Hindawi
Security and Communication Networks
Volume 2020, Article ID 8840058, 11 pages
https://doi.org/10.1155/2020/8840058

Research Article

An Efficient and Effective Approach for Flooding Attack Detection in Optical Burst Switching Networks

Bandar Almaslukh

Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia

Correspondence should be addressed to Bandar Almaslukh; b.almaslukh@psau.edu.sa

Received 27 March 2020; Revised 8 July 2020; Accepted 11 July 2020; Published 5 August 2020

Academic Editor: Muhammad Faisal Amjad

Optical burst switching (OBS) networks are frequently compromised by attackers who can flood the networks with burst header packets (BHPs), causing a denial of service (DoS) attack, also known as a BHP flooding attack. Nowadays, a set of machine learning (ML) methods have been embedded into OBS core switches to detect these BHP flooding attacks. However, due to the redundant features of BHP data and the limited capability of OBS core switches, the existing technology still requires major improvements to work effectively and efficiently. In this paper, an efficient and effective ML-based security approach is proposed for detecting BHP flooding attacks. The proposed approach consists of a feature selection phase and a classification phase. The feature selection phase uses the information gain (IG) method to select the most important features, enhancing the efficiency of detection. For the classification phase, a decision tree (DT) classifier is used to build the model based on the selected features of BHPs, reducing the overfitting problem and improving the accuracy of detection. A set of experiments are conducted on a public dataset of OBS networks using 10-fold cross-validation and holdout techniques. Experimental results show that the proposed approach achieved the highest possible classification accuracy of 100% by using only three features.

1. Introduction

Optical burst switching (OBS) in networks has become an important dynamic sub-wavelength switching technique and a solution for developing the new type of Internet backbone infrastructure [1]. The OBS network mainly consists of three types of nodes, namely, core nodes, ingress, and egress. The core nodes represent the intermediate nodes, which are designed to reduce the processing and buffering of the optical data burst using a control data packet with specific information, namely, burst header packets (BHPs) [2].

In a network with burst traffic, OBS plays an essential role for packet switching with a higher level of necessary details than other existing networks' switching techniques. However, this type of switching is still suffering from several challenges such as security and quality of service (QoS) due to BHP flooding attacks. The function of BHP in OBS is to reserve the unused channel for the arrival of a data burst (DB). This function can be exploited by attackers to send fake BHPs without DB acknowledgment. Such fake BHPs can affect the network and reduce its performance through decreasing bandwidth utilization and increasing data loss, leading to a denial of service (DoS) attack [3], which is one of the most crucial security threats to networks.

Several methods have been proposed to tackle DoS and BHP flooding attacks on OBS networks in the literature and have achieved satisfactory results [4–6]. However, due to the limited capability of OBS core switches, developing a lightweight method that can attain high accuracy with a small number of features is still a challenging issue for developers and researchers.

In this research, an effective and efficient approach is proposed for securing the OBS networks. Thus, the main objective of the work is to develop a lightweight ML model for detecting BHP flooding attacks based on the information gain (IG) feature selection method and a decision tree (DT) classifier. To achieve this objective, two key research

questions are formulated to answer throughout this study. The first research question is does the feature selection method improve the effectiveness of the DT model to detect the BHP flooding attacks. The second research question is does the feature selection method improve the efficiency of the DT model for detecting the BHP flooding attacks. Actually, the lightweight property of the model comes from the fact that only a small number of features are used to build the classifier. The model will be evaluated using a public OBS dataset based on a set of performance metrics such as accuracy, precision, recall, and *F*-measure.

The remainder of the research is organized as follows:

(i) In Section 2, related works are introduced to give details about the proposed approaches and methods of DoS attack on different networks.

(ii) Section 3 presents the proposed approach architecture for detecting the BHP flooding attacks on OBS networks.

(iii) Section 4 explains the experimental setup and results in more detail.

(iv) Section 5 presents the conclusion of the study.

2. Related Works

Nowadays, machine learning (ML) methods have been used in many intrusion detection systems (IDSs) to detect several types of network attacks. However, feature selection methods are also used to select the significant features of network traffic without reducing the performance of the IDSs [7]. Feature selection is the process of selecting the best set of features that can be most effective for classification tasks [8, 9]. The high number of features may decrease the performance and accuracy of many classification problems [10, 11].

In the field of optimization, feature selection methods are classified in three main approaches: embedded, wrapper, and filter methods [12]. For the filter methods, there are two major types of evaluation: subset feature evaluation and groups of individual feature evaluation. In the groups of individual feature evaluation, heuristic or metaheuristic filter methods or even the hybrid of them is utilized for ranking the features and then the best of them is selected based on some thresholds [11, 13]. In contrast, the subset feature evaluation methods find the subset of candidate features using a certain measure or a certain strategy. They compare the previous best subset with the current subset for finding the candidate subset of features. In the groups of individual feature evaluation methods, the redundant features are kept in the final subset of selected features according to their relevance but the group of subset feature evaluation methods removes the features with similar ranks. In general, the filter methods are considered as classifier-independent approaches [13]. The wrapper methods are classifier-dependent approaches that take each time a subset of features from the total features and calculate the accuracy of classifiers to find the best subset. Therefore, they are time consuming compared with filter methods [14]. The embedded methods combine wrapper and filter methods [15]. In this study, a filter-based method is used for feature selection.

In the literature review of intrusion detection, a set of ML and deep learning (DL) methods have been widely used to detect different types of attacks in several works [16–20]. Meanwhile, a set of related works have also been proposed for detecting BHP flooding attacks using different ML methods like the decision tree (DT) method in [21]. This work evaluated the performance of the adopted method using different metrics and reported a 93% accuracy rate in classifying the classes of BHP flooding attack. Liao et al. [22] introduced a classification approach to classify the access patterns of various users using sparse vector decomposition (SVD) and rhythm matching methods. This study demonstrates that the approach is able to distinguish between the intruders and the legal users in the application layer.

