Empirical Comparison of Cross-Validation and Test Data on Internet Traffic Classification Methods

Oluranti Jonathan, Nicholas Omoregbe, Sanjay Misra
Email: jonathan.oluranti@covenantuniversity.edu.ng

Abstract. In this paper, we compare two validation methods that are used to estimate the performance of classification algorithms in a non-problem-specific knowledge scenario. One way to measure the performance of a classification algorithm is to determine its prediction error rate. However, this value cannot be calculated but estimated. In this work, we apply and compare two common methods used for estimation namely: test data and cross-validation. Precisely, we analyze and compare the statistical properties of the K-fold cross-validation and test data estimators of the prediction error rates of six classifiers namely: Naïve Bayes, KNN, Random Forest, SVM, J48, and OneR. From the study, the statistical property of repeated cross-validation tends to stabilize the prediction error estimation which in turn reduces the variance of the prediction error estimator when compared with test data. The NIMS dataset collected over a network was employed in the experimental study.

Keyword: cross-validation, classification, performance metrics, machine learning

1. Introduction

Internet traffic Classification or characterization is the module of network management responsible for associating network traffic with the application generating them [1-3]. Among its benefits are: network monitoring, network security, network resource management etc. Traditional methods of traffic classification which include port-based, payload-load based, host-based, behavior-based are limited in a number of ways such as dynamic port assigning to applications, encryption of application contents, privacy issues, to mention a few [4,5]. Nowadays, it is common to use machine learning techniques that are based on statistical properties of traffic flows such as maximum length of packets, inter-arrival time of packets among others for identifying the traffic flow[2]. However, given a machine learning-based problem for instance, classification, the first thing expected after understanding the problem and the dataset at hand, is to decide on the various machine learning methods to apply. For a classification problem, the selection of appropriate classifiers for which the dataset serves as input is a major issue of concern that should carefully tackled as this has effect on the accuracy of the classification result [6,7].

It therefore becomes necessary to adopt a well-planned action on how to select the appropriate classifiers to use. According to [8], one acceptable way to achieve this is to evaluate the performance of a learning algorithm using its prediction error, a value that indicates the prediction capability of the learning algorithm over an independent test set. As mentioned earlier, this value is not usually calculated but estimated by running various experiments using the dataset provided. Some studies exist that focus on developing appropriate prediction error estimators including hold-out validation, K-fold cross-validation, leave-one-out cross-validation, and repeated K-fold cross-validation [9]. This study focuses on evaluating the performance of a number of classifiers intended for use on the NIMS network traffic dataset using well-known estimators namely; test data and cross-validation. The dataset is applied to each of the six classifiers first based on training/test data formula (Test) and the results noted. The same classifiers are applied again to the training set of the same dataset but this time with 10-fold cross-validation. The results of this operation are also noted for the purpose of comparison. The remaining sections of this work include section two which focuses on related works, Section three presents some discussions on cross-validation as well as experimental setup and analysis of results. Section four describes some the performance measures employed in the study while...
section five discusses some of the results obtained. In section six, the conclusion of the entire study is presented alongside recommendation for future work.

2. Related Works

Several works related to the subject matter have been carried out. In this section we discuss a number of them from which we draw insights into the work carried in this study. In the work of [10], an ensemble of non-related methods was proposed with the aim of identifying Peer-to-Peer (P2P) traffic. The learning machines include Bayes, Support vector machines (SVMs) and Decision trees the voting principle was used to realize the desired results. Experimental results also show that the ensemble method is able to improve the classification accuracy. The authors in [11] also proposed a new semi-supervised learning model based on K-means and Transductive support vector machines to improve the accuracy of P2P traffic identification. The work of [12] focused on a comparative analysis of the classification of botnets command and control traffic as a result of the rising effect of Botnets with modular and flexible structures. In the study, the traffic flow features were carefully combined with some of the selected command and control for better classification using machine learning algorithms. The results obtained suggested that the method applied was very effective as it yielded very good test accuracy and very little training time. Comparison of the performances of a number of classifiers was done and from the results obtained they proposed a rule induction algorithm based on one of the algorithms that participated. The outcome of the experiments carried out pointed to the fact that their proposed method produced better accuracy than the original classifier. In 2014, [13] presented a machine learning approach that accurately classified live traffic using C4.5 decision tree. By collecting 12 features at the start of the flows, without inspecting the packet payload, their method has identified live traffic of different types of applications with 99.8% total accuracy. In another experiment the authors focused on the issue of Peer-to-Peer (P2P) traffic identification in internet network analysis. They applied an ensemble learning model in which they integrated Random Forests and feature weighted Naïve Bayes to P2P traffic identification. For the prediction stage, calculation of scores was done for each category in the model. This was followed by weighted majority voting which they applied to obtain the final output. The effectiveness and stability of the new model was also verified by conducting some experiments. Results have shown that the model achieved a better overall performance and may provide an alternative way to solve the problem of P2P traffic identification.

