Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.026769
Article

Tech Science Press

An Intrusion Detection System for SDN Using Machine Learning

G. Logeswari*, S. Bose and T. Anitha

Department of Computer Science and Engineering, College of Engineering, Guindy, Anna University, Chennai, Tamilnadu, India
*Corresponding Author: G. Logeswari. Email: logeswarig4@gmail.com

Received: 04 January 2022; Accepted: 16 February 2022

Abstract: Software Defined Networking (SDN) has emerged as a promising and exciting option for the future growth of the internet. SDN has increased the flexibility and transparency of the managed, centralized, and controlled network. On the other hand, these advantages create a more vulnerable environment with substantial risks, culminating in network difficulties, system paralysis, online banking frauds, and robberies. These issues have a significant detrimental impact on organizations, enterprises, and even economies. Accuracy, high performance, and real-time systems are necessary to achieve this goal. Using a SDN to extend intelligent machine learning methodologies in an Intrusion Detection System (IDS) has stimulated the interest of numerous research investigators over the last decade. In this paper, a novel HFS-LGBM IDS is proposed for SDN. First, the Hybrid Feature Selection algorithm consisting of two phases is applied to reduce the data dimension and to obtain an optimal feature subset. In the first phase, the Correlation based Feature Selection (CFS) algorithm is used to obtain the feature subset. The optimal feature set is obtained by applying the Random Forest Recursive Feature Elimination (RF-RFE) in the second phase. A LightGBM algorithm is then used to detect and classify different types of attacks. The experimental results based on NSL-KDD dataset show that the proposed system produces outstanding results compared to the existing methods in terms of accuracy, precision, recall and f-measure.

Keywords: Intrusion detection system; light gradient boosting machine; correlation based feature selection; random forest recursive feature elimination; software defined networks

1 Introduction

Web applications are becoming more widespread, and the internet has become an integral component of our everyday lives. As a result, there has also been increasing attention paid to the issue of network security [1]. The identification of anomalous network behavior is an important issue in network security research. Intrusion Detection System (IDS) are used to examine network data and identify anomalous network activities. IDSs are typically categorized into two types: signature-based and anomaly-based detection systems [2].

Signature-based Intrusion Detection Systems identify intrusion by constructing abnormal behavior character libraries and comparing network data. This approach identifies well-known attacks and has a low false alarm rate. However, this method is incapable of detecting zero-day attacks that are new and unknown. Anomaly-based intrusion detection systems build models based on typical network activity and identify intrusions depending on whether the behaviors deviate from the norm [3].

Software Defined Networking (SDN) is a topology which is dynamic, programmable, inexpensive and flexible, making it a good choice for today's high-bandwidth, dynamic real time applications. The network control and forwarding responsibilities are isolated in this topology, enabling network control to be easily configurable while the underlying infrastructure for applications and network services is hidden. Fig. 1 depicts a detailed description of the SDN architecture. The OpenFlow protocol [4] is a well-known standard that is utilized in the development of SDN systems. SDNs are being used in a variety of network applications, ranging from residential and corporate networks to datacenters. SDN characteristics assist in resolving various security challenges in a conventional network and provide us with the ability to govern network traffic at a fine-grained level. However, the SDN architecture offers additional attack vulnerabilities and security threats.

