Vol 6 • Number 2 April - Sep 2015 pp. 275-280 **Impact Factor: 2.5** **Available at** www.csjournals.com

A Data Mining Tool for Network Fault Detection

Poonam Chaudhary & Vikram Singh
Department of Computer Science, Ch. Devi Lal University, Sirsa

Abstract: A new data mining tool detecting faults in the communication networks has been developed over the existing tool WEKA using the eclipse – an integrated development environment that is used to develop programs in diverse languages like java, ruby etc. The present tool has been developed in the java environment.

1. INTRODUCTION

Telecommunication networks are extremely complex configurations of hardware and software. Almost the network elements are capable of at least limited self-diagnosis. These elements may collectively generate millions of status and alarm messages each month (Fawcett & Provost, 1997). Alarms must be analyzed automatically in order to identify network faults in a timely manner for effectively manage the network—or before they occur and degrade network performance. The proactive response is essential to maintaining the reliability of the network. So, volume of the data, and because a single fault may cause various different, seemingly unrelated, alarms to be generated, so the task of network fault isolation is quite difficult. The data mining has a role to play in generating rules for identifying faults (Han, J. et al., 2002).

1.1 Mobile network faults

Mobile network fault can be defined as an abnormal operation or defect at the component, equipment, or sub-system level that is significantly degrades performance of an active entity in the network or disrupts communication. Each and every error is not faults as protocols can mostly handle them. Mostly faults may be indicated by an abnormally high error rate. The fault can be defined as an inability of an item to perform a required function (a set of processes defined for purpose of achieving a specified objective), excluding that inability due to preventive maintenance, lack of external resources, or planned actions.

There is lack of a generally accepted definition of what constitutes behaviour of a normal mobile network fault (Hajji, et al., 2001; Hajji & Far, 2001; Lin & Druzdzel, 1997). Therefore, it is very difficult to characterize the mobile network faults accurately. However, there are estimations (based on statistics of the network traffic) as to what characterize a mobile network fault. The mobile network faults are characterized by transient performance degradation, high error rates, loss of service provision to the customers (i.e., loss of signal loss of connection, etc), delay in delivery of services and getting connectivity.

The main causes of network faults differ from network to network. Managing complex hardware and software systems has always been a difficult task. The Internet and the proliferation of web-based services have increased the importance of this task, while aggravating the problem (faults) in at least four ways (Meira, 1997; Thottan & Ji, 1998; Hood & Ji, 1997; Lazar et al., 1992).

2. DESIGN OF THE TOOL

A new tool atop the WEKA data mining shareware has been designed and developed using eclipse integrated development environment (IDE). The eclipse IDE consists of the package explorer, editor and various others windows. The package explorer shows the projects developed using eclipse. The editor window shows the code of the selected file. The opening screen of the eclipse is shown in the following figure 1.

