

International Research Journal on Advanced Science Hub 2582-4376 Vol. 03, Issue 08 August www.rspsciencehub.com

Check for updates

http://dx.doi.org/10.47392/irjash.2021.231

# Workaround Prediction of Cloud Alarms using Machine Learning

Dhanya S Karanth<sup>1</sup>, H V Kumaraswamy<sup>2</sup>, Rajesh Kumar<sup>3</sup>

Abstract

 <sup>1</sup>PG - Digital Communication, Department of Electronics and Telecommunication Engineering, RV College of Engineering, Bangalore, Karnataka, India
 <sup>2</sup>Professor and Associate Dean, Department of Electronics and Telecommunication Engineering, RV College of Engineering, Bangalore, Karnataka, India
 <sup>3</sup>Technical Leader, CDSTC&SRN, Mobile Networks, Karnataka, Nokia, Bangalore, India

Email: kumaraswamyhv@rvce.edu.in

#### **Article History**

Received: 12 June 2021 Accepted: 17 July 2021

#### Keywords:

Cloud Alarms; Machine Learning; Prediction Cloud-based systems implies to applications, resources or services provided to users as per their requirement through the Internet using a cloud computing provider's server. These clouds triggers alarm events to indicate the health of system. Monitoring these alarms is essential for maintaining the health and continuous functioning of cloud. Because of humungous alarms triggered on daily basis, notifying critical alarms in time and taking required action is quite challenging task. In this paper machine learning model is implemented using decision tree classifier to analyse each alarm and predict if any action required for that alarm or not and also notify the concerned team via creating JIRA tickets.

# 1. Introduction

Cloud-based systems implies to applications, resources or services provided to users as per their requirement through the Internet using a cloud computing provider's server (Anand). Companies make use of cloud-based computing to enhance capacity, improve functionality or add additional services on demand without investing on expensive infrastructure or spend on training of existing support staff. customers storage or software offered via private or public cloud by the service provider. Industrial machines have various alarms are embedded in machine controllers. By employing sensors and machine states to notify to end-users or to keep machines in a specific mode. In particular, sensor data is compared with some predefined threshold values in machine controller and the alarms are triggered frequently (Agrawal et al.). The root causes of system misbehaviour can be detected by

analysing alarm logs and the problems can be fixed. Because of enormous amount of the system log, detecting critical alarms in time and tracing the key reason of system faults turned out be a complex challenge in enhancing the durability of telecommunication network systems and compromising the quality of customer service (Yuan et al.).

The usual way is to avoid system failures in a reactive way is when an internal fault is detected, a monitoring agent triggers a recovery procedure to reduce the problem and a human operator is alerted. But this method carried out after a fault has happened, which may need some extra time until it is notified. when the recovery procedure initiates, the fault may have caused some harm to the system. Alarm events are the indication of defect that occurred by malfunctioning of hardware or software or false operations or users (Yu). Alarm data contains information about fault diagnosis and recov-

ery. Thus, handling of alarm data has prominent impact on operation price and quality of services in telecommunication industry. Faults or unexpected events are unavoidable in critical and advanced systems (Bhowmik, Chandana, and Rudra).

Proactive failure detection is way to detect events prior so that the preventative or recovery measures can be planned to improve system availability (Adamu et al.). Machine learning algorithms applied in different areas of research and resulted in fine performance in learning and understanding the patterns (Wong and Yeh). In case of proactive failure detection, assumption is made prior to occurrence of failure, few parameters of the system can reveal signs of the approaching failure. When these data are collected and analysed by the algorithms, specific characteristics of the system in healthy and faulty states can be adjusted during the training stage and identified at runtime (Vrana and Korenek Sanzo, Avresky, and Pellegrini). Machine learning techniques are proved to be suitable to find out patterns from datasets available and to categorize class of a new sample of knowledge belongs. Nokia's UCIM (Unified Cloud Infrastructure management) tool is used for monitoring of telco clouds located in different locations. alarms triggered by this cloud can be viewed in UCIM tool. Analysing each alarm manually is time consuming and takes huge manual effort (Sun et al.). In this paper machine learning model is built using decision tree classifier to watch the cloud alarms in efficient way and notify the concerned team/person via incident creation

