00	0.00
		Normal	0.33	1.00	0.49		0.33	1.00	0.49
		Probe	0.00	0.00	0.00		0.00	0.00	0.00

Continued on next page

Table 4 – continued from previous page

	ANN Features					LASSO Features			
Model	Accuracy	Class	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
		Web-Attack	0.00	0.00	0.00		0.00	0.00	0.00
		macro avg	0.05	0.14	0.07		0.05	0.14	0.07
NB	0.357	BFA	0.02	0.93	0.03	0.419	0.02	0.90	0.04
		BOTNET	0.00	0.96	0.00		0.00	1.00	0.01
		DDoS	1.00	1.00	1.00		0.92	1.00	0.96
		DoS	0.84	0.07	0.12		0.85	0.06	0.11
		Normal	0.94	0.30	0.46		0.94	0.30	0.46
		Probe	0.30	0.01	0.02		0.16	0.03	0.05
		Web-Attack	0.00	0.00	0.00		0.00	0.00	0.00
		macro avg	0.44	0.47	0.23		0.41	0.50	0.26
MLP	0.968	BFA	0.88	0.48	0.85	0.901	0.97	0.34	0.51
		BOTNET	1.00	0.44	0.77		0.86	0.93	0.89
		DDoS	0.99	1.00	1.00		1.00	1.00	1.00
		DoS	0.90	0.75	0.94		0.74	0.99	0.84
		Normal	1.00	1.00	0.99		0.83	1.00	0.91
		Probe	1.00	0.99	0.96		0.97	0.97	0.97
		Web-Attack	0.95	0.02	0.58		0.00	0.00	0.00
		macro avg	0.96	0.82	0.87		0.79	0.71	0.72
		BFA	0.00	0.00	0.00		0.53	0.31	0.39

Continued on next page

LightGBM	0.920		0.619

Table 4 – continued from previous page

	ANN Features					LASSO Features			
Model	Accuracy	Class	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
		BOTNET	0.00	0.00	0.00		0.00	0.00	0.00
		DDoS	0.97	0.94	0.92		0.00	0.00	0.00
		DoS	0.95	0.89	0.95		0.83	0.78	0.81
		Normal	0.95	0.95	0.95		0.77	0.82	0.82
		Probe	0.94	0.94	0.94		0.90	0.76	0.82
		Web-Attack	0.01	0.09	0.01		0.00	0.00	0.00
		macro avg	0.54	0.54	0.54		0.43	0.39	0.40

4.2 Results Using Extracted Features from RFE and Correlation-based Technique

Table 5 presents the results obtained through the utilization of features extracted using the RFE approach and features derived from the correlation matrix. RF achieved an accuracy of 98.9% when utilizing RFE features and 98.5% when using features derived from the Correlation matrix. The difference in performance arises from the effectiveness of the RFE approach in selecting the most relevant features for the RF model. RFE systematically eliminates less informative features, resulting in a more focused set of attributes for classification. SVM demonstrated superior performance when utilizing the RFE method compared to the Correlation-based feature selection. RFE's ability to identify the most informative features aligns well with SVM's classification approach, resulting in better accuracy with this combination.

LR, DT, and KNN achieved similar levels of accuracy with both RFE and Correlation-based features. This consistency suggests that these models are less sensitive to the feature selection method and perform well with either approach. In contrast, NB showed subpar performance with both RFE and Correlation features, indicating that the independence assumptions of NB may not align with the selected features, resulting in lower accuracy.

MLP displayed strong performance with both RFE and Correlation features, with an accuracy of 97.6% and 97.1%, respectively. The similarity in performance between these two feature selection methods suggests that MLP can effectively utilize a wide range of features. RFE's systematic feature elimination process proved advantageous for RF, SVM, and MLP, while LR, DT, and KNN exhibited robust performance across both RFE and correlation-based feature sets. NB's suboptimal performance indicates a lack of feature-model alignment.

Table 5: Results using extracted features from RFE and correlation-based features.

	RFE Features					Correlation-based Features			
Model	Accuracy	Class	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
RF	0.989	BFA	0.00	0.00	0.00	0.985	0.00	0.00	0.00
		BOTNET	0.00	0.00	0.00		0.00	0.00	0.00
		DDoS	1.00	1.00	1.00		1.00	1.00	1.00
		DoS	0.97	1.00	0.98		0.98	0.98	0.98
		Normal	1.00	0.98	1.00		0.98	1.00	0.99
		Probe	1.00	0.99	1.00		0.99	0.99	0.99
		Web-Attack	0.00	0.00	0.00		0.00	0.00	0.00
		macro avg	0.57	0.57	0.57		0.56	0.57	0.57
SVM	0.573	BFA	0.00	0.00	0.00	0.589	0.00	0.00	0.00
		BOTNET	0.00	0.00	0.00		0.00	0.00	0.00
		DDoS	0.96	0.97	0.98		0.96	0.94	0.91
		DoS	0.39	0.97	0.51		0.39	0.93	0.52
		Normal	0.97	0.35	0.50		0.97	0.35	0.52
		Probe	0.99	0.00	0.00		0.99	0.00	0.00
		Web-Attack	0.00	0.00	0.00		0.00	0.00	0.00
		macro avg	0.45	0.35	0.23		0.44	0.35	0.23
LR	0.578	BFA	0.00	0.00	0.00	0.577	0.00	0.00	0.00
		BOTNET	0.00	0.00	0.00		0.00	0.00	0.00
		DDoS	0.76	1.00	0.86		0.76	1.00	0.87

