Flood Attacks Control in Optical Burst Networks by Inducing Rules using Data Mining

Rabah Alshboul†

Computer Science Department-Al albayt University -Jordan

Summary
One of the key security challenges facing Optical Burst Switching (OBS) network that may influence its resource utilization performance, is flooding Burst Header Packets (BHPs). This problem is usually caused by edge nodes transmitting harmful BHPs that unnecessarily hold network resources causing the network to slowdown or in some cases deny the service. One emerging technology that may reduce this problem is the use of automated classification systems. These systems are based on data mining and are able to automatically label misbehaving nodes that send malicious BHPs before these BHPs impact network resources. This learning technology will not only save time and effort but also increase the accuracy of classification processes. In this paper, we investigate the applicability of rule induction nodes on the hard problem of BHP classification within OBS networks. Specifically, we propose a data mining model that adopts rule induction as a learning strategy to build classifiers that reduce flooding attacks by detecting misbehaving edge nodes early on. Empirical analysis using a recently published dataset reveals that data mining approaches generate promising results with respect to error rate. More importantly, the rule induction approach can detect nodes that are potentially transmitting malicious BHPs more accurately than other approaches such as those that are statistical and probability based.

Key words:
Computer Network; Computer Security; Classification; Data Mining; Flooding BHPs; Probability Models; OBS Network; Rule Induction

1. Introduction

Optical Burst Switching (OBS) is an advanced technology used to improve optical network resources by utilizing the advantages of optical packet switching and wavelength routing [3]. In using OBS for data transmission, burst header packets (BHPs) are usually sent prior to data bursts in order to preserve necessary resources and to ensure maintaining network management resources [19]. However, in many cases, the network gets flooded with harmful BHPs transmitted by potential attackers without sending the BHP's corresponding data burst, thereby wasting network resources. This can cause a notable deterioration of the OBS network performance and may result in a more severe problem named the denial of service [18]. To avoid the problems caused by harmful BHPs it is advantageous to detect edge nodes that are misbehaving and continuously sending these BHPs so that they may be blocked. This may improve the overall performance of the network and reduce flood attacks.

There have been few research works that investigated the flooding attack problem in OBS networks in literature, i.e. [19][21][18] integrated a computer security model that classifies edge nodes based on their historical performance of transmitting BHPs within an OBS network. In their model, domain experts' rules are created primarily by using packet drop rates while running an adjusted NCTUns network simulator. These rules are then employed to classify the nodes and penalize those that are not behaving (transmitting malicious BHPs). In some cases, the nodes get blocked for some time until they modify their behavior at which point the model releases the blockade.

A similar approach based on the statistical analysis of data collected during simulation was proposed by [21]. Using this approach, different rules are created based on the results of statistical analysis. Domain experts were trained to differentiate between edge nodes and identify those that are misbehaving. Empirical analysis revealed that the rules developed were able to reduce the risks of flood attacks and suggested that automating the process of creating rules can indeed boost the detection rate.

Few research studies on BHP flood attacks in OBS networks have been conducted and the majority of these studies rely on human experience and simple statistical analysis. Two major obstacles associated with these studies are:

1) The ways of detecting misbehaving edge nodes is not automated since they rely on the judgement of domain experts, and
2) The detection rate of flood attacks is not high enough

One way to improve the detection of misbehaving edge nodes in OBS networks is to use classification systems derived from techniques such as data mining. Data mining involves discovering important and useful information

Manuscript received February 5, 2018
Manuscript revised February 20, 2018

within databases or datasets to improve decision making [16]. In the context of the flood attack problem, data mining can deal with labeling edge nodes as a typical supervised learning problem in which a classifier is constructed from previous simulations of the OBS network. During simulations, important facts related to edge node performance can be collected, such as packet drop rate, lost bandwidth, packets received, average delay time and utilized bandwidth rate, among others. This data may then be saved in a training dataset. A learning algorithm may then process the training dataset to derive classifiers that may label edge nodes as accurately as possible based on their historical performance. These classifiers can also be utilized by domain experts as well as computer security administrators to verify the results and increase their knowledge and awareness of the OBS network security. This intelligent solution is the concern of this paper.

In this paper, we investigate the utilization of classification algorithms within data mining to minimize the impact of BHP flooding attacks in OBS networks. Specifically, we examine the applicability of the rule induction approach to determine if it can effectively predict misbehaving edge nodes. Rule induction involves constructing rules sets in the format of if-then rules based on search methods from training datasets [22]. We develop a rule induction model that constructs an automated knowledge base to assess domain expert understanding of the way edge nodes are labelled based on their behavior in OBS network. The model adopts Repeated Incremental Pruning to Produce Error Reduction (RIPPER) as a learning method for the rules sets [4].

