International Journal of Software Innovation
Volume 7 • Issue 2 • April-June 2019

Network Support Data Analysis for Fault Identification Using Machine Learning

Shakila Basheer, King Khalid University, Abha, Saudi Arabia
Usha Devi Gandhi, VIT University, Vellore, India
Priyan M.K., VIT University, Vellore, India
Parthasarathy P., VIT University, Vellore, India

ABSTRACT

Machine learning has gained immense popularity in a variety of fields as it has the ability to change the conventional workflow of a process. The abundance of data available serves as the motivation for this. This data can be exploited for a good deal of knowledge. In this article, we focus on operational data of networking devices that are deployed in different locations. This data can be used to predict faults in the devices. Usually, after the deployment of networking devices in customer site, troubleshooting these devices is difficult. Operational data of these devices is needed for this process. Manually analysing the machined produced operational data is tedious and complex due to enormity of data. Using machine learning techniques will be of greater help here as this will help automate the troubleshooting process, avoid human errors and save time for the technical solutions engineers.

KEYWORDS

ID3 Algorithm, Network Support Data, Predictive Analytics, Rule Based Induction, Show Tech Support, Supervised Learning Techniques, Technical Solutions Engineer (TSE)

1. INTRODUCTION

Text Mining is widely used nowadays to mine useful patterns from text. Text mining has a found its use in business, medicine, education, drug discovery, etc. As an effect of this, there is a lot of research going on to analyse natural language (i.e. human produced data). But, nowadays machines also produce enormous amount of operational data which are semi-structured (e.g. system logs, usage logs, error logs etc.). The data produced by machines is the main source of identifying faults in machines. So, it is crucial to analyse machine produced data.

The machine produced data that we are considering for our course of research is 'Show Tech Support'. These are referred as Network Support data files. These files contain operational data of networking devices that are deployed at different sites. The Network Support data are the source of many useful information about the device. It contains information about the software and features that are configured in the device. The contents of support data files vary from time to time as it is a

DOI: 10.4018/IJSI.2019040104

device's operational data. When there is a malfunction or fault in a device, support data files are of use. They are used to trouble shoot the faults in the networking devices. These files are unique to their make and device configuration set up. Support data files are difficult to analyse manually because of the complex nature and enormity data. Also, manual analysis is always prone to errors. So, the need for a more systematic and automated analysis arises.

This paper has outlined ongoing research work in Machine Learning and its relevance to our problem in section 2. In section 3, we have discussed about the methodology used. In section 4, a comprehensive comparison between ID3 algorithm and Rule Based induction is done.

2. LITERATURE SURVEY

2.1. Introduction to Machine Learning

In an attempt to analyse the industrial data/machine produced data research fraternity has done a significant contribution. For industrial data analysis, a strong subject expertise is needed. But, the huge result sets and internal relationships between the workflow is sometimes beyond our subjective knowledge. To overcome this, a more generic framework for processing industrial data is needed. Mr. Mariusz Kamola, in his work (2015) has comes up with a defined set of rules for choosing the most required features for predictive analysis on industrial data. Clearly, the processing framework will differ depending on the use case and type of analysis. So, choice of a suitable Machine Learning algorithm is necessary.

Surya, Nithin, Prasanna, and Venkatesan (2016), gives a brief introduction to machine learning and discusses about various machine learning techniques and pre-processing techniques. The paper discuses about three main topics. They are:

- Types of machine learning
- Machine learning techniques
- Linguistic pre-processing

Types of Machine Learning:

- **Supervised Learning:** In this technique, knowledge is referred from training datasets. Example: classification and regression;
- **Unsupervised Learning:** In unsupervised learning, there is no training datasets. In this technique, knowledge is inferred from input data that are not tagged. Example: clustering and dimensional Reduction;
- **Reinforcement Learning:** A software agent is trained to make suitable decisions to be taken which will be based on the previous experience;
- **Machine Learning:** Techniques discussed are, N-Gram and Markov Models, Neural Networks and Decision Tree classifiers;
- **Linguistic Pre-Processing:** This step is a preparatory step which prepares the process to take place. This will ensure that the text will be in a form that would be understood by the machine. Here the context of a word is understood.

2.2. Applications of Machine Learning

As discussed in section 1, Machine Learning finds it application in a variety of fields. Machine Learning is widely used in Natural Language Computing. Khan and Khan (2016) has done a comprehensive work on using Machine Learning to learn the semantics of natural language. The algorithms used in Machine Learning understand and process numerical data. In their research work,

they have addressed a crucial demand of understanding the context of text data. Use of SEBLA based NLU for semantic computation has been explored in their work.

