The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

Predicting Trouble Ticket Resolution

Kenneth R. Sample, Alan C. Lin, Brett J. Borghetti, Gilbert L. Peterson
Air Force Institute of Technology, 2950 Hobson Way, Wright-Patterson AFB, OH 45433
Email: kenneth.sample@afit.edu, alan.lin@afit.edu, brett.borghetti@afit.edu, gilbert.peterson@afit.edu

Abstract

Many organizations with an in-house information technology department rely on a trouble ticket system to track network issues. The goal of an effective trouble ticket system is to prioritize limited support personnel, responsively address each issue, and maintain user satisfaction. This paper presents a machine learning system that predicts ticket resolution time to provide users with an expected resolution time for their issue upon ticket submission. Classification and regression models were developed using boosted regression trees and artificial neural networks (ANNs). Evaluating on 12,303 trouble tickets, the classification model accuracy from the boosted regression tree was 74.5%. As a regression problem, the ANN model achieved the best result, with a mean absolute error (MAE) of 24.8 hours.

1.0 Introduction

Most enterprise networks use trouble ticket systems to resolve network outages. A trouble ticket is an e-record that documents outage data, actions taken, and workflow. During outages, users are unable to use their equipment and often are unaware of how long resolution will take. This uncertainty leads to wasting resources on work-arounds for short outages or insufficient planning for long outages. Predicting resolution time at the time of trouble ticket submission enables better mitigation decisions.

We explore both classification and regression models to predict trouble ticket resolution time. The prediction uses standardized fields within trouble ticketing systems. Of the algorithms tested, the boosted regression tree model obtained the highest classification accuracy of 74.5%. A feed-forward ANN had the best regression performance with a mean absolute error (MAE) of 24.8 hours.

2.0 Related Work

In the early years of enterprise-level outage management, most automated attempts at diagnosing tickets fell into the category of rule-based reasoning, or expert systems (Lewis and Dreo, 1993). One of the difficulties with analyzing trouble tickets is that most of the relevant information is contained in free-text fields. This can be resolved by 1) using only the portions of the ticket with categorical or numeric fields, or 2) using natural language processing (NLP) techniques to parse free text.

Temprado, et al. (2008) performed classification on trouble ticket data to predict whether a technician was needed onsite and whether its severity would be escalated within its lifetime. In addition to set form fields, they used techniques such as stemmer algorithms, entity-relationship models, frequency recount, and stop lists to extract information from free-form text. The two-class classification predicted whether on-site technicians were needed with 94% accuracy and found that Bayesian Networks, Naïve Bayes, C4.5 Decision Trees, and Decision Tables all produce similar accuracy results, while Decision Stumps and Hyper Pipes were less accurate.

Symonenko, et al. (2006) analyzed tickets manually, with *n*-gram analysis and contextual mining, to identify characteristics of the unique sublanguage related to trouble tickets. They were able to categorize the parts of each ticket (e.g. the complaint, the job referral) with a 1.4% error rate. Medem, et al. (2009) created Trouble Miner, which applied clustering techniques to classify types of tickets to aid troubleshooting. Trouble Miner was able to draw several conclusions, including that over half of all tickets are maintenance-related and that most of these maintenance tickets concern cables and routers.

Potharanju and Nitarotaru (2013)'s NetSieve used NLP, ontology modeling, and knowledge representation techniques to extract data categorizations of problems, troubleshooting activities, and resolution actions from free text. Overall, NetSieve was able to achieve between 89% and 100% accuracy in categorization on the test dataset.

3.0 Data Preparation

The data used in this study consisted of 12,303 trouble tickets; each ticket has 64 distinct fields. Data preparation removed fields that were free text or sparsely populated (< 10% contained data). Table 1 shows the retained fields. Most of the fields are categorical, with only a few numerical fields. Dummy variables were generated for categorical fields with more than two possible values, with one variable for each possibility which contains a 1 if that field contained that value, and a 0 otherwise. In total, 88 features were used in this analysis. Table 2 presents 4 representative tickets. Although all categories could have predictive value, the categorization tiers are most descriptive about the work conducted for the resolution.