To achieve the aim of an improved prediction rate of misbehaving nodes, we test and examine our model with three data mining algorithms named Naïve Bayes, Bayes Net and Decision Tables algorithms [6][7][11]. We test these algorithms on a recently developed dataset related to OBS network security [13]. We seek to determine whether the rule induction models are more effective in differentiating between behaving and misbehaving nodes with respect to different evaluation metrics related to classification problems. These metrics include: predictive accuracy, recall, precision, harmonic mean, and rules content.

The remainder of this paper is organized as follows. Section 2 reviews the related research studies and the considered ML approaches. Section 3 is devoted to experimental settings, data description, and results analysis. Lastly, Section 4 will offer concluding remarks.

2. Literature Review

Few research works related to the use of data mining and intelligent systems have been related to combat flood attacks in OBS networks. In this section, we shed light on closely related studies that applied data mining or machine learning methods.

Berral, et al., (2008)[2] employed a probabilistic machine learning mechanism, i.e. NB, to deal with security issues in computer networks, particularly Distributed Denial-of-Service flood attacks. This type of problem deteriorates network performance since users may lose access to the server and others may face access delays. The authors showed that machine learning can be an effective mechanism in discovering data sources that are sending malicious data and blocking them. NB processed data related to the network performance and shared it with other network elements to generate classifiers.

Rajab (2017)[18] proposed a decision tree model based on machine learning (Quinlan, 1993)[17] to detect the different types of edge nodes in OBS network. The decision tree model was generated after processing historical data collected using an NS2 simulator and consisting of over 20 variables. In the model, each path from the root node to each leaf was converted into a rule for easier interpretation. The authors utilized a number of feature selection methods, prior data processing and model construction in order to identify the most effective variables, i.e. Chi Square and Correlation Features Set [12][10]. Empirical results on an OBS simulation dataset [13][19] showed that the tree model was able to accurately classify edge nodes into two class labels. Moreover, the tree model was also able to decompose the class labels further into four possible values using the rules derived.

Coulibaly et al., (2015)[5] investigated denial of service and data burst redirection attacks in OBS network. The authors proposed a security model based on the Rivest-Shamir-Adleman (RSA) encryption technique. Data produced by an NCTUns simulator was employed to evaluate the security model proposed. The experimental results in regards to burst loss ratio demonstrated that the number of data burst redirection attacks was decreased in normal and high traffic scenarios.

Balamurugan & SiV Asubramanian (2014)[1] studied potential simulated solutions related to the distributed denial of service attacks. The authors performed simple simulations to determine the cause of BHP rejection and acceptation rates once attacks are transmitted from the ingress nodes to the destinations. The empirical results indicated that when the size of active flows increase, this decreases the numbers of rejected BHPs.

An anti-flood attack model that can be integrated into the OBS core was proposed by [19]. This model utilizes ingress nodes' historical performance in transmitting BHPs against the behavior of the network, i.e. allocated resources not being used to detect misbehaving ingress nodes. The ingress nodes that continuously transmit malicious BHPs are blocked until they change their behavior (the BHPs reserve resources that are actually used by data bursts). Experimental eV Aluation based on adopting NCTUns

simulator reported that the anti-flood attack model can detect and block malicious ingress nodes. However, the authors suggested improving expert rules based on in-depth data processing using machine learning.

The NB algorithm [6] was employed to classify Internet traffic data types based on features such as data flow length, port number, time elapsed between two successive flows, among others, by [14]. A class variable called traffic flow was added to the dataset to convert the problem into supervised learning and enable NB algorithm build classifiers. When NB was applied a probabilistic classifier was generated that showed the likelihood of each value of the traffic flow variable to occur. The prediction rate was slightly enhanced by the use of the dimensionality reduction method.

Prathibha & Rejimol-Robinson (2014)[15] reviewed the use of artificial intelligence, neural network and statistical analysis techniques to deal with distributed denial-of-service flood attacks. The review was performed mainly in a theoretical context to reveal advantages and disadvantages of each considered method. The authors showed a conceptual artificial neural network model and its primary phases but without implementing the model and evaluating its performance. [20] reviewed the problem of distributed denial-of-service flood attacks in order to seek its potential impact on network behaviour. The authors pinpointed a number of security network vulnerabilities and then suggested possible solutions to prevent them.