Predictive Analytics is another field of Machine Learning which has gained immense popularity. Radhika R Halde (2016), has done a thorough study of predictive analytics and algorithms that are used in it. The main focus of Halde (2016) is using predictive analytics in Education. The author has used SVM, Neural Networks, Logistic Regression, Linear Regression, Decision trees and Naïve Bayes classifier to build the prediction model and concluded that SVM and Neural Networks predicted the numeric data student more accurately. Whereas, Decision trees are more accurate when it comes to classification.

Having talked about using Machine Learning in understanding natural language. Another interesting application of Machine Learning is to make the machines recognize and analyse facial expression. Nugrahaeni and Mutijarsa (2016), discusses about using Machine Learning for recognizing facial expression and compares the performance KNN, SVM and Random Forest algorithms for classifying facial expression. The author has used facial measures as an input to the classification model. The model will classify facial expression to determine the following: happy, sad, angry, disgust, fear, neutral and surprise.

Another important application of machine learning is malware detection and fault Identification. Mr. Matthew Leeds (2016), has done a significant work in using machine learning for malware detection in Android phones. He has used classification algorithms to detect malwares in Android phones. In Nagaraja and Sadashivappa (2016), Mr. Nagaraaja and Sadashivappa have done a survey on fault diagnosis using Machine Learning algorithms. They have explored the use of Artificial Neural Networks, Fuzzy logic and SVM for fault diagnosis.

2.3. Algorithms Used in Machine Learning

In section 2.2, we have discussed about applications of Machine Learning. It is necessary to know the algorithms that are used in those applications. From the previous section it is clear that for a classification problem, Decision trees are accurate for larger datasets. In this section we have done a study on Decision trees and Rule Induction.

In Vlahovic (2016), the author has given a brief survey of decision trees and has proposed an evaluation framework for decision trees. Decision tree belongs to class of supervised learning algorithms. They have a variety of applications ranging from managerial sciences to knowledge engineering. Decision trees are classified as follows:

- Decision Analysis trees;
- Knowledge representation trees;
- Classification and regression trees; and
- Decision forests.

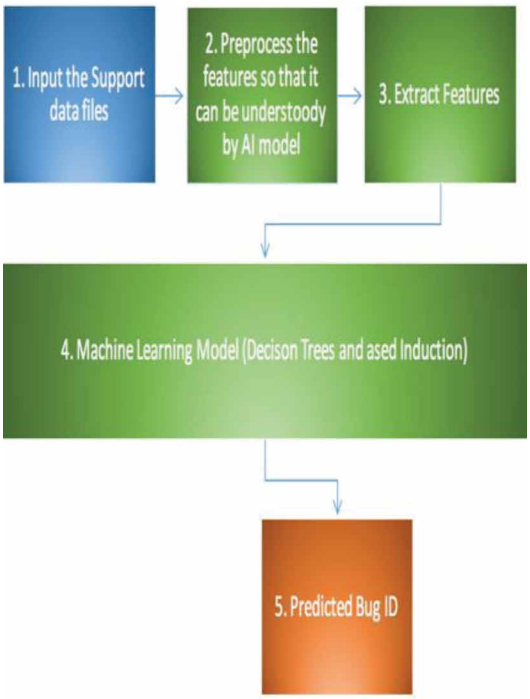
Decision trees were initially used to represent rules that form knowledge/expert systems. Algorithms that are used in decision trees are forward chaining and backward chaining.

The main focus of our work would be to find rules to classify the bugs that might occur in a device. Rule based Induction can be of use here. In Das, Acharjya, and Patra (2014), the authors have proposed a framework for producing decision rules using rule Induction. Rule Induction is a methodology of deriving rules from statistical information based on probabilities. Examples of such rule induction are CN2 (Clark & Niblett, 1989), RIPPER (Cohen, 1995) and C4.5 (Quinlan, 1993).

3. PROPOSED METHODOLOGY

In this section, we discuss about how we would analyse the support data files. Figure 1. shows the module description of the tool. Figure 2. describes the algorithm approached.

Figure 1. System architecture



3.1. Module Description

3.1.1. Input

Network support data files contain information about the networking devices that have been deployed at customer sites. These files serve as a means by which one could analyse the running configurations of the networking devices and trouble shoot if a problem arises. These files are the input to our system.