Xiao et al. [23] offered an effective scheme for detecting a distributed DoS attack (DDoS) using the correlation of the information generated by the data center and the k-nearest neighbors (KNNs) method. They analyzed the flows of data traffic at the center to identify normal and abnormal flows. In [24], the authors proposed an approach for detecting DDoS attacks based on seven features and using an artificial neural network (ANN) method with a radial basis function (RBF). This NN-RBF approach can classify the data traffic into attack or normal classes by sending the IP address of the incoming packets from the source nodes to be filtered in the alarm modules which then decide if these data packets can be sent to the destination nodes.

The authors in [25] applied a data mining method for detecting a DDoS attack using the fuzzy clustering method (FCM) and a priori association algorithm to categorize the data traffic patterns and the status of the network. Another ML approach in [26] used a DT method with a grey relational analysis for detecting DDoS attacks. They also applied the pattern matching technique to the data flows for tracing back the estimated location of the attackers.

Alshboul [27] investigated the use of rule induction nodes for BHP classification in OBS networks. The author applied a set of data mining methods to the public OBS network dataset. He reported that the repeated incremental pruning to produce error reduction (RIPPER) rule induction algorithm, Naïve Bayes (NB), and Bayes Net were able to achieve a predictive accuracy of 98%, 69%, and 85%, respectively.

Chen et al. [28] developed a detection method to identify a DDoS attack using ANN. A set of different simulated DoS attacks were used for training the ANN model to recognize abnormal behaviors. Li et al. [29] offered different types of ANN models, including learning vector quantization (LVQ) models, to differentiate traffic associated with DDoS attacks from normal traffic. The authors converted the values of the dataset features into a numerical format before feeding them into the ANN model.

In [30], the authors presented a probabilistic ANN approach for classifying the different types of DDoS attacks. They categorized the DDoS attacks and normal traffic by applying radial basis function neural network (RBF-NN)

coupled with a Bayes decision rule. Nevertheless, the approach concentrated on the events of unscrambling flash crowds generated by DoS attacks.

Li and Liu [31] proposed a technique that integrates the network intrusion prevention system with SVM to improve the accuracy of detection and reduce the incidents of false alarms. In [32], Ibrahim offers a dynamic approach based on distributed time-delay ANN with soft computing methods. This approach achieved a fast conversion rate, high speed, and a high rate of anomaly detection for network intrusions.

Gao et al. [33] introduced a data mining method for analyzing the piggybacked packets of the network protocol to detect DDoS attacks. The advantage of this method is to retain a high rate of detection without manual data construction. Hasan et al. [34] proposed a deep convolutional neural network (DCNN) model to detect BHP flooding attacks on OBS networks. They reported that the DCNN model works better than any other traditional machine learning models (e.g., SVM, Naïve Bayes, and KNN). However, due to the small number of samples in the dataset and the limited resource constraints of OBS switches, such deep learning models are not effective tools to detect BHP flooding attacks and they are not computationally efficient to run in such network.

3. Proposed Approach

The proposed approach in this paper consists of two main phases: feature selection and classification. The input of the approach is a set of OBS dataset features collected from network traffic. The output of the approach is a class label of the BHP flooding attacks. The flowchart of the proposed approach is illustrated in Figure 1.

In the feature selection phase of the approach, the input features of OBS network traffic are prepared for processing by using the information gain (IG) feature selection method. The purpose of IG is to rank the features and discover the merit of each of them according to the information gain evaluation of the entropy function. The output of the feature selection phase is a scored rank of features in decreasing order according to their merit, whereby adding any feature decreases the features merit.

This is then followed by the classification phase, in which the dataset with selected features will be used to train and test the DT classifier to detect attacks on OBS networks. The output of the classification phase is a DT trained model that is able to classify the BHP flooding attacks and return the class label of that attack. The following sections explain the methods used in the two phases of the proposed approach.

3.1. Information Gain (IG) Feature Selection Method. Information gain (IG) is a statistical method used to measure the essential information for a class label of an instance based on the absence or presence of the feature in that instance. IG computes the amount of uncertainty that can be reduced by including the features. The uncertainty is usually calculated by using Shannon's entropy (E) [35] as

$$E(D) = \sum_{i=1}^{n} P_i \log_2 (P_i), \tag{1}$$

where n represents the number of class labels and P_i is the probability that an instance i in a dataset D can be labeled as a class label c by computing the proportion of instances that belong to that class label for the instance i as follows:

$$\frac{|D_{i \in C}|}{D}. \tag{2}$$

A selected feature f divides the training set into subsets $D_1, D_2, \ldots, D_v$ according to the values of f, where f has v distinct values. The information required to get the exact classification is measured by

$$\text{Reminder}(f) = \sum_{j=1}^{v} \frac{|D_j|}{D} \times E(DJ), \tag{3}$$

where $|D_j|/D$ represents the weight of j^{th} subset, $|D|$ is the number of instances in the dataset D, $|D_j|$ is the number of instances in the subset D_j, and $E(D_j)$ is the entropy of the subset D_j. Therefore, the IG of every feature is calculated as

$$\text{IG}(f) = E(D) - \text{Reminder}(f). \tag{4}$$

After calculating the IG for each feature, the top k features with the highest IG will be selected as a feature set because it reduces the information required to classify the flooding attack.