Continued on next page

Table 5 – continued from previous page

	ANN Features					LASSO Features			
Model	Accuracy	Class	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
		DoS	0.86	0.09	0.16		0.21	0.10	0.14
		Normal	0.48	0.99	0.65		0.56	0.97	0.71
		Probe	0.37	0.00	0.00		0.07	0.00	0.01
		Web-Attack	0.00	0.00	0.00		0.00	0.00	0.00
		macro avg	0.35	0.30	0.24		0.23	0.30	0.25
KNN	0.993	BFA	0.81	0.88	0.84	0.992	0.82	0.63	0.70
		BOTNET	1.00	1.00	1.00		0.93	1.00	0.96
		DDoS	1.00	1.00	1.00		1.00	1.00	1.00
		DoS	1.00	0.99	1.00		0.98	0.99	1.00
		Normal	1.00	1.00	1.00		1.00	1.00	1.00
		Probe	1.00	1.00	1.00		0.99	0.99	1.00
		Web-Attack	0.44	0.42	0.18		0.56	0.12	0.19
		macro avg	0.88	0.85	0.85		0.89	0.81	0.84
DT	0.326	BFA	0.00	0.00	0.00	0.326	0.00	0.00	0.00
		BOTNET	0.00	0.00	0.00		0.00	0.00	0.00
		DDoS	0.00	0.00	0.00		0.00	0.00	0.00
		DoS	0.00	0.00	0.00		0.00	0.00	0.00
		Normal	0.33	1.00	0.49		0.33	1.00	0.49
		Probe	0.00	0.00	0.00		0.00	0.00	0.00

Continued on next page

Table 5 – continued from previous page

	ANN Features					LASSO Features			
Model	Accuracy	Class	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
		Web-Attack	0.00	0.00	0.00		0.00	0.00	0.00
		macro avg	0.05	0.16	0.07		0.05	0.14	0.07
NB	0.372	BFA	0.04	0.90	0.04	0.383	0.02	0.90	0.04
		BOTNET	0.01	0.97	0.00		0.00	0.99	0.01
		DDoS	1.00	0.99	1.00		0.91	0.95	0.95
		DoS	0.86	0.04	0.10		0.82	0.06	0.11
		Normal	0.91	0.29	0.43		0.91	0.35	0.46
		Probe	0.32	0.01	0.02		0.16	0.04	0.07
		Web-Attack	0.00	0.00	0.00		0.00	0.00	0.00
		macro avg	0.43	0.44	0.21		0.45	0.47	0.25
MLP	0.976	BFA	0.88	0.48	0.85	0.971	0.97	0.34	0.51
		BOTNET	1.00	0.44	0.77		0.86	0.93	0.09
		DDoS	0.99	1.00	1.00		1.00	1.00	1.00
		DoS	0.90	0.75	0.94		0.97	0.91	0.89
		Normal	0.99	1.00	0.99		0.99	1.00	0.99
		Probe	1.00	0.99	0.96		0.97	0.97	0.97
		Web-Attack	0.95	0.23	0.58		0.00	0.00	0.00
		macro avg	0.95	0.86	0.87		0.96	0.75	0.79
		BFA	0.00	0.00	0.00		0.53	0.31	0.39

Continued on next page

LightGBM	0.886					0.853			

Table 5 – continued from previous page

	ANN Features					LASSO Features			
Model	Accuracy	Class	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
		BOTNET	0.00	0.00	0.00		0.00	0.00	0.00
		DDoS	0.97	0.94	0.92		0.00	0.00	0.00
		DoS	0.91	0.90	0.93		0.83	0.78	0.81
		Normal	0.92	0.99	0.95		0.77	0.86	0.82
		Probe	0.91	0.90	0.94		0.93	0.76	0.84
		Web-Attack	0.02	0.01	0.05		0.00	0.00	0.00
		macro avg	0.52	0.56	0.53		0.44	0.37	0.41

4.3 Results Using Extracted Features from SelectKBest and Polynomial Features

Table 6 presents the results of various ML models when utilizing two different feature selection methods: SelectKBest features and Polynomial features. The results are displayed in terms of accuracy, precision, recall, and F1 score.

RF using SelectKBest features, achieved an accuracy of 97.8%. It displayed excellent performance in classifying DDoS, DoS, and Normal data, with F1 scores close to 1.0, indicating high precision and recall for these classes. However, it performed poorly in classifying BFA, BOTNET, and Web-Attack, as indicated by zero precision, recall, and F1 score. The macro average (average across all classes) F1 score for RF was 0.57. KNN achieved a high accuracy of 99.3% with SelectKBest features. It excelled in classifying BFA, BOTNET, DDoS, DoS, Normal, and Probe, with F1 Scores close to 1.0. However, it performed poorly in classifying Web-Attack, as indicated by a lower F1 score. The macro average F1 score for KNN was 0.85. The results for the Polynomial features are displayed in the same manner as SelectKBest features, showing accuracy, precision, recall, and F1 score for each model across different classes.

Table 6: Results using extracted features from SelectKBest and Polynomial features.

	SelectKBest Features					Polynomial Features			
Model	Accuracy	Class	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
RF	0.978	BFA	0.00	0.00	0.00	0.990	0.00	0.00	0.00
		BOTNET	0.00	0.00	0.00		0.00	0.00	0.00
		DDoS	1.00	1.00	1.00		1.00	1.00	1.00
		DoS	0.97	1.00	0.98		0.98	0.98	0.98
		Normal	1.00	0.98	1.00		0.98	1.00	0.99
		Probe	1.00	0.99	1.00		0.99	0.99	0.99
		Web-Attack	0.00	0.00	0.00		0.00	0.00	0.00
		macro avg	0.57	0.57	0.57		0.56	0.57	0.57
SVM	0.562	BFA	0.00	0.00	0.00	0.574	0.00	0.00	0.00
		BOTNET	0.00	0.00	0.00		0.00	0.00	0.00
		DDoS	0.96	0.97	0.98		0.96	0.94	0.91
		DoS	0.39	0.97	0.51		0.39	0.93	0.52
		Normal	0.97	0.35	0.50		0.97	0.35	0.52
		Probe	0.99	0.00	0.00		0.99	0.00	0.00
		Web-Attack	0.00	0.00	0.00		0.00	0.00	0.00
		macro avg	0.45	0.35	0.23		0.44	0.35	0.23
LR	0.576	BFA	0.00	0.00	0.00	0.01	0.01	1.00	0.01
		BOTNET	0.00	0.00	0.00		0.00	0.00	0.00
		DDoS	0.76	1.00	0.86		0.00	0.00	0.00

