ISSN 2007-9737

Detection of Flooding Attack on OBS Network Using Ant Colony Optimization and Machine Learning

Mohamed Takieddine Seddik[1], Ouahab Kadri[2], Chakir Bouarouguene[1], Houssem Brahimi[1]

[1] University of Batna 2, LASTIC, Faculty of MI,
Department of Computer Science,
Algeria

[2] University of Batna 2, Laboratory of Automation and Manufacturing Engineering,
Algeria

{m.seddik, o.kadri }@univ-batna2.dz,
{chakir.bouarougueneb12, houbrahimi05}@gmail.com

Abstract. Optical burst switching (OBS) has become one of the best and widely used optical networking techniques. It offers more efficient bandwidth usage than optical packet switching (OPS) and optical circuit switching (OCS).However, it undergoes more attacks than other techniques and the Classical security approach cannot solve its security problem. Therefore, a new security approach based on machine learning and cloud computing is proposed in this article. We used the Google Colab platform to apply Support Vector Machine (SVM) and Extreme Learning Machine (ELM)to Burst Header Packet (BHP) flooding attack on Optical Burst Switching (OBS) Network Data Set.

Keywords. Optical burst switching, support vector machine, extreme learning machine, burst header packet, cloud computing.

1 Introduction

Optical burst switching (OBS), as the name suggests, is a dynamic data switching technique. It is hybridization between full optical packet switching (OPS) and optical circuit switching (OCS). In this technique, the first step before sending the data, we send the header independently in a reserved optical channel [12]. The name of this step is delayed reservation. It has become the basic technique to construct the future technology of network communication. Today, optical media is an important element in telecommunications since it offers high bandwidth. Besides, Optical Burst Switching (OBS) significantly optimizes bandwidth usage and does not require the use of optical memory. It is based on data transmission in the form of large packets or bursts (burst) preceded by a Burst Header Packet (BHP) as a signaling header. The time (offset time, OT) separating the data header serves for the conversion and processing of the header at each intermediate node. Therefore, the use of memory is not essential for the routing of data. As we said OBS ensures the burst of traffic. Moreover, its effectiveness is comparable to other switching techniques. Flood attacks represent a key challenge for OBS in terms of quality of service. BHP flood attack is based on sending continuous BHP to a server without the rest of the data. Therefore, all Wavelength-division multiplexing (WDM) channels are reserved by the emitted BHPs.

The consequence of this attack is denied of services (DoS) [13]. It was initially targeted to environments of TCP/IP connections networks. It prevents receiving communications during the attack since all the channels are waiting for data that never arrived.

This work consists in proposing a new security approach based on a model of nodes classification. This model is established to prevent BHP flood attacks.

The learning database used comprises four classes. These classes depend on several parameters that characterize the bandwidth states and Packet types.

doi: 10.13053/CyS-25-2-3939

The rest of our paper is divided as follows. Section 2 presents the related works of flooding attacks on the OBS network. Section 3 presents the main lines of the problem of flooding attacks in OBS. Section 4, presents some material and methods related to our studies. Section 5summarizes the different stages of our approach. In Section 6, our approach is evaluated through the BHP flooding attack on the OBS network dataset. Finally, we give a general conclusion and an overview of our future work.

2 Related Works

In the literature, several solutions have been proposed to solve the security problem of OBS networks. In this section, we summarize the most relevant work related to this issue.

Rajab et al. [10] have used a decision tree rule learning approach to counter burst header packet flooding attacks in the Optical Burst Switching network. They proposed a decision tree-based architecture as an appropriate solution. It contains a learning algorithm that extracts rules from tree models using data processed from several simulation runs. The rules can classify the Misbehaving edge nodes into four sub-class labels with 87% accuracy, including Misbehaving-Block (Block), Behaving-No Block (No Block), Misbehaving-No Block (M-No Block), and Misbehaving-Wait (M-Wait).

Yayah et al. [9] have proposed an Intelligent Offset Time (IoT) algorithm that adapts offset time based on the condition of the network and the traffic. Burst size, destination and burst queuing delay are used as parameters of IOT's fuzzy input. This proposed algorithm is created contrary to adaptive offset time algorithms and the conventional offset time.

Hasan al. [1] have proposed a deep convolutional neural network (DCNN) model to expect BHP flooding of Optical Burst System (OBS) and they have compared the performance with other machine learning techniques such as Naïve Bayes, SVM, and KNN. Moreover, to estimate the performance of their model, they have used a large set of standard performance metrics generally used in the design of classification problems.

The metrics they have used include Classification Accuracy, Sensitivity, Precision, Specificity, Negative Predictive Value, False Positive Rate, False Negative Rate, F1-Score and Error Rate of Classification.