3.2. Decision Tree Method. Decision tree (DT) is a tree-like model of decisions with possible consequences that is commonly used in the fields of data mining, statistics, and machine learning [36]. In machine learning, the goal of DT is to build a model that predicts or classifies the value of a target class based on a learning process from several input features. The tree model that has a target class label with discrete values is called a classification tree model. In this model, the tree leaves constitute the values of the class label and the tree branches constitute aggregations of features that produce this class label.

DT learning is a simple process to represent the features for predicting or classifying instances. DT models are created by splitting the input feature set into subsets that establish the successor nodes of the children, thereby establishing the tree root node. Based on a set of splitting rules on the values of the features, the splitting process for each derived subset is repeated in a recursive manner [36]. This recursive manner is stopped when the splitting process no longer adds values to the predictions or when the subset of nodes have all the same values of the target class label.

The DT can be described also as a mathematical model to support the categorization, description, and generalization of a given dataset.

Assume the dataset comes in the form of records as follows:

$$(x, y) = (x_1, x_2, x_3, \ldots, x_k, y), \tag{5}$$

where the variable y is a dependent target variable that we need to generalize or classify. The vector x consists of the features $x_1, x_2, x_3, \ldots, x_k$, which are led to the variable y.

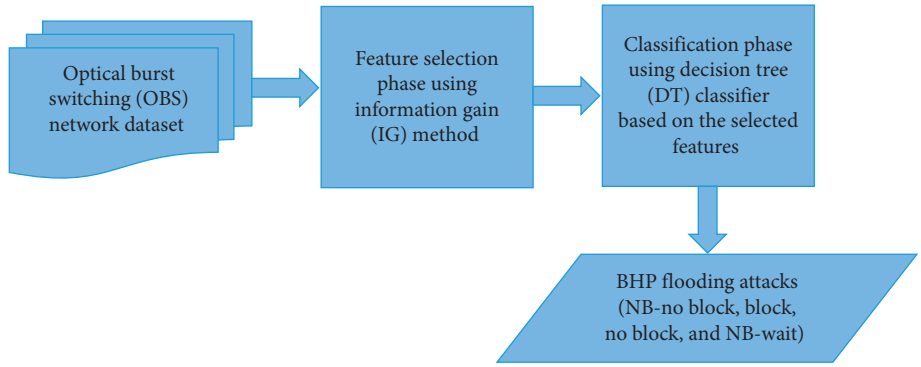

Figure 1: Flowchart of the proposed approach.

In principle, the DT is based on the C4.5 algorithm [37], which is an updated version of the ID3 algorithm [38]. C4.5 can avoid the overfitting problem of ID3 by using the rule-post pruning technique to convert the building tree into a set of rules.

DT is used in the proposed approach because it is simple, very intuitive, and easy to implement. Furthermore, it deals with missing values, requires less effort in terms of data preprocessing, and does not need to scale or normalize the data [36].

4. Experiments and Discussion

The experiments of this research are implemented using a popular open source tool called the Waikato Environment for Knowledge Analysis (Weka) software [39], which offers a rich toolbox of machine learning and data mining methods for preprocessing, analyzing, clustering, and classification. It offers Java-based graphical user interfaces (GUIs). The implementation was performed on a laptop with an Intel Core i7 CPU processor, 2.0 GHz, 8 GB RAM, and a Windows 10 64 bit operating system. Due to the scarcity of OBS historical data, the experiments were conducted on a public optical burst switching (OBS) network dataset [1].

4.1. OBS Network Dataset Description. The OBS network dataset is a public dataset, available from the UCI Machine Learning Repository [1]. It contains a number of BHP flooding attacks on OBS networks. There are 1,075 instances with 21 attributes as well as the target class label. This target label has four types of classes, which are NB-no block (not behaving-no block), block, no block, and NB-wait (not behaving-wait). All dataset features have numeric values except for the node status feature that takes a categorical value out of three values: B (behaving), NB (not behaving), and potentially not behaving (PNB). The description of the dataset features is given in Table 1.

Table 2 shows the number of instances for each class in the dataset, while Figure 2 shows the distribution of instances over different types of BHP flooding attacks. This figure is deduced from the dataset.

4.2. Evaluation Measures. The experimental results will be evaluated using four evaluation measures. These measures are precision, recall, F-measure, and accuracy. The following equations show how these evaluation measures are computed:

$$\begin{aligned} \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{recall}\,(\text{sensitivity}) &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ F - \text{measure} &= 2 * \frac{(\text{precision} * \text{recall})}{(\text{precision} + \text{recall})}, \\ \text{accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \end{aligned} \tag{6}$$

where FP is the number of false positives, FN is the number of false negatives, TP is the number of true positives, and TN is the number of true negatives.

4.3. Results and Comparisons. In this section, the experimental results for both the feature selection and classification phases of the proposed approach are given in detail. The average rank score and average merit of features from the IG feature selection method are shown in Table 3 and are based on a 10-fold cross-validation with stratified sampling in order to guarantee that both training and testing sets have the same ratio of classes.

In Table 3, the dataset features are ranked in decreasing order according to their significance to target classes. The reason behind this variation in the feature significance is that the target class has four categorical labels, and for each label, different values for each feature are assigned. Therefore, the rank score from the IG method determines how much each feature contributes to the target class label.

The rank scores in Table 3 show that the "packet received," "10-run-AVG-drop-rate," and "flood status" features have higher scores than all the other features. Thus, the hypothesis that those first three features (packet received, 10-run-AVG drop-rate, and flood status) are more influential and more correlated to the labels of target class will be checked experimentally in the following paragraphs.

Table 1: The description of the dataset features.