3. Experimental design

The general methodology adopted is experimental and comprises of a number of steps beginning with description of the dataset used for the experiments. The specific steps employed in this work are as follows:

i) Divide the given dataset into training/test data (70:30)

ii) Run the six classifiers over the training set using 10-fold cross validation method as highlighted in figure 1.

iii) Run another set of operations comprising of training and supplied test data using the same set of classifiers used in as highlighted in figure 1.

iv) Compare the performance of the various classifiers for cross–validation and supplied test dataset in terms of Accuracy, Precision, Recall, AUC, and F-measure

iv) Present findings in tables and charts with proper analysis and further discussion

3.1 NIMS dataset Description

NIMS data set is one of the classical datasets used for the study. It includes packets collected from a tested network and made available by the original authors for download. The data set consists of SSH servers outside connection and application behaviors traffic such as DNS, HTTP, SFTP and P2P traffic. The detailed characteristics of the dataset are presented in Table 1 below.
### Table 1. NIMS Dataset description

| s/n | Name | No. of Instances |
|-----|------|------------------|
| 1   | DNS  | 38,016           |
| 2   | FTP  | 1,728            |
| 3   | HTTP | 11,904           |
| 4   | LFD  | 2,557            |
| 5   | RFD  | 2,422            |
| 6   | SCP  | 2,444            |
| 7   | SFTP | 2,412            |
| 8   | SHELL| 2,491            |
| 9   | TELNET | 1,251          |
| 10  | X11  | 2,355            |

*Figure 1. Workflow for Application of Classifiers to Test Data and 10-Fold Cross-Validation*
4. **Performance measures**

In order to select the algorithm that is best suitable for classifying the traffic patterns of NIMS dataset, it is important to evaluate the classification performances of suite of classifiers SVMs, Naïve Bayes, Decision Tree and Random Forest, KNN, etc. on experimental dataset. The classification performance of a classifier is generally measured in terms of several metrics like classification accuracy, kappa value, F-measure or by analyzing the predicted results in a confusion matrix. But the selection of evaluation metrics should be made in accordance to the class proportions in the experimental data and also the classification problem that we are aiming to solve [14]. In our study, we have considered classification accuracy and F-measure to evaluate the classification performances of our five chosen classification algorithms.

- **Accuracy**

  Classification accuracy is the number of correct predictions made divided by total number of predictions made, multiplied by 100 to turn it to a percentage. It is the percentage of correct classifications made by the classifier.

\[
\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad \text{---------- equation 1}
\]

- **F1 score**

  The F1 Score is a measure of a test's accuracy. It considers both precision (p) and recall (r) of the test to compute the score.

\[
\text{F1 Score} = \frac{2*(\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}} \quad \text{---------- equation 2}
\]

- **Precision:** It is the number of positive predictions divided by the total number of positive class values predicted by the classifier.

\[
\text{Precision} = \frac{TP}{TP+FP} \quad \text{---------- equation 3}
\]

- **Recall:** It is the number of positive predictions divided by the number of positive class values in the test data.

\[
\text{Recall} = \frac{TP}{TP+FN} \quad \text{---------- equation 4}
\]

5. **Results and Analysis**

The experiments were carried out using a laptop computer with the following configuration Intel Core (i7) CPU @3.72GHz, 16 GB RAM, x64-based processor and 64-bit Windows 10 OperatingSystem. Some of the algorithms were implemented in R language while some others were done in WEKA, a popular machine learning tool used for feature selection and Classification.
Table 2. Classification Performance Comparison Using 10-fold Cross Validation (CV) vs Test Dataset (TD)