Tang et al. [14] have proposed a Distributed Denial-of-Service (DDoS) attack situation assessment method via an optimized cloud model based on the influence function. Firstly, according to the state change characteristics of the IP addresses, which are accessed by new and old users respectively, they have defined a fusion feature value. Then, based on this value, they have established a V-Support Vector Machines (V-SVM) classification model to analyze network flow for identifying DDoS attacks. Secondly, according to the change of new and old IP addresses, they have proposed three evaluation indexes.

And, they have proposed an index weight calculation algorithm to measure the importance of different indexes. According to the fusion index, which is optimized by the weighted algorithm, they have defined the Risk Degree (RD) and calculate the RD value of each network node. Then they have obtained the situation information of the whole net work according to the RD values, which are from each network node with different weights. Finally, the whole situation information is classified via a cloud model to quantitatively assess the DDoS attack situation.

Ismail et al. [2] have proposed a model to detect flooding based DoS attacks in a cloud environment. It has been suggested consisting of three phases. (1) The first phase is to model the normal traffic pattern for baseline profiling and (2) the second phase is the intrusion detection processes and (3) finally the prevention phase. The used method for detection is the covariance matrix mathematical model.

The studies cited above consider the problem of security as a simple phenomenon. However, in reality, it is a complex and dynamic problem. It is complex since it depends on several factors. It is dynamic since several scenarios may lead to the same result. This study addresses this problem in another way. The proposed approach combines several techniques that offer a good precision of classification, optimization of parameters and speed of calculation.

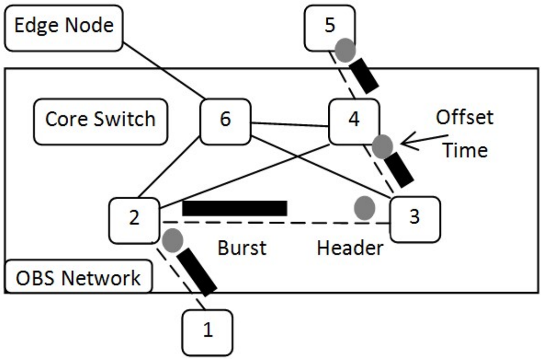

Fig. 1. Optical Infrastructure

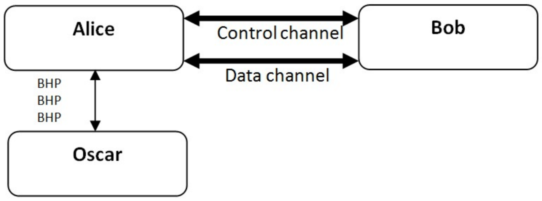

Fig. 2. The scenario of BHP flooding attack

3 The Problem of Flooding Attacks in OBS

3.1 Optical Burst Switching (OBS) Network

Today, the optical medium is important in telecommunications due to its high bandwidth. In addition, Burst Switching - Optical Burst Switching (OBS) - optimizes in an important way the bandwidth usage and does not require optical memory. It is based on dividing the data into large packets or bursts (burst). Before sending those bursts, it reserves the optical channel by sending a burst header packet (BHP). It contains the time (offset time, OT) [8]. It unravels the data header serves for the conversion and processing of the header at each middle node. Therefore, the use of memory is not necessary.

3.2 Flooding Attacks

The result of a flooding attack is a denial of service (Dos). It applies as part of the TCP protocol and consists of sending a succession of synchronization requests to the target (BHP).

Firstly, the attacker sent a huge amount of Burst header packets (BHP) to the server for reserving the optical channel. Secondly, the server sent back connection acceptance and waiting for the bursts [8]. The waiting time is set during the server configuration and it depends on the latency of the network. Finally, the attacker did not send them. Consequently, the resources will be reserved waiting for the bursts, and with several repetitions of previous incomplete connections, the server will not be able to open other connections. That leads to a denial of service.

The weak point of OBS against flooding attack is the release of channel reservation with both implicit and explicit types. In an implicit release, we add the length of burst and the offset time to BHP. On the other hand, in explicit mode, the reservation ends when the node receives an end message (REL). This message is sent after the burst.

In conclusion, the successful repeated transmission of BHP (flooding attack) performs the network overload and denial of service DOS.

4 Material and Methods

4.1 Fault Diagnosis

The main purpose of a diagnostic system is to determine whether the system operation is normal or abnormal. If a malfunction has occurred, the diagnostic system must determine with great precision the affected parts, the type of malfunction and the actions to be performed to repair the fault [3].

For some complex systems, it is very hard to find a mathematical typical model. To deal with this problem, the connection data are used to build a classification model.

Each record is shown as an array of values with category identification.

4.2 Cloud Computing

Nowadays, there is a considerable development in the field of cloud computing. Several companies have invested in this area, which has led to the creation of several incompatible applications.