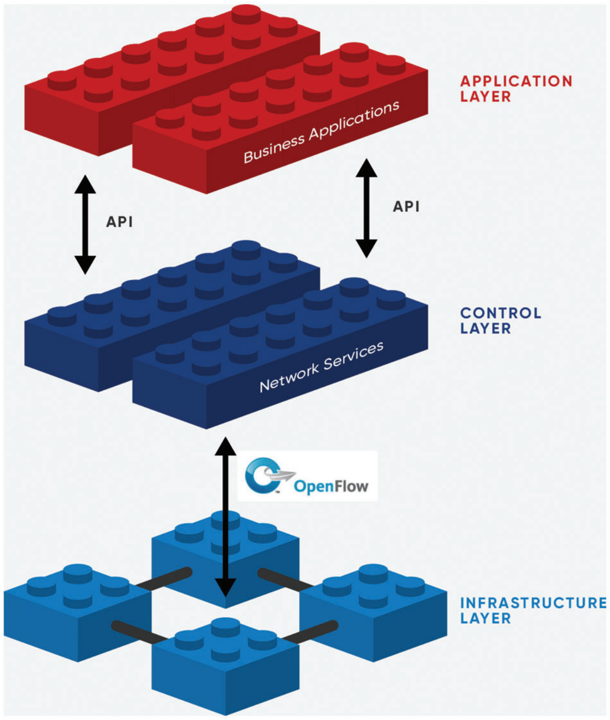

Figure 1: Software defined networking architecture

In SDN, Kreutz et al. [5] presented seven threat vectors. The control-data interface and the control-application interface are two that are unique to SDN and are connected to the controller. Controller is insecure because it creates a single point of attack and failure. Encrypting the communication connection with the Transport Layer Security protocol (TLS) is also ineffective and optional. In the SDN architecture, several attacks are possible. SDN security is a serious challenge due to the broad diversity types of SDN deployment. To secure SDNs, different machine learning algorithms has been used.

Machine learning is a field in computer science which originated from the pattern recognition and computational learning theories of artificial intelligence. It deals with the design and development of algorithms for learning and forecasting data. When the training data is labelled, machine learning is classed as supervised learning, unsupervised learning is when the training data is unlabeled, and semi-supervised learning is when the training data is a mix of labelled and unlabeled data [6]. Decision Tree (DT), Random Forest (RF), XGBoost, K-Nearest Neighbor (KNN), K-Mean Clustering, Support Vector Machine (SVM) and Ensemble Methods are some of the most commonly used techniques for IDS.

In order to facilitate the comprehension of the proposed work, the rest of the paper is structured into different sections. Section 2 illustrates the related work for SDN that is based on machine learning algorithms. Section 3 depicts the proposed HFS-LGBM IDS. Section 4 discusses and evaluates the results of the proposed system using various evaluation metrics. Finally, Section 5 summarizes the conclusions and makes recommendations for further research.

2 Related Works

Machine learning techniques have become an essential component of attack security measures. These machine learning-based technologies are capable of distinguishing between normal and attack traffic with extreme accuracy. There are multiple research projects that provide attack detection and prevention methods against various SDN threats.

Singh et al. [7] suggested a machine learning-based DDoS defense technique in SDN. The developed system is divided into three modules: flow statistics collection, feature extraction, training, and network traffic categorization. In this security system, several machine learning classifiers such as Support Vector Machine, Decision Tree, Logistic Regression and Nearest Neighbors are investigated for identifying normal and attack traffic. For attack detection, the classifier with the highest accuracy and lowest false positive rate is chosen. Chen et al. [8] presented an attack detection method based on the XGBoost classifier. By exploiting the controller's characteristics, the traffic collector and classification model were deployed in the controller. The purpose of the research was to resolve the treat concerns on the controller.

Researchers presented a crossbred IDS method in [9] that combines two different evolutionary algorithms such as Artificial Fish Swarm and Artificial Bee Colony. The crossbred method was used to produce anomaly detection criteria that are based on a limited subset of characteristics generated by the primary combined process. The inspired alternatives also outperform traditional and profound learning algorithms for detecting network threats, but comparisons are more difficult due to a lack of knowledge on methodologies and processes.

In [10], an extreme gradient boosting classifier was employed to differentiate between two types of attacks: normal and DoS. POX SDN, an open source SDN infrastructure for testing and developing SDN-based approaches, was used as a controller to analyse and verify the detection strategy. To improve computations by generating structure trees, the XGBoost term was added and integrated with the logistic regression technique. Two normalization techniques such as logarithmic and min- max were used. For an intrusion detection application, the SVM Classifier [11] was combined with the principal component analysis (PCA) technique. In this approach, the NSL-KDD dataset is employed to train and optimize the model for detecting anomalous patterns. To tackle the diversity data scale ranges with the fewest misclassification concerns, a Min-Max normalization approach was developed.

The researchers in [11] have proposed a framework that relies on voting based ensemble model for the attack detection. Ensemble model is a combination of multiple machine learning classifiers for prediction of final results. In the proposed work, three ensemble models such as Voting-CMN, voting-RKM and voting -CKM are proposed and analysed. The authors in [12] integrated the benefits of machine learning with intrusion detection systems to provide high detection rate and to defend networks from attackers. To detect attack anomalies, the presented system employs a Grid Search approach combined with SVM.

Their experimental research shows that they have made significant progress in identifying practically all conceivable network attacks in an SDN-based cloud environment.

In [13], new features are incorporated to build an ML-based model capable of detecting a DDoS attack on the SDN controller. In addition to traffic data, the additional characteristics are retrieved from packet headers. The experimental findings suggest that the proposed work can identify the attack rapidly. Sultana et al. [14] conducted a review on several current researches on machine learning algorithms that use SDN to construct NIDS. The study also included tools for developing NIDS models in an SDN environment.