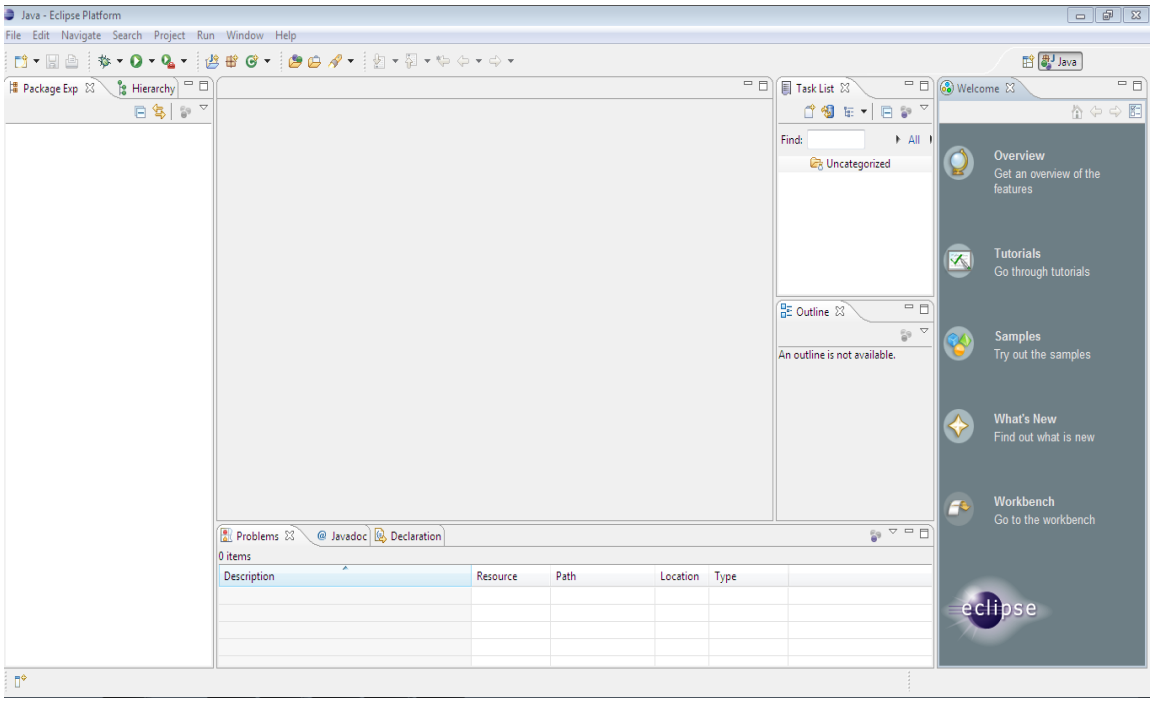

Figure 1 The eclipse IDE

Logic for the tool – in form of algorithm – has been added creating a new class in the required package. The simulation develops proposed algorithm under the *weka.classifier.function* package. A new java class named *Proposed.java* is created as depicted in fig. 2.