Therefore, this incompatibility has caused a limit in interoperability. The ideal solution is to create standards. Thus, developers follow current standards when creating new applications. In keeping with this solution, we have developed our application by respecting the current standards.

Cloud computing is mainly based on virtualization. it allows developers to create the network and its different virtual machines. The creation of several different platforms also causes the problem of interoperability.

For example, if our application requires the addition of machines with a specific platform. Thus, the deployment of the new structure takes a considerable amount of time.

Here are some suggestions to solve this problem.

Open Virtual Machine Format (OVMF) is a storage standard that supports multiple virtualization platforms [5]. It helps to ensure portability, integrity, and automate the installation and configuration phases of virtual machines.

Open Grid Forum (OGF) is a community of developers of Open Cloud Computing Interface (OCCI) which is a set of protocols and interfaces for application deployment and network management [6].

The Storage Networking Industry Association Cloud Data Management Interface (SNIA CDMI) is interested in user interaction with data and data encryption.

Any cloud computing application is characterized by five essential properties:

- The availability of services is dependent on the needs of the user.
- The system is remotely tuned by the user using an easy-to-use interface.
- Cloud servers use a digital transmission network capable of delivering high data rates and providing fast access from any location.
- The use of multiple storage servers to save copies of the same information, and it allows the user to choose the closest server to its geographical area and consequently offer the user quick access to information.
- The addition or the suppression of a machine is realized with the creation of a small script that allows the supplier to invoice the customer according to the duration and the number of resources used. A stopped processing unit is not charged.

There are several classifications of cloud applications but here we cite one that is based on four different models:

- In Private Cloud, Infrastructure is placed, used and managed by a single company.
- In Community Cloud, the infrastructure is shared by several organizations that have a strong relationship with each other and are managed by organizations or by a third party.
- In Public Cloud, the infrastructure is accessible to the public by paying a usage bill.
- Hybrid Cloud is a combination of previous models.

4.3 Neural Networks

Extensive attention has been revealed in the literature in the use of neural networks for the problem of fault diagnosis.

Artificial neural networks have been proposed for many different problems such as classification and function approximation problems.

Several neural network types have been used to solve the problem of fault diagnosis. We can differentiate two dimensions of the use of neural networks; the activation function of the network such as sigmoidal, radial basis and so on; and the learning approach such as supervised and unsupervised learning.

When we use a supervised learning mode in multilayer neural networks, the model is configured in the logic that the problem is optimized to the estimation of the connection weights. The values of connection weights are calculated by training the network and the difference between the desired and computed values is used to guide the search. This supervised learning represents the appropriate way to realize the fault diagnosis system because the networks can give a good classification.

Unsupervised learning uses estimation techniques. This type of neural network is called self-organizing. We do not have any learning through the procedure of self-organization, and, it is only needed to present a set of motivation observations frequently to the input layer of the network.

In our study, we have used TensorFlow. It is a Machine Learning Library Created by the Google Brain Team in 2011 [4]. It is a collection of Deep Learning features. It is an open-source application. It allows for creating software components in graphical forms. The chart nodes represent the mathematical operations, and the borders represent multidimensional data arrows communicated between them.

4.4 Support Vector Machines

In our wrapper approach, we have used SVM as a classifier. SVM is an attractive learning algorithm first introduced by Vapnik [15]. It has a competitive advantage Compared to neural networks and decision trees.

Given a set of data S ={(x_1, y_1),…,(x_i, y_i),…,(x_m, y_m)}. Here $x_i \in R^N$ is features vector and $y_i \in$ {-1, +1} is a class label. The goal of the SVM is to find a_n of the form:

$$w\varphi(x) + b = 0 \;\; with \;\; y_i(w\varphi(x_i) + b) \geq 1 - \xi_i \, . \tag{1}$$

That separating the S set of training data into two classes (positive and negative) (figure 5). In general, S cannot be partitioned by a linear hyper-plane.

However, S can be transformed into a higher dimensional feature space for making it linearly separable.

The mapping $\varphi(x)$ need not be computed explicitly; instead, an inner product Kernel of the form:

$$\mathrm{K}(\mathrm{x_i}, \mathrm{x_j}) = \varphi(\mathrm{x_i}).\varphi(\mathrm{x_j})\, . \tag{2}$$

To solve the optimal hyperplane problem, we can construct a Lagrangian and transform into the dual. Then, we can equivalently maximize:

$$\sum_{\mathrm{i}=1}^{\mathrm{m}} \alpha_{\mathrm{i}} - \frac{1}{2}\sum_{\mathrm{i,j}=1}^{\mathrm{m}} \alpha_{\mathrm{i}}\alpha_{\mathrm{j}}\mathrm{y_i}\mathrm{y_j}\,\mathrm{K}(\mathrm{x_i}, \mathrm{x_j})\, , \tag{3}$$

subject to:

$$\begin{gathered}\sum_{i=1}^{m} a_i \mathrm{y_i} = 0\, , \\ 0 \leq a_i \leq C\, .\end{gathered} \tag{4}$$

For a test example z, we define the decision function as follow:

$$sign\left(\sum_{\mathrm{i}=1}^{\mathrm{m}} \alpha_{\mathrm{i}}\mathrm{y_i}\,\mathrm{K}(\mathrm{z_i}, \mathrm{z}) + b\right), \tag{5}$$

where:

– w is the weight vector.

