Identifying Missed Monitoring Alerts based on Unstructured Incident Tickets

Liang Tang and Tao Li
School of Computer Science
Florida International University
Miami, FL, USA
Email: {ltang002, taoli}@cs.fiu.edu

Larisa Shwartz
Operational Innovations
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
Email: lshwart@us.ibm.com

Genady Ya. Grabarnik
Dept. Math & Computer Science
St. John's University
Queens, NY, USA
Email: grabarng@stjohns.edu

***Abstract*—Automatic system monitoring is an efficient and reliable mean for problem detection in enterprise IT infrastructures. The performance of monitoring systems depends on their configurations specified by the system administrators. In dynamic and large IT environments, the IT infrastructures are frequently changed to meet various business requirements, so the configurations may not be always consistent with the updated status. Misconfigurations can lead to false positive (false alarms) and false negative (missing alerts) for the system administrators. The false negatives can cause serious system faults. This paper presents an automatic approach for discovering the false negatives from incident tickets that are created by humans. The discovered results help the system administrators correct the misconfigurations and minimize the false negatives in future. This approach applies a text classification model for analyzing the descriptions of incident tickets and identifying the corresponding system issues. The domain knowledge for describing those issues can be incorporated to assist with this model. Experiments are conducted on real system incident tickets from a large enterprise IT infrastructure. The experimental results demonstrate the effectiveness of the proposed approach.**

I. Introduction

Modern IT infrastructures are maintained by automating routine maintenance procedures, including problem detection, determination and resolution. System monitoring provides an effective and reliable means for problem detection. Coupled with automated ticket creation, it ensures that a degradation of the vital signs, defined by acceptable thresholds or monitoring conditions, is flagged as a problem candidate and sent to supporting personnel as an incident ticket. Defining monitoring conditions (situations) requires the knowledge of a particular system and its relationships with other hardware and software systems. Continuous updating of IT infrastructures also leads to a number of system alerts that are not captured by system monitoring (false negative). The false negatives eventually cause system faults, such as system crashes and data loss, which are extremely harmful to enterprise users. When missed or misconfigured monitoring alerts are discovered, the system administrators would identify and correct monitoring settings.

In system and networking management, many previous studies focus on new problem detection methods for minimizing the false negatives [24] [16] [12] [18] [5]. In reality, it is not easy to change the methods in existing monitoring software products, such as IBM Tivoli monitoring [1]. The performance of problem detection also depends on the configurations for those methods, because the suitable configurations require the domain knowledge of particular systems. In this paper, we present an approach for improving the configurations of existing monitoring systems to minimize the false negatives. This approach is based on a combined analysis of system incident tickets that were created manually. It utilizes a text classification model to automatically discover the false negatives in manual tickets, where the selective labeling is able to incorporate the domain knowledge in system management and make the training process become more efficient. This approach provides optimization and generality: it can be trained against various time windows and focused training data. We conducted experiments to understand the coverage and precision of the proposed method, as well as training efficiency in comparison to known methods.

II. False Negative Ticket Discovery

In this section, we present our method for identifying the missed monitoring alerts based on the textual incident tickets.

A. Problem Formulation

In IT Service management, manual incident tickets are the short textual messages which describe system incidents and are written by the system administrators, helpdesk or end-users. Those system incidents can be any type of system issues, such as the system alerts and customer requests. The system alerts are about high utilization of the disk space, crashes of a database and so on. The customer requests are about resetting database passwords, installing a new web server and so on. Therefore, discovering the system alert tickets from a collection of textual tickets is a binary text classification problem. Given a incident ticket, our method classifies it into "1" or "0", where "1" indicates this ticket is a system alert situation and "0" indicates it is not.

