

PAPER • OPEN ACCESS

Comparative Analysis of Machine Learning techniques for Network Traffic Classification

To cite this article: Oluranti Jonathan *et al* 2021 *IOP Conf. Ser.: Earth Environ. Sci.* **655** 012025

View the [article online](#) for updates and enhancements.

You may also like

- [Machine learning for condensed matter physics](#)
Edwin Bedolla, Luis Carlos Padierna and Ramón Castañeda-Priego
- [Network Traffic Classification Based on Deep Learning](#)
Jun Hua Shu, Jiang Jiang and Jing Xuan Sun
- [A Classification Method for Network Traffic Based on Semi-supervised Approach](#)
Yun Liu, Zhiqiang Zhu and Pengfei Zhong




The Electrochemical Society
Advancing solid state & electrochemical science & technology

242nd ECS Meeting
Oct 9 – 13, 2022 • Atlanta, GA, US
Presenting more than 2,400 technical abstracts in 50 symposia


ECS Plenary Lecture featuring M. Stanley Whittingham,
Binghamton University
Nobel Laureate – 2019 Nobel Prize in Chemistry

 Register now!

Comparative Analysis of Machine Learning techniques for Network Traffic Classification

Oluranti Jonathan¹, Sanjay Misra², Victor Osamor¹

¹Department of Computer and Information Sciences, Covenant University, Ota, Nigeria

²Department of Electrical and Electronics Engineering, Covenant University, Ota, Nigeria

Email: jonathan.oluranti@covenantuniveristy.edu.ng

sanjay.misra@covenantuniversity.edu.ng

victor.osamor@covenantuniversity.edu.ng

Abstract. Network traffic classification is the operation of giving appropriate identification to the every traffic flowing through a network. Several methods have been applied in the past to achieve network traffic classification including port-based, payload-based, behavior based and so on. These methods have been found to one limitation or the other. Nowadays, attention is now on Machine Learning(ML) methods that rely on the statistical properties of the traffic flows generated. However, ML methods do not perform well when confronted with large-scale traffic data having large number of features and instances. Feature selection is employed to remove non-relevant and redundant features before passing the data to ML classifiers. In this study, network traffic classification using ML methods is demonstrated from two perspectives: one that involves feature selection and one that does not. A number of performance metrics are considered including runtime, accuracy, recall, precision and F- score. The experimental results indicate that the classification without features has an average accuracy and runtime of 94.14% and 0.52 seconds respectively. On the other hand, the method with feature selection has accuracy of 95.61% and average of 0.25 seconds for the runtime. The improvement obtained reflects the importance of applying only relevant and non-redundant features to the ML methods. Thus it recommended that feature selection be included in the network classification process to guarantee an optimal accuracy result.

Keywords: Feature Selection, Machine Learning, Network Traffic Classification.

1. Introduction

Network traffic classification can be described as the process of attributing traffic instances or elements to the applications or kinds of applications that generated them [1]. Accurate and timely classification of network traffic data is of key significance to service providers and network operators providing several benefits that include: network security monitoring, congestion avoidance, Internet Protocol (IP) management, Quality of Service (QoS) enforcement, bandwidth management, estimation of bills for usage, to mention a few [1][2].

Several techniques have been applied to classify network traffic data, the most common being port-based technique. The port-based technique makes use of the port number assigned to applications by the Internet Assigned Number Authority (IANA) [3]. The limitation of this technique comes from the use of dynamic port numbers instead of the originally assigned or well-known port numbers [3]. Also, applications can hide under another application such that their own port number is not captured but that of the host application [4]. The second known technique is the payload-based technique. This technique also known as deep packet inspection (DPI) works by inspecting the entire payload of each packet to discover the signature pattern of the packet[5]. The DPI method has the issue of packet encryption which makes it difficult to detect some applications and /or their correct attributes during the classification process [1][4][5]. The technique also makes it possible for third parties who have no connection with the traffic to inspect the payload of each packet (that is, the payload is visible) [3][5]. Due to the limitations of previous techniques,



attention is now toward ML techniques that make use of statistical properties of the traffic flows[1][5]. However, ML techniques do not perform efficiently when confronted with dataset that possess non-relevant and redundant features [1][3][5]. Table 1 presents a summary of the network traffic classification methods. The table details the characteristics and limitations of the different classification methods.