– b is the bias term.

– C is the punishment parameter.

– α is the Lagrange multiplier.

4.5 Ant Colony Optimization (ACO)

Ant colony optimization (ACO) is based on the cooperative behavior of real ant colonies, which can find the shortest path from their nest to a food source.

ACO algorithms can be applied to any optimization problems that can be characterized as follows [7]:

1. A finite set of components C = { c_1, c_2, …, c_N} is given.

2. A finite set of L of possible connections/transitions among the elements of C is defined over a subset C' of the Cartesian product C×C, L={c_ic_j}|(c_i, c_j) ∈ C'}, $|L| \leq N^2_{c'}$.

3. For each $l_{c_ic_j} \in$ L a connection cost function $J_{c_ic_j} \equiv J$ ($l_{c_ic_j}$, t), possibly parameterized by some time measure t, is defined.

4. A finite set of constraints $\Omega \equiv \Omega$(C, L, t) is assigned over the elements of C and L.

doi: 10.13053/CyS-25-2-3939

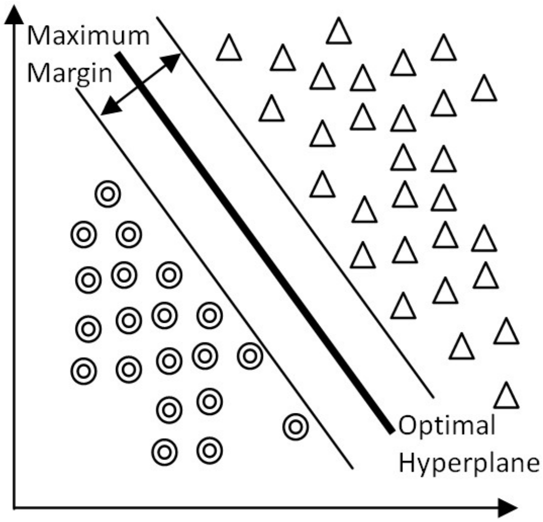

Fig. 3. Two-Class SVM used in linear classification

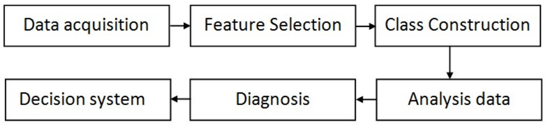

Fig. 4. The architecture of a classification model for detection of flooding attack on the OBS network

5. The states of the problem are defined in terms of sequences $s = (c_i, c_j,\dots,c_k, \dots)$ over the elements of C or of L. S' is a subset of S. The elements in S' define the problem's feasible states.

6. A neighborhood structure is assigned as follows: the state s_2 is said to be a neighbor of s_1 if s_1 and s_2 are in S and the state S_2 can be reached from s_1 in one logical step, that is, if c_1 is the last component in the sequence determining the state s_1, it must exists $c_2 \in C$ such that $l_{C1C2} \in L$ and $s_2 \equiv \langle s_1, c_2 \rangle$.

7. A solutionΨ is an element of S' satisfying all the problem's requirements. A solution is said multi-dimensional if it is defined in terms of multiple distinct sequences over the elements of C.

8. A cost $J_\Psi(L, t)$ is associated with each solutionΨ. $J_\Psi(L, t)$ is a function of all the costs J_{CiCj} of all the connections belonging to the solution.

It is worth mentioning that ACO makes a probabilistic decision in terms of the artificial pheromone trails and the local heuristic information. This allows ACO to explore a larger number of solutions than greedy heuristics. Another characteristic of the ACO algorithm is the pheromone trail evaporation, which is a process that leads to decreasing the pheromone trail intensity over time. Pheromone evaporation helps in avoiding the rapid convergence of the algorithm towards a sub-optimal region.

In the next section, we present our proposed approach and explain how it is used for the detection of flooding attacks on OBS.

5 Detection of Flooding Attack on OBS Network

We will explain in detail the different stages of our approach. Figure 4 summarizes these steps.

Our detection system consists of 5 main steps which are: data acquisition, generation of characteristics, parameter selection, classification, and system evaluation.