There are two challenges for building the classification model. First, the manual ticket data is highly imbalanced. Most system alerts are captured by the automatic monitoring system. Very few of them are missed but recorded by the system administrators. At the same time, manual tickets are responsible for tracking customer requests and other issues. Hence, most of manual tickets are customer requests. Only very few are about system situation alerts. Second, labeled data is very limited. Most system administrators are only working on some types of incident tickets. Only a few senior experts can label all tickets. It is difficult to obtain a large amount of labeled tickets for training the classification model.

B. Selective Ticket Labeling

It is time-consuming for human experts to scan all manual tickets and label their classes. In our approach, we select only a small proportion of tickets for labeling. The selection is crucial to the highly imbalanced data. The situation tickets are very rare. If the selected ticket set contains none of them, the classification model cannot be trained well. Before we obtain the class labels, however, we do not know in advance if a ticket is related to the monitoring situations or not. We took the classic approach of utilizing the domain words in system management. There are some proper nouns or verbs that indicate the scope of the system issues. Table I lists examples of the domain words with their corresponding situations. The domain words can be easily obtained from the system administrators and related documents. The domain words can also be obtained from the taxonomy of the system management.

TABLE I: Domain Word Examples

Situation Issue	**Words**
DB2 tablespace Utilization	DB2, tablespace
File System Space Utilization	space,file
Disk Space Capacity	space,drive
Service Not Available	service,down

The ticket selection first computes the score of each ticket and ranks all the tickets based on the score. Then, it selects the top k tickets in the ranked list. The top k tickets are chosen as the training tickets, where k is the predefined number of training tickets in total. Given a ticket T, the score is computed as follows:

$$score(T) = \max\{|w(T) \cap M_1|, ..., |w(T) \cap M_l|\},$$

where $w(T)$ is the word set of ticket T, l is the number of predefined situations, M_i is the given domain word set for the i-th situation, $i = 1, ..., l$. Intuitively, the score is the largest number of the matching words between the ticket and domain words. The number of matching words is usually used to measure the degree of a ticket matching with a concept.

TABLE II: An Example of Situation Ticket

xxx needs to be cleaned up both C: and F: drives Getting an alert of low **disk space** on xxx. What should happen is that Fusion should be moved to the IBM controlled OS **drive**. But@ to free up enough space to not cause issues...

For instance, a situation ticket T about a low disk space alert is shown in Table II, where the names of the system administrator and servers are replaced by "xxx" due to the privacy issue. The bold words are the matched word with the domain words. Let M_i denote the domain words for the low disk space monitoring situation, $w(T) \cap M_i = \{\text{"}space\text{"}, \text{"}disk\text{"}, \text{"}disk\text{"}\}$, then $score(T) \geq 3$. In the evaluation section of this paper, we consider a baseline method that only uses the domain words to identify situation tickets by simple word-matching.

C. Classification Model Building

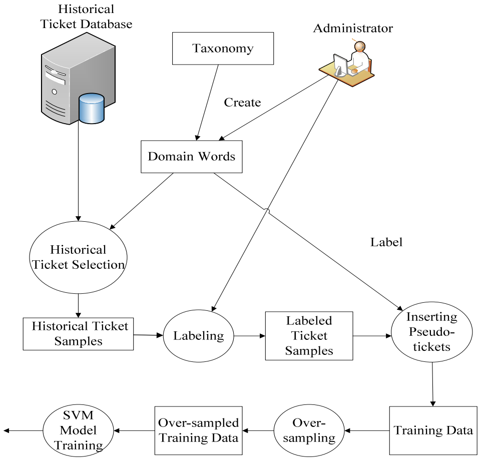


Fig. 1: Flow Chart of Classification Model Creation

The situation ticket is identified by applying a SVM classification model [20] on the ticket texts. We choose the SVM as the classification model because it is known to perform best in the text classification [11]. For the training process, we have two types of labeled data: 1) the selected training tickets, and 2) the domain words. Each domain word is treated as a *pseudo* ticket that only contains one word. All *pseudo* ticket are considered as situation tickets, so the label of each *pseudo* ticket is "1".