The fundamental task of an intrusion detection system is to analyse huge amounts of network traffic data. To solve this problem, a well-organized classification system is required. Support Vector Machine (SVM) and Naive Bayes are two machine learning algorithms used in the proposed system [15] for resolving categorization issues. To increase the efficiency of identifying known and unexpected attacks, the research [16] develops a multi-level hybrid intrusion detection model that employs support vector machines and extreme learning machines.

A modified K-means technique is also proposed for creating a high-quality training dataset, which helps classifiers perform better. The modified K-means method is used to create new tiny training datasets that represent the full original training dataset, reducing classifier training time and improving intrusion detection system performance [17]. provides a complete analysis of machine learning-based intrusion detection algorithms that have been published in the literature. This study complements existing intrusion detection surveys and serves as a reference for researchers working on machine learning-based intrusion detection systems.

3 Proposed Method

This section discusses the proposed IDS to detect the attack in SDN. A high-level view of the deployment of HFS-LGBM based IDS in SDN architecture is presented in Fig. 2. The OpenFlow switches on the controller unit are normally managed by the SDN controller. The SDN controller has the ability to request all network data whenever it is needed. As a result, the proposed HFS-LGBM intrusion detection system is deployed in the SDN Controller.

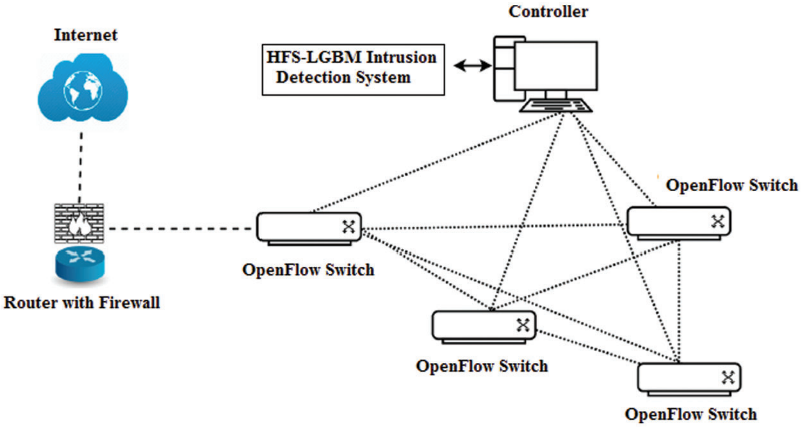

Figure 2: Proposed HFS-LGBM based IDS in SDN

The architectural diagram of the HFS-LGBM system is presented in Fig. 3. In this paper, the HFS algorithm that combine two feature selection algorithms such Correlation based Feature Selection and Random Forest Recursive Feature Elimination is proposed.

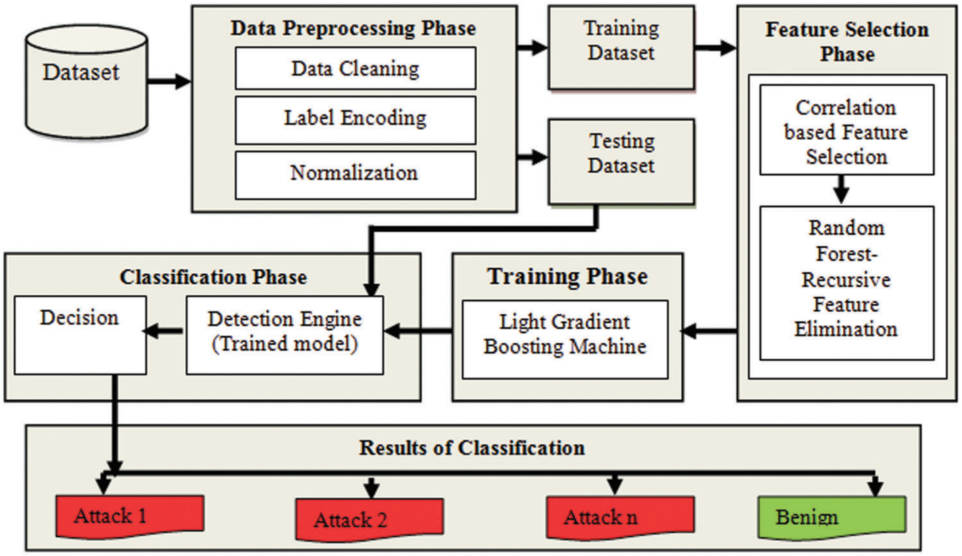

Figure 3: The proposed HFS-LGBM based IDS

3.1 Data Preprocessing Phase

Data preprocessing is the most crucial task which simplifies and improves the efficiency of data mining methods. Data is usually derived from a variety of sources and can be noisy, excessive, incomplete or contradictory. As a result, it is necessary to transform unprocessed data into useful information for investigation and disclosure. The pre-preprocessing phases in this research include the following processes, which are detailed in the following sections:

3.1.1 Data Cleaning

To achieve accurate prediction, data cleaning is the process of eliminating or correcting inaccurate, duplicate, or incomplete records and filling missing values within provided datasets. To avoid deceiving the models during training, the white or blank spaces of multi-class labels are identified and removed.

3.1.2 Label Encoding

Several datasets have categorical variables, that are incompatible with most machine learning algorithms. As a result, it is necessary to convert these categorical into numerical values. Each categorical value is assigned an integer value using one-hot and ordinal encoding techniques.

3.1.3 Normalization

The effectiveness of classification models can be significantly influenced by feature imbalance scales. As a result, it's critical to standardize these discrepancies within the dataset's features such that both negligible and dominating values are within an appropriate limits. The minimum-maximum technique is used to systematically normalize dataset features within the normalized range of [0, 1], making the data easier to interpret. The minimum-maximum method's equation is as follows:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

where X' denotes the normalized result, X_{min} and X_{max} denotes the minimum and maximum value of feature X and the original data sample value to be normalized is represented by X.

3.2 Feature Selection Phase

The proposed HFS approach is described in this section. Fig. 4 depicts the flow diagram for the HFS approach.

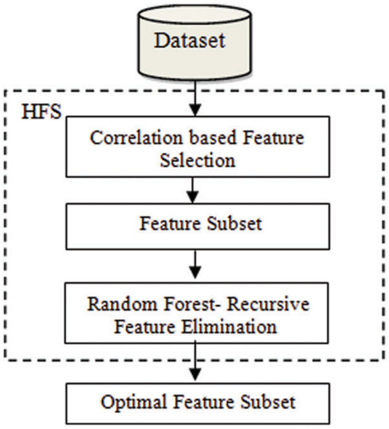
Dataset
HFS
Correlation based Feature Selection
Feature Subset
Random Forest- Recursive Feature Elimination
Optimal Feature Subset

Figure 4: Flow diagram for HFS approach

The proposed HFS Approach is composed of two stages. The CFS is used to minimize the size of the feature set in the first stage by eliminating noisy irrelevant and redundant features [18]. The RF-RFE is used in the second stage to find the optimal feature subset from the reserved feature set.

3.2.1 Correlation Based Feature Selector (CFS)

One of the ways in machine learning is to select features for predicting outcomes based on their connection, and such a feature selection methodology may be favorable to regular machine learning algorithms. It is advantageous if a feature conforms to or anticipates a class. A distinguishing feature (X_i) is perceived to be relevant if and only if some probability (P)x_i and y exist such that P(X_i = xi) > 0, as shown in Eq. (2).

$$P(Y = y \mid X_i = x_i) \neq P(Y = y) \tag{2}$$

Superfluous features must be removed in addition to insignificant features, according to experimental evidence from the feature selection literature. If a feature is overly intertwined with one or more other features, it is considered superfluous. Furthermore, this yielded a hypothesis for feature selection, which is a usable, acceptable characteristic feature subgroup consisting of features that are strongly connected with class but dissociated from one another. If the association among an individual feature and an extrinsic variable is renowned in a provided feature set, and the inter-relationship among each other pair of features is known, then Pearson's correlation coefficient Eq. (3) can be used to calculate the relationship between the complicated test consisting of the total features and the extrinsic variable.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})\overline{(y_i - y)}}{\sqrt{\sum_{=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3}$$

The observed and average values of the features examined are denoted by x_i and $\bar{x}$, respectively.

The dataset class's observed and average values are defined by y_i and $\bar{y}$.

If a set of n features is picked, the correlation coefficient would be used to investigate the link between the set and the class while accounting for feature inter-correlation. As the link between features and classes expands, the feature group becomes more important. Furthermore, it lowers as inter-correlation increases. The aggregated correlation coefficient across features and output variables is defined as r_{ny} = p(X_n,Y), and the aggregate among varied features is defined as r_{nn} = p (X_n,X_n). Eq. (4) gives the group correlation coefficient for determining the significance of the feature subset.

$$J(X_{n,}Y) = \frac{nr_{ny}}{\sqrt{n + (n-1)r_{nn}}} \tag{4}$$

In the correlation-based feature selection technique, the Pearson's correlation coefficient is used to allow the inclusion or deletion of one feature at a time.