Most of the research works conducted on BHP flood attacks were primarily based on the behavior of ingress nodes. Domain experts developed rules either using experience or statistically based on the behavior of certain variables. More importantly, the majority of the articles in this domain investigated distributed denial-of-service flood attacks. The use of data mining detection models is still rare in tackling BHPs flood attacks in OBS networks. There are limited attempts in the use of probabilistic models and recently the adoption of the decision tree model was proposed.

This study will explore a new direction in investigating the classification problems of ingress nodes by utilizing a new rule-based classifier, comparing its performance with other classifiers such as probabilistic models. We believe that data mining has not been explored enough to develop smart solutions to combat flood attacks in OBS networks.

3. The Rule based Model for BHP Flood Attacks

One of the most effective learning approaches in classification is rule induction [8][22]. Using this approach, occurrences in the training dataset are categorized into positive and negative, based on the class values. For instance, for a class C1, all instances that occur with C1 are considered positive instances whereas the remaining instances that occur are considered negative instances to C1. The learning algorithm discovers rules that belong to the different available class labels in the format of If-then, i.e. If age <60 AND Annual_Income > 90000 AND Gender = Male THEN Credit Card=Yes. In each rule, the "If" part (rule's body) contains variables' values in a conjunctive form and the "then" part (the consequent) contains one class value. These rules typically cover instances in the training dataset and should discriminate among instances with respect to class labels.

In learning the classification models from the training dataset, rule induction algorithms such as RIPPER seek for the rule that best predicts part of the training set, then continuing the search for more rules from the remaining unclassified training cases. This process continues until no rules can be discovered. This approach guarantees each data occurrence is covered at least by a single rule.

We propose the model shown in Figure 1, which predicts the type of ingress nodes in OBS networks and hence reduces the risk of BHP flooding. In our model, different simulation runs are collected after running the NCTUn simulator multiple times to collect features related to the ingress nodes and the behavior of the network. These variables are then stored and processed using V A feature selection method [25].

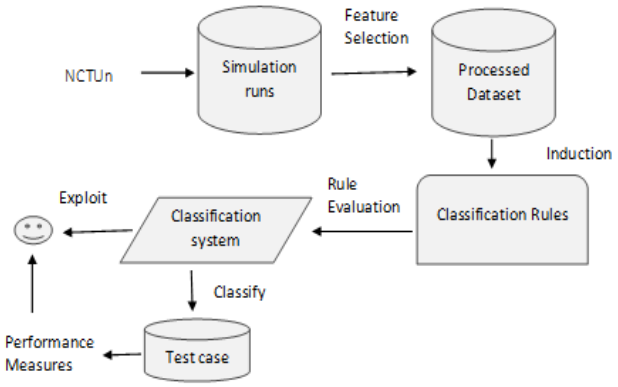

Fig. 1. The proposed data mining framework for flood attacks in OBS network

The V A method ranks features based on their significance within the class label in a training dataset. The key reason for using feature selection is to remove any feature redundancy since multiple features recorded during the simulation runs are dependent on each other. Therefore, using a feature selection technique such as V A discards weak features and prevents them from taking rule in building the classifier. Choosing V A feature selection was based on the fact that it combines the scores of three common filtering methods, namely Chi Square, Information Gain and Correlation Features Set [12][17][10]. Thus, V A method stabilizes the scores of variables in a dataset and reduces result disparities.

We adopt a learning algorithm based on rule-based data mining. Algorithm 1 divides the training dataset into two sets, i.e. growing and pruning. During the training phase, the rules are discovered from growing subsets. Since the initial rules set may overfit the training dataset, the learning algorithm implements a pruning method that simplifies each rule, if necessary, by discarding literals from the rule's body. In using this pruning procedure, each rule discovered is evaluated in the pruning subset, possibly trimming some of the literals in the rule's body. This happens if the test yields a reduction of the error rate in the pruning data subset. The pruning procedure terminates when discarding a literal from the current rule's body does not improve the accuracy of the data subset. The main steps involved in building the set of rules is shown in algorithm 1. The formula utilized for rule pruning is shown in Equation 1.

$$F(p,n) \equiv \frac{pos - neg}{N + P} \quad (1)$$

where neg and pos and are the number of negative and positive data classified by the rule. N and P are the number of negative and positive data in the pruning data subset.