The focus of this study is to investigate the application of ML methods to a multi-class network dataset. The study is approached from two perspectives. The first part considered direct application of ML methods to the NIMS multi-class network traffic data. In the second part, feature selection is introduced to investigate the performance of the ML algorithms after the redundant and irrelevant features are removed. Afterwards, a comparative analysis of the two approaches is presented based on a number of metrics. Charts are also used to further explain the findings of the study. In all five ranker-based feature selection methods and four ML algorithms are used in the various experiments carried out.

The rest of the study is presented as follows. In Section two, core aspects of the study are described ML and Feature Selection). Section three presents the materials and methods employed in the study. In Section four, the experimental results and discussions are presented while Section five concludes the study.

Table 1: Overview of Network Traffic Classification Methods

S/N	Classification Method	Characteristics/Advantages	Disadvantages
1	Port-Based [5][8]	<ul style="list-style-type: none"> • Traffic identification done using port numbers allocated by IANA. • Method is fast and low-resource consuming • Supported by many network devices 	<ul style="list-style-type: none"> • Due to growing number of application, there is tendency to use unpredictable port numbers • The method may not be suitable for applications not registered by IANA • Method may not work well with applications that use dynamically allocated port numbers.
2	Deep Packet Inspection Method [5][9]	<ul style="list-style-type: none"> • Inspects the actual pay-load of the packet. • Identification not based on port number • Method provides more accurate result compared to port-based techniques • Method is quite suitable for P2P traffic 	<ul style="list-style-type: none"> • Method is slow and requires much processing power- high computational cost • Signatures must be kept up-to-date, as the applications change very frequently • Not easy to apply to encrypted traffic. • The method also suffers from violation of privacy policies and regulations
3	Machine Learning Method based Statistical Analysis of attributes [5][6][7]	<ul style="list-style-type: none"> • Method is based on analysis of statistical properties of the flow comprising the packets. • Attributes of flow such as packet size, packet inter-arrival times etc may be used. • Method is fast and consumes less processing power • It can also detect the class of yet unknown applications • Method is also able to identify encrypted traffic. 	<ul style="list-style-type: none"> • There is usually the need to preprocess the traffic data before classification • The performance of the ML algorithms may be affected by too many features especially when they either not relevant or redundant. Feature selection is usually employed to select optimal feature sets.

2. Methodology

In this section, a description of the various ML algorithms and feature selection methods is provided. The dataset used is also described.

2.1 NIMS Dataset

NIMS dataset is one of the network traffic datasets that are freely available for research purpose. It is a collection of several internet/network applications such LFD, RFD, SCP, SFTP, SHELL, and X11. [8]. The NIMS dataset was obtained from Network Information Management and Security Group. The original dataset included applications with only few instances which were deleted in order to avoid an imbalance situation which could lead to wrong accuracy results. Table 2 contains the breakdown of the final copy of dataset used in this study.

Table 2: Description of the NIMS Dataset

S/N	Class Name	Number of Samples
1	LFD	2,557
2	RFD	2,422
3	SCP	2,444
4	SFTP	2,412
5	SHELL	2,491
6	X11	2,355
Total		14,681

2.2 Machine Learning Algorithms

Four ML algorithms were employed in this study. The algorithms include SVM, Naïve Bayes, K-NN and C4.5 -Decision Tree.

2.2.1 Support Vector Machine (SVM)

This is a supervised ML algorithm that can be used for both regression and classification purposes though it is more common in classification. The equations for the SVM are presented in Equations 1, 2 and 3 respectively.

In SVM, we want to maximize:

$$\text{Margin} = \frac{2}{\|\vec{w}\|^2} \quad (1)$$

which is equivalent to minimizing:

$$L(w) = \frac{\|\vec{w}\|^2}{2} \quad (2)$$

but subjected to the following constraints:

$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 \end{cases} \quad (3)$$

2.2.2 C4.5 – Decision Tree

Decision trees are tree-based classifiers for samples represented as feature-vectors. A decision tree forecasts the value of a target attribute using a number of input attributes. Building a decision tree is all about finding attribute with the highest information gain[9].

