WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Comparative Analysis of the Predictive Performance of Six Selected Machine Learning Classifiers when Used as a Tool for Network Fault Management

Okolo C C[1], Ezeugbor I. C[2], Ngene C C[2], Iwegbuna O N[2], Chukwuogo O E[2]

[1]Electronics Development Institute, Federal Min. Of Science And Technology,Awka Capital Territory, Anambra State, Nigeria
[2]Department of Computer Science, Nnamdi Azikiwe University, Awka, Anambra State. Nigeria

ABSTRACT

Network troubleshooting is a significant process that has undergone many studies. The first step in achieving a solution to it is represented in collecting information. Syslog messages sent by almost all network devices include a large amount of data that is related to the network problems. From studies, it was discovered that analyzing syslog data can be a guideline for network problems and their causes. Detecting network problems can become better if the problems detected have already been classified based on the network layers. The classifications of syslog data can be achieved when the syslog messages that describe the network problems for each layer is identified. The work is aimed at proposing a method for classifying the syslog messages which will identify the network problems. This classification is done with respect to the network layers. This makes use of data mining instrument for the classification of the syslog messages. The description part of the syslog message was used execute the classification process and relevant syslog messages were also identified. The features were also selected to train the classifiers. Six classification algorithms were learned; LibSVM, SMO, KNN, Naïve Bayes, J48, and Random Forest. A real data set was achieved from an educational network device and it was used for the prediction stage. From findings, the LibSVM performs better than other classifiers with respect to the rate of probability of the classified instances when at the range of 89.90%–32.80%. The validation results shows that the rate of probability of the correctly or well classified instances is >70%.

Keywords: Machine Learning, Classification Algorithms, Network Fault Management, SVM, Network Management

INTRODUCTION

Most organizations rely on networks to manage their transactions. Any failure or error occurring in the network shall negatively affect the achievements, productivity, and service quality of organizations. There have been a lot of challenges in the management of networks especially with respect to detecting and classifying network problems. The detecting and classification of network problems are very essential in the maintenance processes [1]. Therefore, it is very important to diagnose and confirm the causes of network problems thereby addressing such problems and to reduce the chances of experiencing it in the future. Network troubleshooting can be important process which can be done when systematic approach is applied. Applying such approach will reduce any inaccuracy associated with the troubleshooting process and also contribute in time reduction during this process. Diagnosis and recovery of any problem experienced by a system begins with reviewing the system’s log files [2]. These files present the activity record of the system. They present the sources of problems. Syslog messages are comprise of a large amount of data which concerns the network problems. Almost every network types of equipment, such as the routers, switches etc send data [3]. Network troubleshooting is executed through the help of Layered Model [4]. Through this model, problems are normally described in terms of a specific model layer [4].

Network errors could be distributed into the network layers depending on the TCP/IP model (network access layer, Internet layer, transport layer, application layer) [5]. Therefore, network problems can be classified based on their causing layers, and the layer elements problems. Any layer that causes any sort of network problems can be detected and addressed by carrying out the classification process. The detection and classification of network problems will put into consideration; incurring cost, devoting time and exerting effort [6]. These are needed to extract the data related to the network problems and undergo the classification of this data with respect to the network layers. There are some challenges with the classification of the syslog messages; possession of various types of logs, increment in the number of network elements and the log format [6]. To classify syslog messages, there shall be a good domain knowledge of each log format, the component of each of the layer and its potential problems.

The application of a machine learning techniques on syslog data will enhance the opportunity to detect syslog messages. This study compares the predictive performance of six selected machine learning classifiers with each other [7]. To enable the detection of the network fault and also analyze the classification framework, a comparison is conducted. This is done to provide recommendations to the model selection. The study is centered more on the classification of syslog data with respect to the TCP/IP layers. This classification process in this paper is aimed at enhancing the processes of the network troubleshooting and efficiency of the maintenance processes.