Continued on next page

Table 6 – continued from previous page

	ANN Features					LASSO Features			
Model	Accuracy	Class	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
		DoS	0.86	0.09	0.16		0.00	0.00	0.00
		Normal	0.48	0.99	0.65		0.00	0.00	0.00
		Probe	0.37	0.00	0.00		0.00	0.00	0.00
		Web-Attack	0.00	0.00	0.00		0.00	0.00	0.00
		macro avg	0.35	0.30	0.24		0.00	0.14	0.00
KNN	0.993	BFA	0.81	0.88	0.84	0.992	0.82	0.63	0.70
		BOTNET	1.00	1.00	1.00		0.93	1.00	0.96
		DDoS	1.00	1.00	1.00		1.00	1.00	1.00
		DoS	1.00	0.99	1.00		0.98	0.99	1.00
		Normal	1.00	1.00	1.00		1.00	1.00	1.00
		Probe	1.00	1.00	1.00		0.99	0.99	1.00
		Web-Attack	0.44	0.42	0.18		0.56	0.12	0.19
		macro avg	0.88	0.85	0.85		0.89	0.81	0.84
DT	0.326	BFA	0.00	0.00	0.00	0.326	0.00	0.00	0.00
		BOTNET	0.00	0.00	0.00		0.00	0.00	0.00
		DDoS	0.00	0.00	0.00		0.00	0.00	0.00
		DoS	0.00	0.00	0.00		0.00	0.00	0.00
		Normal	0.33	1.00	0.49		0.33	1.00	0.49
		Probe	0.00	0.00	0.00		0.00	0.00	0.00

Continued on next page

Table 6 – continued from previous page

	ANN Features					LASSO Features			
Model	Accuracy	Class	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
		Web-Attack	0.00	0.00	0.00		0.00	0.00	0.00
		macro avg	0.05	0.16	0.07		0.05	0.14	0.07
NB	0.57	BFA	0.04	0.90	0.04	0.203	0.01	1.00	0.01
		BOTNET	0.01	0.97	0.00		0.00	0.00	0.00
		DDoS	1.00	0.99	1.00		0.98	0.57	0.72
		DoS	0.86	0.04	0.10		0.36	0.05	0.09
		Normal	0.91	0.29	0.43		0.95	0.16	0.27
		Probe	0.32	0.01	0.02		0.00	0.00	0.00
		Web-Attack	0.00	0.00	0.00		0.00	0.00	0.00
		macro avg	0.43	0.44	0.21		0.33	0.25	0.16
MLP	0.934	BFA	0.88	0.48	0.85	0.962	0.63	0.47	0.54
		BOTNET	1.00	0.44	0.77		1.00	1.00	1.00
		DDoS	0.99	1.00	1.00		1.00	1.00	1.00
		DoS	0.90	0.75	0.94		0.99	0.87	0.93
		Normal	0.99	1.00	0.99		0.91	1.00	0.95
		Probe	1.00	0.99	0.96		1.00	0.99	0.99
		Web-Attack	0.95	0.23	0.58		0.16	0.07	0.10
		macro avg	0.95	0.86	0.87		0.81	0.77	0.79
		BFA	0.00	0.00	0.00		0.26	0.37	0.31

Continued on next page

LightGBM	0.874		0.978

Table 6 – continued from previous page

	ANN Features					LASSO Features			
Model	Accuracy	Class	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
		BOTNET	0.00	0.00	0.00		1.00	0.34	0.45
		DDoS	0.97	0.94	0.92		0.98	0.99	0.99
		DoS	0.91	0.90	0.93		0.89	0.78	0.81
		Normal	0.92	0.99	0.95		0.99	0.86	0.82
		Probe	0.91	0.90	0.94		0.99	0.99	0.99
		Web-Attack	0.02	0.01	0.05		0.02	0.05	0.03
		macro avg	0.79	0.81	0.82		0.65	0.68	0.66

4.4 Results of Proposed Lightweight Hybrid Model

This section showcases the findings of the LwHM's effectiveness in the context of intrusion detection and offers a comparative analysis of its performance using supervised models and feature selection techniques.

The LwHM demonstrated exceptional performance across various evaluation criteria as shown in Table 7. Its utilization presents an effective solution to overcome the inherent limitations of single models in intrusion detection. Notably, within the domain of Web-attacks, the LwHM exhibited significantly improved performance in terms of accuracy, recall, and F1 score. When the LwHM leveraged RFE techniques, it achieved outstanding results. The system achieved a comprehensive accuracy rate of 99.93%, showcasing impeccable precision, recall, and F1 scores for many categories of attacks, including BOTNET, DDoS, DoS, Normal, and Probe. Furthermore, the LwHM's performance using the RFE method attained a noteworthy accuracy rate of 98%. In the classification of online attacks, the proposed Lightweight Model demonstrated a remarkable 100% accuracy rate when employing correlation-based features. Overall, the LwHM exhibits outstanding performance with diverse feature selection techniques, underscoring its adaptability and effectiveness in intrusion detection, and positioning it as a robust alternative to traditional methods.

The hybrid approach of the LwHM, which leverages both KNN and DT, enables it to achieve superior performance when compared to individual models or single feature selection techniques. By averaging the probabilities obtained from these two models, the LwHM delivers highly accurate predictions, making it a robust and effective solution for intrusion detection.

Table 7: Results of the proposed LwHM using various feature selection techniques.