2.2.3 Naïve Bayes, NB

Naïve Bayes is a very common, simple and efficient classifier in ML. The Naïve Bayes classifier is based on the application of Bayes' theorem with strong (naïve) individuality assumptions[10]. This is classifier is most suitable for non-numeric data and robust to noise and irrelevant features.

2.2.4K-Nearest Neighbors (K-NN)

K-NN classifies an instance by considering the majority of the surrounding instances. Whatever class or label that majority of the neighbors of a particular instance belong, that is the class assigned to that instance[11].

2.3Feature Selection Methods Used

Five popular ranker-based feature selection algorithms were used for this study. Ranker-based feature selection methods are computationally efficient and are based on different metrics thereby suitable for making some comparisons.

2.3.1 Information Gain (IG)

IG is a very common univariate filter technique. It evaluates features based on the information they have gained from other features. IG first classifies all the features, and then following a threshold, a certain number of features are selected based on the order obtained [12]. The formula for IG is given in Equation 4.

$$IG(t) = -\sum_{i=1}^m p(c_i) \log p(c_i) + p(t) \sum_{i=1}^m p(c_i | t) \log p(c_i | t) + p(\bar{t}) \sum_{i=1}^m p(c_i | \bar{t}) \log p(c_i | \bar{t}) \quad (4)$$

2.3.2 ReliefF

ReliefF is a multivariate ranking-based method that works by arbitrarily sampling an instance and then finding its nearest neighbor from the same and opposite class. The idea is that a useful feature should be able to separate between samples that belong to different classes.

2.3.3 Symmetrical Uncertainty

Symmetric Uncertainty (SU) focuses on estimating the correlation between the features and the target class [13]. The formula for symmetric uncertainty is given in Equation 5.

$$SU = \frac{H(X) + H(Y) - H(X/Y)}{H(X) + H(Y)} \quad (5)$$

2.3.4 Gain Ratio

Gain ratio (GR) is an extension of the IG feature selection method. GR is able to estimate how much information is needed to decide which branch an instance belongs [14]. The formula for gain ratio is as given in Equation 6. This value represents the potential information generated by splitting the training data set.

$$Gain\ Ratio = \frac{Gain\ (Attribute)}{Intrinsic_info(Attribute)} \quad (6)$$

2.4 Evaluation Techniques

2.4.1 Confusion Matrix (CM)

Here, the confusion matrix (CM) is used to summarize the performance of the classification model. We also used it to determine the correctness and accuracy of the model. The four elements of the CM used include True Positive (TP), True Negative (TN), False Positive (FP), and False negative (FN). Because accuracy metric alone can give misleading results, other metrics like precision, recall and F-measure are also employed especially since the dataset contains classes that have varying number of instances [15].

2.4.2 Accuracy

This refers to the number of correctly predicted sample by the model data divided by all available predictions. The formula is given in Equation 7.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (7)$$

2.4.3 Precision

This metric returns the proportion of data predicted as true, that are actually true. The formula to accomplish this is given in Equation 8.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (8)$$

2.4.4 Recall or Sensitivity

This metric returns the proportion of data that are actually positive, that was predicted as positive. The formula used in realizing recall value is given in Equation 9.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (9)$$

2.4.5 F1 Score – (F-measure)

This is a single metric that can represent both recall (R) and precision (P). The equation for F1-score is given in Equation 10

$$\text{F1 Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

2.5 Experimental Setup

In order to carry out the experiments for all the scenarios, a Dell laptop running Windows 10 with 12Gb Ram and corei7 series was used. Implementation of the ML algorithms and feature selection methods were carried out using Python programming tools and environment. The experiments were carried out in two stages. First stage considered application of ML classifiers directly without feature selection. The second stage was done with feature selection preceding the application of the four ML classifiers. Figure 1 represents the methodology workflow of the study. The figure presents the two separate ways in which the experiments were conducted. In the first experiment, the dataset was divided into training/test subsets in the ratio 70/30. Then the ML classifier was trained first on the 70% percent (training subset), after which the testing set was applied. In the second experiment, the NIMS dataset was subjected to feature selection in order to remove the redundant and non-relevant features. The resulting dataset was then divided into training/test subsets in the usual ratio of 70/30 percent. Several classification experiments were conducted and the results were compared for the two scenarios.