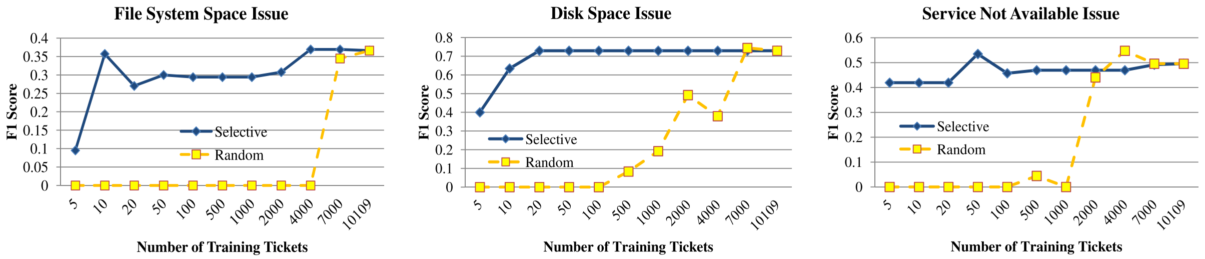
Although the selected training tickets have a better chance of containing the situation tickets, the set of the situation tickets is still the minority of all selected training tickets. The highly imbalanced training data affects the performance of the SVM classification model [7]. The over-sampling technique is applied before we do the SVM model training [7]. Figure 1 shows the flow chart for building the SVM classification model. The model training is an off-line process, so the running time is not a concern. The classification model is updated for one month or three months, since the configurations of production systems are known to change as often as every one or three months.

III. Evaluation

In this section, we present our empirical studies on the real system incident ticket data.

A. Experiment Data

The experimental data is collected from a large customer account in IBM IT service center. This account consists of over 1,000 monitored servers and network devices. The data set contains two months' tickets, for one month, the number of manual tickets is 9584, and for the other month, the number of manual tickets is 10109. We apply the first month's tickets as the training data, the second month's tickets are used as the testing data. The ground truth of those situation issues is labeled by humans.



(a) File system Space

(b) Disk space

(c) Service not available

Fig. 2: F1 Score

B. Baseline Methods

We compare our method with two baseline methods in terms of the accuracy and the amount of training tickets. Certainly, if the accuracy is higher and the training tickets are fewer, the method is better. The first baseline method is the word-match method based on the domain words to identify the situation tickets. To compare with this method, we would like to validate whether our method enhances the domain knowledge (domain words) or not. The second method is to randomly select a subset of tickets for labeling and training the SVM model. To compare with this method, we would like to show that our method based on the domain knowledge (domain words) is more efficient in the model training.

C. Effectiveness

The effectiveness is evaluated by the F1 score of the discovery of monitoring alerts that were missed, which are the the standard accuracy metrics and used in classification problems [19]. The incident tickets collected from one month are used as the testing data. We first test the accuracy of the word-match method. For each monitoring situation, we only provide 1 or 2 domain words. Table III shows the tested accuracy of the word-match method with the given domain words. This word-match method has a high recall but a very low precision.

TABLE III: Accuracy of the word-match method

Situation	Words	Precision	Recall	F1Score
File System Space	space,file	0.0341	0.8	0.0654
Disk Space	space,drive	0.1477	0.9565	0.2558
Service Not Available	service,down	0.1941	0.75	0.3084

Figure 2 shows the F1 scores for the three monitoring situations by using SVM models. Another set of incident tickets collected from different month is used as the training data. Our method is denoted as “Selective”, the second baseline method is denoted as “Random”. The domain words for “Selective” methods are the same as the words used by the word-match method and shown in Table III. As shown by Figure 2, the “Random” method can only achieve the same F1 score of ”Selective” when the number of training tickets is larger than 5000, while the ”Selective” only uses about 50 training tickets. In other words, only when the training data is very large, the “Random” method can achieve a good accuracy.