LITERATURE REVIEW

Many significant studies were conducted about the network detection issues from log data using data mining techniques and proposed a system for the detection of runtime problems by mining console logs. The latter researchers converted free-text console logs into numerical features. These features were analyzed through using Principal Component Analysis (PCA). A similar approach was adopted by Liu and Jiangang [8]. Those researchers used PCA learning algorithm for detecting the anomalies existing in syslog messages. They created features which capture various correlations among different types of log messages. On the hand, Tongqing et al. developed an automated tool syslog digest that transforms the massive volume of routers syslog messages into a smaller number of meaningful network events. Then, they identified the signature of syslog messages that capture the network behavior over a period of time. They grouped these messages based on their nature and severity. Fukuda used syslog messages to detect the unusual events occurring in a network. That was done through assigning a global weight, based on a global appearance of a message type in the whole data set. Kimura et al. [6] analyzed two types of data: SNS data from tweeter and syslog messages. That was done to detect and diagnose any network failure. The latter researchers used non-negative matrix factorization (NMF) machine learning algorithm in order to analyze syslog messages. They supported the vector machine in order to analyze tweeter messages. All of the aforementioned studies aimed to detect network problems only. However, they did not classify the problems in the aim of increasing the efficiency of the maintenance and troubleshooting problems. The present study aimed to propose methods to analyze syslog data, detect network problems and diagnose their causes, and classify these problems in terms of network layers.

METHODOLOGY

Fault classification techniques

Machine learning technique and supervised learning techniques are used for text classification. Through using these techniques, the class labels shall be defined early. The most popularly used algorithms for classification include K-Nearest Neighbour (k-NN), Naive Bayes (NB), Support Vector Machines (SVM) and Decision Tree algorithms.

Support vector machine

The support vector machine is a supervised learning technique that is applicable to both classification and regression, by applying linear classification techniques to non-linear data. SVM is a machine learning technique. It adopts the structural risk minimization principle. The SVM divides the training data set into two classes and it makes its decision depending on the ”Support Vectors”. Through this, the effective elements are selected from the training set. There are two main methods which have been proposed for this purpose; one-against-one method, one-against. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to the other.

K-nearest neighbor KNN

The KNN is one of the simplest supervised machine learning algorithms used mostly for classification. KNN classifies a data point with respect to similarity measure (i.e. the distance function). That KNN classification is quite straightforward. The KNN is also seen as Lazy Learning, and its process of induction is slow to run time. In KNN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with object being assigned to the class most amont its k nearest neighbors.

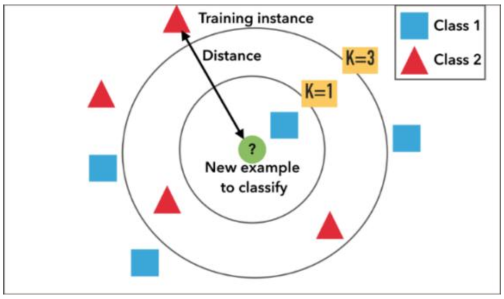

Figure 1: classifying new example depending on Training instance distance

Decision Trees

Decision Tree (DT) or recursive partitioning model refers to a decision support tool. It uses a tree-like graph of decisions and their possible consequences. DT creates a type of flowchart which consists of nodes (called “leafs”) and a set of decisions to be made based on a node (called “branches”). The leaf and the branch structure forms a hierarchical representation that mimics the form of a tree. Decision tree learning is one of the most widely used methods for carrying out the inductive inference [28]. It is an important tool for carrying out the predictive analysis process. The main problem in the DT classification algorithm is represented in the way of constructing the optimal classifier [29]. In general, the DT classifier can be built from a set of features. In the classification task, the size of search space is exponential and the accuracy of some trees is more precise than other trees. The key idea is to use a decision tree to partition the data space into cluster regions and empty regions.

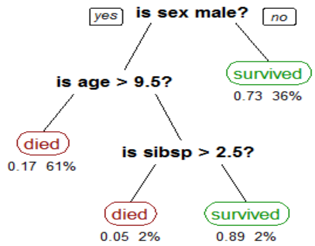

Figure 2:Tree like representation of datain Decision tree

J48 algorithm

J48 is an algorithm used to generate a decision tree that is generated by Quinlan's C4.5 [30]. C4.5 algorithm is used for building a decision tree. It is an extension of the ID3 algorithm that uses a predictive machine-learning model. J48 generates rules that identify the identity of the data. For gaining the equilibrium of flexibility and accuracy, the decision tree is progressively generalized.