One of the vital procedures invoked by the learning model is the optimization phase of RIPPER in which the initial classification model (set of rules) gets tested to potentially boost predictive power. For each rule in the initial model, the algorithm generates two substitute rules: the replacement and the revision. The former rule is created by growing a new rule from the growing set and then eliminating items from its body until reaching a minimized error rate on the entire model. The revision rule is created by inserting new items into the rule body until reaching an increased error rate. Finally, RIPPER decides which rules should be included in the final classifier, among the initially discovered rule, its revision rule or its replacement rule. Using the minimum description length is a principle criterion.

```
Input: A dataset
Output: A classifier with rules
R: Rule, R_S: Rule_Set, S:Storage
1.  procedure Learn_Rule (pos, neg)
2.  begin
3.  S ← {}
4.  while pos do
5.  split (pos, neg)
6.  R_S ← GrowRule(pos, neg)
7.  R_S ← PruneRule(r, pos, neg)
8.  if Error (R_S) on (pos, neg) >= 50% then
9.  return R_S
10. else
11. S ← R_S
12. Discard data instances covered by r from (pos,neg)
13. endif
14. end while
15. return R_S
16. end
```

Algorithm 1. Learning algorithm based on incremental reduced error pruning

Below are the main distinguishing features of the learning model:

1) The model is formatted by simple If-then rules that are easy to understand by different users, such as Network Administrators, and can be exploited as a knowledge base for flood attacks
2) The model utilizes straightforward metrics to generate rules based on the rule error rates and the minimum description length principle
3) The model results in a reduced number of rules after the optimization procedure is complete, hence the classifiers generated are easy to manage and control by domain experts
4) 4) Highly predictive accuracy classifiers that can improve the manual process of classifying ingress nodes in OBS networks.

3. Empirical Analysis

3.1 Data and Simulation Details

The dataset utilized in the experiments was published recently by [19] at the University of California Irvine data repository [13]. The dataset contains 21 features plus the class variables (four possible values), and over 1,000 data instances. Therefore, the dataset, which represents the problem of classifying edge nodes in an OBS network can be seen as a multi-class dataset. The distribution of class labels in the dataset are 500, 120, 155 and 300 instances for NB-No-Block, Block, No-Block and NB-Wait respectively. NB denotes a non-behaving node.

The dataset was created by running NCTUns simulator on NSFNET topology [24] over 100 times in which certain features such as node number, bandwidth allocated to each node, bandwidth lost by each node, packet drop rate, average delay time and more, were collected. Before running the NCTUns, the following parameters were set according to [19]: Link bandwidth to 1,000 MB/s, Propagation delay to 1 μs, maximum burst length to 1.500 bytes, number of DB channels to 2, number of BHP channels to 1, bit error to 0.0, and transport layer protocol to UDP.

The simulation performed on the NCTUns reflect the real scenarios that may occur on the OBS network in order to minimize biased data. Thus, various different bandwidth to the edge nodes have been allocated during simulation runs so that normal and abnormal cases are taken into consideration. For example, there are simulation runs in which the allocated bandwidth to the node was 1,000 Mbps and decreases to 900 Mbps, then to 800 Mbps, then to 700 Mbps until it reaches a minimum of 100 Mbps. Conversely, there are runs in which the allocated bandwidth to the node starts with 100 Mbps, increases to 200 Mbps, then to 300 Mbps, until it reaches a maximum level of 1,000 Mbps [18].

To ensure that there were BHP flooding cases in the data collection, the load of the network was changed during each simulation run and the malicious edge node was placed randomly at different locations on the topology. Lastly, one or multiple legitimate edge nodes and one malicious node were utilized during each simulation run.

A sample of 4 data transmissions in a simulation run with two edge nodes and just ten features is shown in Table 1 for demonstration purposes[13]. The sample data represents three different situations in which during the first data transmission, node No. 9 is associated with higher packet drop rate than node No. 3 and therefore it was labeled "Block," whereas node 3 was allowed to transmit packets and was labeled, "Non-Behaving-No-Block" (NB-No-Block). However, in data transmission Number 2, node No. 3 has no flood status and so was given higher priority to send packets whereas node No. 9 was blocked given that it has considerable flooding status. Finally, in data transmissions 3&4, node No. 3 was labeled "No Block" since it has no flood status. Node No. 9 was also given permission to transmit data since it was associated with acceptable packet drop rate, in spite of being identified as misbehaving and with flood status greater than zero.