D. Case Study

The case study is used to demonstrate how our method enhances the domain knowledge from the experts for identifying the situation tickets. Table IV shows a set of discovered situation issues from the manual tickets. The first column is the situation name (standard name convention in IBM Tivoli Monitoring). The second column is the situation ticket. For privacy issues, the administrators’ names and the server names are replaced by “xxx”. To explain why our method is better than the simple word-match method, we list some words that have the largest weights in the SVM model for disk space situation (see Table V). “free”, “full”, “clear” are the common adjective and verb terms for describing the disk space issues, which are not in the domain words. “Missing”, “access”, “client” and “migration” are about other system issues and customer requests, so their weights are negative. Therefore, the SVM model is able to gather more useful knowledge from the training tickets.

TABLE V: Terms Weights for Disk Space Situation

Positive Term	Weight	Negative Term	Weight
space	8.9775	missing	-1.4643
free	3.6807	access	-1.1659
full	3.6643	client	-1.0950
clear	3.0330	migration	-1.0270
...	...	...	...

IV. Related Work

Each enterprise system producer has its own system management solution [2] [4] [3], which allows to monitor or detect faults, configure, account for, tune performance and secure (FCAPS) software systems, data centers, private and public clouds. In the centralized location events generated by monitoring are filtered, summarized, analyzed by enterprise consoles using either rule based (like Tivoli Enterprise Console [4]), or case based (codebook, [26]), or AI based [10], [15], [8], [9]. If necessary, a warning or error ticket is generated for the IT personnel to verify, and sometimes address, possible system or network issue.

A significant amount of scientific research was devoted to the automated or semi-automated generation and verification of the ticket generating configuration of enterprise consoles [23], [6], [17], [14], [13], [25]. This direction also includes work with false positives events [21] and the framework for the processing of false positives [22]. Much less was done in the

TABLE IV: Identified False Negatives

Situation	Ticket
dsp_3ntc_std	Please clear space from E drive xxxx-fa-ntfwwfdb Please clear space from E drive xxxx-fa-ntfwwfdb.it is having 2 MB free...
fss_rlzc_std	/opt file system is is almost full on us97udb010ampsb Hi Team@/opt file system is almost full. Please clear some space /home/dbasso¿df -h /optFilesystem...
svc_3ntc_std	RFS101681 E2 Frontier all RecAdmin services are down Frontier RecAdmin services are not running on the batch server Kindly logon to the server : xxx.xxx.155.183/xxx ...
dboffln_3oqc_std	DB2 is not connectable from xxxxx Hi Team@Can you please look into why we are unable to connect to Porfolio XRef DB.Server : xxxx12DB Instance : sec_mastId : ipxrbtchWhile...
dboffln_3oqc_std	Unable to login to DB server Hi Team@We had raised a request 131443 for access on the E1 and E2 serversE1 - Full access@ to read/write/execute programs Hostname Server xxxxx xxx.xxx.147.194

direction of this paper, or identification of the false negatives. Joint consideration of automatically generated alarms and open tickets differs from the majority of the research in the area and provides additional information to be used as the ground truth. By the definition of the false negatives the only indication of fault is a service ticket opened by the request of system user.

V. Conclusion

This paper investigates the methods for improving the configurations of the monitoring system to minimize the false negatives (missing alerts). A text classification based approach is proposed in this paper to discover the false negative alerts from manual tickets. The discovered results help the system administrators to find and correct the misconfigured monitoring conditions in the monitoring system. To efficiently build the classification model, this approach incorporates the domain knowledge in system management, which helps us to find effective training data from a large amount of historical tickets. We conduct experiments on system incident tickets from real and large IT environments. The experiment results demonstrated the benefits of this approach for improving the monitoring quality in real IT environments.

References

[1] IBM Tivoli Monitoring. http://www-01.ibm.com/software/tivoli/products/monitor/.

[2] OpenView, HP (retrieved on 04/22/2013). http://www8.hp.com/us/en/software/enterprise-software.html.