Feature	Accuracy	Class	Precision	Recall	F1 score
ANN	0.9992	BFA	0.98	0.98	0.98
		BOTNET	1.00	1.00	1.00
		DDoS	1.00	1.00	1.00
		DoS	1.00	1.00	1.00
		Normal	1.00	1.00	1.00
		Probe	1.00	1.00	1.00
		Web-Attack	0.98	0.98	0.98
		macro avg	0.99	0.99	0.99
LASSO	0.9978	BFA	0.88	0.86	0.87
		BOTNET	1.00	1.00	1.00
		DDoS	1.00	1.00	1.00
		DoS	1.00	1.00	1.00
		Normal	1.00	1.00	1.00

Continued on next page

Table 7 – continued from previous page

Feature	Accuracy	Class	Precision	Recall	F1 score
		Probe	1.00	1.00	1.00
		Web-Attack	0.86	0.70	0.76
		macro avg	0.96	0.94	0.95
RFE	0.9993	BFA	0.98	0.99	0.99
		BOTNET	1.00	1.00	1.00
		DDoS	1.00	1.00	1.00
		DoS	1.00	1.00	1.00
		Normal	1.00	1.00	1.00
		Probe	1.00	1.00	1.00
		Web-Attack	0.98	0.98	0.98
		macro avg	0.99	1.00	0.99
Correlation	0.9990	BFA	0.97	1.00	0.98
		BOTNET	1.00	1.00	1.00
		DDoS	1.00	1.00	1.00
		DoS	1.00	1.00	1.00
		Normal	1.00	1.00	1.00
		Probe	1.00	1.00	1.00
		Web-Attack	1.00	0.98	0.99
		macro avg	1.00	1.00	1.00
SelectKBest	0.9992	BFA	0.98	0.99	0.99
		BOTNET	1.00	1.00	1.00
		DDoS	1.00	1.00	1.00
		DoS	1.00	1.00	1.00
		Normal	1.00	1.00	1.00
		Probe	1.00	1.00	1.00
		Web-Attack	0.98	0.95	0.96
		macro avg	0.99	0.99	0.99
Polynomial	0.9992	BFA	0.97	0.98	0.97
		BOTNET	1.00	1.00	1.00
		DDoS	1.00	1.00	1.00
		DoS	1.00	1.00	1.00

Continued on next page

Table 7 – continued from previous page

Feature	Accuracy	Class	Precision	Recall	F1 score
		Normal	1.00	1.00	1.00
		Probe	1.00	1.00	1.00
		Web-Attack	0.98	0.93	0.95
		Macro avg	0.99	0.99	0.99

4.5 LwHM Results Representation using Confusion Matrices

A confusion matrix is a valuable tool used to assess the performance of a classifier by presenting a detailed breakdown of its predictions, including both accurate and inaccurate ones. In the context of our proposed method, the confusion matrix results using different feature sets provide insights into the model's prediction accuracy. When employing ANN features (Figure 4a), the model exhibits slightly lower accuracy with 97.7% for Web-attack predictions and 98.2% for BFA attacks, while achieving 100% accuracy for all other targets.

On the other hand, when using LASSO features (Figure 4b), the model's performance is comparatively lower, with 69.8% accuracy for Web-attack predictions and 85.8% for BFA attacks. However, it demonstrates better results for other target categories. Utilizing RFE features (Figure 4c), the model showcases high accuracy, achieving 97.7% for Web-attack and an impressive 99.1% for BFA attacks. Similarly, correlation-based features (Figure 4d) yield excellent results, with 97.7% accuracy for Web-attack and 99.5% for BFA attacks. SelectKBest features (Figure 4e) also deliver strong performance, with 95.3% accuracy for Web-attack and 99.1% for BFA attacks.

However, employing polynomial features (Figure 4f) results in a lower accuracy of 93.0% for Web-attack predictions and 97.7% for BFA attacks, which is comparatively poorer compared to other attacks. These confusion matrix results offer a comprehensive view of the model's performance across different feature sets, aiding in the evaluation and selection of the most effective approach for intrusion detection. In summary, the models exhibit some variance in their performance, with BFA and Web-attack predictions showing slightly lower accuracy compared to other attack categories.