To detect node performance with respect to predictive accuracy of our rule-based model, different classification data mining techniques have been employed. In particular, we have compared with probabilistic and other rule induction approaches on the problem of detecting nodes that may cause flood attacks. For the probabilistic approach, we have selected Bayes Net

Table 1. Sample of four transmissions in a simulation run for 10 features based on the OBS network dataset (Lichman, 2013)

SR #	N #	UBR	PDR	BA	ADTPS	PRR	LB	RB	FS	Label
1	3	0.821038	0.190381	1000	0.004815	0.809619	177.9625	108304320	0.023485	'NB-No Block'
1	9	0.275513	0.729111	100	0.004815	0.270889	72.44875	3529440	0.460725	Block
2	3	0.923707	0.090383	900	0.000633	0.909617	68.664	106489200	0	'No Block'
2	9	0.368775	0.63771	100	0.000592	0.36219	63.1125	4720320	0.439255	Block
3	3	0.905217	0.10867	800	0.000487	0.89133	75.82625	92708760	0	'No Block'
3	9	0.514687	0.484142	100	0.003098	0.505858	48.53125	6590880	0.291742	'NB-No Block'
4	3	0.912327	0.102633	700	0.000423	0.897367	61.371225	81776160	0	'No Block'
4	9	0.555424	0.444076	100	0.004677	0.555924	43.4875	7243200	0.122299	'NB-No Block'

Feature Descriptions

- **SR**: Simulation Run Number
- **N #**: Sending Node Number
- **UBR**: Utilized Bandwidth Rate
- **PDR**: Packet Drop Rate
- **BA**: Bandwidth Allocated
- **ADTPS**: Average Delay Time Per Second
- **PRD**: Packet Received Rate
- **LB**: Lost Bandwidth
- **RB**: Received Byte
- **FS**: Computed Flood Status
- **Label**: The class label assigned by domain expert

and Naïve Bayes algorithms [6][11] and for the rule induction we have selected Decision Table [7]. The key task for the data mining algorithms is to construct classification systems that are able to classify ingress nodes into one or more of the available labels in a given training dataset. This system will provide benefits in two ways:

a) It will utilize improvement of resource utilization and management in OBS networks, and
b) It provides new patterns and classification models that can empower network administrators, computer security specialists and researchers interested in network management and security within OBS networks.

All experiments to build classificaiton systems have been conducted using the Waikato Environment for Knowledge Analysis (WEKA) machine learning Java platform [9]. WEKA is a common open source platform that contains developed algorithms related to dimensionality reduction, classification, clustering, association rule, data visulaization, and distributed data analytics. In all data processing experiments, the ten-fold cross validation technique was adopted to build and test classifiers. This testing technique is common in supervised learning in which the input dataset is divided into 10 blocks. The data mining algorithm is trained on 9 blocks and tested on the one remaining data block. The same process gets repeated ten times to produce one single average error rate [23]. The experiments were performed on a computing machine with 2.0 GHz processor and 8 GB RAM.

3.2 Results Analysis

The dataset used in experiments testing data mining algorithms is a multi-class classification data tool. For that reason, confusion matrix evaluation measures that complement the nature of the problem have been adopted. Using the confusion matrix displayed in Table 2, precision, predictive accuracy, and recall[26] are utilized to guage the performance of learning in the considered algorithms for building BHP anti-flood models. Table 2 contains the possible decisions for a test instance during the classification process of edge nodes. Predictive accuracy (Equation 2)[26] is one of the known evaluation measures in supervised learning the computes the number of test data that have been allocated the right label by the classification algorithm from the total number of test data. Recall (Equation 3)[26] corresponds to the rate of test data that have been classified to the correct class such as (C1) from the total of test data that are actually linked with class C1. Lastly, precision is the rate of classified test data to C1 to the total predicted C1 cases (Equation 4)[26]. Lastly, the harmonic mean metric (F1)[26] takes into account both recall and precision. F1 estimates the weighted average of precision and recall (false positives & false negatives).

$$Accuracy = \frac{|TP + TN|}{|TP + TN + FP + FN|} \quad (2)$$

$$Recall = \frac{|TP|}{|TP + FN|} \quad (3)$$

$$Precision = \frac{|TP|}{|TP + FP|} \quad (4)$$

$$F1 = 2 \times \frac{|Precisoin \times Recall|}{|Precisoin + Recall|} \quad (5)$$

Figure 2 shows the classifiers' predictive accuracies generated by data mining algorithms on the OBS network dataset. The accuracy figures clearly demonstrate that rule induction classifiers produce better accuracy rates than probabilistic based classifiers (Bayes Net, Naïve Bayes) at least on the dataset we consider. In particular, our model outperformed Naïve Bayes and Bayes Net in terms of accuracy rates by 29.77% and 12.47% respectively. This significant difference can be attributed to the learning mechanism employed, i.e. RIPPER, in which the algorithm not only searches for rules that maximize the data coverage but also prunes them in an excessive manner. The learning algorithm splits the training dataset into growing and pruning subsets and learns the rules from the growing subset. Then the set of rules are tested on the pruning data subset to remove rules overlapping as well as rules with a high potential error rate. These procedures ensure fewer redundant rules in the final classifier. On the other hand,