[3] System Center, Microsoft (retrieved on 04/22/2013). http://www.microsoft.com/en-us/server-cloud/system-center/default.aspx.

[4] Tivoli, IBM (retrieved on 04/22/2013). http://www-01.ibm.com/software/tivoli/.

[5] A. Bouillard, A. Junier, and B. Ronot. Hidden anomaly detection in telecommunication networks. In *Proceedings of CNSM*, pages 82–90, 2012.

[6] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection for discrete sequences: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 24(5):823–839, 2012.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[8] P. Desnoyers. *Distributed Data Collection: Archiving, Indexing, and Analysis*. ProQuest, 2008.

[9] C. Gupta. Event correlation for operations management of largescale it systems. In *Proceedings of the 9th international conference on Autonomic computing*, pages 91–96. ACM, 2012.

[10] D. W. Gurer, I. Khan, R. Ogier, and R. Keffer. An artificial intelligence approach to network fault management. *SRI International*, 86, 1996.

[11] T. Joachims. Text categorization with support vector machines: Learning with many relevant features, 1998.

[12] Y. Liao and V. R. Vemuri. Using text categorization techniques for intrusion detection. In *USENIX Security Symposium*, pages 51–59, 2002.

[13] S. Ma, J. L. Hellerstein, C.-s. Perng, and G. Grabarnik. Progressive and interactive analysis of event data using event miner. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 661–664. IEEE, 2002.

[14] P. Marcu, G. Grabarnik, L. Luan, D. Rosu, L. Shwartz, and C. Ward. Towards an optimized model of incident ticket correlation. In *Integrated Network Management, 2009. IM '09. IFIP/IEEE International Symposium on*, pages 569–576, 2009.

[15] K. F. Navarro, E. Lawrence, and J. Debenham. Intelligent network management for healthcare monitoring. In *Trends in Applied Intelligent Systems*, pages 72–81. Springer, 2010.

[16] A. J. Oliner, A. Aiken, and J. Stearley. Alert detection in system logs. In *Proceedings of IEEE ICDM*, pages 959–964, 2008.

[17] C.-S. Perng, D. Thoenen, G. Grabarnik, S. Ma, and J. Hellerstein. Data-driven validation, completion and construction of event relationship networks. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 729–734, New York, NY, USA, 2003. ACM.

[18] R. K. Sahoo, A. J. Oliner, I. Rish, M. Gupta, J. E. Moreira, S. Ma, R. Vilalta, and A. Sivasubramaniam. Critical event prediction for proactive management in large-scale computer clusters. In *Proceedings of ACM KDD*, pages 426–435, 2003.

[19] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1984.

[20] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2005.

[21] L. Tang, T. Li, F. Pinel, L. Shwartz, and G. Grabarnik. Optimizing system monitoring configurations for non-actionable alerts. In *Network Operations and Management Symposium (NOMS), 2012 IEEE*, pages 34–42. IEEE, 2012.

[22] L. Tang, T. Li, F. Pinel, L. Shwartz, and G. Grabarnik. Recommending resolutions for problems identied by monitoring. In *International Symposium on Integrated Network Management (IM), 2013 IEEE*, pages 1–8. IEEE, 2013.

[23] W. Xu, L. Huang, A. Fox, D. Patterson, and M. Jordan. Mining console logs for large-scale system problem detection. In *Workshop on Tackling Computer Problems with Machine Learning Techniques (SysML), San Diego, CA*, 2008.

[24] W. Xu, L. Huang, A. Fox, D. A. Patterson, and M. I. Jordan. Online system problem detection by mining patterns of console logs. In *Proceedings of IEEE ICDM*, pages 588–597, 2009.

[25] K. Yamanishi and Y. Maruyama. Dynamic syslog mining for network failure monitoring. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 499–508. ACM, 2005.

[26] S. A. Yemini, S. Kliger, E. Mozes, Y. Yemini, and D. Ohsie. High speed and robust event correlation. *IEEE communications Magazine*, 34(5):82–90, 1996.