Random forests

Random Forest (RF) is an ensemble classifier that uses many decision tree models, ensemble models combine the result of different classifier. The result resulting from the ensemble model is usually different from one resulting from the individual model [31]. RF employs the DT concept by producing a large number of decision trees. The approach first selects a random sample of the data and identifies a key set of features in order to grow each decision tree. Then, the decision trees have their own error rate determined. After that, the collections of decision trees are compared to find the joint set of variables that produce the strongest classification model. In order to make the prediction of the class label of a case y through majority voting, the equation below shall be used [31]: I(y) = argmaxc(∑N n Thn (y) = c) Where I the indicator function and hn the nth tree of the RF.

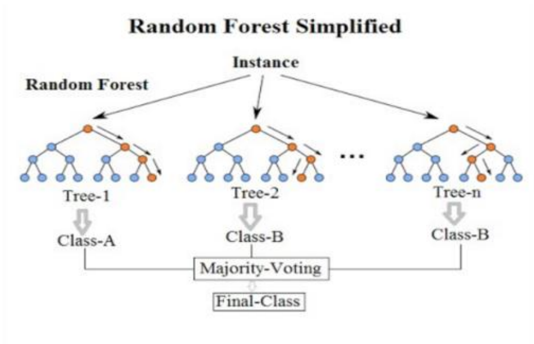

Figure 3:Random forest

Naïve bayes algorithm

Naïve Bayes classifier is a simple probabilistic classifier that relies on bayes. Theorem with independence assumptions between predictors is particularly suited when the dimensionally of the inputs is high. It is mainly used in case the categorical input variables are compared to numerical variables. In other words, Naïve Bayes operates by assuming independence (i.e. the presence of some feature will not affect the other features "Naïve Bayes" classifier is considered as a "supervised learning" algorithm. It relies on the probability model. In many practical applications, the method of maximum likelihood is used in parameter estimation for "Naïve Bayes" models. Naïve Bayes model is easy to build with no complicated iterative parameter estimation. That makes it useful for the large data sets. Despite its simplicity, the Naive Bayes model operates surprisingly well. It is widely used because it often outperforms the classification methods that are more sophisticated. Naïve Bayes Algorithm is a statistical classifier that predicts class membership probabilities, such as: the probability that a given tuple belongs to a particular class. It uses the prior probability of each category given that here is no information provided about the item. As a result, categorization produces a posterior probability distribution over the possible categories.

Likelihood Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

Posterior Probability Predictor Prior Probability

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figure 4: Naive Bayes Theorem

Implementing fault classification algorithms

Data set Cisco syslog manual was used to identify related syslog messages and the result could be applied to other syslog data from different vendors, as all vendors describe network problems using almost same terms and vocabularies. Cisco IOS devices have more than 500 facilities represented by integers. Facility simply is a service provides classification of the sources that generate syslog messages. The most common facilities are: IP, Open Shorted Path First (OSPF), SYS Operating System, IP security (IPsec), Rout Switch Processor (RSP), Interface (IF) [3]. The related messages are identified depending on the symptoms and causes of network problems related to the network layer. The task of identifying related messages from the manual requires reading and searching the manual in order to extract them. Searching in the manual is done by using the symptoms and the causes of each problem as key words to identify related problems. Loss of connectivity, performance lower than baseline, high collision counts, attenuation, bad cable, disconnected cables, damaged cables, improper cable types, cable length exceeds the design limit for the media, and cable fails are the symptoms and causes of cable problems that could be used as key words for searching in the manual, which include error message, its explanation, and the recommended action. The following are the extracted messages that describe cable problems from the Cisco syslog manual:

%PIX|ASA-1-101001: (Primary) Failover cable OK.
%PIX|ASA-1-101002: (Primary) Bad failover cable.
%PIX|ASA-1-101003: (Primary) Failover cable not connected (this unit).
%PIX|ASA-1-101004: (Primary) Failover cable not connected (other unit).
%PIX|ASA-1-101005: (Primary) Error reading failover cable status.

The goal is to identify syslog messages that describe network problems related to each network layer. Network problems were identified for each layer and key words that described each problem were extracted to be used for searching in Cisco syslog manual. Figure 1 illustrates problems of each network layer, the key words used for searching in Cisco syslog manual, and examples of extracted syslog messages.

Classification phases

The classification process is an attempt made in order to label a set of unlabeled conditions (i.e. assign them to classes). That is done based on a set of conditions along with using known labels. The classification approach that is commonly adopted involves two stages as follows;

[1] Training the classifier through using the pre-labeled set of conditions, and using a selected subset of probes which are considered as the selected features.