Table 2: Confusion matrix for a two class classification problem

	Predicted Class	
Domain Assigned Class	Yes	No
Yes	True Positive (TP)	False Positive (FP)
No	False Negative (FN)	True Negative (TN)

despite probabilistic classifiers being computationally efficient since they estimate the likelihood of which class can be assigned to each test data, using simple estimated probabilities researchers assumed that classes are independent and have no overlap. However, in the case of the BHP flooding attack problem, classes do overlap in the training instances, as we will see shortly. Therefore many of the test data are misclassified by Naïve Bayes and Bayes Net probabilistic algorithms. This explains the larger error rates produced by these algorithms on the OBS network dataset.

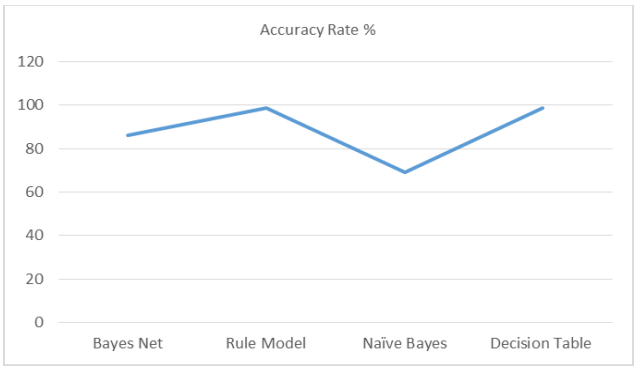

Fig. 2. Classification accuracies in % generated by the data mining algorithms

Figure 3 displays the recall and precision rates produced by the four data mining algorithms on the OBS network dataset. Precision denotes the number of accurately predicted test instances from all that have been predicted whereas recall denotes the number of accurately predicted instances in all instances intended to be accurately predicted. The recall and precision rates are consistent with the predictive accuracy rates generated earlier. To be exact, Naïve Bayes classifiers are associated with the lowest recall and precision rates when compared with those of the other data mining algorithms. The reason for the low rates of precision and recall for Naïve Bayes is that this probabilistic classifier fails largely to predict the values for two specific class labels (NB-Wait, NB-No-Block). These large misclassifications are attributed to the similarity in the situations in which these class labels are assigned to the training cases during the data collection phase (simulation runs). Edge nodes transmitting data instances that belong to these two class labels are considered misbehaving since they have flood status values greater than zero and due to having a large packet drop rate. This behavior creates a burden on Naïve Bayes during the classification phase in which the algorithm gets confused and hence frequently misclassifies test data among these two class labels. To confirm this theory, we listed the confusion matrix figures produced by Naïve Bayes below.

According to the figures listed in Naïve Bayes confusion matrix, there are 169 test instances that should belong to NB-No-Block class label that have been incorrectly predicted to the other class labels, especially to NB-Wait. In addition, there are 145 incorrectly predicted test instances that should belong to NB-Wait. These test instances have been largely assigned the wrong class label, i.e. NB-No-Block. Overall, the significant difference in recall and precision rates for Naïve Bayes was due to the high false positives and true negative rates caused because of the overlapping between NB-Wait and NB-No-Block in the data instances.

```
=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 331  34  15 120 |   a = NB-No Block
   0 101   0  19 |   b = Block
   0   0 155   0 |   c = No Block
  69  76   0 155 |   d = NB-Wait
```

Since we have an imbalanced dataset with respect to class labels, we considered a measure that takes into account both precision and recall and it is called the F1 measure (Figure 4). The F1 rates derived by the rule induction algorithms are high and individual scores of F1 per class label is consistent for both our learning model and Decision Table algorithm. This indicates that these two algorithms perform well in imbalanced datasets alongside balanced datasets. On the other hand, we noticed that F1 scores for class labels in Naïve Bayes fluctuates. For example, the F1 score obtained by the No-Block class by Naïve Bayes was 95.4%, which is pretty good. However, the F1 rates obtained by the same algorithm on the NB-Wait was 52.2%, which is relatively low. The fluctuation in the F1 scores per class for the probabilistic algorithms is due to the fact that these algorithms work well in situations when class labels are different in data instances, and poor when there are similarities among their data instances.