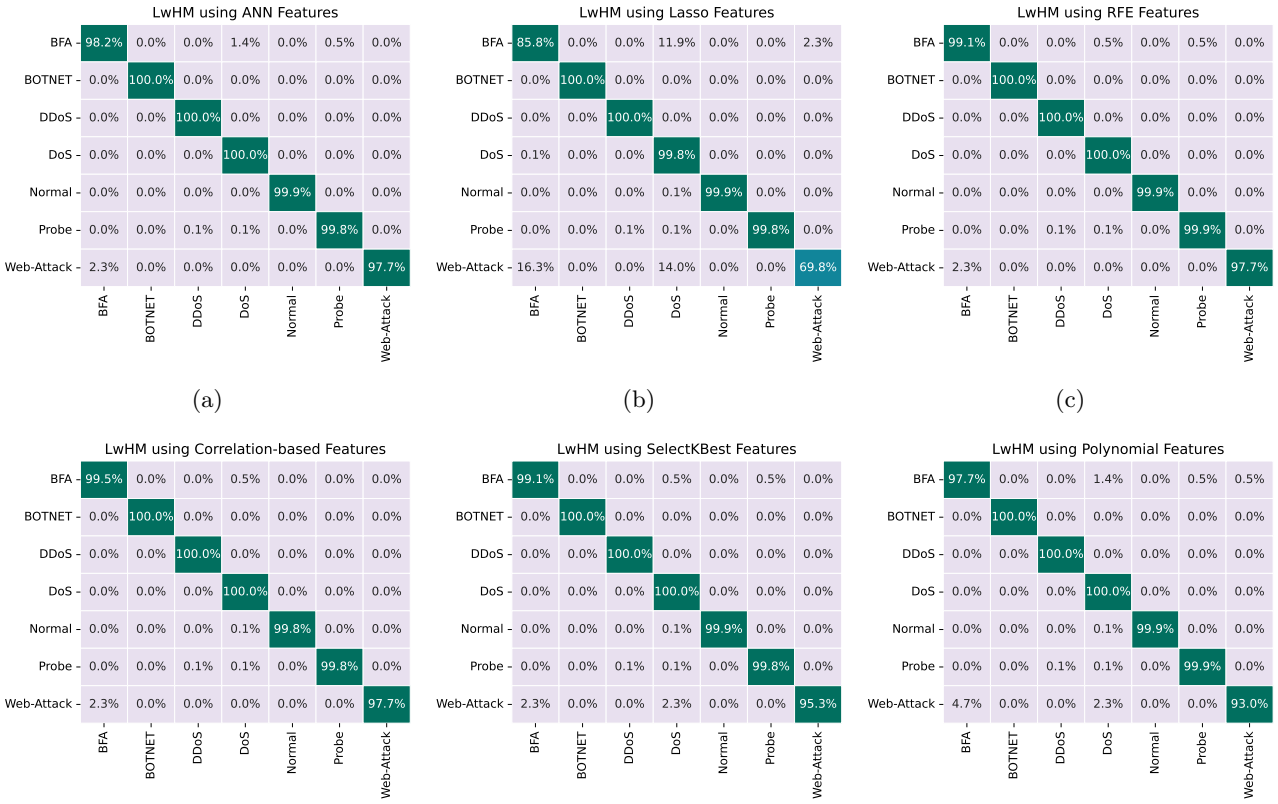

Fig. 4: Confusion matrices for the proposed LwHM using, (**a**) ANN features, (**b**) LASSO features, (**c**) RFE features, (**d**) Correlation-based features, (**e**) SelectKBest features, and (**f**) Polynomial features.

Figure 5 visualizes the overall accuracy of LwHM using different feature selection techniques. The LwHM exhibited superior performance while utilizing RFE features. The LwHM demonstrated a minimum accuracy of 99.78% when employing the LASSO method, while the highest accuracy was attained utilizing the RFE technique.

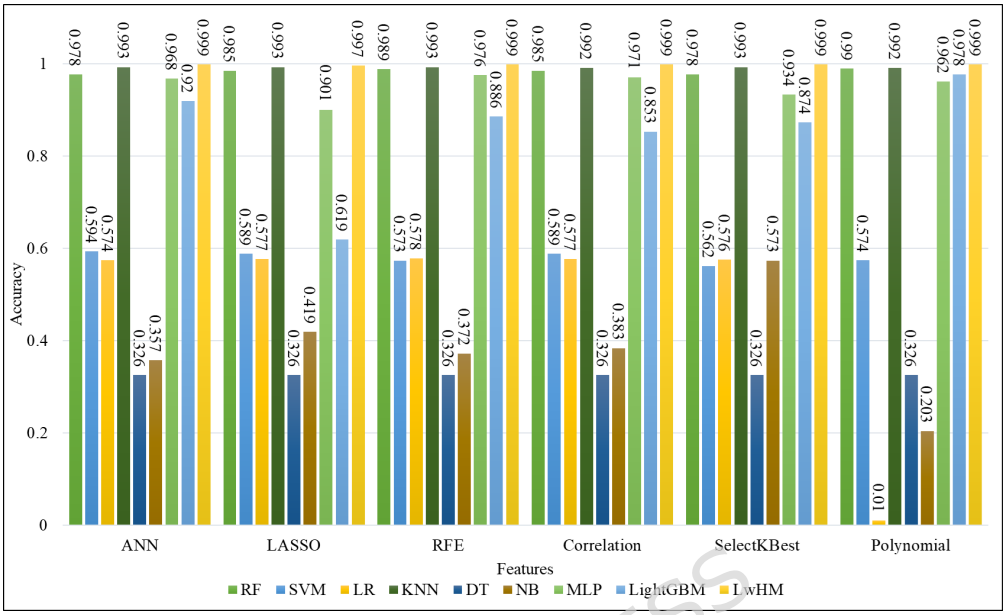

Fig. 5: Overall accuracy of LwHM using different feature selection techniques.

4.6 K-fold Cross-validation Results

One widely adopted approach to validate ML model's performance is k-fold cross-validation. In this approach, the training dataset is divided into 'K' smaller subsets for cross-validation. The dataset is partitioned into 'K' subsets, and the model is trained using 'K-1' of these subsets while excluding one subset in each iteration [44]. This process is repeated 'K' times. The excluded subset is referred to as the validation dataset. The k-fold cross-validation technique is a powerful tool that serves multiple purposes. It allows for the manipulation of hyperparameters, assessing the model's effectiveness, and fine-tuning configurations to enhance performance [44, 45]. Moreover, it imparts strong generalization capabilities, improving the robustness and efficiency of machine learning models when applied to new data.

Table 8 shows the results of k-fold cross-validation experiments for the proposed LwHM. Each technique, such as ANN, LASSO, RFE, correlation, SelectKBest, and Polynomial, is evaluated based on its mean accuracy, which measures how consistently the model makes correct predictions, and the standard deviation, reflecting the degree of accuracy variation across different validation sets. The high mean accuracy

values, consistently close to or above 99%, indicate that the model excels at making accurate predictions across all techniques. Moreover, the minimal standard deviations, mostly around 0.01 or lower, signify that the model's performance remains stable and dependable, strengthening its suitability for robust intrusion detection applications.

Table 8: K-fold cross-validation results using LwHM.

Technique	Mean accuracy	Standard deviation
ANN	99.8	+/- 0.00
LASSO	99.5	+/- 0.02
RFE	99.9	+/- 0.00
Correlation	99.6	+/- 0.01
SelectKBest	99.8	+/- 0.00
Polynomial	99.9	+/- 0.01

4.7 Computational Complexity of LwHM

The proposed LwHM combines KNN and DT models. Time complexity of KNN for training is $o(1)$ as it requires no explicit training while for the prediction phase it takes $o(n \times d)$ as it computes distance between the given sample and all n training samples. For DT, training time complexity is $o(x \times d \times log\ n)$ while prediction phase complexity is $o(long\ n)$. For the hybrid LwHM model, the complexity is $O(n \cdot d \cdot log\ n)$ for training and $O(n \cdot d)$ for prediction.