Table 1: List of Data Features with Example Data

Field Name	Example Data	Unique Values
Assnd_Spt_Org	Infrastructure Support, Client Service Center	28
Cat_Tier_1	Repair/Restore, Request, Move/Change	8
Cat_Tier_2	Workstation, Network/Infrastructure, Messaging	13
Cat_Tier_3	Connectivity, Hardware/Appliance, User Account	29
Impact_Number	1,2,3,4,5	3
Reported_Source	Walk In, Email, Direct Input	9
Service_Type	Service Restoration, Infrastructure Event	4
Urgency_Number	1,2,3,4,5	4
Res_Time	326.76,299.62	5790
Res_Time_Cat	0,1,2	3

Table 2: Example Ticket Data

Assnd_Spt_Org	Cat_Tier_1	Cat_Tier_2	Cat_Tier_3	Impact_No	Reported_Src	Svc_Type	Urgency_No	Res_Time (hrs)	Res_Time_Cat
Infrastructure Support	Repair/Restore	Workstation	Connectivity	4	Systems Mgmt	User Svc Restoration	4	358	2
Client Service Center	Create/Add	Share Drive/SAN	Share Drive Access	4	Systems Mgmt	User Svc Request	4	24	0
Information Assurance	Provision/Enable	Share Drive/SAN	Share Drive Access	4	[Blank]	User Svc	4	176	2
Operations	Deprovision/Disable	Account Mgmt	User Account	4	Walk In	User Svc Request	4	181	2

A histogram of the resolution time, shown in Figure 1, identified resolution-time bins for classification. Notably, there are two large spikes; tickets tend to be resolved within 5 hours or between 24 and 120 hours. However, a small percentage of tickets had resolution times greater than 120 hours. This observation suggested partitioning tickets into three resolution-time categories: less than 24 hours, between 24-120 hours, and over 120 hours. Successfully classifying tickets into these categories provides the user with a sufficient granularity about their ticket resolution time, since it matches up closely with human work timelines (same day, work week, next week).

4.0 Methodology

This work evaluated several classification and regression models to predict trouble ticket resolution time. Although the use of artificial neural networks has not been conducted in this domain, its effectiveness in other domains prompted its exploration.

4.1 Classification

Several machine learning techniques were used to solve the 3-class problem: logistic regression, linear discriminant analysis (LDA) (James, et al., 2013), boosted regression trees (Elith, et al., 2008), and ANNs (Goodfellow, Bengio, and Courville, 2016). Five-fold cross-validation was conducted for all methods, both for testing and selecting hyperparameters. For logistic regression and linear discriminate analysis (LDA), no hyperparameter optimization was conducted. For regression trees, optimization was conducted to find the best maximum depth and learning rate using the combinations of the following values: Max Depth = [1,2,3,4,5], Learning Rate = [0.01,0.05,0.1,0.2].

Figure 2 shows the core architecture of all ANNs used. Since the data has no particular topological structure or sequencing and truth values are known for outputs, a fully connected feed-forward network was used (Goodfellow, Bengio, and Courville, 2016). A rectified linear unit (ReLU) activation function was used in the hidden layers, which fed into a 3-node output layer with a Softmax activation function. The Adam optimization function was always used with a cyclic learning rate—although the base learning rate was a tested hyperparameter, the max learning rate was always equal to five times the base. Dropout was conducted between hidden layers, using the dropout rates indicated in Table 3.

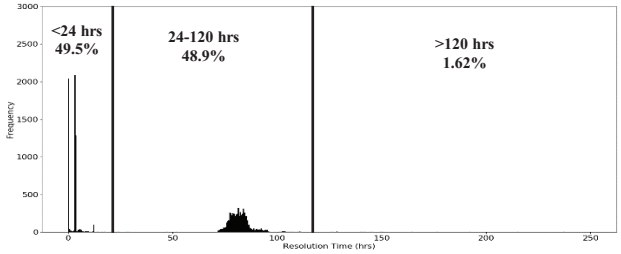

Figure 1: Histogram of Data Resolution Time Frequency with 3-Classification Bins and Percentage of Tickets in Each Bin

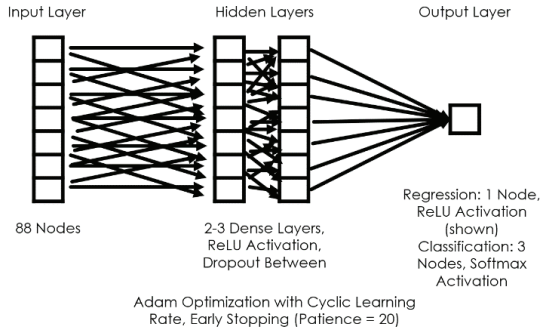

Figure 2: Architecture Diagram for Regression and Classification ANNs.