No.	Feature name	Feature description
1	Node	It is a numeric feature representing the number of node that sends the data traffic.
2	Utilized bandwidth rate	It is a numeric feature representing the rate of bandwidth used.
3	Packet drop rate	It is a numeric feature representing the rate of packet drop.
4	Reserved bandwidth	It is a numeric feature denoting the initial reserved bandwidth assigned to a given node.
5	Average delay time per sec	It is a numeric feature denoting the average delay time per second for each node. It is also called end-to end delay feature.
6	Percentage of lost packet rate	It is a numeric feature representing the percentage rate of lost packets for each node.
7	Percentage of lost byte rate	It is a numeric feature representing the percentage rate of lost bytes for each node.
8	Packet received rate	It is a numeric feature representing the packet received rate per second for each node based on the reserved bandwidth.
9	Used bandwidth	It is a numeric feature represents the bandwidth used or what each could reserve from the reserved bandwidth.
10	Lost bandwidth	It is a numeric feature denoting the lost amount of bandwidth by each node from the reserved bandwidth.
11	Packet size byte	It is a numeric feature denoting the packet size in bytes allocated explicitly for each node to transmit. For instance, if the data size is 1440 bytes and there are 60 bytes for (IP header 40 bytes) + (UDP header 20 bytes), then all headers will be added to the data size to get 1500 byte as follows: packet size = ((data size 1440 bytes) + (IP header 40 bytes) + (UDP header 20 bytes)) = 1500 bytes.
12	Packet transmitted	This is a numeric feature representing the total packets transmitted per second for each node based on the reserved bandwidth.
13	Packet received	This is a numeric feature representing the total packets received per second for each node based on the reserved bandwidth.
14	Packet lost	This is a numeric feature representing the total packets lost per second for each node based on the lost bandwidth.
15	Transmitted byte	This is a numeric feature representing the total bytes transmitted per second for each node.
16	Received byte	It is a numeric feature denoting the total bytes received per second for each node based on the reserved bandwidth.
17	10-run-AVG-drop-rate	This is a numeric feature representing the rate of average packets that drop for 10 consecutive iterations and runs.
18	10-run-AVG-bandwidth-use	It is a numeric feature representing the average bandwidth that is utilized for 10 consecutive iterations and runs.
19	10-run-delay	This is a numeric feature representing the time of average delay for 10 consecutive (run) iterations.
20	Node status	This is a categorical feature. It is an initial classification of nodes based on the rate of packet drop, used bandwidth, and average delay time per second. The categorical values are B for behaving, NB for not behaving, and PNB for potentially not behaving.
21	Flood status	This is a numeric feature that represents the percentage of flood per node. It is based on the packet drop rate, medium, and high level of BHP flood attack in case behaving (B).
22	Class label	This feature is a categorical feature that represents the final classification of nodes based on the packet drop rate, reserved bandwidth, number of iterations, used bandwidth, and packet drop rate. The categorical values of the class label are NB-no block, block, no block, and NB-wait

Table 2: The number of instances for each class in the OBS network dataset.

Class label	NB-no block	Block	No block	NB-wait	Total
No. of instances	500	120	155	300	1075

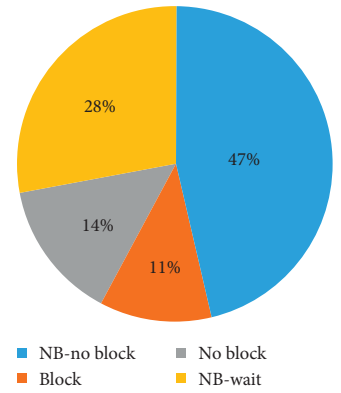

Figure 2: The distribution of instances for each class in the OBS network dataset.

To accept or reject this hypothesis, the evaluation results of the DT method are presented using all features and the combinations of the three selected features. These evaluation results are reported based on the holdout and 10-fold cross-validation techniques. For the holdout technique, the dataset is divided into 75% for training and 25% for testing. Before applying the DT method for classifying the types of BHP flooding attacks and getting the results, an analysis of the DT

Table 3: Rank score of IG feature selection method for all features in the dataset.

Feature name	Feature no.	Average rank	Average merit
Packet received	13	1.8 ± 1.17	1.402 ± 0.079
10-run-AVG-drop-rate	17	4.1 ± 1.64	1.306 ± 0.06
Flood status	21	4.2 ± 1.78	1.309 ± 0.052
Used bandwidth	9	4.3 ± 3.66	1.285 ± 0.191
10-run-AVG-bandwidth-use	18	4.3 ± 1.62	1.282 ± 0.122
Received byte	16	5.3 ± 3.35	1.241 ± 0.163
Packet lost	14	6.7 ± 3.32	1.175 ± 0.15
Packet drop rate	3	9.3 ± 2	1.053 ± 0.083
Packet received rate	8	9.8 ± 1.6	1.018 ± 0.043
Percentage of lost byte rate	7	9.8 ± 1.08	1.018 ± 0.044
Utilized bandwidth rate	2	10 ± 2.32	1.05 ± 0.074
Percentage of lost packet rate	6	10.8 ± 1.08	1.009 ± 0.017
Average delay time	5	12 ± 2.79	0.9 ± 0.19
Reserved bandwidth	10	12.8 ± 1.66	0.899 ± 0.1
Node status	20	14.9 ± 0.3	0.488 ± 0.004
10-run-delay	19	16.2 ± 0.98	0.36 ± 0.104
Full bandwidth	4	17.2 ± 0.6	0.146 ± 0.007
Transmitted byte	15	17.7 ± 0.46	0.146 ± 0.007
Packet transmitted	12	18.8 ± 0.6	0.146 ± 0.007
Node	1	20.1 ± 0.3	0.017 ± 0.006
Packet size byte	11	20.9 ± 0.3	0 ± 0

parameters is investigated to tune and select the best values of these parameters.

