Effective fault detection approach for cloud computing

Pola Ashritha¹, M Banusri¹, R Namitha¹, Dr J Shiny Duela²

¹Student at CSE Department in SRMIST, Bharathi Salai, Ramapuram, Chennai-600089, India.
²Associate Professor of CSE Department in SRMIST, Bharathi Salai, Ramapuram, Chennai-600089, India.

Abstract. In cloud computing, accessibility to data anytime is crucial, acquiring data and maintaining that data without any loss or incursion is an essential task. A cloud service must have the potential to recognize unexpected faults and respond effectively. Hence, a system to identify faults is developed which recognizes anomalies using various techniques and algorithms. Several different types of faults occur in cloud computing which causes the poor performance of cloud computing. The various types of faults occurred are collected and classified using a fuzzy one class support vector machine and long short term memory(LSTM) algorithm. Comparative analysis of accuracy and precision is done with various algorithms like Naive Baye Algorithm, Decision Tree Algorithm, K-Neighbors Algorithm, and Logistic Regression Algorithm. Experimental results show that Logistic regression gives the best accuracy, precision and performance for detecting faults among the aforementioned algorithms. The efficacy of our model is illustrated in experimental results.

1. Introduction

Cloud computing has tremendously advanced over the recent years. Many of the internet applications are moving towards cloud computing. Cloud services help us to use and work applications over the internet without having to download. Tremendous amount of data is stored and handled with the help of the cloud. Cloud computing must be able to provide great security to this data to avoid corruption and malign data. It provides attributes like scalability, reliability, storage and helps with management of sizable data. Hence to solve this problem fault tolerance is being used to achieve high performance in cloud computing. Faults cause error occurrence which leads to the failure of the system. Avoiding failures is essential to improve cloud failures. Innumerable types of faults [1],[5],[8],[11],[12],[13] that occur in cloud services are classified into three main categories: software fault, byzantine fault and hardware fault.

1. Faults in Hardware part - Faults that arise due to power failure, storage or fault in memory processor, switch or router fault, disk failure, fault in secondary components etc. These faults can be fixed manually and to avoid hardware faults from occurring proper maintenance must be done.

2. Software faults - This can be further classified into network faults, data faults, process faults. Network faults Arise due to packet loss, message does not reach destination, packet failure, packet corruption, configuration failure, link failure, network congestion. Data faults are occurred due to jammed data, incomplete data, unreadable data. These faults may occur due to intrusion, breach of
firewall, malicious virus and software bugs. Process faults are Memory leak, CPU hog, middleware failure, application failure, when nodes do not respond on time or error page emerges instead of the actual log in page causing timing faults. Faults in the OS, hypervisor or VM.

3. Byzantine faults - Byzantine fault Tolerance (BFT) is the ability to tolerate Byzantine faults. This type of fault mostly occurs in a distributed system [6]. All the nodes in a distributed system must agree on a decision. When the nodes do not follow the same decision because of various reasons like communication problems between the nodes or confusion among them, it causes failure in the distributed system. To rectify this failure a system must be able to ignore the nodes that fail to communicate or act maliciously and must work with the remaining nodes assuming that they work on the same decision. Server replication and petri nets can also help in reducing the risk of byzantine failure.

Fault recognition and system recuperation nowadays in cloud systems is a critical problem. A cloud service must have the ability to recognize the fault and respond to it within a short period of time. Many varieties of faults arise in cloud computing and due to this it causes poor performance in the cloud. To solve this issue and to reduce the possibility of non-success, an optimal fault tolerance mechanism is used. Long short term memory is used to classify the faults.

Numerous solutions have been given to this fault detection such as mathematical methods, statistical method [1], density-based method, support vector machine[4], Scale-free network topology, kernel-based method, decision tree method [2], random forest method[7][9]. A fault tolerance is an optimized approach in which a model is drawn to tolerate the faults based on the authenticity of each node and it can be put back if the performance is not excellent [3]. A study [7] To analyze the experimental performance of the numTrees and numFeatures parameters using the Random forest method. The study [5] have used many tools like ASSURE, where this works for the rescue points when the user expects the failure. All this technique will help the author to reduce the failure of the nodes, and to improve the reliability. SVM-grid [4] is used to predict the performance and to decrease time cost. The model proposed will mainly predict the faults present in the cloud system. The author [4] has used the detection performance to improve the utilization of the proposed system.

2. System Model

Cloud computing is becoming a rising, important and innovative platform for web server applications, where cloud computing has its own technical challenges and that is very increasing because of the fault in web pages. In this study, we are focusing on the faulty webpages as an input and we propose an effective software fault detection method where our system is created to detect fault based on the availability of the computer node. Here the cloud master is the main feature in this environment where it records the number of virtual nodes created and also gives the direct connection to the Fault indulged server. The virtualization helps us to manage the records of the virtual nodes; the new virtual node is created for each physical node that is present in the resources of the physical servers. For a single physical server a set of virtual nodes is created. The cloud information system is used to collect all contained data of the nodes IDs, servers IDs and the rate, where this record help us to keep track the nodes identity and tasks that are set to the virtual nodes of the individual physical servers. There are innumerable faults that occur in cloud systems like system faults, software faults, system faults, byzantine faults are detected by all the techniques and that are collected and analyzed in the database that is present in the data collection set. The data which lies in an outer radius and an abnormal length is collected into a single data that is known as the outliers. To detect these outliers, a FOCSVM is used. LSTM is used to select the vital features and to
classify the data. Logistic regression algorithm is used to detect whether the data is normal/anomaly.