Overall, the results obtained in regards to recall, precision and F1 pinpoint to a similarity in assigning class labels for two particular classes (NB-Wait, NB-No Block) during the construction of the dataset. This issue has a clear impact on the performance of probabilistic classifiers such as Bayes Net and Naïve Bayes. The reason these algorithms' performance was affected is attributed to Bayes Theorem assumption of class independence, which is not valid in the case of the BHP flood attack problem. However, this issue was overcome by rule induction algorithms. It seems that rule induction algorithms are more tolerant toward noisy and unbalanced datasets such as the one considered in this paper and therefore are more predictive. The considered data mining algorithms seem comfortably able to predict edge nodes that are associated with block and no-block class labels. Nevertheless, probabilistic algorithms struggles in classifying edge nodes that are labeled NB-Wait or NB-No Block as explained earlier. One possible way to improve the probabilistic classifiers' performance is by collecting more data for these two class labels during the simulation run.

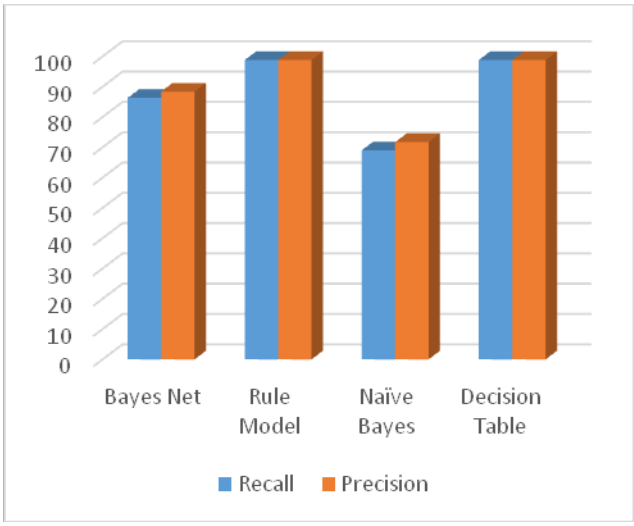

Fig. 3. recall and precision rates in % generated by the data mining algorithms

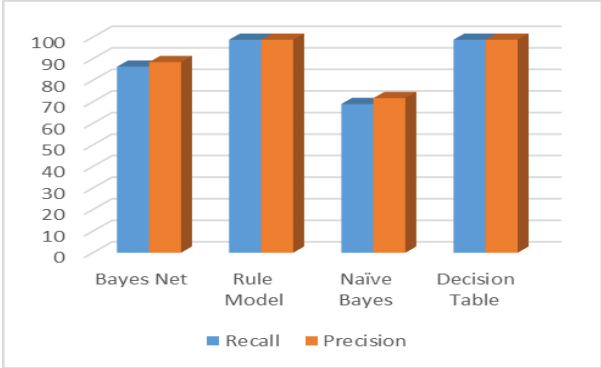

Fig. 4. F1 rates in % generated by the data mining algorithms

4. Conclusions

OBS networks can be seen as the upcoming technology for the Internet, enhancing network resource utilization when contrasted with other networks. Despite the economic benefits offered by an OBS network, it becomes susceptible when malicious BHPs are transmitted by misbehaving nodes, causing flood attacks and in some cases denial of service. This problem has clear negative implications on OBS network performance and thus it is advantageous to block edge nodes that might cause flood attacks as early as possible. In this paper, a classification system based on induced automated rules has been proposed. This system adopts learning methods in data mining that ensure only highly effective rules that detect misbehaving edge nodes are developed. The model was trained on a flood attack dataset that was gathered using an NCTUns network simulator and consists of over twenty different variables

associated with network performance as well as the edge nodes. Empirical results using four different data mining algorithms on the dataset showed that rule induction classification systems are more accurate than those of probabilistic techniques in predicting the type of edge nodes that will likely allow BHP flood attacks. To be exact, a RIPPER rule induction algorithm was able to derive classifiers with predictive power above 98% whereas Naïve Bayes and Bayes Net derived classifiers with 69% and 85% respectively from the dame dataset. The recall, precision and harmonic mean results also revealed that the performance of the derived classification systems by rule induction approach are better than a probabilistic approach, at least for the flood attack problem, in OBS networks. Moreover, the classification systems results in simple useful rules that is easy to understand and exploit by the end-user. In the future, we will investigate using multiple class labels with voting mechanisms to enhance the predictive rate of the classification performance of edge nodes.