The network state is represented by a form vector x of d parameters such that $X = (x_1, x_2 \dots x_d)$. The vector X is represented by a point in representation space. The parameters of the form vector X are derived from the executions of network simulations. For each new observation, we have to decide among M classes that correspond to areas in the representation space, grouping similar shapes.

We have proposed a set of modifications to the algorithm proposed by Weiqing et al. [16] to make it faster and more efficient. In our detection system, the stopping criterion is the number of features obtained by the algorithm where the quality of the solution does not improve if we add a new feature or increase the number of iterations.

We have applied a binary coding, a string of 1s and 0s, to represent the information. The number of parameters is the dimension of the solution. The general form of a vector is: $V = (x_1, x_2 \dots x_n)$. Where x_i takes the value 1 if the parameter is present in the vector of the solution, otherwise it takes the value 0 if the parameter does not belong to the solution.

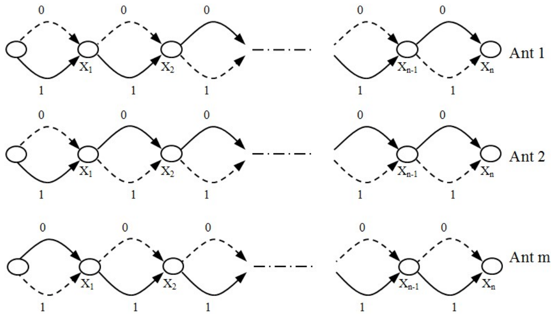

Fig. 5. The possible solution obtained by the ACO algorithm

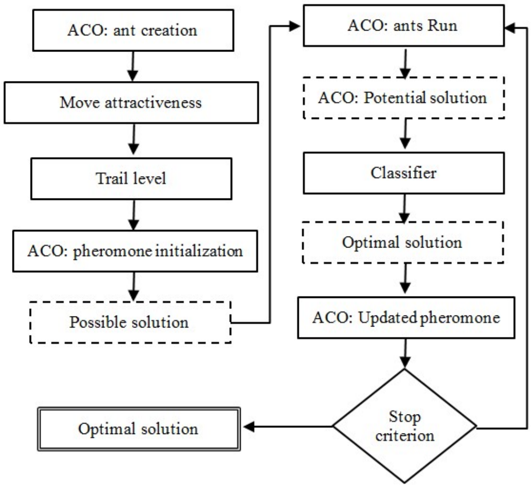

Fig. 6. Optimization of flooding attack detection on OBS network

Each path of an ant is a solution that consists only of the parameters whose positions equal the value 1:

$$P_m(i,j)=\begin{cases}\dfrac{\left[\tau_{(i,j)}\right]^{\alpha}\cdot\left[\eta_{(i,j)}\right]^{\beta}}{\sum_{k\in S_m(i)}\left[\tau_{(i,j)}\right]^{\alpha}\cdot\left[\eta_{(i,j)}\right]^{\beta}} & if\ j\in S_m(i)\\ 0 & otherwise\end{cases}, \tag{6}$$

where:

$\tau_{(imp)}$: The amount of pheromone deposited by an ant on the arc (*i,j*).

α : The pheromone parameter.

$\eta_{(i,j)}$: The heuristic parameter of the arc (*i,j*). The heuristic parameter value is fixed and does not change during program execution, is determined by $\eta_{(i,j)}=1/l_{(ij)}$, where $l_{(ij)}$ is the cost for an ant to move between i and j.

β : The heuristic parameter.

$S_m(i)$: The set of vertices that remain to be visited by an ant that is on vertex i.

The amount of pheromone is updated using the following rule:

$$\tau_{(i,j)}\leftarrow\rho\cdot\tau_{(i,j)}+\Delta\tau_{(i,j)}, \tag{7}$$

where:

ρ : The evaporation parameter.

$\Delta\tau_{(i,j)}$: The amount of pheromone added.

In our study, we have performed the ACO algorithm using two kinds of classifier SVM and ELM.

6 Performance Analysis

6.1 Dataset

In this study, we used a Burst header packet flooding attack on an optical burst-switched network data set that is available in the UCI machine learning repository [11]. This database includes 1075 observations spread over 4 classes which are NB-No Block, Block, No Block, and NB-Wait. Each observation includes 22 attributes. The database also contains missing values.

6.2 Features Selection

To generate a sub-table, we performed a preprocessing which is the selection of parameters.

This step consists in carrying out several simulations to fix the parameters of the ACO algorithm and the parameters of the kernel function.

The best pair of (C, γ) is (2^3, 2^{-5}). Knowing that we have used the intervals [2^3, 2^{11}], [2^{-12}, 2^2] as search spaces. The following table presents some instances of the obtained database.

A3. Packet Drop Rate: This is the normalization of a percentage of lost packet rate (numeric).