Practically, the DT classifier (J48) in Weka performs the pruning process based on a set of parameters, which are the subtree raising, the confidence factor, and the minimal number of objects. The default values of these parameters are true, 0.25, and 2, respectively. The subtree raising is the parameter that can be used to move the node of the tree upwards towards the root that can replace other nodes during the pruning process. Confidence factor is a threshold of acceptable error in data through pruning the DT and this value should be smaller. However, in the proposed approach, the values of subtree raising and confidence factor parameters are set to have the default values. The minimal number of objects is very important parameter to represent the minimal number of nodes in a single leaf. It is used to obtain smaller and simpler decision trees based on the nature of the problem. For tuning the minimal number of objects parameter, we try a set of different values for selecting the best value of this parameter. Figure 3 shows the accuracies of proposed approach at different values of minimal number of objects in the range from 2 to 5. These accuracies are obtained using the holdout technique with 75% training and 25% testing.

As shown in Figure 3, it is clear that the best values of minimal number of objects in a single leaf are 1 and 2 that generate a simple and accurate DT model. The value of this parameter is set to be 2 to make the DT model moderately simple.

Once the values of DT parameters are selected, the evaluation results of the proposed approach are reported in

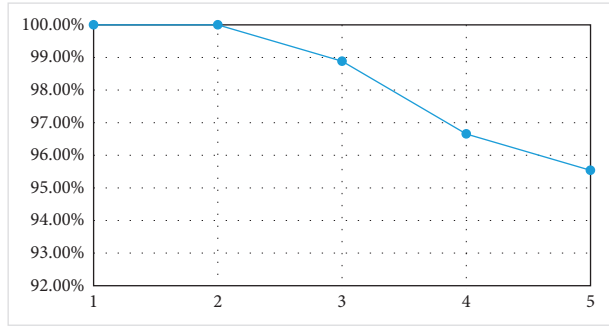

Figure 3: The accuracies of proposed approach at different values of minimal number of objects.

next tables and figures. Table 4 presents the evaluation results of the holdout technique for classifying BHP flooding attacks using all features in the dataset. Figure 4 shows the confusion matrix of classification for the 25% testing set.

Table 5 illustrates the evaluation results of the holdout technique for classifying the BHP flooding attacks using the first three selected features (packet received, 10-run-AVG-drop-rate, and flood status) of the dataset, and Figure 5 shows the confusion matrix of this evaluation result.

From Tables 4 and 5, as well as from Figures 4 and 5, it is clear that the selected features improved the values of evaluation measures for the DT method to classify the BHP flooding attacks. Moreover, for efficiency, detecting attacks using only three features is more efficient for the OBS core switches, which have limited resources.

To validate the evaluation results, other experiments for the DT classification method based on the 10-fold cross-validation technique were conducted using all features and using the first three selected features from the IG feature selection method. Table 6 shows the evaluation results, and Figure 6 shows the confusion matrix for classifying the BHP flooding attacks using all features based on the 10-fold cross-validation technique.

Similarly, Table 7 and Figure 7 present the evaluation results and the confusion matrix, respectively, for classifying the BHP flooding attacks using the first three selected features based on the 10-fold cross-validation technique.

The evaluation results in Tables 6 and 7 and Figures 6 and 7 validate the evaluation results of the 10-fold cross-validation technique that confirm the remarkable performance of the proposed approach. After further investigation, the evaluation results of the DT classification methods using one and two features from the first three selected features are compared with the previous results of the holdout and the 10-fold cross-validation techniques and are shown in Figure 8.

Table 8 shows and summarizes a comparison between the proposed approach and the recent related works on the OBS network dataset. In this comparison, we can see that the proposed work achieves the highest accuracy result with a small number of features compared to all these recent works.

The results presented in Figure 8 and Table 8 prove the hypothesis of the proposed approach that says that the first three selected features using the IG method are more

Table 4: Evaluation results of holdout technique using all features of the dataset.

	Evaluation measure				
Class label	FP rate	Precision	Recall	*F*-measure	Accuracy
NB-no block	0.039	0.950	1.000	0.975	
Block	0.000	1.000	1.000	1.000	
No block	0.000	1.000	1.000	1.000	97.7695
NB-wait	0.000	1.000	0.925	0.961	
Weighted avg.	0.017	0.979	0.978	0.978	

	NB-no block	Block	No block	NB-wait
NB-no block	115	0	0	0
Block	0	31	0	0
No block	0	0	43	0
NB-wait	6	0	0	74

Figure 4: The confusion matrix of classification for the 25% testing set using all features of the dataset.

Table 5: Evaluation results of holdout technique using the first three selected features of the dataset.

	Evaluation measure				
Class label	FP rate	Precision	Recall	*F*-measure	Accuracy
NB-no block	0.000	1.000	1.000	1.000	
Block	0.000	1.000	1.000	1.000	
No block	0.000	1.000	1.000	1.000	100
NB-wait	0.000	1.000	1.000	1.000	
Weighted avg.	0.000	1.000	1.000	1.000	

	NB-no block	Block	No block	NB-wait
NB-no block	115	0	0	0
Block	0	31	0	0
No block	0	0	43	0
NB-wait	0	0	0	80

Figure 5: The confusion matrix of classification for the 25% testing set using the first three selected features of the dataset.

Table 6: Evaluation results of 10-fold cross-validation technique using all features of the dataset.

	Evaluation measure				
Class label	FP rate	Precision	Recall	*F*-measure	Accuracy
NB-no block	0.007	0.992	1.000	0.996	
Block	0.000	1.000	1.000	1.000	
No block	0.000	1.000	1.000	1.000	99.6279
NB-wait	0.000	1.000	0.987	0.993	
Weighted avg.	0.003	0.996	0.996	0.996	

influential and more correlated to the labels of BHP flooding attacks than any of the other features.