3.1.2. Pre-Processing

Pre-processing is an important task that has to be carried out before Analysis. Here, we have cleansed the data to eliminated null values and extreme values. Outlier analysis was done and subject expertise was applied to decide what should be included in the dataset and what should be ignored.

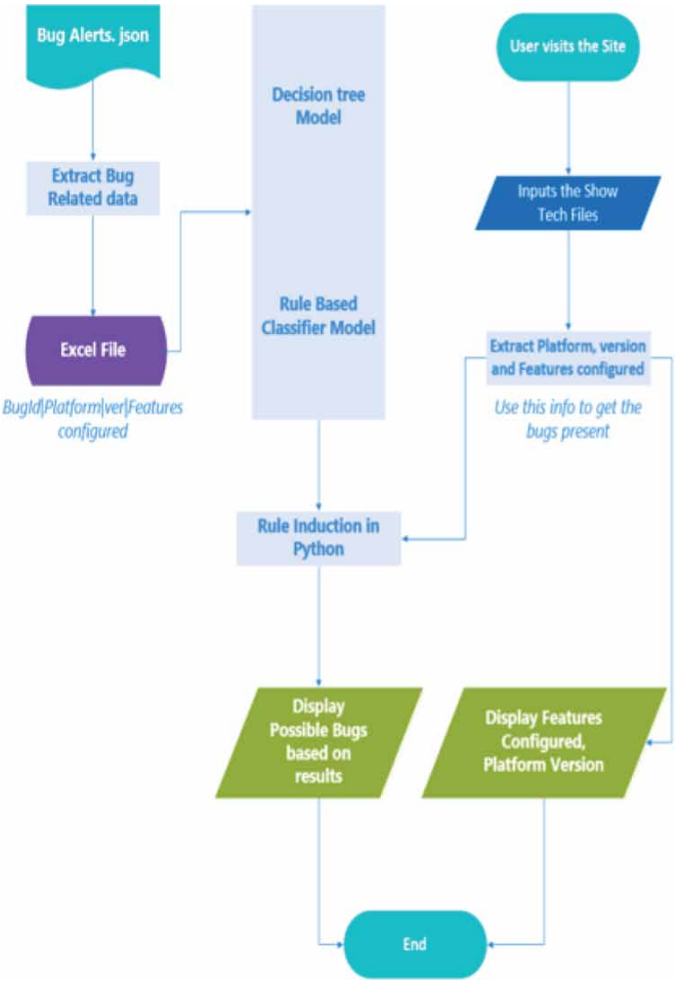
3.1.3 Extract Features

In this step, subset of the features that are suitable for building the learning model are extracted from the files that are inputted in step1. In this case, Device configuration (platform and software version running on a device) and software features that are enabled or configured on a device are extracted.

3.1.4. ID3 Model and Rule Based Induction Model

The Machine Learning model that is built here is a supervised learning model. A classified dataset of bugs based on platform and software version coupled with specifics of the bug is the excellent source for identifying a bug. One way to identify the nature of a bug is to start by knowing the cause for a bug. The attribute selection step depends on strong domain knowledge and use case. A dataset containing Bug Id, Hardware Platform, software version and Features causing the bug is used to build ID3 model and Rule Based Induction model. Bug Id is the label or class that will be predicted

Figure 2. Flow chart



at the end of analysis. Software version, Hardware platform and features causing the bug are the classifying attributes in our case.

Rules are constructed based on the results of Rule Based Induction and ID3 Decision trees. Features extracted in step 3 is given as an input to the rules and based on the input we will be directed through the decision tree to predict the bug ID.

3.1.5. Output

Output of this process will be predicted bug Id. This prediction is based on a device's configuration.

3.2. Software Requirements

The system is built using python 2.7. The language has many programming constructs that help in building both large scale and small-scale software. The language is designed to support many programming styles e.g. Object oriented, functional programming and imperative programming. It has a broad range of standard library which can be used for a variety of purpose. Scikit library is used for developing the Machine Learning model.