4.8 Comparison of Proposed LwHM with Other State-of-the-Art Approaches

In this section, the performance of the proposed LwHM is compared with other state-of-the-art methods to evaluate its efficacy and performance. All selected studies work on the same dataset, which is used in this study for experiments. For example, the study [13] employed network detection classifiers trained using the RF method for intrusion detection on an 11-feature dataset utilizing ADASYN. The experiment achieved an impressive accuracy of 98.5%. The study [14] employed a supervised learning approach using a 1D-CNN model for intrusion detection, combining max-pooling with convolutional layers to reduce the computational cost, increase output size, and raise feature count. This model was compared against RF, SVM, LSTM, and a hybrid system using both balanced and unbalanced training datasets. In study [16], metaheuristic optimization and machine learning were utilized to develop a voting classifier. The Whale Optimization Algorithm and Dipper Throated Optimizer, metaheuristic optimizers, were used to build a reliable voting classifier for network attacks. The study [17] employed a 1D-DCNN for attack classification as an intrusion detection system, achieving superior results compared to contemporary deep learning approaches. In study [20], a CNN architecture was utilized within the RNN framework to detect and analyze SDN attacks.

The authors employed DT-ML-based ensemble learning for the classification of DDoS attacks in SDN-based system and achieved 95.2% accuracy [46]. The authors used an ensemble DL-based method for the detection of DDoS attacks in intrusion detection system with superior performance [47]. Nalayini et al.,[48] utilized adaptive transformer for efficient detection of intrusion with SDN environment. Also, transformer with the combination of DL based models was adopted by the Wang et al., [49] for attack detection using hybrid technique. While these state-of-the-art studies employ complex architectures and require substantial training data and effort, our LwHM stands out for its simplicity, efficiency, and exceptional results for intrusion detection in SDN. The proposed approach outperforms all others not only in terms of accuracy but also across all other evaluation matrices, where other studies often exhibit poorer results in areas such as recall, precision, and F1 score, as shown in Table 9.

Table 9: Comparison of proposed approach with other state-of-the-art approaches.

Ref.	Year	Methods	Results
[13]	2021	RF	98.5% precision, 92.3% recall, 95.3% f1 score
[14]	2020	1D CNN	91.2% accuracy, 87.5% precision, 96.1% recall, 91.5% f1 score
[15]	2023	RF	95.1% AUC
[16]	2022	1D-DCNN	99.7% accuracy
[17]	2023	CNN	99.3% accuracy, 62% precision
[20]	2019	SVM	95.7% accuracy
[50]	2023	LightGBM	98.7% accuracy, 97.4% precision, 97.9% recall, 98.2% f1 score
[51]	2018	GRU-RNN	89% accuracy
[46]	2024	Ensemble model	95.2% accuracy on their own dataset
[47]	2025	Ensemble Model	99.8% accuracy
[48]	2025	Adaptive transformer	99.3% overall accuracy.
[49]	2021	Transformer+DL	99.8% accuracy.
Proposed	2025	LwHM + RFE	99.9% accuracy, 99.9% precision, 100% recall, 99% f1 score

4.9 Discussion

The existing datasets are observed to contain numerous redundant or ineffective features, hampering the classifiers' ability to make precise predictions, especially in the context of multi-class data. To address these limitations, this study introduces the LwHM, combining KNN and DT to enhance the classification performance of intrusion detection models.

During the experiments, individual ML models are combined with these feature selection strategies. Notably, when ANN features are used alongside the KNN model, an accuracy of 99.3% is achieved. However, the ANN-KNN model exhibits lower precision, recall, and F1 scores for both BFA and Web-attacks. Furthermore, the RF model shows a recall of 28% in the Web-attack class, indicating subpar performance. Comparatively, using LASSO, the RF model demonstrates accuracy on par with ANN features but a lower recall (12%) compared to ANN. The DT model yields low accuracy across various feature selection approaches.

However, it is essential to highlight that in the previous approaches, ML models demonstrated strong performance in terms of accuracy metrics; however, other evaluation metrics reveal poorer outcomes because of imbalanced datasets. Consequently, the proposed LwHM, integrating KNN and DT, emerges as a superior approach, addressing the limitations of single models. The LwHM excels in precision, recall, and F1 score, particularly in the realm of Web-attacks. The model's utilization of RFE features stands out, achieving an overall accuracy of 99.93% and perfect precision, recall, and F1 scores for various attack categories, including BOTNET, DDoS, DoS, Normal, and Probe. This signifies a significant performance improvement, with RFE features reaching an accuracy rate of 98%. Leveraging correlation-based features, the model attains higher accuracy in the detection of Web-attacks as well. The proposed framework does not rely only on KNN; it employs a lightweight hybrid model combined with feature selection strategies that reduce the dimensionality and decrease the computational cost. In addition, the reported values for individual ML models are 0.00 because individual models failed to correctly identify any instances. This was addressed using the lightweight hybrid approach and achieved substantially better performance for certain attacks.

5 Conclusion

This study addresses the vulnerability of SDNs to various attacks by introducing a novel approach that leverages a hybrid machine learning mechanism. The methodology begins with the extraction of significant features using prominent feature engineering techniques, including LASSO, ANN, Polynomial, SelectKBest, and RFE, among others. Following feature extraction, preprocessing is employed to clean and normalize the data. Subsequently, the LwHM is trained for attack detection. Empirical investigations validate the effectiveness of our method in swiftly and effectively detecting and mitigating attacks in SDN environments.