4.4. Result Analysis. For analyzing the results and linking the results with conclusion, we show how the proposed feature selection method can improve the model from three different angles: reducing overfitting, improving accuracy, and reducing training and testing (prediction) time.

From the definition of the overfitting problem, it occurs when the training errors are low or very low and the validation errors are high or very high. Therefore, reducing the

	NB-no block	Block	No block	NB-wait
NB-no block	500	0	0	0
Block	0	120	0	0
No block	0	0	155	0
NB-wait	4	0	0	296

Figure 6: The confusion matrix of classification for the 10-fold cross-validation testing sets using all features in the dataset.

Table 7: Evaluation results of 10-fold cross-validation technique using the first three selected features of the dataset.

	Evaluation measure				
Class label	FP rate	Precision	Recall	*F*-measure	Accuracy
NB-no block	0.000	1.000	1.000	1.000	
Block	0.000	1.000	1.000	1.000	
No block	0.000	1.000	1.000	1.000	100
NB-wait	0.000	1.000	1.000	1.000	
Weighted avg.	0.000	1.000	1.000	1.000	

	NB-no block	Block	No block	NB-wait
NB-no block	500	0	0	0
Block	0	120	0	0
No block	0	0	155	0
NB-wait	0	0	0	300

Figure 7: The confusion matrix of classification for the 10-fold cross-validation testing sets using the first three selected features of the dataset.

overfitting problem requires to reduce the gap between the training and validation error. To show how the proposed method can reduce the overfitting problem, we depict the training error against the validation error in Figure 9 with different sets of features, which are ordered according to rank score given in Table 3. The training percentage is set to 75%, and the validation percentage is 25%. We notice that the gap between the training and validation error is decreased as the number of features is decreased until the gap reaches zero approximately when using the three selected features of the proposed method. We also notice that the overfitting problem is eliminated with 14 and 7 features. In our opinion, the overfitting problem is eliminated with 14 and 7 features because of an implicit pruning functionality implemented by the used decision tree algorithm (J48). In addition, it is clear that the accuracy is improved by the three selected features.

To evaluate the efficiency of the proposed feature selection approach, the average time of building and testing the DT model is computed. The DT model is trained on 75% of the dataset which consists of 806 instances and tested on 25% of the dataset which consists of 269 instances. Table 9 shows the computed average time of training and testing the DT model using all features and using our three selected features.

As shown in Table 9, we can see that the DT model has a lower average time for training and testing using our three selected features than using all features. In terms of time complexity, represented by O notation, the overall average time of the DT method is $O(m \times n)$, where m is the number of features and n is the number of instances [40]. Because the number of features in classification problems is limited, the running time will be $O(C \times n)$, where C is a constant time. Therefore, the time complexity of the DT method is $O(n)$ for classification problems. The advantage of the proposed approach is that it reduces the number of features to three features (reducing C), which leads to faster running time compared with using all features. This confirms that the approach is able to detect the attacks more efficiently, especially in congested network with limited computing resources.

We can conclude that reducing the features to three and using the pruning process of the DT classifier helped the proposed approach to reduce the overfitting problem and classify the OBS flooding attacks. Consequently, all performance results clarified the effectiveness and efficiency of the DT model based on selected features to classify BHP flooding attacks. This reveals that the proposed approach is more accurate and suitable for real-time detection in the limited computing capability of OBS core switches.

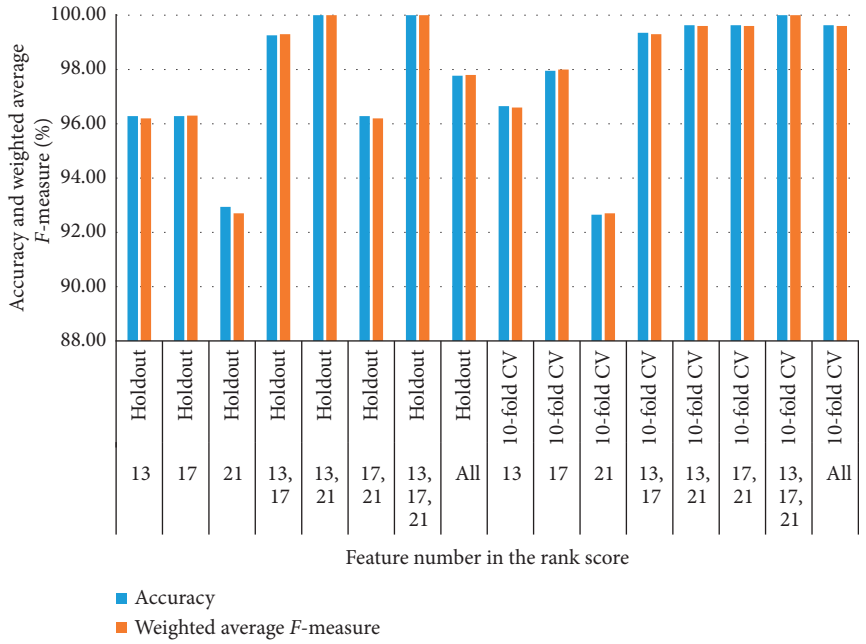

Figure 8: Evaluation results of accuracy and *F*-measure for the DT classification method using a combination of the three selected features compared to all features.

Table 8: Accuracy and number of features of the proposed BHP flooding attack detection approaches using OBS network dataset.