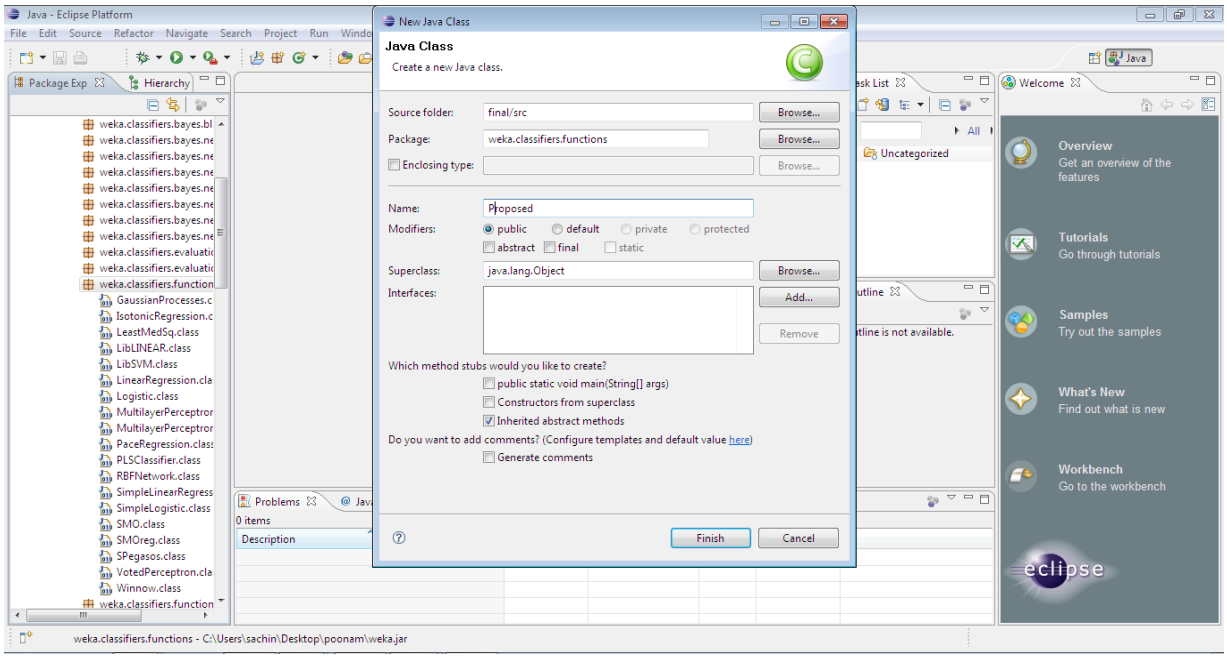

Figure 2: Creation of new class in eclipse

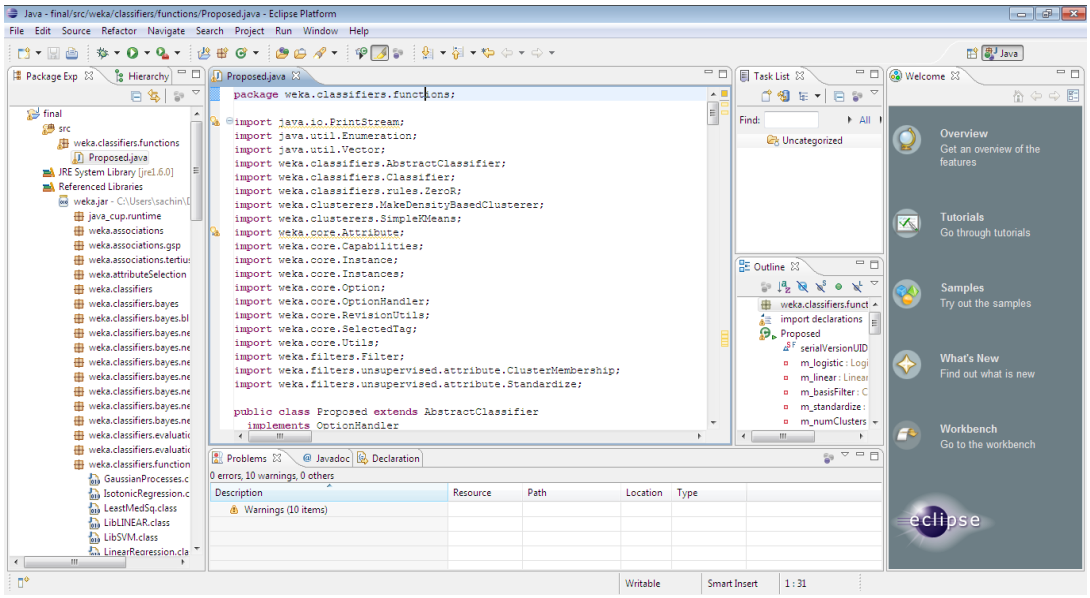

Figure 3: Code of proposed algorithm

The java code of the proposed algorithm is entered or copy-pasted in the blank java file. Figure 3 shows the eclipse editor window containing the code of proposed algorithm. New algorithm gets displayed in side pane alongside the existing algorithm in WEKA tool. Bare WEKA does not consist of the "Proposed" algorithm whereas the "Proposed" tab is available in the left side pane of the WEKA explorer in fig. 4.