Algorithm 1: Correlation based Feature Selection

```
INPUT:
Dt— Dataset for training
Y—The predictor
k— No. of features to select
OUTPUT:
$F_i$— Choosen feature set
BEGIN:
$F_o = Ø$
i = 1
while $|F_i| < k$ do
if $|F_i| <k-1$ then
$F_i = CFS(F_{i-1}, Dt, Y)$
else
  best-ranked feature f' is added to $F_{i-1}$
end if
i = i + 1
end while
END
```

3.2.2 Random Forest Recursive Feature Elimination (RF-RFE)

RFE is a greedy algorithm-based feature-ranking approach. In order to achieve the most significant features, RFE is used to eliminate the least significant features from the entire feature one by one in each iteration. The recursion is mandatory because significant characteristic features for some processes might change dramatically while evaluating beyond an alternate subset of features during step-wise elimination. This mainly pertains to features which are closely linked. The final feature set is built in the order in which the features are rejected. Simply extracting the first n features from this ranking is the feature selection strategy.

Random Forest is a prediction approach that utilizes an ensemble of models. It constructs a composite predictor by integrating a large number of independent prediction trees. To construct the final forecast of a dataset, an absolute rule is employed among the predictors' options. Furthermore, in order to generate unrelated and diverse insights, each tree is built using a subset of the preparation set's dataset. Additionally, the algorithm incorporates random contingency in the search for optimal splits in order to optimize the dissimilarity among the trees.

Algorithm 2: Random Forest Recursive Feature Elimination

```
Input:
$T_0$=[$t_1$, $t_2$,…..$t_k$]- Dataset for training
A=[$a_1$, $a_2$,….$a_k$]–Set of k features
Ranking Method R(T,A)
D = [1, 2,… n] — Subset of features
Output:
Final optimal feature set $A_d$
Begin:
D = [1, 2,… n]
$A_d$=[ ]
While D ≠[ ] do
Repeat for x in {1:k}
   Ranking feature set using R(T,A)
   D(a*) ← A's last ranked feature
   $A_d$ (k-x+1)←D(a*)
   D($A_d$) ← D($A_d$)–D(a*)
end while
END
```

The proportion of relevance of distinctive features is linked with the recursive feature elimination process in the RF-RFE technique. The RF-RFE approach is based on the notion of frequently generating a random forest model and picking the appropriate or worst operational feature. Eliminate the feature and repeat the process again with the remaining features. This process is repeated until all of the features in the dataset have been utilized. After that, the features are ranked in the order they are deleted. , a greedy optimization technique is used to find the best performing feature subset.

3.3 Training Phase

The LightGBM classification algorithm is used to train the data selected by the HFS algorithm. LightGBM is a high-performance gradient boosting machine learning technique that is fast, distributed, and open-source. It employs histogram-based methods to improve training while consuming less memory. LightGBM is very efficient and precise, enables parallel learning, and works well with huge datasets. LightGBM is primarily comprised of two enhanced algorithms: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). LightGBM employs an enhanced histogram technique: the EFB algorithm reduces the number of features, while the GOSS algorithm reduces the number of samples each round of training. Various data instances in GOSS play different roles in calculating information gain, with a higher gradient instance contributing more to the information gain. To preserve the precision of information gain estimates, GOSS retains instances with high gradients and removes instances with low gradients at random [19]. Eq. (5) depicts the mathematical analysis in GOSS.

$$\widehat{V}_j(d) = \frac{1}{n}\left(\frac{\left(\sum_{x_i \in A_l} g_i + \frac{1-a}{b}\sum_{x_i \in B_l} g_i\right)^2}{n_l^i(d)} + \frac{\left(\sum_{x_i \in A_r} g_i + \frac{1-a}{b}\sum_{x_i \in B_r} g_i\right)^2}{n_r^i(d)}\right) \tag{5}$$

where $\widehat{V}_j(d)$: Estimated variance gain over the subset A∪B.

$A_l : \{x_i \in A : x_{ij}\}$

$A_r : \{x_i \in A : x_{ij} > d\}$

$B_l : \{x_i \in B : x_{ij} \leq d\}$

$B_r : \{x_i \in B : x_{ij} > d\}$

$\frac{1-a}{b}$: Coefficient to normalize the size of A^c to the sum of gradients over B.