[2] Using the trained classifier in order to label other conditions

Regarding the training stage, the files of training dataset along with a different number of features were prepared. The training dataset is derived from the syslog messages which were extracted from the Cisco syslog manual during phase two and processed (cleaned, removed stop words, stemmed, removed duplicated words). The training dataset was in ARFF extension, to be used in the Weka data mining tool. The classification algorithm was applied several times through using the same dataset. With different number of features. The obtained results were compared in order to identify the best feature and identify the best classification model. First, the algorithm was applied to four training data files with a different number of features. The first training file represents the data the employs all the features (terms of vector space). The second training file represents the data that employs 1000 features which have the highest frequency. The third training file represents the data using 500 features which have the highest frequency. The fourth training file represents the data using 200 features which have the highest frequency. The accuracy rate was calculated. In addition, the results were compared to identify the best number of features. That was done to use these features for representing syslog data files. The accuracy rate indicates that the ratio of correctly classified instances; and the performance of the classifier model would be better when reaching a high accuracy rate. The accuracy rate was calculated through using the following equation:. accuracy rate $= n/N$ Where n is the number of correctly classified instances , and N is the number of all classified instances.

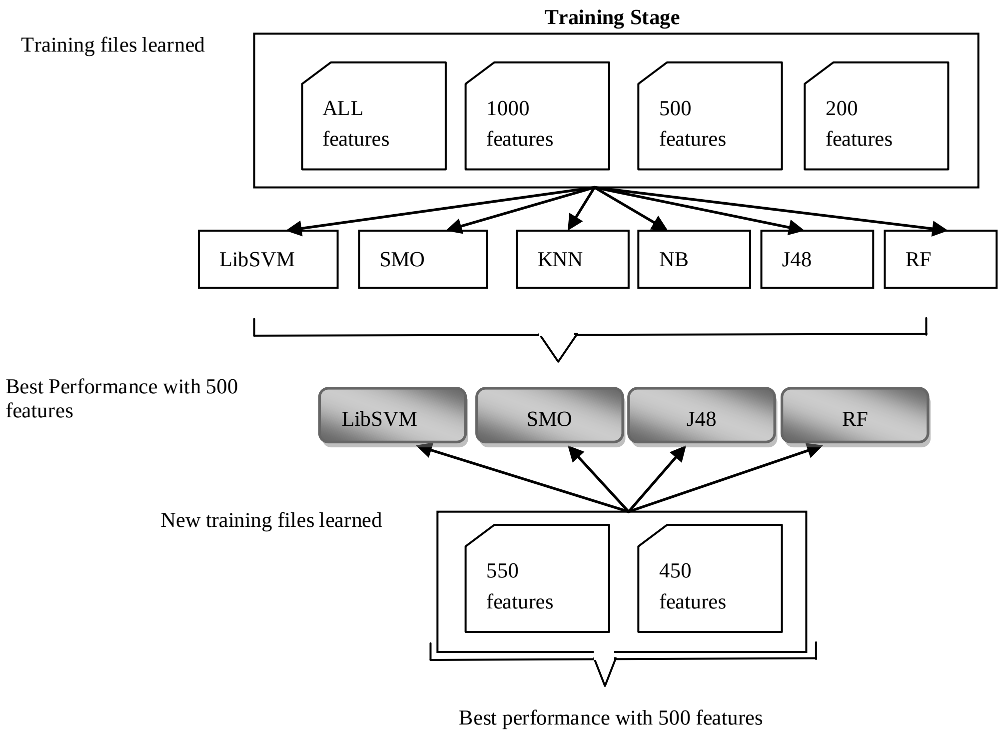

Figure 7: Process of training stage

The results showed that the performance of six classifiers scored the highest values by using a training file that was represented with 500 features, Figure 3 illustrates the process of training stage. They also showed that four classifiers: libSVM, RF. SMO and J48 had recorded performance value higher than NB, and KNN. The latter four algorithms (i.e. libSVM, RF. SMO and J48) were relearned by another two training files with different features number. The relearning process used 1 file which represents data that uses 550 features. It also used another file which represents data using 450 features. The relearning process aims to make sure that using 500 features is the best number of features to be used for representing datasets. The results of the four classification algorithms were compared in terms of the accuracy rate, and performance through using a training file that was represented with 500 features.