In addition to the ANN characteristics above, there were several other hyperparameters that could take on multiple values, which led to the testing of many combinations of hyperparameter values: hidden layer width and depth, learning rate, and dropout. Table 3 shows a list of these hyperparameters. In the validation phase, ANNs for regression and classification were created for all possible hyperparameter combinations, and a 5-fold cross-validation was used to determine the best performing ANNs. The best performing ANN for each fold was then given test data.

Table 3: ANN hyperparameter values tested

Hyperparameters	Values
Base Learning Rate	0.001, 0.0001
Hidden Layers	2,3
Nodes per Hidden Layer	8,10,12
Dropout Level	0.0,0.1,0.2

4.2 Regression

Linear regression, lasso regression (Tibshirani, 2007), boosted regression trees, and ANNs were used. Parameters for each technique were tuned to produce the lowest MAE value for comparison to capture the totality of prediction error for each ticket. Again, 5-fold cross-validation was used on all models. The lasso weight penalty hyperparameter (alpha) was selected from the following range: $\alpha = [0.00001,0.0001,0.001,0.01,0.1,1,10]$. Regression tree boosting was conducted across all combinations of the following values of max depth and learning rate: Max Depth = [1,2,3,4,5], Learning Rate = [0.01,0.05,0.1,0.2]. The process used for regression ANNs was identical to that for classification, with the exception of the output layer. As seen in Figure 2, the core ANN architecture, the output layer consists of one node with a ReLU activation function. Correspondingly, the hyperparameter values tested for classification can be found in Table 3.

5.0 Results

The primary measure of models will be test set accuracy for the 3-classification ANN and MAE for the regression ANN. However, with uneven distribution of data, a raw accuracy score is misleading. Cohen's Kappa statistic will be calculated to better contextualize the performance of the classification models given the data distribution.

5.1 Classification

Table 4 presents the 3-class classification results for the four algorithms. Logistic regression and LDA achieved test set accuracies of 72.7% and 71.3%, respectively. The boosted regression tree classified the data most effectively, with an accuracy of 74.5%. The ANN also performed well, yielding an average accuracy of 73.9%.

Table 4: 3-Class Classification Accuracy Results for All Techniques

Technique	Accuracy
Logistic Regression	72.7%
LDA	71.3%
Boosted Regression Trees	**74.5%**
ANN	73.9%

Cohen's kappa analysis was conducted on both the classification ANN and the boosted regression tree model. The expected accuracy was 48.9% for the ANN and 48.7% for the regression tree. With observed accuracies of 73.9% and 74.5%, respectively, this results in kappa values of 0.489 and 0.503 for the ANN and regression tree. Although there are no standard interpretations of kappa values, values between 0.4 and 0.6 indicate moderate agreement (Landis and Koch, 1977) and that values between 0.4 and 0.75 represent "fair to good agreement beyond chance" (Fleiss, 2003).

Table 5 presents a confusion matrix of the ANN to better understand the sort of classification these models are performing, and the errors that they made. The confusion matrix highlights the model's tendency to predict many tickets between 24 and 120 hours, which makes it identify these tickets with an accuracy of 85.7%. However, this also leads to poor accuracy on shorter and longer tickets; tickets which were resolved in less than 24 hours were only correctly identified 64.2% of the time, and tickets longer than 120 hours were only identified 11.1% of the time.

Table 5: Confusion matrix of the classification ANN's performance on test data

		Predicted Label (hours)		
		<24	24-120	>120
True Label (hours)	<24	**0.642**	0.353	0.005
	24-120	0.139	**0.857**	0.003
	>120	0.567	0.322	**0.111**

5.2 Regression

The test MAE for all three models are shown in Table 6. The linear regression model performed poorly, with a MAE of 1.30×10^{12} hours, indicating that a linear model is insufficient for this data, possible due to the prevalence of dummy variables and irrelevant features. When optimizing lasso regression, an α of 0.00001 was found to be the best fit on the validation data. This value corresponds to a severe reduction of the features, which supports the poor linear regression results. The lasso model achieved a MAE of 38.97 hours on the test data, falling short of the 24-hour objective, but verges on usable, as an average of 39 hours

away from the actual resolution time between one and two working days.

The boosted regression tree model was found to be optimal at a max depth of 4 and a learning rate of 0.05. This model achieves a MAE of 37.86 hours on the test data, making it just better than the lasso model, but still not within the objective range of 24 hours. Finally, the ANN approach proved to be superior, producing a MAE of 24.78 hours on test data, almost achieving the goal of 24 hours. Although some tickets took over 300 hours to be resolved, the ANN never predicted a resolution time of over 150 hours—performance is poor on very high values, although they are uncommon. Further, when the actual resolution time is close to 0 there are many misclassifications where the model predicts between 60 and 100 hours. However, a large proportion of points occur between 24 and 120 hours, and most of them are classified somewhere in that range.