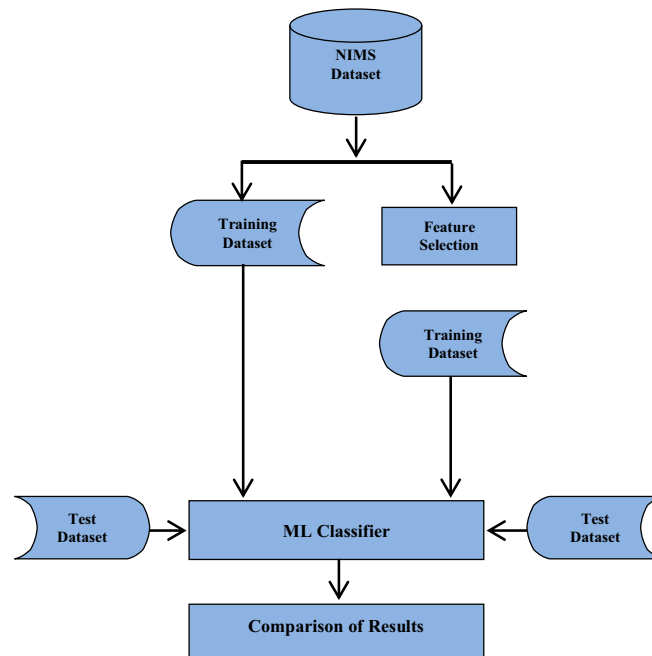


Figure 1: Methodology Workflow of the study

3. Results and Discussion

3.1 Result for Classification Without Feature Selection

The first sets of experiments were carried out by applying the ML classifiers directly to the NIMS dataset. This means all available features were taken into consideration. Five metrics were used to determine the performance of the classifiers which include accuracy, recall, precision, f-measure and runtime. The first four metrics were directly estimated from the confusion matrix values obtained for each classification algorithm. The runtime was obtained by observing the start time and stop of the entire process. The results obtained for this set of experiments are presented in Table 3. As can be seen from the table, SVM was found to spend the highest time during the process of classification. This is not surprising as it has been confirmed that SVM works better with binary class dataset for classification. The NIMS dataset used is a multi-class type of dataset.

The table also indicates KNN and C4.5 as performing well with good results for all metrics including runtime. NB algorithm performed a little lower than the remaining algorithms as indicated in Table 3. The reason for this is that NB is more suitable for non-number(categorical) data compared to numeric data.

Table 3: Classification Performance (Without feature Selection) on the NIMS dataset

Classifier	Accuracy	Precision	Recall	F-Measure	Runtime(s)
NB	85.64	0.873	0.871	0.858	0.38
KNN	98.12	0.993	0.983	0.96	0.18
SVM	94.47	0.934	0.951	0.952	1.35
C45	98.25	0.975	0.984	0.983	0.16

3.2 Result for Classification With Feature Selection

The second sets of experiments were carried out for which feature selection methods were involved before applying classification algorithms. In order to determine the set of features that will likely improve accuracy and other metrics, a set of features was selected after the five feature selection methods were applied. This set comprised of features common to all the five feature selection methods for a reasonable threshold like 15 out of the 21 features of the NIMS dataset. These features were then applied to each of the four ML classifiers for both training and testing of the classifier. Table 4 shows the list of top 10 features from the five feature selection methods used. The table indicates the names as well as the rank of the features. The performance results of the second sets of experiments are presented in Table 5. As can be seen, there is clear improvement for all the five metrics especially runtime and accuracy. While some classifiers had slight improvement like KNN and C4.5, others like NB and SVM had significant improvement.

Table 4: Top ten Features returned by five Ranker-based Feature Selection Methods