Regarding the prediction stage, four classification algorithms based on the training stage. These algorithms are: libSVM, RF. SMO, and J48. They were used as classifiers to classify a testing dataset in the prediction stage. These algorithms show a good performance during the training stage. Testing dataset, obtained from UUM network devices, was processed and represented with 500 features in ARFF files; this 500 of features is the best number to be used as shown in the results of training in Figure 4. Classifiers results were compared in terms of probability rate. It was found that LibSVM is the best model to be used for syslog data classification. Probability rate indicates the proportion of accuracy that the classified instance relays to the specific class. Probability rate for each classified instance is displayed in the table of classification results. It is calculated by Weka.

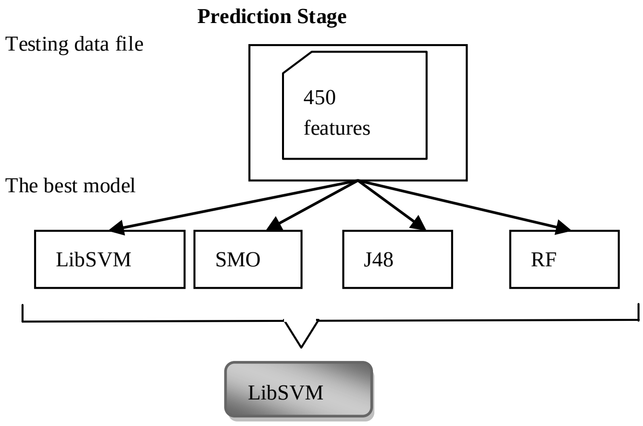

Figure 8: Process of prediction stage

RESULTS

Results of training stage

The obtained results were compared for identifying the best number of features and the best models. Six classification algorithms were applied to the above training dataset files and performance rate (i.e. the accuracy rate). The latter rate represents the ratio of the instances that are correctly classified for each classifier. The results of the performance of the six algorithms through using above training datasets are presented below:

Table 1: Accuracy rate of six classifiers using training files represented by different numbers of features

Traininig file	SVM(Lib)	SVM (SMO)	Naïve bayes	KNN	J48	Random forest
All features	73.80%	71.10%	41.10%	64.30%	64.30%	70.70%
1000 features	74.10%	71.10%	52.90%	64.30%	64.30%	73.00%
500 features	74.50%	71.10%	60.10%	64.30%	69.30%	75.70%
200 features	65.40%	65.40%	55.50%	49.80%	58.20%	67.70%

As shown in Table 1, the performance values recorded through the training file indicate that the 500 features show the highest values. That's because the number of the correctly classified instances are higher than the values of other files. That means that this file includes the best number of features that can be used for the prediction stage. As shown in Figure 5, it should be noted that the accuracy rate of Random Forest, LibSVM, SMO, and J48 show the highest values. The results were compared with the results of the previous training dataset that includes 500 features.

Table 2: Accuracy rate of four classifiers using training files represented by different numbers of features

Traininig file	SVM(Lib)	SVM (SMO)	J48	Random forest
550 features	71.10%	70.30%	64.30%	71.10%
500 features	74.50%	71.10%	64.30%	75.70%
450 features	71.10%	70.40%	64.30%	71.10%

Table 2 displays the accuracy rate of each classifier of the four ones. That was done through using training files represented by different numbers of features. As shown in Table 2, the accuracy rate of the algorithms show fewer values when using training dataset files of 550 features and 450 features. That means that the training dataset file including 500 features include the best features to be used during the prediction stage. It is clearly seen that the results of the experiments conducted during the training stage indicate that the features of number 500, had the best performance by applying Random Forest, LibSVM, SMO, and J48 classification algorithms.

Results of prediction stage

In the prediction stage, four classification algorithms: Random Forest, LibSVM, SMO, and J48 were applied to the testing dataset. Classification algorithms were chosen for good performance during the training stage. An obtained testing dataset consists of 2610 instances (syslog message) from firewalls and switches devices involving a short period of time (less than one minute). The testing dataset was preprocessed similar to training dataset (cleaned, removed stop words, stemmed, removed duplicated words). Testing data set file was generated with 500 features in ARFF format to be classified using the Weka data mining tool. The results of the comparison between the four classification algorithms-Random Forest, LibSVM, SMO, and J48-involving testing dataset are presented in Table 3.

Table 3. Prediction stage results for four classifiers.