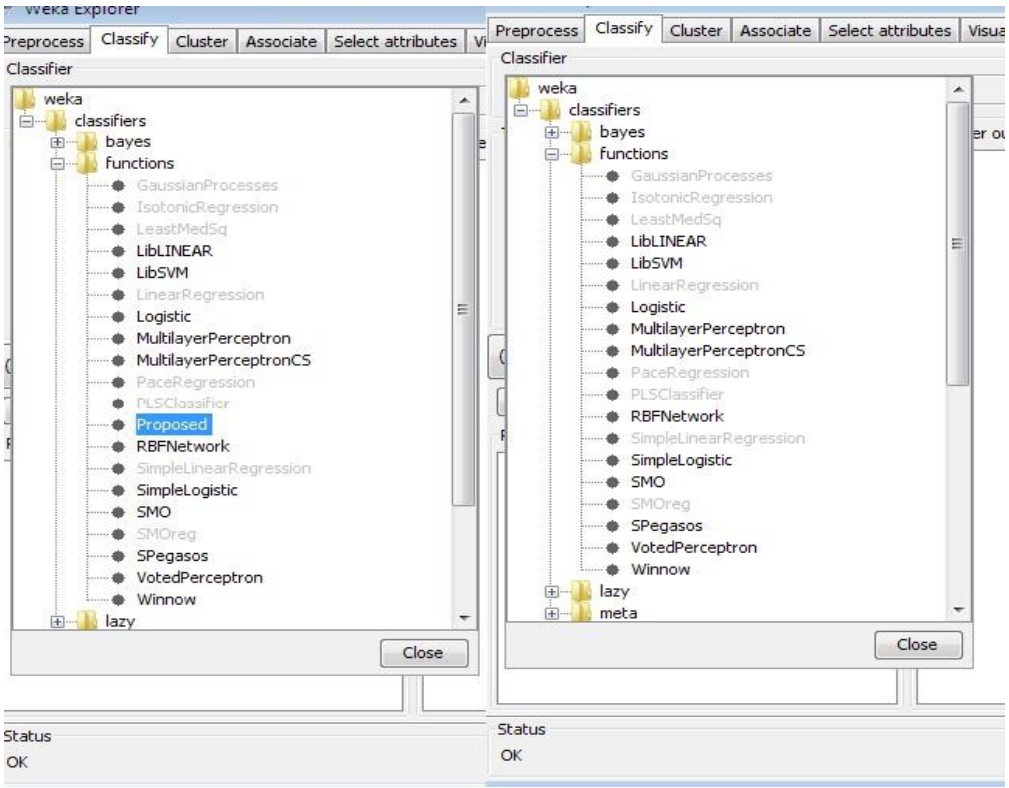

Figure 4: Proposed tool in WEKA explorer

3. PERFORMANCE EVALUATION

In the present work precision and recall, ROC and lift have been chosen as the performance metrics for estimating the accuracy of the proposed tool. Each of these was used where appropriate in the analysis of the performances. Apart from the aforementioned performance criteria, the speed and the robustness of the classifier tool have also been taken into consideration.

Precision and recall: The general percentage accuracy as a performance measure has been proven to be misleading (Provost et al., 1998). For example, a classifier that labels all regions as the majority class will achieve an accuracy of 94%, because 96% of the majority may belong to that region. Accordingly, the classifier may have incorrectly classified some of the minority class instance as the majority because of the bias nature of the dataset but may appear to be accurate. It is therefore imperative to compare the accuracy using an alternative method - precision and recall – as derived below.

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP+FP}} \times \mathbf{100}\% \qquad (1)$$

$$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP+FN}} \times \mathbf{100}\% \qquad (2)$$

Where, TP, TN, FP, and FN are as represented in the confusion matrix. *Precision* in this context refers to the actual percentage of responses to mails that were predicted by the classification model, which translates into the returns on cost of mailing; and the *recall*, on the other hand, measures the percentage of customers that were identified and needed to be targeted.

ROC analysis: An alternative method that was used to evaluate classifier performance is the Receiver Operating Characteristic (ROC) analysis (Provost and Weiss, 2001. It compares visually the performance of the classifier across the entire range of probabilities. It shows the trade-off between the false-positive rate on the horizontal axis of a graph and the true-positive rate on the vertical axis. From the values obtained from the confusion matrix above, the true-positive rate and false positive rate could be defined as equations (3) and (4) respectively

$$\frac{\mathbf{TP}}{\mathbf{TP+FN}} \qquad (3)$$

$$\frac{\mathbf{FP}}{\mathbf{FP+TN}} \qquad (4)$$

As a standard method for evaluating classifiers, the primary advantage of ROC curve is that they are used to evaluate the performance of a classifier independent of the naturally occurring class distribution or error cost. A good classifier must achieve high TP rate and at the same time less FP rate.

Lift chart analysis: Lift chart is a well-known tool used for the evaluation of the effectiveness of a predictive model in the marketing domain. It displays a graphical representation of the change in lift that a mining model may cause, which enables marketers to determine the predictive model that will produce the highest hit rate. It is similar to the ROC curve described above. The difference is in the axis. The horizontal axis represents the population size (customers) or possible mailing expressed as a percentage, while the vertical axis represents the actual customers that will respond to the offer. This makes it easier to examine the returns each technique will produce at a glance when the mailing cost and the returns on the investment are known. From the confusion matrix above, y axis represents TP. The horizontal axis is as shown in equation (5)

$$\mathbf{PopulationSize} = \frac{\mathbf{TP+Fp}}{\mathbf{TP+FP+TN+Fn}} \times \mathbf{100}\% \qquad (5)$$

Speed: Speed refers to the time it takes a classifier to be trained. This translates to how demanding the algorithm is, in terms of the resources it will require to run. To measure this, the CPU time was used to assess how long it takes during building and training. This was recorded in seconds and computed. The WEKA time stamped is the time it took to build and train a model.

Robustness: Because real world datasets are never perfect and could suffer from noise, which may have a negative impact on the interpretation of the output, it is always important for a classifier to have the capability of handling these without affecting the accuracy of the performance in general. To evaluate the impact of noise on the performance of the techniques, variable noise was introduced into the dataset. Variables were randomly selected; missing and unknown values were inserted from 5% to 25% with a step of 5%. The idea is to contaminate the datasets (Zhou et al., 2004). After this the classifiers are then re-applied to the porous dataset and the performance is measured and compared.