Ref.	Year	Approach	# Of features	Accuracy (%)
[34]	2018	Naïve Bayes	21	79
[21]	2018	Features selection using chi-square testing (CHI) + decision tree for classification	7	87
[34]	2018	Support vector machine	21	88
[34]	2018	K-nearest neighbor	21	93
[27]	2018	Repeated incremental pruning to produce error reduction (RIPPER) rule induction algorithm	21	98
[4]	2019	Features selection using Pearson correlation coefficient (PCC) + semisupervised machine learning with k-mean	8	95.6
[34]	2018	Deep convolutional neural network (DCNN)	21	99
This work	2020	Features selection using information gain (IG) + decision tree for classification	**3**	**100**

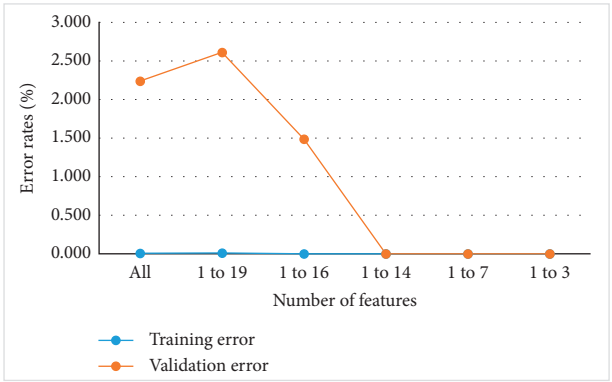

Figure 9: Training and validation error rates with different sets of features.

Table 9: Average time of training and testing the DT model using all features and using our three selected features.

Dataset features	Average time taken to build the model (seconds)	Average time taken to test the model (seconds)
Using all features	0.22	0.03
Using our three selected features	0.01	0.001

5. Conclusion and Future Work

In this paper, an effective and efficient approach using the information gain (IG) feature selection method and the decision tree (DT) classifier is proposed to detect BHP flooding attacks on OBS networks. The approach starts with selecting the most important features of OBS network traffic to improve the accuracy and efficiency of attack detection in OBS switches that have limited resources. A set of experiments is conducted on an OBS network dataset using 10-fold cross-validation and holdout techniques to evaluate and validate the approach. The experimental results demonstrate that the proposed approach can classify the class labels of OBS nodes with 100% accuracy by using only three features. The comparison with recent related works reveals that the proposed approach is suitable for OBS network security in terms of effectiveness and efficiency.

One of the limitations of the proposed approach is the lack of evaluation on more OBS datasets that can be varied in size and types of attacks due to unavailable OBS datasets other than the dataset used in the experiments of this study. Moreover, because the proposed approach is based on the decision tree method for classification, the training time is relatively expensive in case of large training datasets. However, by reducing the number of features of the proposed approach and the emergence of high-speed processors, this limitation is no longer a major problem. In future work, a large set of OBS network data will be collected for further evaluation of the proposed approach and will be made available for researchers in the field. This is due to lack of public OBS network datasets other than the dataset used in this research work.

Data Availability

The OBS-network dataset used in this study is publicly available at the UCI Machine Learning Repository [1].

Conflicts of Interest

The author declares that there are no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Acknowledgments

This study was supported by the Deanship of Scientific Research at Prince Sattam Bin Abdulaziz University.

References

[1] A. Rajab, C.-T. Huang, M. Al-Shargabi, and J. Cobb, "Countering burst header packet flooding attack in optical burst switching network," *Information Security Practice and Experience*, Springer, in *Proceedings of the International Conference on Information Security Practice and Experience*, pp. 315–329, Springer, Melbourne, Australia, December 2016.

[2] T. Venkatesh and C. Murthy, *An Analytical Approach to Optical Burst Switched Networks*, Springer, Boston, MA, 2010.

[3] N. Sreenath, K. Muthuraj, and G. V. Kuzhandaivelu, "Threats and vulnerabilities on TCP/OBS networks," in *Proceedings of the 2012 International Conference on Computer Communication and Informatics*, pp. 1–5, IEEE, Coimbatore, India, January 2012.

[4] M. K. H. Patwary and M. Haque, "A semi-supervised machine learning approach using K-means algorithm to prevent burst header packet flooding attack in optical burst switching network," *Baghdad Science Journal*, vol. 16, no. 3 Supplement, pp. 804–815, 2019.

[5] A. D. A. Rajab, "A machine learning approach for enhancing security and quality of service of optical burst switching networks," University of South Carolina, Columbia, South Carolina, Dissertation, 2017.

[6] S. S. Chawathe, "Analysis of burst header packets in optical burst switching networks," in *Proceedings of the 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, pp. 1–5, IEEE, Cambridge, MA, USA, November 2018.

[7] A. Alazab, M. Hobbs, J. Abawajy, and M. Alazab, "Using feature selection for intrusion detection system," in *Proceedings of the 2012 International Symposium On Communications and Information Technologies (ISCIT)*, pp. 296–301, IEEE, Gold Coast, QLD, Australia, October 2012.

[8] J. Li, K. Cheng, S. Wang et al., "Feature selection: a data perspective," *ACM Computing Surveys*, vol. 50, no. 6, pp. 1–45, 2017, https://arxiv.org/abs/1601.07996.

[9] N. Dessì and B. Pes, "Similarity of feature selection methods: an empirical study across data intensive classification tasks," *Expert Systems with Applications*, vol. 42, no. 10, pp. 4632–4642, 2015.

[10] E. J. Keogh and A. Mueen, "Curse of dimensionality," in *Encyclopedia of Machine Learning and Data Mining*, Springer, Boston, MA, USA, 2017.

[11] S. S. Kannan and N. Ramaraj, "A novel hybrid feature selection via Symmetrical Uncertainty ranking based local memetic search algorithm," *Knowledge-Based Systems*, vol. 23, no. 6, pp. 580–585, 2010.

[12] G. Chandrashekar, F. Sahin, and E. Engineering, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[13] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," *Knowledge and Information Systems*, vol. 34, no. 3, pp. 483–519, 2013.