Table 6: Regression MAE Results

Technique	MAE (hrs)
Linear Regression	1.30×10^{12}
Lasso Regression	38.97
Boosted Regression Trees	37.86
ANN	**24.78**

5.3 Feature Importance

Regression trees and lasso regression models both provide a list of the importance of all feature values. Table 7 lists the top 5 most important features for each model. None of the top 10 features from each technique overlap. The lack of feature correspondence across models indicates that very few individual features have a high degree of correlation with the underlying phenomenon.

Table 7: Most Important Features out of 88 total features, Sorted by Average Importance Rank

Feature	Lasso Importance Rank	Regression Tree Importance Rank	Average Importance Rank
Assigned_Support_Org: [redacted]	13	3	8
Assigned_Support_Org: [redacted]	12	5	8.5
Assigned_Support_Org: Network Operations	19	1	10
Reported_Source: System Management	8	14	11
Reported_Source: Direct Input	2	25	13.5
Assigned_Support_Org: [redacted]	21	9	15
Assigned_Support_Org: Cable/Antenna Systems	26	4	15
Assigned_Support_Org: [redacted]	18	13	15.5
Cat_Tier_3: Computer Account	29	2	15.5
Reported_Source: Phone	6	31	18.5

6.0 Conclusions and Future Work

Accurate prediction of trouble ticket resolution time enables organizations to make better decisions when an issue occurs. We explored classification and regression models to predict resolution times. For classification, the best models achieved accuracy up to 74.5%. In the case of regression, the ANN achieved the lowest MAE of 24.8 hours. This improvement suggests the possibility of deploying a model to the network that proactively estimates user outage duration.

This work shows that for some trouble ticket systems analysis can be conducted with only fixed fields, without requiring free text. Future work includes gathering data from other networks to examine generalizability and find utility in the tickets' data fields to integrate free text analysis to improve upon current results or in instances where standardized fields are unavailable.

7.0 References

Elith, J., Leathwick, J. R. and Hastie, T. 2008. A Working Guide to Boosted Regression Trees. *Journal of Animal Ecology*, 77(4), 802–813.

Fleiss, J. L., Levin, B., and Paik, M. C. 2003. *Statistical methods for rates and proportions*. 598-626. John Wiley & Sons, Inc.

Goodfellow, I., Bengio, Y., Courville, A. 2016. *Deep Learning*. MIT Press.

James, G., Witten, D., Hastie, T. and Tibshirani, R. 2013. *An Introduction to Statistical Learning with Applications in R*, *Current medicinal chemistry*. New York Heidelberg Dordrecht London: Springer.

Landis, J. R. and Koch, G. G. 1977. The Measurement of Observer Agreement for Categorical Data, *Biometrics*, 33(1), 159.

Lewis, L. and Dreo, G. 1993. Extending Trouble Ticket Systems to Fault Diagnostics, *IEEE Network*, 7(6), 44–51.

Lu, C. N., Tsay, M. T., Hwang, Y. J. and Lin, Y. C. 1994. Artificial Neural Network Based Trouble Call Analysis', *IEEE Transactions on Power Delivery*, 9(3), 1663–1668.

Medem, A., Akodjenou, M. I. and Teixeira, R. 2009. Trouble Miner: Mining Network Trouble Tickets, *2009 IFIP/IEEE International Symposium on Integrated Network Management-Workshops,* 113–119.

Potharaju, R. and Nita-rotaru, C. 2013. Juggling the Jigsaw: Towards Automated Problem Inference from Network Trouble Tickets, *10th USENIX Symposium on Networked Systems Deisgn and Implementation*. 127–141.

Symonenko, S., Rowe, S. and Liddy, E. D. 2006. Illuminating Trouble Tickets with Sublanguage Theory, *Proceedings of the Human Language Technology Conference of the NAACL*, (June), 169–172.

Temprado, Y., Molinero, F. J., García, C. and Gómez, J. 2008. Knowledge Discovery from Trouble Ticketing Reports in a Large Telecommunication Company, *2008 International Conference on Computational Intelligence for Modelling Control and Automation, CIMCA 2008*, 37–42.

Tibshirani, R. 2007. Regression Shrinkage and Selection via the Lasso, *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 58(1), 267–288.