4. RESULTS AND CONCLUSION

Ericson data set has been used for comparing the performance of proposed tool vis-à-vis J48 and MLP tools available in the public domain. Tables 1, 2 and 3 list the results of performance of three tools when run on the same test data.

Table 1: Parameter analysis using J48

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	1	0.458	1	0.629	0.406	Major
	0	0	0	0	0	0.182	minor
	0	0	0	0	0	0.425	Minor
	0	0	0	0	0	0.212	Critical
Weighted Avg.	0.458	0.458	0.21	0.458	0.288	0.374	

Table 2: Parameter analysis using MLP

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.636	0.692	0.438	0.636	0.519	0.439	Major
	0.5	0.023	0.667	0.5	0.571	0.733	minor
	0.235	0.29	0.308	0.235	0.267	0.49	Minor
	0	0	0	0	0	0.233	Critical
Weighted Avg.	0.417	0.422	0.365	0.417	0.38	0.46	

Table 3: Parameter analysis using new Algorithm

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.182	0.231	0.4	0.182	0.25	0.51	Major
	0.5	0.091	0.333	0.5	0.4	0.807	minor
	0.235	0.194	0.4	0.235	0.296	0.566	Minor
	0.4	0.465	0.091	0.4	0.148	0.544	Critical
Weighted Avg.	0.25	0.23	0.362	0.25	0.268	0.558	

From the results contained in the above listed tables it can be made out that proposed tool yields lesser true-positive rates of 0.25 against the 0.458 and 0.417 of J48 and MLP respectively. Similarly, lesser false-positive rates of 0.23 have been yielded against the 0.458 and 0.422 of J48 and MLP respectively. As far precision rate goes the performance of proposed tool (0.362) is comparable to MLP (0.365) whereas it lesser than J48 (0.21). Recall rate of proposed tool (0.25) is far less than both J48 (0.458) and MLP (0.417).

REFERENCES

1. Provost, F., *"Learning when training data are costly: The effect of class distribution on tree induction*". Journal of Artificial Intelligence Research 2003; 19:315- 354.
2. Fawcett, T., Provost, F. ,*"Adaptive fraud detection. Data Mining and Knowledge Discovery*" 1997; 1(3):291-316.
3. Han, J., Altman, R. B., Kumar, V., Mannila, H., Pregibon,*" D. Emerging scientific applications in data mining*". Communications of the ACM 2002; 45(8): 54-58.
4. Hajji, B. & Far, B. H. (2001), *"Continuous Network Monitoring for Fast Detection of Performance Problems", Proceedings of 2001 International Symposium on PerformanceEvaluation of Computer and Telecommunication Systems*, July 2001.

5. Hajji, B.; Far, B. H. & Cheng, J. (2001), *"Detection of Network Faults and Performance Problems*", *Proceedings of the Internet Conference*, Osaka, Japan, Nov. 2001.
6. Lin, Y. & Druzdzel, M. J. (1997), *"Computational Advantages of Relevance Reasoning in Bayesian Belief Networks", Proceedings of the Thirteenth Annual Conference inUncertainty in Artificial Intelligence (UAI-97)*, pp. 342-350, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1997.
7. Meira, D. M. (1997), *"A Model for Alarm Correlation in Telecommunications Networks",PhD Thesis*, Federal University of Minas Gerais, Belo Horizonte, Brazil, Nov. 1997.
8. Thottan, M. & Ji, C. (1998), *"Proactive Anomaly Detection Using Distributed Intelligent Agents"IEEE Network*, Sept./Oct. 1998.
9. Hood, C. S. & Ji, C. (1997), *"Proactive Network Fault Detection"*, *Proceedings of the IEEE INFOCOM*, pp. 1139-1146, Kobe, Japan, April 1997.
10. Lazar, A.; Wang, W. & Deng, R. (1992), " *Models and algorithms for network fault detection and identification: A review", Proceedings of IEEE ICC, Singapore, pp.999-1003, November 1992.*
11. Zhao, Q. & Xu, Z., *"A novel approach to fault detection and isolation based on wavelet analysis and neural network,"Electrical and Computer Engineering*, vol. 1, pp. 572–577, May. 2002.