Algorithm	Layer 1	Layer 2	Layer 3	Layer 4
LibSVM	172 (6.59%)	2383 (91.30%)	55 (2.11%)	0 (0%)
SVM (SMO)	134 (5.13%)	2351 (90.08%)	126 (4.83%)	0 (0%)
J48	250 (9.58)%	2320 (88.89%)	40 (1.53%)	0 (0%)
RF	170 (6.51%)	2395 (91.76%)	43 (1.65%)	2 (0.08%)

Table 3 shows the number of instances, classified into each layer with the percentage of all testing dataset sample. Four algorithms Random Forest, LibSVM, SMO, and J48 gave convergent results; LibSVM, SMO, and J48 algorithms classified all instances into three classes (layer1, layer2, and layer3), while RF algorithm classified all instances into four classes (layer1, layer2, layer3, and layer4), layer4 got only two instances, as shown in Figure 7. In fact, the obtained results comply with the fact saying that lower layer of TCP/IP model are where the most of issues occur. Therefore, it is advisable in network management process, to start troubleshooting from the physical layer and gradually proceed to the upper layers.

The following table compares the range of probability rate of classified instances for results of Random Forest, LibSVM, SMO, and J48.

Table 4: Probability range of classified instances for used classifiers.

Algorithm	Layer 1	Layer 2	Layer 3	Layer 4
LibSVM	(72.20 – 32.80)%	(67.00 – 33.00)%	(89.90 – 36.20)%	0.00%
SVM (SMO)	(50.00 – 40.30)%	(55.00 – 40.30)%	(50.00 – 40.30)%	0.00%
J48	(60.50 – 45.30)%	(50.00 – 40.00)%	(50.00 – 40.30)%	0.00%
RF	(79.00 – 26.00)%	(78.00 – 34.00)%	(58.00 – 36.00)%	38.00%

As shown in Table 4, the results of LibSVM algorithm show probability rates higher than others classifiers. LibSVM gave higher values for both maximum and minimum rate in each layer range, except for the first layer range whose maximum value is less than RF. Depending on the previous results, the results of LibSVM algorithm is considered as the best. The result is divided into two parts: one for the lower probability to be validated as informational messages, and the other for the higher probability to be validated as problems messages. Each part is compared to the training dataset to validate the result. The following table shows the numbers of instances and their percentage with a probability rate >= 50% and < 50% for each class.

Table 5: Probability rate for classified instances using LibSVM

Probability	Layer 1	Layer 2
50	49 (1.88%)	2218 (84.98%)
50	123 (4.71%)	165 (6.32%)

As shown in Table 5, layer1, and layer3 have small numbers of instances with probability rate with probability rate >= 50%. However, layer2 have large numbers of instances with a probability rate < 50%. This is because of repeated messages for the same problem with one different word.

Results of validation phase

Based on the results of the training stage, the classification algorithms that show good performance are: libSVM, RF SMO, and J48. These four algorithms were used during the prediction stage. They were applied to the testing dataset. During this phase, the results of the prediction stage were validated. The validation process was carried out through comparing the instances of each class to the corresponding training dataset. That was done to make sure that each instance belongs to its class, and refers to a problem in one layer. LibSVM classifier has classified all instances of testing dataset into three layers- (layer1, layer2, layer3).

In Layer 1 validation, the classifier had classified 49 instances with a probability rate >= 50%, to layer1. These instances were compared to the instances that belong to class one in training dataset. Only three instances indicated that there is a network problem. The probability rate of these instances is > 70%. These three messages described the problem of "TCP connection to firewall server had been lost, restricted tunnels are now allowed full network access". That was repeated three times. By referring to syslog manual, this problem indicates that the TCP connection to the security appliance server was lost. Thus, that requires checking the server and network connections. The latter problem belongs to the first layer. This layer is a network access layer.

In Layer 2 validation, the classifier classified 2218 instances with a probability rate of >= 50%, to layer2. These instances were compared with the instances belonging to class two in the training dataset. No instance in the classified instances indicated that there is a network problem. The probability rate of them is < 70%.

In Layer 3 validation, the classifier had classified 8 instances with a probability rate that is >= 50%, to layer3. One instance among the all classified instances indicated that there is a network problem. The probability rate is > 70%. This message described the problem of "No translation group found for protocol src".