A4. ReservedBandwidth: Initially reserved bandwidth assigned (given) to each node, the

Table 1. Instances of the obtained database

A3	A4	A17	A18	A19	A20
1000	0	0.7	0	0	B
100	0	0.2	0	0.4	NB
900	0	0.8	0	0	B
100	0	0.3	0	0.4	NB
800	0	0.8	0	0	B
100	0	0.4	0	0.2	NB
700	0	0.8	0	0	B
100	0	0.5	0	0.1	PNB
900	0	0.8	0	0	B
100	0	0.4	0	0.1	PNB

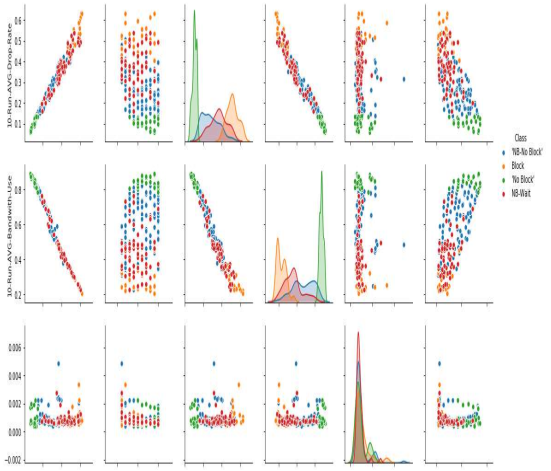

Fig. 7. Class positions by attributes

user (usr) in the experiments assign these values. (numeric).

A17. 10-Run-AVG-Drop-Rate: Average packet drop rate for 10 consecutive (run) iterations (numeric).

A18. 10-Run-AVG-Bandwidth-Use: Average Bandwidth utilized for 10 consecutive (run) iterations (numeric).

A19. 10-Run-Delay: Average delay time for 10 consecutive (run) iterations (numeric).

A20. Node Status' {B, NB, P NB}: initial classification of nodes based on a Packet Drop Rate, Used_Bandwidth, and anAverage_Delay_Time_Per_Sec. B = Behaving, NB = Not Behaving and P NB = Potentially Not Behaving (Categorical).

6.3 Classification

The source datasets are in ARFF format and we have to transfer them in CSV format, for that we created a script to generate a CSV file, and then we have to add the first line with the attribute names. After this step, Data can be manipulated by Python and Excel.

To upload files to google colab, Pandas is a python library that allows manipulating data as tables with labels of variables (columns) and individuals (lines) called DataFrames. We read and write these data frames from or to a tabular file using a pointer on pandas. Graphs can be easily drawn from these data frames using mat plot lib.

The next step is selecting the columns that present the six classification attributes with the NumPy library commands. To illustrate the relationships between these features, we created pairwise graphics with the Seaborn library that adds to Matplotlib, replaces some default settings and features, and adds new features. Seaborn corrects three Matplotlib defects.

In our model, shuffling data aims at reducing possible drifts and keeping the model general. A part of the data is reserved for the prediction test. The steps of preparation are as follows:

- Put all the columns of 8 first lines in the variable data_to_predict.
- Then we select the classes of *data_to_predict* and put them in the *predict_class* variable as NumPy array.
- Assigning attributes of *data_to_predict* except for the classes to the prediction variable
- Finally, we put the rest of the variables in data2.

We will continue our work with data2 (train and test the neural network).

The text is rarely exploitable as it is, it must be converted to digital format. The simplest treatment is to convert each value to a numeric value.

In Python, this operation is performed by the *LabelEncoder* function of Scikit-learn, which is a free Python library for machine learning. It is developed by many contributors, particularly in the academic world, by French institutes of higher education and research such as Inria and Télécom Paris Tech. It includes functions for estimating random forests, logistic regressions, classification algorithms, and support vector machines. It is designed to harmonize with other free Python libraries, including NumPy and SciPy.

Our model is a Perceptron Multilayer (PMC), which is undoubtedly the architecture most frequently used today.

Each neuron will calculate a weighted sum of its inputs that it will transmit to a transfer function f to produce its outputs. For each layer of the neural network in a PMC network, there is also a bias term. An equal to 1., a bias connects to neurons via a weight, usually called threshold. Neurons and biases are organized in a structure of non-looped layers (feed-forward). The network can, therefore, be interpreted simply as an input-output model, the weights and the thresholds being the free (adjustable) parameters of the model.

These networks can model even very complex functions, where the number of layers and the number of units in each layer will determine the complexity of the function. When designing Multilayer Perceptrons, it is important to specify the number of hidden layers as well as the number of units in these layers. But it is also important to choose the activation functions and learning methods.