| Classification Algorithm | Testing Method | Accuracy (%) | Precision | Recall | F-Measure | AUC  |
|--------------------------|----------------|--------------|-----------|--------|-----------|------|
| Naïve Bayes              | 10-CV          | 97.6         | 0.964     | 0.956  | 0.953     | 0.998|
|                          | TD             | 95.91        | 0.966     | 0.959  | 0.956     | 0.997|
| KNN                      | 10-CV          | 99.52        | 0.995     | 0.995  | 0.995     | 0.999|
|                          | TD             | 97.53        | 0.995     | 0.995  | 0.995     | 0.999|
| Random Forest            | 10-CV          | 99.88        | 0.999     | 0.999  | 0.999     | 1.000|
|                          | TD             | 98.75        | 0.999     | 0.999  | 0.999     | 1.000|
| SVM                      | 10-CV          | 96.73        | 0.970     | 0.967  | 0.967     | 0.997|
|                          | TD             | 95.50        | 0.969     | 0.965  | 0.965     | 0.998|
| J48                      | 10-CV          | 99.79        | 0.998     | 0.998  | 0.998     | 0.999|
|                          | TD             | 94.85        | 0.998     | 0.998  | 0.998     | 0.999|
| OneR                     | 10-CV          | 93.60        | 0.935     | 0.936  | 0.935     | 0.949|
|                          | TD             | 94.32        | 0.942     | 0.943  | 0.942     | 0.955|

Figure 2. Performance Comparison of Six Classifiers using NIMS Dataset (CV vs Test Data) - Accuracy
Figure 3. Performance Comparison of Six Classifiers using NIMS Dataset (CV vs Test Data) – Precision

Figure 4. Performance Comparison of Six Classifiers using NIMS Dataset (CV vs Test Data) – Recall
**Figure 5.** Performance Comparison of Six Classifiers using NIMS Dataset (CV vs Test Data) – Area Under ROC

**Table 3.** Precision - Classification Performance Per Application using CV-10 and Test Data

| Application | Naïve Bayes | KNN | Random Forest | SVM | J48 |
|-------------|-------------|-----|---------------|-----|-----|
| CV-10 TD    | CV-10 TD    | CV-10 TD    | CV-10 TD    | CV-10 TD    | CV-10 TD    |
| Dns         | 0.999 0.999 | 1.000 1.000 | 1.000 1.000 | 1.000 1.000 | 1.000 1.000 |
| Ftp         | 0.999 0.758 | 0.985 0.970 | 1.000 1.000 | 0.955 0.894 | 0.998 1.000 |
| Hftp        | 0.987 0.992 | 0.999 0.999 | 1.000 0.997 | 0.996 0.996 | 0.996 0.995 |
| LFWD        | 0.971 0.994 | 0.995 0.997 | 0.997 1.000 | 0.999 0.997 | 1.000 1.000 |
| RFWD        | 1.000 1.000 | 0.997 0.988 | 0.995 1.000 | 1.000 1.000 | 1.000 0.997 |
| Scp         | 0.817 0.833 | 0.963 0.945 | 0.999 0.987 | 0.627 0.572 | 0.988 0.990 |
| Sftp        | 0.561 0.615 | 0.956 0.974 | 0.990 1.000 | 0.767 0.788 | 0.994 1.000 |
| Shell       | 0.878 0.884 | 0.971 0.986 | 1.000 0.986 | 0.841 0.850 | 0.990 0.983 |
| Telnet      | 1.000 1.000 | 0.995 1.000 | 0.990 1.000 | 0.964 0.988 | 0.992 1.000 |
| X11         | 0.987 0.954 | 1.000 1.000 | 0.999 1.000 | 0.999 0.994 | 0.993 0.997 |

**5.1. Discussion**

From the accuracy measure perspective, presented in Table 2, the applied machine learning classifiers give maximum classification accuracy results with Random Forest machine learning classifier providing the overall maximum classification accuracy for NIMS dataset, 99.88%, which is effective
classification accuracy value as compared to other machine learning classifiers’ accuracy results. On the other side, all applications are classified with very effective values as shown in Table 3; however, DNS, X11, LFWD, RFWD, and HTTP show better precision indicating that the selected classifiers work quite well for them. For the precision, the maximum values obtained include 1.000 for a number of classifiers and applications. However, the highest value of precision of value 100% was recorded on the Random Forest classifier. This agrees with the values on Table 2 with respect to accuracy.