Coefficient to normalize the size of Ac to the sum of gradients over B.

To identify the split point, the estimated $\widehat{V}_j(d)$ is employed over a smaller instance subset rather than the accurate $V_j(d)$across all the instances. Simultaneously, LightGBM employs the EFB approach to reduce model complexity by combining exclusive features into a single feature. The logistic regression obtained in Eq. (6) is the loss function [19] in our model. This function is found to be an excellent calibration statistic function for training our detector since it penalizes the difference between real and anticipated odds. Furthermore, it calculates the relative uncertainty between the classes predicted by our method and the actual classes.

$$Logloss = -\frac{1}{N}\sum_{i=i}^{N} y_i \log(\hat{y}) + (1 - y_i)\log(1 - \hat{y}) \tag{6}$$

where i represent the given observation, y_i represent the true value and $\hat{y}$ represent the probability of prediction.

3.4 *Classification Phase*

Furthermore, based on the clusters produced during the training phase, the final trained model in the classification phase with the average of probability rule and voting technique is applied to categorize the test dataset as benign or various attack types inside the test dataset.

4 Experimental Results and Analysis

4.1 *Mininet Implementation*

This paper host VMware's Mininet Virtual Machine. Mininet is a Python-based open source network emulator that generates a virtual networking architecture that connects virtual hosts through various devices such as switches, links, and controllers. It comes with Linux network software and is capable of supporting OpenFlow for custom routing and SDN. Because mininet must be installed on a Linux server, we picked Oracle VM VirtualBox for our simulations. The simulation was run on a PC running 64-bit Ubuntu 18.04 LTS on a Core-i7 with 16 GB of RAM.

4.2 Dataset Description

The authors of [20] presented an improved version of KDDCup'99, termed NSL-KDD, to address the previously noted issues [1]. Despite the NSL-KDD dataset's severe intrinsic challenges, such as the insufficient representation of contemporary low footprint attack scenarios, it is still regarded the most recommended IDSs assessment dataset due to its unique feature of maximizing predictions for classifiers. It is made up of four attack categories with 41 attributes each and a single labeled class that distinguishes between malicious and normal network traffic. The dataset is divided into training and testing dataset. The training set consists of 1, 25, 973 records and the testing set consists of 22, 544 records.

4.3 Experimental Results of HFS-LGBM Intrusion Detection System

The experiments were carried out using the WEKA environment and the NSL-KDD dataset. The proposed HSF-LGBM IDS is evaluated using traditional metrics such as precision, recall, accuracy and F1-score.

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

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

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

$$F1 - score = 2 \times \frac{precision\ X\ recall}{precision + recall} \tag{10}$$

The proposed method is compared with the existing machine learning classifiers. Support Vector Machine (SVM), Linear Regression (LR), Random Forest, Catboost, Xgboost, LightGBM, and the proposed HFS-LGBM are the machine learning classification algorithms compared in Tab. 1. The comparison is pictorially presented in Fig. 5. From the analysis, it is clear that the proposed HFS-LGBM performs better in terms of accuracy, precision, recall and F-score.

Table 1: Comparison of various machine learning classification algorithms

Algorithm	Accuracy	Precision	Recall	F-measure
SVM	0.822	0.8284	0.8263	0.8284
LR	0.8492	0.8431	0.8321	0.8332
Catboost	0.9163	0.9239	0.9351	0.9542
Xgboost	0.9631	0.9335	0.9407	0.9666
LightGBM	0.9869	0.9553	0.967	0.9805
HFS-LGBM	0.9872	0.9745	0.9792	0.9823

The optimal selected features of NSL-KDD dataset in presented in Tab. 2. In this paper, the proposed HFS- LGBM method is compared with the other feature selection algorithms such as chi, information gain, correlation based feature selection. The results are shown in Tab. 3 and Figs. 6–9. The proposed algorithm has achieved better performance in terms of accuracy, precision, recall and F-measure.