In our case the number of input variables is six (6 attributes), the number of the output variable is four (4 classes) with six hidden layers of ten neurons.

In Python we will start with the sequential model of Keras (a linear stack of layers), the model must know what form of input it must wait. This is the reason why the first layer of a sequential model must receive the characteristics of the input form (the other layers can determine the form of their inputs by inference).

In the case of a multi-class classification problem, the softmax function can be used as an activation function. This is a generalization of the sigmoid.

If the output for the class k is sufficiently larger than those of the other classes, its activation will be close to 1 while the activation of the others will be close to 0. We can also consider that it is a version differentiable from the maximum, which will help us greatly for classification and learning.

Before training our model, we need to configure the learning process by calling the compile method which accepts three arguments:

- Optimizer: It can be an optimizer defined by its name, for example, rmsprop or adagrad, or an instance of the Optimizer class.
- Loss: This is the cost function that the model will use to minimize errors, it can be defined by its name.
- Metrics: a list of metrics. In the case of a classification problem, you will use metrics = ['accuracy'] but this argument can specify another metric.
- In our model, the error correction function is the cross-entropy function. In information theory, the cross-entropy between two probability laws measures the average number of bits needed to identify an event from the "set of events", the optimization function is ADAM, and the metric is precision.

The fit function is a technique of analyzing an experimental curve, consisting of building a curve from mathematical functions and adjusting the parameters of these functions to approximate the measured curve. Therefore also speaks of adjustment of parameters. The term curve fitting, profile fitting or simply fitting is often used to designate this method; franglais is often used to "fitter a curve" to say "adjust a curve".

Regression methods are used (Regression is a set of statistical methods widely used to analyze the relationship of a variable to one or more others). In simple cases, it is multi-linear regression if the law is linear for all the parameters or of polynomial regression when one uses a polynomial to simulate the phenomenon (the physical parameters being able to be deduced from the coefficients of the polynomial).

Conventional regression methods make it possible to determine the parameters from calculations on the data but are inapplicable if the function is too complex. It is then necessary to work by trial and error to approach a solution, in

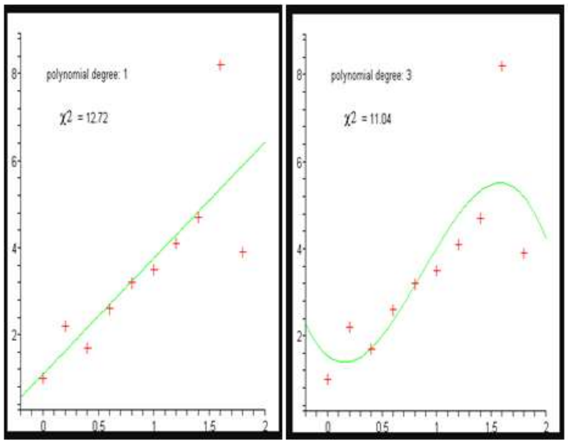

Fig. 8. Multilinear and Linear Regression with Fit Function

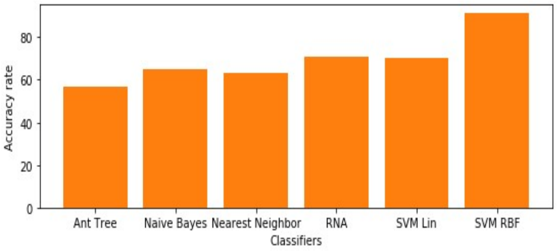

Fig. 9. Detection rate on the testing dataset

the sense of the least-squares method. The solution is not necessarily unique.

Batch_size is the number of samples for the learning function update and the data is used ten (10) times.

The name of the evaluation function in Keras is evaluated to associate with our model, its return value is the metric (in our case accuracy or precision) and the inputs are the attributes and the classes of tests.

we can also make predictions to classify new data with the predict_classes function on the two variables we reserved at the beginning (prediction and predict_class).

After examining the results of the different SVM and RNA methods and comparing their performance (precision or accuracy), the results are illustrated as follows.

For the SVM: we have shown the results of the two methods Kernel Gaussian (Acc = 91%) and the linear kernel (Acc = 70%), we can see that the best method to classify our data in comparison with the other methods is the Gaussian kernel, which performed much better. However, there is no absolute rule as to the best kernel in each scenario. This is to test all the kernels and select the one that gives the best results on the test data set.

For the RNA: (Acc = 71%) note that the accuracy may be lower than this rate because of the random initialization of the weights associated with the variables.

Figure 9 shows the classification rate obtained using machine learning algorithms. We used a sub dataset randomly generated

So the best method advocated by our study to diagnose flooding attacks in OBS is the Gaussian kernel of SVM.

This result is logical and this is due to the nonexistence of an ideal way for the choice of the optimal topology, hence the need for very advanced research in deep networks.