#### 2. Methodology



# FIGURE 1. Dataflow Diagram of Proposed Method

Figure 1. shows the Dataflow Diagram of Proposed Methodology followed, when a new alarm comes in UCIM tool, collect the alarm from the database and filter out the alarms for different category of alarms and prepare data set for training. analyse the data and carryout pre-processing like balancing the data, removing the duplicates etc. Once the data is ready model needs to be chosen. Three algorithms such as Decision tree classifier, random forest algorithm and naïve bayes classifier are chosen at this initial phase, whichever algorithm gives the highest accuracy will be finalized for the deployment. Model will pick all the active alarms in real time and analyse the root cause of the alarm to suggest the work around. An Internal JIRA ticket will be created with respective to the cloud located lab for the quick action.

For implementation, three separate modules are considered, Input module includes the data collection and pre-processing of data. Building a model module includes evaluation of algorithms, model selection and implementation of model. Notification module is responsible for JIRA creation.

#### 3. Implementation

Main technologies used in this implementation is Python, SQL, Scikit-learn, REST APIs.

Python is general purpose, versatile programming language. It has multiple libraries which can be used in building machine learning algorithm.

• Rest API Representational State Transfer application programming interface. When a client request is made via a REST API, it transfers a representation of the state of the resource to the requester or endpoint.

• Scikit-learn is a free software machine learning Library for the Python programming language. It features various classification and regression algorithms.

Cloud alarms are collected from UCIM tool via Rest APIs and stored in an external database. Preprocessing refers to the conversions applied on data prior to use of them the in algorithm. Pre-processing is method conversion of the raw data into polished or clean data set. When data is collected from different sources, the format may not be same or may not be in a desired format. It may also contain duplicate values and redundant values and some algorithm will not take null values and in case of classification algorithms training data needs to have balanced data for each category. Hence data should be processed to be unique, balanced and correct before using the data in machine learning methods for training and testing. Since response of Rest API is in JSON format, it is converted to CSV file using python component. The pre-processing includes removal of missing values, removal of outliers, data visualization, data transformation, balancing of data etc. Figure 2. shows Use case diagram of input module.



FIGURE 2. Use case diagram of input module

Feature selection is identifying and choosing the input features that are most relevant to the target variable. Selection of feature is done by evaluating feature importance of each feature by assigning scores to input features. Importance indicates the relative importance of every feature when deciding a prediction. After evaluating all the features available with cloud alarms such as alarm ID, name, description, timestamp, rack and location four features name, description, alarm ID and rack are considered for the training. The data considered for training the model is analyzed for its correctness, uniqueness and balance.

In order to choose the suitable algorithm for building the model three algorithms such as decision tree, random forest and naïve Bayes classifier are evaluated for its accuracy of its prediction. Cloud alarms are collected and accuracy of three classifiers is tested for one week period and Average Accuracy obtained by different Algorithm results are tabulated in Table 1.

**TABLE 1.** Average Accuracy obtained by different Algorithms

Algorithms	Average Accuracy (%)				
Decision-Tree	98				
Naïve Bayes	85				
Random forest	96				

Once the results predicted by the model are obtained, analysis of cloud alarms are carried out

and report is generated. Referring to the report if the alarm is critical then the notification is sent to the respective authority. Before creating the ticket, need to check whether the rack or cloud in maintenance mode or any planned activity scheduled so that unnecessary ticket can be reduced. Creation of Jira incident/ticket is done by using python Jira. Client module. These tickets contain alarm ID location, description of alarms

#### 4. Result



FIGURE 3. Plot of accuracy of three models

**TABLE 2.** Average Accuracy obtained by theModel

Week	Average Accuracy (%)
week 1	83
week 2	85
week 3	90
week 4	95

In this paper three supervised learning algorithms were evaluated for accuracy of cloud alarm prediction. Figure shows the average accuracy of prediction of model with cloud alarms collected for one week period using three supervised learning algorithms such as Random Forest, Naïve Bayes and Decision tree classifier. Decision tree algorithm gave higher accuracy of 98%.Figure 3. Reflects the plot of accuracy of three models and Table 2. shows the Average Accuracy obtained by the Model.