4. RESULTS AND DISCUSSIONS

The experimental setup for comparing ID3 and rule-based induction algorithms was mimicked using Rapid Miner Studio 7.4. The experiment was carried out in three segments:

1. A training dataset of size 100 was used which contained 5 class variables. The results indicate that Rule Based Induction was approximately 2% more accurate than ID3 algorithm. This is shown in Figure 3 and Figure 4 as a confusion matrix;
2. A training dataset 0f size 200 with 5 class variables showed that ID3 algorithm is approximately 14% accurate than Rule based Induction. The results are shown in Figure 4 and Figure 5 as a confusion matrix;
3. A training dataset of size 347 with 5 class variables showed that ID3 algorithm outperforms Rule Based Induction by 5%. This is shown in Figure 6 and Figure 7 as a confusion matrix.

Figure 3. Confusion matrix for rule-based induction (data size:100) accuracy: 33.33%

accuracy: 33.33%

	true 87719	true 102923	true 109929	true 112650	true 117804	class precision
pred. 87719	3	0	0	0	0	100.00%
pred. 102923	4	6	4	3	12	20.69%
pred. 109929	0	0	5	3	4	41.67%
pred. 112650	0	0	0	0	0	0.00%
pred. 117804	0	0	0	0	1	100.00%
class recall	42.86%	100.00%	55.56%	0.00%	5.88%	

Figure 4. Confusion matrix for ID3 algorithm (data size:100) accuracy: 31.11%

accuracy: 31.11%

	true 87719	true 102923	true 109929	true 112650	true 117804	class precision
pred. 87719	7	0	0	0	8	46.67%
pred. 102923	0	6	9	6	5	23.08%
pred. 109929	0	0	0	0	3	0.00%
pred. 112650	0	0	0	0	0	0.00%
pred. 117804	0	0	0	0	1	100.00%
class recall	100.00%	100.00%	0.00%	0.00%	5.88%	

Figure 5. Confusion matrix for rule-based induction (data size:200) accuracy: 48.89%

accuracy: 48.89%

	true 87719	true 102923	true 109929	true 112650	true 117804	class precision
pred. 87719	0	0	0	0	0	0.00%
pred. 102923	7	6	9	6	1	20.69%
pred. 109929	0	0	0	0	0	0.00%
pred. 112650	0	0	0	0	0	0.00%
pred. 117804	0	0	0	0	16	100.00%
class recall	0.00%	100.00%	0.00%	0.00%	94.12%	

Figure 6. Confusion matrix for ID3 algorithm (data size:200) accuracy: 62.22%

accuracy: 62.22%

	true 87719	true 102923	true 109929	true 112650	true 117804	class precision
pred. 87719	7	0	0	0	0	100.00%
pred. 102923	0	6	6	0	8	30.00%
pred. 109929	0	0	0	0	0	0.00%
pred. 112650	0	0	3	6	0	66.67%
pred. 117804	0	0	0	0	9	100.00%
class recall	100.00%	100.00%	0.00%	100.00%	52.94%	

Figure 7. Confusion matrix for rule-based induction (data size:347) accuracy: 75.56%

accuracy: 75.56%

	true 87719	true 102923	true 109929	true 112650	true 117804	class precision
pred. 87719	7	0	0	0	0	100.00%
pred. 102923	0	6	9	0	0	40.00%
pred. 109929	0	0	0	0	0	0.00%
pred. 112650	0	0	0	4	0	100.00%
pred. 117804	0	0	0	2	17	89.47%
class recall	100.00%	100.00%	0.00%	66.67%	100.00%	

Figure 8. Confusion matrix for ID3 algorithm (data size:347) accuracy: 80.00%

accuracy: 80.00%

	true 87719	true 102923	true 109929	true 112650	true 117804	class precision
pred. 87719	7	0	0	0	0	100.00%
pred. 102923	0	6	9	0	0	40.00%
pred. 109929	0	0	0	0	0	0.00%
pred. 112650	0	0	0	6	0	100.00%
pred. 117804	0	0	0	0	17	100.00%
class recall	100.00%	100.00%	0.00%	100.00%	100.00%	