Through extensive experiments, it is found that feature engineering plays a crucial role in enhancing performance, particularly RFE for network datasets. Its iterative process makes it robust against noisy and irrelevant features, effectively reducing the impact of noisy data. Additionally, the findings suggest that ensemble models with diverse natures can yield significant results. The proposed LwHM achieved an impressive 99.93% accuracy by utilizing DT and KNN outperforming existing approaches. Furthermore, the conclusions emphasize that proper feature engineering and model selection, tailored to the data's nature, can lead to good results with minimal computational cost, avoiding the complexities associated with deep learning models.

Despite the significance of this study, it has some limitations. First, it is important to note that this study exclusively explores a specific hybrid technique. Second, the InSDN dataset exhibit class imbalance and experiments are conducted on original data to reflect realistic network traffic conditions. Future research can delve into additional hybrid approaches, such as weighted voting and stacking generalization, to further enhance results. Secondly, the use of an imbalanced dataset can lead to potential

issues with model overfitting. In the future work, we aim to address data balancing to mitigate the risk of overfitting.

Declarations

Funding

This study is funded by the European University of Atlantic and the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2026R746),Princess Nourah bint Abdulrahman University, Riyadh,Saudi Arabia.

Conflict of Interests

The authors declare no conflict of interest.

Availability of Data

The data can be requested from the corresponding authors.

Acknowledgement

The authors extend their gratitude to the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2026R746),Princess Nourah bint Abdulrahman University, Riyadh,Saudi Arabia.

Authors Contributions

KK conceptualization, formal analysis, writing - the original manuscript.
MM data curation, methodology, writing - the original manuscript.
JCME funding acquisition, investigation, methodology.
CEUR project administration, visualization, software.
NAS investigation, visualization, validation.
IA validation, supervision, writing - review and editing.
All authors reviewed the manuscript and approved it.

References

[1] Xia, W., Wen, Y., Foh, C.H., Niyato, D., Xie, H.: A survey on software-defined networking. IEEE Communications Surveys & Tutorials **17**(1), 27–51 (2014)

[2] Jammal, M., Singh, T., Shami, A., Asal, R., Li, Y.: Software defined networking: State of the art and research challenges. Computer Networks **72**, 74–98 (2014)

[3] Madi, T., Alameddine, H.A., Pourzandi, M., Boukhtouta, A.: Nfv security survey in 5g networks: A three-dimensional threat taxonomy. Computer Networks **197**, 108288 (2021)

[4] Hubballi, N., Suryanarayanan, V.: False alarm minimization techniques in signature-based intrusion detection systems: A survey. Computer Communications **49**, 1–17 (2014)

[5] Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E.: Anomaly-based network intrusion detection: Techniques, systems and challenges. computers & security **28**(1-2), 18–28 (2009)

[6] Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp **1**, 108–116 (2018)

[7] Sommer, R., Paxson, V.: Outside the closed world: On using machine learning for network intrusion detection. In: 2010 IEEE Symposium on Security and Privacy, pp. 305–316 (2010). IEEE

[8] Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A.A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. computers & security **31**(3), 357–374 (2012)

[9] Elsayed, M.S., Le-Khac, N.-A., Dev, S., Jurcut, A.D.: Machine-learning techniques for detecting attacks in sdn. In: 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT), pp. 277–281 (2019). IEEE

[10] Elsayed, M.S., Le-Khac, N.-A., Jurcut, A.D.: Insdn: A novel sdn intrusion dataset. Ieee Access **8**, 165263–165284 (2020)

[11] Yang, Z., Liu, Z., Zong, X., Wang, G.: An optimized adaptive ensemble model with feature selection for network intrusion detection. Concurrency and Computation: Practice and Experience **35**(4), 7529 (2023)

[12] Alamin Talukder, M., Fida Hasan, K., Manowarul Islam, M., Ashraf Uddin, M., Akhter, A., Abu Yousuf, M., Alharbi, F., Moni, M.A.: A dependable hybrid machine learning model for network intrusion detection. arXiv e-prints, 2212 (2022)

[13] Chen, Z., Zhou, L., Yu, W.: Adasyn- random forest based intrusion detection model. In: 2021 4th International Conference on Signal Processing and Machine Learning, pp. 152–159 (2021)

[14] Azizjon, M., Jumabek, A., Kim, W.: 1d cnn based network intrusion detection with normalization on imbalanced data. In: 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), pp. 218–224 (2020). IEEE

[15] Khafaga, D.S., Karim, F.K., Abdelhamid, A.A., El-kenawy, E.-S.M., Alkahtani, H.K., Khodadadi, N., Hadwan, M., Ibrahim, A.: Voting classifier and metaheuristic optimization for network intrusion detection. CMC-COMPUTERS MATERIALS & CONTINUA **74**(2), 3183–3198 (2023)

[16] Rizvi, S., Scanlon, M., McGibney, J., Sheppard, J.: Deep learning based network intrusion detection system for resource-constrained environments. In: Springer, pp. 1–7 (2023)

[17] Wang, Z., Ghaleb, F.A.: An attention-based convolutional neural network for intrusion detection model. IEEE Access (2023)

[18] Kim, J., Shin, Y., Choi, E., *et al.*: An intrusion detection model based on a convolutional neural network. Journal of Multimedia Information System **6**(4), 165–172 (2019)

[19] Elaeraj, O., Leghris, C.: Toward an appropriate approach for intelligent intrusion and detection systems. Journal of Artificial Intelligence & Cloud Computing. SRC/JAICC-109. DOI: doi. org/10.47363/JAICC/2022 (1) **109**, 2–10 (2022)

[20] Saleh, A.I., Talaat, F.M., Labib, L.M.: A hybrid intrusion detection system (hids) based on prioritized k-nearest neighbors and optimized svm classifiers. Artificial Intelligence Review **51**, 403–443 (2019)

[21] Duy, P.T., Khoa, N.H., Do Hoang, H., Pham, V.-H., *et al.*: Investigating on the robustness of flow-based intrusion detection system against adversarial samples using generative adversarial networks. Journal of Information Security and Applications **74**, 103472 (2023)