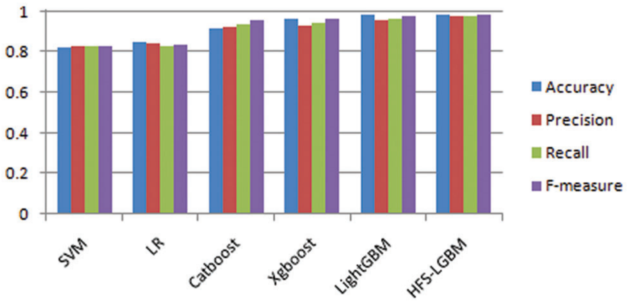

Figure 5: Comparison of various machine learning classification algorithms

Table 2: Selected optimal features

No	Selected features
4	Flag
5	Src_bytes
6	Dst_bytes
15	Min.Packet.Length
17	Radiotap.channel.type.cc
26	Srv_serror-rate
30	Diff_srv_rate
29	Same_sev_rate

Table 3: Comparison of classification indexes of feature selection algorithms

Algorithm	Accuracy	Precision	Recall	F-measure
LightGBM	0.9869	0.9553	0.967	0.9805
IG+LGBM	0.9778	0.9051	0.9047	0.9034
CHI+LGBM	0.9725	0.9314	0.9211	0.9287
CFS+LGBM	0.9524	0.7589	0.7058	0.7251
HFS-LGBM	0.9872	0.9745	0.9792	0.9823

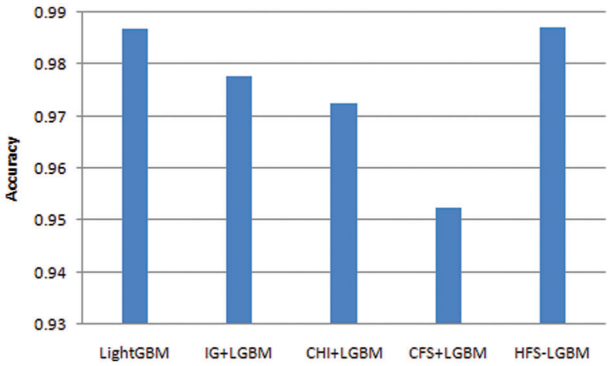

Figure 6: Comparison of the accuracy of HFS-LGBM with various feature selection algorithms

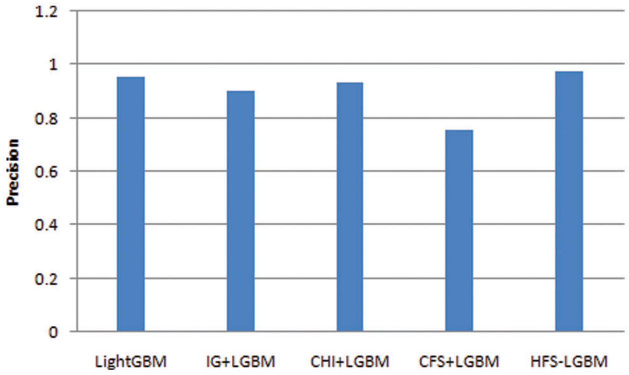

Figure 7: Comparison of the precision of HFS-LGBM with various feature selection algorithms

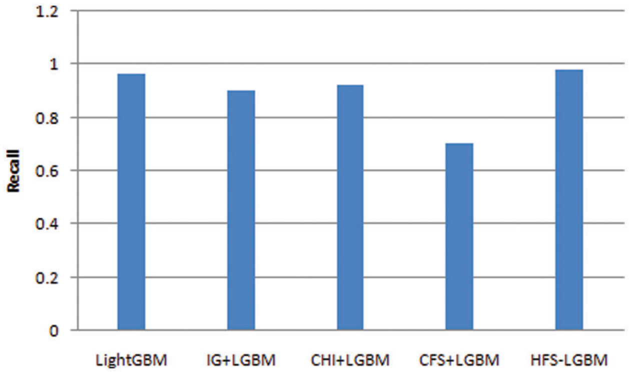

Figure 8: Comparison of the recall of HFS-LGBM with various feature selection algorithms

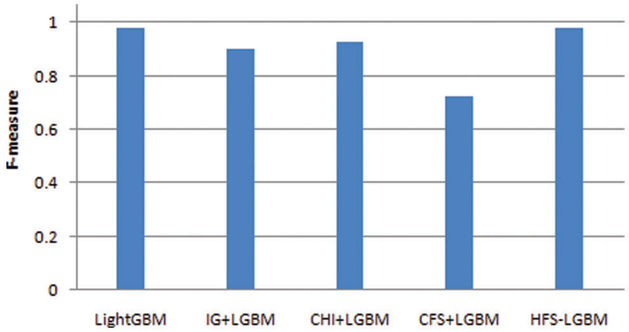

Figure 9: Comparison of the F-measure of HFS-LGBM with various feature selection algorithms