Dataset	FS Method	Feature Names	Feature Values/Rankings
NIMS	CFS	{maxbpctl, stdbpctl, stdfpctl, maxfpctl, stdfiat, meanfpctl, stdbiat, meanfiat, meanbkctl, meanbiat}	{0.383, 0.363, 0.327, 0.326, 0.325, 0.318, 0.310, 0.305, 0.298, 0.269}
	GR	{maxfpctl, maxbpctl, minfiat, meanbkctl, meanfpctl, stdbpctl, maxfiat, maxbiat, stdfpctl, totalfvol}	{0.937, 0.588, 0.483, 0.453, 0.442, 0.403, 0.353, 0.348, 0.334, 0.293}
	IG	{maxbiat, maxfiat, meanfpctl, stdbpctl, meanbkctl, stdfpctl, minfiat, meanfiat, minbiat, stdbiat}	{2.476, 2.475, 1.992, 1.960, 1.724, 1.610, 1.572, 1.428, 1.425, 1.410}
	RELIEFF	{meanfpctl, stdbpctl, maxfpctl, stdfpctl, maxfiat, maxbpctl, durattion, maxbiat, stdbiat, stdfiat, meanbkctl}	{0.203, 0.181, 0.174, 0.169, 0.153, 0.136, 0.127, 0.122, 0.112, 0.111}
	SU	{maxfpctl, meanfpctl, meanbkctl, minfiat, stdbpctl, maxfiat, maxbiat, maxbpctl, stdfpctl, stdbiat}	{0.617, 0.562, 0.54, 0.539, 0.526, 0.515, 0.510, 0.455, 0.435, 0.370}

Table 5: Performance Result of four classifiers after feature selection on NIMS Dataset

Classifier	Accuracy	Precision	Recall	F-Measure	Runtime(s)
NB-FS	87.34	0.888	0.893	0.894	0.16
KNN-FS	99.72	0.997	0.997	0.997	0.09
SVM-FS	95.58	0.923	0.927	0.912	0.65
C4.5-FS	99.81	0.998	0.998	0.998	0.07

3.3 Comparison of Machine Learning Approaches

In Table 6, the performance of the two sets of experiments on ML classifiers is presented for comparison purpose. Considering runtime of the four classification algorithms, there is general improvement ranging from 50 %- 58%. Although this time does not include the time spent in carrying out feature selection to select the most useful feature from the available ones. The highest improvement in runtime being NB. The table indicates general improvement in all other metrics when feature selection was applied before the classification process. Figure 2

Table 6: Comparison of ML Approaches (FS = feature selection)

Classifier	Accuracy	Precision	Recall	F-Measure	Runtime(s)
NB	85.64	87.3	87.1	85.8	0.38
NB-FS	87.34	88.8	89.3	89.4	0.16
KNN	98.12	99.3	98.3	96	0.18
KNN-FS	99.72	99.7	99.7	99.7	0.09
SVM	94.47	93.4	95.1	95.2	1.35
SVM-FS	95.58	92.3	92.7	91.2	0.65
C45	98.25	97.5	98.4	98.3	0.16
C4.5-FS	99.81	99.8	99.8	99.8	0.07

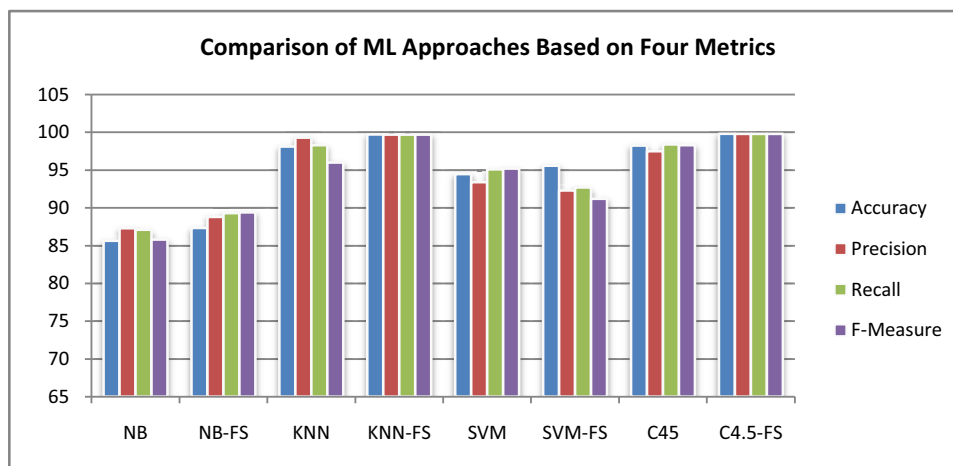


Figure 2: Comparison of the Performance of ML Algorithms before and after the application of Feature Selection Methods