References

[1] Balamurugan A.M. and SiV Asubramanian A. (2014) Modeling the Performance of DDoS Attack in Optical Burst Switched Networks. Aust. J. Basic & Appl. Sci., 8(18): 479-482, 2014

[2] Berral, J.L., Poggi, N., Alonso, J., GaV Aldà, R., Torres, J., and Parashar, M., (2008) Adaptive distributed mechanism against flooding network attacks based on machine learning, Proceedings of the 1st ACM Workshop on Workshop on AISec, October, pp.43–50.

[3] Chen, Y., Qiao, C., and Yu, X.: Optical burst switching: A new area in optical networking research. IEEE Network 18(3), 16–23 (2004).

[4] Cohen, W., 1995. Fast Effective Rule Induction. In Proceedings of the Twelfth International Conference on Machine Learning. Tahoe City, California, 1995. Morgan Kaufmann.

[5] Coulibaly, Yahaya, et al. "Secure burst control packet scheme for Optical Burst Switching networks." Broadband and Photonics Conference (IBP), 2015 IEEE International. IEEE, 2015.

[6]

[7] Duda, R. O., and Hart P. E., (1973). Pattern Classification and Scene Analysis. New York: John Wiley & Sons.

[8] Friedman, N., Geiger, D. and Goldszmidt, M. (1997) Bayesian Network Classifiers. Machine Learning - Special issue on learning with probabilistic representations, 29(2-3), pp.131-63.

[9] Fürnkranz J., and Widmer, G., (1994). Incremental reduced error pruning. In Machine Learning: Proceedings of the Eleventh Annual Conference, New Brunswick, New Jersey,1994. Morgan Kaufmann.

[10] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I., (2009). The WEKA Data Mining Software: An Update. In SIGKDD Explorations, 11(1).

[11] Hall, M., (1999) Correlation-based Feature Selection for Machine Learning. Thesis, department of computer science, Waikaito University, New Zealand.

[12] Kohavi R. (1995) The Power of Decision Tables. In: 8th European Conference on Machine Learning, 174-189, 1995.

[13] Liu, H., and Setiono, R. (1995). Chi2: Feature Selection and Discretization of Numeric Attribute. Proceedings of the Seventh IEEE International Conference on Tools with Artificial Intelligence, November 5-8, 1995, pp. 388.

[14] Lichman, M., (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[15] Moore, A. W., and Zuev, D., (2005) Internet traffic classification using Bayesian analysis techniques. In ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS) 2005, Banff, Alberta, Canada, June 2005, vol. 33, No 1, pp. 50-60.

[16] Prathibha, R. C., and Rejimol Robinson, R. R., (2014)A Comparative Study of Defense Mechanisms against SYN Flooding Attack. International Journal of Computer Applications (0975 – 8887). Volume 98– No.18, July 2014.

[17] Qabajeh, I., Thabtah, F., Chiclana, F. (2015) A dynamic rule induction method for classification in data mining. Journal of Management Analytics, 2(3): 233–253

[18] Quinlan, J., (1993) C4.5: Programs for machine learning. San Mateo, CA: Morgan Kaufmann.

[19] Rajab, A., (2017) Burst Header Packet (BHP) flooding attack on Optical Burst Switching (OBS) Network Data Set. PhD dissertation, University of California Irvine Data Repository, 2017.

[20] Rajab, A., Huang CT, Al-Shargabi M., Cobb J (2016) Countering Burst Header Packet Flooding Attack in Optical Burst Switching Network. ISPEC 2016: Information Security Practice and Experience pp 315-329.

[21] Shakhov V. (2012) Dods flooding attacks in obs networks. 2012. 2012 7th International Forum on Strategic Technology (IFOST). Tomsk, Russia.

[22] Sliti, M., and Boudriga, N., (2015) BHP flooding vulnerability and countermeasure. Photonic Network Communications, 29(2), pp.198-213.

[23] Thabtah, F., Qabajeh, I., and Chiclana, F., (2016) Constrained dynamic rule induction learning. Expert Systems with Applications 63, 74-85.

[24] I.H. Witten, E. Frank, M.A. Hall, C.J. Pal, Data Mining: Practical Machine Learning Tools and Technique, Morgan Kaufmann (2016).

[25] NSFNET : [Online]. Available: http://nsl.csie.nctu.edu.tw/nctuns.html [accessed January 09, 2018].

[26] Kamalov, F., and Thabtah, F.(2017).A feature selection method based on ranked vector scores of features for classifcation. Ann, Data Sci.(2017),1-20. Berlin: Springer Berlin Heidelberg.

[27] https://en.wikipedia.org/wiki/Confusion_matrix