5 Conclusion

SDNs enhance networking flexibility by separating the control plane and data plane and removing network architectural privacy while opening and scheduling networks. SDN, as a dynamic network, threatens the technological future. In this paper, a novel HFS-LGBM IDS is proposed. The HFS approach combines the advantages of two algorithms such as correlation based feature selection and Random Forest Recursive Feature Elimination. Mininet is used to construct a static software network. To evaluate the proposed system, NSL-KDD dataset is used. The evaluation results of the proposed system are compared with various feature selection and machine learning classification algorithms. According the results of the experiments, the proposed HFS-LGBM has obtained better results in accuracy, precision, recall and f-measure.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Islam, P. Hridi, M. Hossain and H. Narman, "Network anomaly detection using light GBM: A gradient boosting classifier," in *Int. Telecommunication Networks and Applications Conf. (ITNAC)*, Melbourne, Australia, pp. 1–7, 2020.

[2] J. Ren, J. Guo, W. Qian, H. Yuan, X. Hao *et al.,* "Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms," *Security and Communication Networks*, vol. 2, pp. 251–262, 2019.

[3] M. A. Jonas, M. S. Hossain, R. Islam, H. S. Narman and M. Atiquzzaman, "An intelligent system for preventing SSL stripping-based session hijacking attacks," in *MILCOM 2019 - 2019 IEEE Military Communications Conf. (MILCOM)*, China, vol. 2, pp. 1–6, 2019.

[4] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson *et al.,* "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[5] D. Kreutz, F. M. Ramos and P. Verissimo, "Towards secure and dependable software-defined networks," in *HotSDN '13: Proc. of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, India, vol. 3, pp. 55–60, 2013.

[6] A. Abubakar and B. Pranggono, "Machine learning based intrusion detection system for software defined networks," in *2017 Seventh Int. Conf. on Emerging Security Technologies (EST)*, Canterbury, UK, vol. 15, pp. 138–143, 2017.

[7] P. K. Singh, S. Kumar Jha, S. K. Nandi and S. Nandi, "ML-Based approach to detect DDOS attack in V2I communication under SDN architecture," in *TENCON 2018 - 2018 IEEE Region 10 Conf. (2018)*, Canda, vol. 10, pp. 0144–0149, 2018.

[8] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu *et al.,* "XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud," in *2018 IEEE Int. Conf. on Big Data and Smart Computing (BigComp)*, India, vol. 11, pp. 251–256, 2018.

[9] V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," *Computer Networks*, vol. 136, pp. 37–50, 2018.

[10] S. Y. Mehr, and B. Ramamurthy, "An SVM based DDoS attack detection method for RYU SDN controller," in *Proc. of the 15th Int. Conf. on Emerging Networking Experiments and Technologies (2019)*, Japan, vol. 12, pp. 72–73, 2019.

[11] S. T. Ikram and A. K. Cherukuri, "Improving accuracy of intrusion detection model using PCA and optimized SVM," *CIT. Journal of Computing and Information Technology*, vol. 24, no. 2, pp. 133–148, 2016.

[12] R. Swami, M. Dave and V. Ranga, "Voting-based intrusion detection framework for securing software-defined networks," *Concurrency and Computation: Practice and Experience*, vol. 2, pp. 512–532, 2020.

[13] W. G. Gadallah, N. M. Omar and H. M. Ibrahim, "Machine learning-based distributed denial of service attacks detection technique using new features in software-defined networks," *International. Journal Computer Network and Information Security*, vol. 3, pp. 15–37, 2021.

[14] N. Sultana, N. Chilamkurti, W. Peng and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Network Application*, vol. 12, pp. 493–501, 2019.

[15] A. Halimaa and K. Sundarakantham, "Machine learning based intrusion detection system," in *2019 3rd Int. Conf. on Trends in Electronics and Informatics (ICOEI)*, Canada, vol. 3, pp. 916–920, 2019.

[16] W. L. Al-Yaseen, Z. A. Othman and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," *Expert Systems with Applications*, vol. 67, pp. 296–303, 2017.

[17] A. S. Dina and D. Manivannan, "Intrusion detection based on machine learning techniques in computer networks," *Internet of Things*, vol. 16, pp. 215–223, 2021.

[18] G. Farahani, "Feature selection based on cross-correlation for the intrusion detection system," *Security and Communication Networks*, vol. 2020, pp. 152–167, 2020.

[19] M. Osman, J. He, F. M. M. Mokbal, N. Zhu and S. Qureshi, "ML-LGBM: A machine learning model based on light gradient boosting machine for the detection of version number attacks in RPL-based networks," *IEEE Access*, vol. 9, pp. 83654–83665, 2021.

[20] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Malaysia, vol. 3, pp. 1–6, 2009.