In Validation of instances with low probability, the instances that have a probability validation that's less than 50% were compared to the training dataset. There isn't any instance pointing to the network problems. It described network events only. The result of the validation process seems to be acceptable. The testing data sample was for a short period of time, less than one minute.

CONCLUSION AND FUTURE WORK

Network fault detection and identification is essential for making well-informed decisions by network administrators. The present paper proposes a method for detecting and classifying network problems, in terms of network layers. That was done through analyzing syslog data. To meet the study's goals, the researchers conducted a comparison of five different machine learning classifiers. This comparison was conducted for detecting and classifying network problems. The researchers compared classifiers based on their accuracy and error rates. It was found that (Random Forest, LibSVM, SMO, and J48) show a good performance during the training stage. These algorithms show the following rate respectively of correctly classified instances: 75.70%, 74.50%, 71.10%, and 69.30%. LibSVM algorithm classified instances with a probability rate that is higher than the counterpart rate of RF, SMO, and J48 classifiers. The probability rates of classified instances through using LibSVM are in the range of 89.90% - 32.80%. The validation results show that the probability rate of the correctly classified instances is > 70%. The results of the comparison between the algorithms indicate that the SVM classifiers show the best performance during the experiments of the present study.

REFERENCES

[1]Bob V. Accessing the WAN, CCNA Exploration Companion Guide. India, 2008.

[2] Son HS, Lee JH, Kim TY, Lee SG. Proceedings of the Second International Conference on Electrical Systems, Technology and Information; Singapore; 2018. page 439-446.

[3]Makhtar, M., Nawang, H., & Shamsuddin, S. (2017). Analysis on students performance using Naïve Bayes classifier. J. Theor. Appl.Inf. Technol. 95(16):3993–3999, 2017.

[4] Joseph S. Network Troubleshooting Tools: Help for Network Administrators. USA: O'Reilly Media, Inc.,2001.

[5] Wilkins, Sean. Designing for Cisco Internetwork Solutions (DESGN) Foundation Learning Guide: (CCDADESGN 640-864). Pearson Education, 2011.

[6] Kimura T, Takeshita K, Toyono T, Yokota M, Nishimatsu K et al. Network Failure Syslog and SNS.International Journal of Pure and Applied Mathematics 2018; 119 (12): 9543-9551.

[7] Viktor M, Kenneth C. Big Data: A Revolution That Will Transform How We Live, Work, and Think.Boston, New York: Eamon Dolan/Mariner Books Houghton Mıın Harcourt, 2013.

[8] Badr, G., Algobail, A., Almutairi, H., & Almutery, M. (2016). Predicting Students Performance inUniversity Courses: A Case Study and Tool in KSU Mathematics Department. Procedia Computer Science 82 (2016) 80 89.

[9] David SK, Saeb AT, Al Rubeaan K. Computer Engineering and Intelligent Systems 2016; 4 (13): 28-38.

[10] Liu M, Jiangang Y. An improvement of TFIDF weighting in text categorization. In: International Proceedings of Computer Science and Information Technology; Singapore; 2012. pp. 44-47.

[11] He Q, Veldkamp BP. Classifying unstructed textual data using the Product Score Model: an alternative text mining algorithm. In: Eggen TJHM, Veldkamp BP (editor). Psychometrics in practice at RCEC. Netherlands: RCEC, Cito, University of Twente. 2012, pp. 47-62

[12]https://networklessons.com/cisco/ccie-routing-switching/cisco-ios-syslog-messages

[13]https://insidebigdata.com/2017/07/14/modern-problem-collecting-network-data/

[14]https://quizlet.com/be/264951735/ccna2-chapter10-flash-cards/

[15]https://insidebigdata.com/2017/07/14/modern-problem-collecting-network-data/

[16]http://www.csun.edu/~twang/595DM/Slides/Week4.pdf

[17] http://journals.tubitak.gov.tr/elektrik/issues/elk-20-28-3/elk-28-3-17-1907-84.pdf

[18]https://www.ukessays.com/essays/computer-science/image-to-voice-converter-is-software-computerscience-essay.php

[19]https://www.aitimejournal.com/@akshay.chavan/a-comprehensive-guide-to-decision-tree-learning

[20] https://www.cisco.com/c/ja_jp/td/docs/sec/multi-functionsecur/asa5500adaptivesecurappli/smg/004/393802/3938-02-2.pdf

[21] 1443 MADI et al./Turk J Elec Eng & Comp Sci