[14] M. M. Kabir, M. M. Islam, and K. Murase, "A new wrapper feature selection approach using neural network," *Neurocomputing*, vol. 73, no. 16–18, pp. 3273–3283, 2010.

[15] A. Mirzaei, Y. Mohsenzadeh, and H. Sheikhzadeh, "Variational relevant sample-feature machine: a fully Bayesian approach for embedded feature selection," *Neurocomputing*, vol. 241, pp. 181–190, 2017.

[16] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Information Sciences*, vol. 513, pp. 386–396, 2020.

[17] F. A. Khan and A. Gumaei, "A comparative study of machine learning classifiers for network intrusion detection," in *Proceedings of the International Conference on Artificial Intelligence and Security*, pp. 75–86, Springer, New York, NY, USA, July 2019.

[18] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "TSDL: a two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373–30385, 2019.

[19] M. Alqahtani, A. Gumaei, H. Mathkour, and M. B. Ismail, "A genetic-based extreme gradient boosting model for detecting intrusions in wireless sensor networks," *Sensors*, vol. 19, no. 20, p. 4383, 2019.

[20] A. Derhab, M. Guerroumi, A. Gumaei et al., "Blockchain and random subspace learning-based IDS for SDN-enabled industrial IoT security," *Sensors*, vol. 19, no. 14, p. 3119, 2019.

[21] A. Rajab, C.-T. Huang, M. Al-Shargabi, and Networking, "Decision tree rule learning approach to counter burst header packet flooding attack in optical burst switching network," *Optical Switching and Networking*, vol. 29, pp. 15–26, 2018.

[22] Q. Liao, H. Li, S. Kang, C. Liu, and C. Networks, "Application layer DDoS attack detection using cluster with label based on sparse vector decomposition and rhythm matching," *Security and Communication Networks*, vol. 8, no. 17, pp. 3111–3120, 2015.

[23] P. Xiao, W. Qu, H. Qi, and Z. Li, "Detecting DDoS attacks against data center with correlation analysis," *Computer Communications*, vol. 67, pp. 66–74, 2015.

[24] R. Karimazad and A. Faraahi, "An anomaly-based method for DDoS attacks detection using RBF neural networks," in *Proceedings of the International Conference on Network and Electronics Engineering*, vol. 11, pp. 44–48, Noida, India, December 2011.

[25] R. Zhong and G. Yue, "DDoS detection system based on data mining," in *Proceedings of the 2nd International Symposium on Networking and Network Security*, pp. 2–4, Jinggangshan, China, April 2010.

[26] Y.-C. Wu, H.-R. Tseng, W. Yang, and R.-H. Jan, "DDoS detection and traceback with decision tree and grey relational analysis," in *Proceedings of the 2009 Third International Conference on Multimedia and Ubiquitous Engineering*, pp. 306–314, IEEE, Qingdao, China, June 2009.

[27] R. Alshboul, "Flood attacks control in optical burst networks by inducing rules using data mining," *IJCSNS International Journal of Computer Science and Network Security*, vol. 18, no. 2, pp. 160–167, 2018.

[28] J.-H. Chen, M. Zhong, F.-J. Chen, and A.-D. Zhang, "DDoS defense system with turing test and neural network," in *Proceedings of the 2012 IEEE International Conference on Granular Computing*, pp. 38–43, IEEE, Hangzhou, China, August 2012.

[29] J. Li, Y. Liu, and L. Gu, "DDoS attack detection based on neural network," in *Proceedings of the 2010 2nd International Symposium on Aware Computing*, pp. 196–199, IEEE, Tainan, China, November 2010.

[30] V. Akilandeswari and S. M. Shalinie, "Probabilistic neural network based attack traffic classification," in *Proceedings of the 2012 Fourth International Conference on Advanced Computing (ICoAC)*, pp. 1–8, IEEE, Chennai, India, December 2012.

[31] H. Li and D. Liu, "Research on intelligent intrusion prevention system based on snort," in *Proceedings of the 2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering*, vol. 1, pp. 251–253, IEEE, Changchun, China, August 2010.

[32] L. M. Ibrahim, "Anomaly network intrusion detection system based on distributed time-delay neural network (DTDNN)," *Journal of Engineering Science and Technology*, vol. 5, no. 4, pp. 457–471, 2010.

[33] N. Gao, D.-G. Feng, and J. Xiang, "A data-mining based DoS detection technique," *Chinese Journal of Computers*, vol. 29, no. 6, pp. 944–951, 2006.

[34] M. Z. Hasan, K. M. Z. Hasan, and A. Sattar, "Burst header packet flood detection in optical burst switching network using deep learning model," *Procedia Computer Science*, vol. 143, pp. 970–977, 2018.

[35] C. Largeron, C. Moulin, and M. Géry, "Entropy based feature selection for text categorization," in *Proceedings of the 2011 ACM Symposium On Applied Computing*, pp. 924–928, ACM, Taichung, Taiwan, March 2011.

[36] J. Friedman, "A recursive partitioning decision rule for nonparametric classification," *IEEE Transactions on Computers*, vol. C-26, no. 4, pp. 404–408, 1977.

[37] J. R. Quinlan, *C4. 5: Programming For Machine Learning*, Morgan Kaufmann, vol. 38, p. 48, San Mateo, CA, USA, 1993.

[38] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[39] A. Marzano, D. Alexander, O. Fonseca et al., "The evolution of bashlite and mirai IoT botnets," in *Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC)*, pp. 00813–00818, IEEE, Natal, Brazil, June 2018.

[40] J. Bekker and J. Davis, "Estimating the class prior in positive and unlabeled data through decision tree induction," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, Febraury 2018.