However, Random Forest machine learning classifier gives very effective results in terms of Classification accuracy. The details are shown in Table 2 and Table 3. In NIMS dataset, DNS application is classified more of 100% when compared to other traffic applications and the applied machine learning classifiers give very accurate identification results for DNS. The applied machine learning classifiers give very accurate performance results for NIMS dataset, but Naïve Bayes machine learning classifier give slightly lower accuracy and precision values for Scp, Sftp and Shell applications.

6. Conclusion
This study aimed at answering the pertinent question of which classification model to select for the problem at hand. In attempt to answer the question, comparison of two estimation methods namely cross-validation and test data was carried out. The study demonstrates the possibility of applying classification algorithms of machine learning to network traffic patterns. The results obtained among other things suggest that all the classification methods applied are capable of being used with NIMS dataset. However, based on the question of which of the six methods to select, Random Forest is topmost and therefore the best suitable method for classifying the traffic patterns of NIMS dataset. In the future we plan to include some statistical tests like Friedman and Nemenyi to ascertain the similarities of the measurements obtained from the experiments carried out for the classifiers.

Acknowledgement
The authors of this research do appreciate the Covenant University Centre for Research, Innovation, and Discovery (CUCRID) for its support for this research.

References
[1] Nguyen T.T. and Armitage G. (2008). A Survey of Techniques for Internet Traffic Classification using Machine Learning. IEEE Communications Surveys & Tutorials, 10 (4): 56-76, IEEE Press, Piscataway, New Jersey, USA.

[2] Callado A., Kamienski C., Szabó G, Péter B. G., Kelner J. Fernandes S., and Sadok D. (2009). A Survey on Internet Traffic Identification", IEEE Communications Surveys & Tutorials, 11(3):37-52, IEEE Press, Piscataway, New Jersey, USA.

[3] Zhang J., Xiang Y., Wang Y., Zhou W., Xiang Y., Guan Y. (2013) Network traffic Classification using correlation information. IEEE Trans Parallel Distrib Syst 24(1):104

[4] Dainotti A., Pescap’e A., and Claffy K.C.(2012) Issues and future directions in traffic characterization. Network, IEEE, 26(1):35 –40
[5] Fahad, A., Tari, Z., Khalil, I., Habib, I., Alnuweiri, H., “Toward an efficient and scalable feature selection approach for Internet traffic classification”, Computer Networks, vol. 57, issue 9, pp. 2040–2057, 2013.

[6] Sai S.K. (2017). On Evaluating Machine Learning Approach for Efficient Classification of Traffic Patterns, Masters Thesis.

[7] Baron G. (2016) Comparison of Cross-Validation and Test Sets Approaches to Evaluation of Classifiers in Authorship Attribution Domain. In: Czachórski T., Gelenbe E., Grochla K., Lent R. (eds) Computer and Information Sciences. ISCIS 2016. Communications in Computer and Information Science, vol 659. Springer, Cham

[8] Baron, G., Harezlak, K.: On Approaches to discretization of datasets used for evaluation of decision systems. In: Czarnowski, I., Caballero, A.M., Howlett, R.J., Jain, L.C. (eds.) Intelligent Decision Technologies 2016, vol. 57, pp. 149–159. Springer, Cham (2016)

[9] Liang, Chengchao, F. Richard Yu, and Xi Zhang. "Information-centric network function virtualization over 5G mobile wireless networks." IEEE Network 29.3 (2015): 68-74.

[10] Ding Li, Li Mao, and Xiao Feng Wang (2014) "Classification of P2P traffic based on a heteromorphic ensemble learning model." Applied Mechanics and Materials. Vol. 687. Trans Tech Publications, 2014.

[11] Ding Yaojun, and Wandong Cai. (2012). P2P traffic identification via k-means based transductive support vector machine. Journal of Xi'an Jiaotong University 2 (2012): 007.

[12] Barthakur P., Manoj D. and Mrinal K. G. "An efficient machine learning based classification scheme for detecting distributed command & control traffic of P2P botnets." International Journal of Modern Education and Computer Science 5.10 (2013): 9.

[13] Shafiq, M., Xiangzhan Y, and Asif Ali L. (2014). WeChat Text Messages Service Flow Traffic Classification Using Machine Learning Technique. 'IT Convergence and Security (ICITCS), 2016 6th International Conference on. IEEE, 2016.

[14] Stapor Katarzyna. (2017). Evaluating and Comparing Classifiers: Review, Some Recommendations and Limitations. International Conference on Computer Recognition Systems. Springer, Cham.