7 Conclusion

We have presented in this paper two methods to detect flooding attacks on the OBS network. We used an ant colony optimization algorithm to select a feature subset. This step is mandatory to reduce computing time. It increases the detection accuracy.

We have evaluated the accuracy of our approach using a UCI dataset. Both classifiers SVM and RNA achieve a reasonable detection rate.

Two main routes may be followed in future work. The first is to try another recent approach which is deep learning. The second is studying the impact of missing data on the detection rate.

Acknowledgments

The research has been generously supported by the Laboratory of Automation & Production Engineering and National Agency for Development of Research and Technological Development Results [PRFU project: C00L07UN050220200003]. The authors would like to express their sincere appreciation for all the support provided.

References

1. **Saini, H.S., Wason, A. (2019).** Fallacious node algorithm for performance enhancement in optical-burst-switching networks. Journal of Optical Communications, Vol. 40, No. 3, pp. 239–245. DOI: 10.1515/joc-2017-0078.
2. **Tallón-Ballesteros, A.J., Riquelme, J.C., Ruiz, R. (2019).** Semi-wrapper feature subset selector for feed-forward neural networks: Applications to binary and multi-class classification problems. Neurocomputing, Vol. 353, pp. 28–44. DOI: 10.1016/j.neucom.2018.05.133.
3. **Rajab, A., Huang, C.T., Al-Shargabi, M. (2018).** 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. DOI: 10.1016/ j.osn.2018.03.001.
4. **Rahmat-Samii, Y., Manohar, V., Kovitz, J.M., Hodges, R.E., Freebury, G., Peral, E. (2019).** Development of highly constrained 1 m ka-band mesh deployable offset reflector antenna for next generation cubesat radars. IEEE Transactions on Antennas and Propagation, Vol. 67, No. 10, pp. 6254–6266. DOI: 10.1109/TAP.2019.2920223.
5. **Hasan, R., Sion, R., Winslett, M. (2009).** Preventing history forgery with secure provenance. ACM Transactions on Storage (TOS), Vol. 5, No. 4, pp. 1–43. DOI: 10.1145/1629080.1629082.
6. **Tang, D., Kuang, X. (2019,).** Distributed Denial of Service Attacks and Defense Mechanisms. IOP Conference Series: Materials Science and Engineering, Vol. 612, No. 5.
7. **Ismail, M.N., Aborujilah, A., Musa, S., Shahzad, A. (2013).** Detecting flooding based DoS attack in cloud computing environment using covariance matrix approach. Proceedings of the 7th international conference on ubiquitous information management and communication, pp. 1–16.
8. **Patwary, M.K.H., Haque, M.M. (2020).** A semi-supervised approach to detect malicious nodes in OBS network dataset using gaussian mixture model. Inventive Communication and Computational Technologies, pp. 707–719.
9. **Kadri, O., Mouss, L.H., Mouss, M.D. (2012).** Fault diagnosis of rotary kiln using SVM and binary ACO. Journal of Mechanical Science and Technology, Vol. 26, No. 2, pp. 601–608. DOI: 10.1007/s12206-011-1216-z.
10. **Matthews, J., Garfinkel, T., Hoff, C., Wheeler, J. (2009).** Virtual machine contracts for datacenter and cloud computing environments. Proceedings of the 1st Workshop on Automated Control for Datacenters and Clouds, pp. 25–30. DOI: 10.1145/1555271.1555278.
11. **Nelson, M.R. (2009).** Building an open cloud. Science, Vol. 324, No. 5935, pp. 1656–1657. DOI: 10.1126/science.1174225.
12. **Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., Johnson, B.A. (2019).** Deep learning in remote sensing applications: A meta-analysis and review. ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 152, pp. 166–177. DOI: 10.1016/j.is prsjprs.2019.04.015.
13. **Vapnik, V. (2013).** The nature of statistical learning theory. Springer Science & Business Media.
14. **Osuna-Enciso, V., Espinoza-Haro, J.I., Oliva, D., Hernández-Ahuactzi, I.F. (2018).** Offshore wind farm layout optimization via differential evolution. Computación y Sistemas, Vol. 22, No. 3, pp. 929–941. DOI: 10.13053/cys-22-3-2668.
15. **Xiong, W., Wang, L., Yan, C. (2006).** Binary ant colony evolutionary algorithm. International Journal of Information Technology, Vol. 12, No. 3, pp. 10− 20.
16. **Rajab, A., Huang, C.T., Al-Shargabi, M., Cobb, J. (2016).** Countering burst header packet flooding attack in optical burst switching network. International Conference on Information Security Practice and Experience, pp. 315–329.

Article received on 20/02/2020; accepted on 17/02/2021.
The corresponding author is Mohamed Takieddine Seddik.