Model is implemented with decision tree classifier and prediction accuracy is monitored for four weeks with live data and tabulated in table II. Cloud Alarm is fed model and classified as one of the labelled outputs as 0-Action to be taken and 1-No action required as shown in Table 3. as Output prediction

Alarm ID	Description	Name Severit	Time Stamp	Uuid	rack	Ip	IoC	Output
31xxx	The state of a ser- vice is	Service Critical fail- ure	2020-03- 20T13:56:22.	2a86e1f2- edef-4220- 8ce5-	zxys	10.xx.xx.xx	Hz	Action to be taken
31xxx	The state of the cluster is degraded	Cluster Major degraded	115Z 2019-02- 16T07:17:21. 645Z	b786bfeffdea C7a53498- 59dc- 459f-ae6a- 4ea27bb71377	zxys b	10.xx.xx.xx	Hz	Action to be taken

#### **TABLE 3.** Output prediction

# 5. Conclusion

This paper evaluates three supervised machine learning algorithms such as random forest, decision tree and Naïve Bayes classifier with available alarm events collected. Decision tree classifier is chosen as suitable application for this specific requirement after accuracy of different algorithms are evaluated and compared. This model gathers active alarms from clouds via Rest APIs and predicts whether action is required or not and creates a JIRA tickets to concerned team with the accuracy 95.73%. By using this model, manual effort of analysing each alarm is reduced and also alarms generated by clouds which are in maintenance mode or scheduled for planned maintenance can be excluded which in turn reduce burden on analysis and ticket creation

## **ORCID** iDs

H V Kumaraswamy https://orcid.org/0000-0002-5260-4549

## References

- Adamu, Hussaini, et al. "An Approach to Failure Prediction in a Cloud Based Environment". 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud) (2017).
- Agrawal, Vaibhav, et al. "Log-Based Cloud Monitoring System for OpenStack". 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService) (2018).
- Anand, Manu. "Cloud Monitor: Monitoring Applications in Cloud". 2012 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM) (2012).

- Bhowmik, Madhuparna, Tulasi Sai Siri Chandana, and Bhawana Rudra. "Comparative Study of Machine Learning Algorithms for Fraud Detection in Blockchain". 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (2021).
- Sanzo, Pierangelo Di, Dimiter R. Avresky, and Alessandro Pellegrini. "Autonomic rejuvenation of cloud applications as a countermeasure to software anomalies". *Software: Practice and Experience* 51.1 (2021): 46–71. 10.1002/spe.2908; %20https://dx.doi.org/10.1002/spe.2908.
- Sun, Meijie, et al. "More Flexible: A Free-Customization Technique Based on Log Visualization". 2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA) (2020).
- Vrana, Roman and Jan Korenek. "Efficient Acceleration of Decision Tree Algorithms for Encrypted Network Traffic Analysis". 2021 24th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS) (2021).
- Wong, Tzu-Tsung and Po-Yang Yeh. "Reliable Accuracy Estimates from k-Fold Cross Validation". *IEEE Transactions on Knowledge and Data Engineering* 32.8 (2020): 1586–1594. 10.1109/ tkde.2019.2912815;%20https://dx.doi.org/10. 1109/tkde.2019.2912815.
- Yu. "An Improved Random Forest Algorithm Based on Attribute Compatibility". IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (2019).
- Yuan, Yue, et al. "An Approach to Cloud Execution Failure Diagnosis Based on Exception Logs in OpenStack". 2019 IEEE 12th International Conference on Cloud Computing (CLOUD) (2019).

S © Dhanya Karanth et (†) (CC al. 2021 Open Access. This article is distributed under of the Creative Comthe terms 4.0 mons Attribution International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and

reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

**Embargo period:** The article has no embargo period.

To cite this Article: Karanth, Dhanya S, H V Kumaraswamy, and Rajesh Kumar. "Workaround Prediction of Cloud Alarms using Machine Learning." International Research Journal on Advanced Science Hub 03.08 August (2021): 164– 168. http://dx.doi.org/10.47392/irjash.2021.231