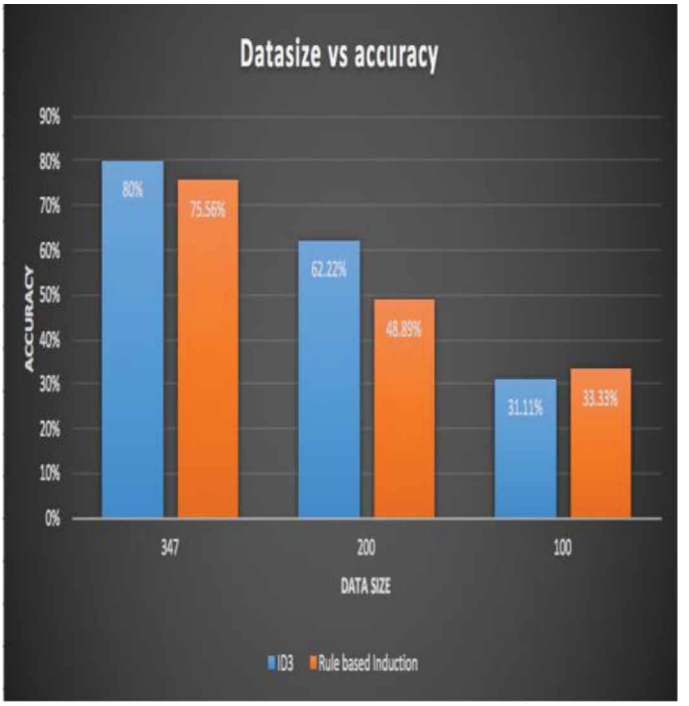
In this Confusion Matrices given below, the column headers are true values for the class variable and the row headers are the predicted class variables. For e.g. let us consider class variable 87719. From the matrix, it is clear that it has occurred three times and in all the trials it is predicted correctly. The class variable 102923 is predicted correctly 6 times but predicted as class 87719, 109929, 112650 and 117804 for four, four, three, twelve times.

The accuracy comparison graph for ID3 and Rule Based Induction is shown in Figure 9.

5. CONCLUSION AND FUTURE WORK

A prediction model for identifying faults in devices deployed at remote sites is built. The accuracy of the model is largely affected by the feature selection and the data size. ID3 model is proved to be

Figure 9. Comparison of accuracy between ID3 algorithm and rule-based induction



more accurate for larger datasets as the classification variables are selected based on the Information entropy and Gini index whereas in Rule Based Induction vague probabilities are used to define rules. As a future scope for this paper, use of Support Vector Machines and Artificial Neural Networks for Operational/Machine Produced data will be explored.

REFERENCES

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, *3*(4), 261–283. doi:10.1007/BF00116835

Cohen, W. (1995). Fast effective rule induction. In *Proceedings of the twelfth international conference on machine learning* (pp. 115 -123).

Das, T. K., Acharjya, D. P., & Patra, M. R. (2014). Business Intelligence from online product review-a rough set based rule induction approach. In Contemporary Computing and Informatics. doi:10.1109/IC3I.2014.7019662

Kamola, M. (2015). Analytics of industrial operational data inspired by natural language processing. In *IEEE International Congress on Big Data* (pp. 681-684). doi:10.1109/BigDataCongress.2015.108

Khan, E. (2016). Machine Learning Algorithms for Natural Language Semantics and Cognitive Computing. In *International Conference of Computational Science and Computational Intelligence* (pp. 1146-1151).

Leeds, M. (2016). Preliminary Results of Applying Machine Learning Algorithms to Android Malware Detection. In *International Conference of Computational Science and Computational Intelligence* (pp. 1070-1073). doi:10.1109/CSCI.2016.0204

Nagaraja, P., & Sadashivappa, G. (2016). Fault Diagnosis of Circuits Using Statistical Parameters and Implementation using Classifiers- A Survey. In *International Conference on Communication and Signal Processing* (pp. 2162-2166).

Nugrahaeni, R. A., & Mutijarsa, K. (2016, August). Comparative analysis of machine learning KNN, SVM, and random forests algorithm for facial expression classification. In *International Seminar on Application for Technology of Information and Communication (ISemantic)* (pp. 163-168). IEEE.

Quinlan, J. R. (1993). C4.5: programs for machine learning.

Radhika, R. (2016). Application of Machine Learning Algorithms for Betterment in Education System. In *International Conference on Automatic Control and Dynamic Optimization Techniques* (pp. 1110-1114).

Surya, K., Nithin, R., Prasanna, S., & Venkatesan, R. (2016). A Comprehensive Study on Machine Learning Concepts for Text Mining. In *International Conference on Circuit, Power and Computing Technologies*. doi:10.1109/ICCPCT.2016.7530259

Vlahovic, N. (2016, May). An evaluation framework and a brief survey of decision tree tools. In 2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 1299-1304). IEEE.

Priyan M.K. is currently pursuing PhD in VIT University Vellore. He has published 10 papers in reputed journals.