4. Conclusion

This study focused on the design of a methodology for the comparison of the performance of various ML approaches. Scenarios considered included direct application of ML methods to network traffic dataset for classification and scenario that combined feature selection and ML methods for classification. Five different metrics namely accuracy, recall, precision, F-measure, and runtime were used to determine and compare the performance of four selected ML classifiers for two scenarios. The KNN and C4.5 decision tree algorithms performed quite well for both scenarios. The other two namely NB and SVM only improved after feature selection was involved before the classification process. The major reasons

advanced for the poor performance of the NB algorithm is the fact it is suitable most for categorical data rather than numeric data that the employed dataset is comprised. For the SVM algorithm, the low performance is due to the fact that the algorithm is well-suited for binary class dataset for which simple hyper-plane can efficiently differentiate the instance available. The NIMS dataset used is a multi-class dataset. Thus, from the results obtained, we conclude that feature selection is able to improve network traffic classification when the right features are selected. Also, necessary care is required when selecting ML methods to apply to a given type of dataset as not all ML algorithms are suitable for any type of data. In the future, we plan to include other types of feature selection methods like wrapper to evaluate and compare the performance of ML techniques using some additional metrics as well.

Acknowledgement

The authors wish to appreciate the Center for Research, Innovation, and Discovery (CU-CRID) of Covenant University, Ota, Nigeria for funding this research.

References

- [1] Boutaba, R., Salahuddin, M., A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., & Caicedo, O., M. (2018). A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications*, 9(1), 1-99.
- [2] Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, 300, 70–79. doi:10.1016/j.neucom.2017.11.077
- [3] Nguyen, T., T., & Armitage, G. (2008). A Survey of Techniques for Internet Traffic Classification using Machine Learning. *IEEE Communications Surveys and Tutorials*, 10(4), 56-76, IEEE Press, Piscataway, New Jersey, USA.
- [4] Callado, A., Kamienski, C., Szabó, G., Péter, B., G., Kelner, J., Fernandes, S. & Sadok, D. (2009). A Survey on Internet traffic identification. *IEEE Communications Surveys and Tutorials*, 11(3), 37-52.
- [5] Foremski, P. (2013). On different ways to classify Internet Traffic. A short review of selected publications. *Theoretical and Applied Informatics*, 25(2), 119-136
- [6] Fan, Z., & Liu, R. (2017). Investigation of Machine Learning Based Network Traffic Classification. International Symposium on Wireless Communication Systems (ISWCS), Bologna, Italy, pp 1-6.
- [7] Shafiq, M., Yu, X., Laghari, A., Yao, L., Karn, N., & Abdessamia, F. (2016). Network Traffic Classification Techniques and Comparative Analysis Using Machine Learning Algorithms. 2nd IEEE International Conference on Computer and Communications, Chengdu, China, pp 2451-2455.
- [8] Fahad, A., H., (2015). Designing an accurate and efficient classification approach for network traffic monitoring. *Ph.D Thesis*, RMIT University, Melbourne, Australia.
- [9] Alshammari, R. & Zincir-Heywood, A., N. (2011). Can encrypted traffic be identified without port numbers, IP Addresses and Payload Inspection? *Journal of Computer Networks*, Elsevier, 2011.
- [10] Rokach, L., & Maimon, O. (2008). Data mining with decision trees: theory and applications. *World Scientific Publishing Company Inc.* ISBN 978-9812771711
- [11] Bhatia, N., & Ashev, V. (2010). Survey of Nearest Neighbor Techniques. *International Journal of Computer Science and Information Security*, 8(2), 1-4.
- [12] Nielsen, T., D., & Jensen, F., V. (2009). Bayesian Networks and Decision Graphs. In: Springer Science and Business Media.
- [13] Quinlan, J., R. (2014). C4.5: Programs for Machine Learning. In: Elsevier, Machine Learning, Morgan Kaufmann Publishers
- [14] Witten, I., H., & Frank, E. (2005). Data mining: practical machine learning tools and techniques with

java implementations (2nd Edition). Morgan Kaufmann Publishers, San Francisco.

[15]Praveena, R., Valarmathi, M., L., &Sivakumari. (2011). Gain ratio based feature selection method for privacy preservation. *ICTACT Journal on Soft Computing*. 10.21917/ijsc.2011.0031.