[22] Yao, W., Hu, L., Hou, Y., Li, X.: A two-layer soft-voting ensemble learning model for network intrusion detection. In: 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), pp. 155–161 (2022). IEEE

[23] Sultana, N., Chilamkurti, N., Peng, W., Alhadad, R.: Survey on sdn based network intrusion detection system using machine learning approaches. Peer-to-Peer Networking and Applications **12**, 493–501 (2019)

[24] Mohammed, A.R., Mohammed, S.A., Shirmohammadi, S.: Machine learning and deep learning based traffic classification and prediction in software defined networking. In: 2019 IEEE International Symposium on Measurements & Networking (M&N), pp. 1–6 (2019). IEEE

[25] Sharma, N., Shambharkar, P.G.: Multi-attention deepcrnn: an efficient and explainable intrusion detection framework for internet of medical things environments: N. sharma, pg shambharkar. Knowledge and Information Systems **67**(7), 5783–5849 (2025)

[26] Sharma, N., Shambharkar, P.G.: Blockchain-based framework for secure medical data sharing and disease diagnosis using optimized deep belief networks. Cluster Computing **28**(16), 1029 (2025)

[27] ElSayed, M.S., Le-Khac, N.-A., Albahar, M.A., Jurcut, A.: A novel hybrid model for intrusion detection systems in sdns based on cnn and a new regularization technique. Journal of Network and Computer Applications **191**, 103160 (2021)

[28] Elsayed, R.A., Hamada, R.A., Abdalla, M.I., Elsaid, S.A.: Securing iot and sdn systems using deep-learning based automatic intrusion detection. Ain Shams Engineering Journal **14**(10), 102211 (2023)

[29] Elsayed, M.S., Le-Khac, N.-A., Jurcut, A.D.: Insdn: A novel sdn intrusion dataset. Ieee Access **8**, 165263–165284 (2020)

[30] Mondal, B., Banerjee, A., Gupta, S.: A review of sqli detection strategies using machine learning. machine learning **6**(S2), 9664–9677 (2022)

[31] McNamee, J.M.: A bibliography on roots of polynomials. Journal of Computational and Applied Mathematics **47**(3), 391–394 (1993)

[32] Labani, M., Moradi, P., Ahmadizar, F., Jalili, M.: A novel multivariate filter method for feature selection in text classification problems. Engineering Applications of Artificial Intelligence **70**, 25–37 (2018)

[33] Jaakonmäki, R., Müller, O., Vom Brocke, J.: The impact of content, context, and creator on user engagement in social media marketing. In: Proceedings of the Annual Hawaii International Conference on System Sciences, vol. 50, pp. 1152–1160 (2017). IEEE Computer Society Press

[34] Tu, W., Ma, Z., Ma, Y., Dopfel, D., Zhang, N.: Suppressing anterior cingulate cortex modulates default mode network and behavior in awake rats. Cerebral cortex **31**(1), 312–323 (2021)

[35] Chatterjee, S., Dey, D., Munshi, S.: Integration of morphological preprocessing and fractal based feature extraction with recursive feature elimination for skin lesion types classification. Computer methods and programs in biomedicine **178**, 201–218 (2019)

[36] Hadem, P., Saikia, D.K., Moulik, S.: An sdn-based intrusion detection system using svm with selective logging for ip traceback. Computer Networks **191**, 108015 (2021)

[37] Maalouf, M.: Logistic regression in data analysis: an overview. International Journal of Data Analysis Techniques and Strategies **3**(3), 281–299 (2011)

[38] Sahin, E.K.: Assessing the predictive capability of ensemble tree methods for

landslide susceptibility mapping using xgboost, gradient boosting machine, and random forest. SN Applied Sciences **2**(7), 1308 (2020)

[39] Goethals, S., Martens, D., Evgeniou, T.: The non-linear nature of the cost of comprehensibility. Journal of Big Data **9**(1), 1–23 (2022)

[40] Granik, M., Mesyura, V.: Fake news detection using naive bayes classifier. In: 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), pp. 900–903 (2017). IEEE

[41] Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K.: Knn model-based approach in classification. In: On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings, pp. 986–996 (2003). Springer

[42] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y.: Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems **30** (2017)

[43] Taud, H., Mas, J.: Multilayer perceptron (mlp). Geomatic approaches for modeling land change scenarios, 451–455 (2018)

[44] Wong, T.-T., Yeh, P.-Y.: Reliable accuracy estimates from k-fold cross validation. IEEE Transactions on Knowledge and Data Engineering **32**(8), 1586–1594 (2019)

[45] Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L., Ridella, S., *et al.*: The'k'in k-fold cross validation. In: ESANN, pp. 441–446 (2012)

[46] Oyucu, S., Polat, O., Türkoğlu, M., Polat, H., Aksöz, A., Ağdaş, M.T.: Ensemble learning framework for ddos detection in sdn-based scada systems. Sensors **24**(1), 155 (2023)

[47] Mahesh, D., Tallapally, S.K.: Advanced sdn-based network security: an ensemble optimized deep learning-based framework for mitigating ddos attacks with intrusion detection. Cluster Computing **28**(5), 331 (2025)

[48] Nalayini, C., Soumya, T., Lalitha, S., Tamijetchelvy, R.: A novel adaptive transformer based quantum intrusion detection system for software defined networks. Scientific Reports **15**(1), 36505 (2025)

[49] Wang, H., Li, W.: Ddostc: A transformer-based network attack detection hybrid mechanism in sdn. Sensors **21**(15), 5047 (2021)

[50] Logeswari, G., Bose, S., Anitha, T.: An intrusion detection system for sdn using machine learning. Intelligent Automation & Soft Computing **35**(1), 867–880 (2023)

[51] Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R., Ghogho, M.: Deep recurrent neural network for intrusion detection in sdn-based networks. In: 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), pp. 202–206 (2018). IEEE