**Figure 1.** Model for fault detection system

### 3. Fault Collection in Cloud Computing

The main objective of detecting faults is to avoid disruptions from occurring which may lead to the failure of the system. Fault detection is a major issue in real-time cloud applications as a lot of research has been done on developing a reliable system. Fault detection is very essential because a few minutes of failure in a cloud system may lead to loss of money and valuable data. Faults that occur in the software part of the system cannot be fixed manually. Few strategies and techniques [1][5] like pinging strategy, heartbeat tactic, intrusion check, and keepalive technique are used to recognize the faults and collect that data.

1. **PINGING STRATEGY** - An ACK(Acknowledgement) signal is sent to the detector node continuously from computer nodes. If a message fails it will be considered as a faulty node. Pinging strategy helps in detecting damaged nodes but it is hard to detect malign nodes or faults that occur due to buffer overflow.

2. **HEARTBEAT TACTIC** - In heartbeat tactic a periodic signal is sent to other computer nodes from a detector node. The signal is sent in regular intervals. If the receiving node does not send back a signal within a given time then it is considered to have failed. The detector node considers it as a fault.

3. **INTRUSION CHECK** - Intrusion is the behavior of a computer node [2] with corrupt intent, which tries to affect other nodes in the network. Nodes that behave unusually are recognized and are compared with a model of normal behavioral nodes. This comparison identifies malicious nodes. Techniques like [5] checkpoints, task resubmission, S-Guard can be used to reduce the risk of malign intent.

4. **KEEPALIVE TECHNIQUE** - In keepalive technique a message is sent within regular intervals of time from one node to another to check whether the connection between nodes still exists. If the connection is cut then it will be considered as that particular link between the nodes is severed.

Acquiring the data required to develop a fault detection system is fundamental. Most of the current
models use normal data/few data to train their systems because of various challenges. Enormous variety of abnormalities and faults exists in cloud systems, obtaining the data can be very costly. A system has more than thousands of nodes to store details about and check. So, A faulty cloud based web application is developed. Virtualization is done for the cloud based application with the help of a hypervisor. Hypervisor helps to create virtual machines (VM), thus a virtual environment is created [13]. A cloud master has a direct connection to service providers and virtual nodes. The cloud master documents the details about all the virtual nodes. A cloud information system keeps track of all the tasks allocated to the virtual nodes. The data in cloud information systems is used to detect faults by applying different strategies. The concept of cloud master, cloud information system and various different strategies helps in identifying faults in nodes.

4. Detection of Faults

4.1. Data Collection and Classification

A fault detection technique must be able to recognize obstacles for the effective and smooth maintenance of the cloud system. It helps to maintain the integrity of the original data without corrupting it. Enormous amount of data is stored and made available always. To ensure a disruption free service a fault detection model is developed. Collecting the data required to train the model is crucial. The data is collected with the help of cloud master, cloud information system, intrusion check, pinging strategy, heartbeat tactic, and keepalive technique. Most of the existing models only use the error samples collected using these strategies. Developing an effective system also requires outlier detection. Outlier is a data that varies unusually from existing data. The maximal or the minimal data or sometimes both are considered as outliers. In some conditions both maximal and minimal values are not considered as outliers if the sample mean is nearby to the max and min values. An outlier may lead to bad data or scientifically significant data. Outliers are sometimes used to identify intrusions. Observations of outliers assist in avoiding ambiguous assumptions. Recognizing outlier data can be a complicated task. Complex algorithms and statistical models are necessary. In this study we propose a FOCSVM to identify outliers.

A one class support vector machine (OCSVM) is a supervised learning algorithm that examines the data and maps it. OCSVM [1] is trained with normal data then a conclusive boundary is marked. If a new data is recognized it is either a normal data or it is in the exterior side of the boundary. The exterior side of the boundary is considered as abnormal data. OCSVM is not very effective in detecting outliers. So, the combination of fuzzy logic and one class support vector machine is used in spotting the outliers. A fuzzy logic is similar to human reasoning which also allows intermediate responses between yes and no: Fuzzy logic can assist with uncertainty and fuzziness with data. Outlier detection can be improved by applying fuzzy logic with OCSVM. Fuzzy one class support vector machine is used in obtaining the required data set and also improves the efficiency of the training data by detecting outliers.

The training data required has been acquired. Now, the faults are to be classified and extraction of important features is done applying a [13] LSTM approach. LSTM helps in dealing with vanishing and exploding gradient factors. The LSTM consists of an internal system called gate. There are three types of gates in the LSTM model. The forget gate, input gate and output gate. The forget gate decides which input data should be kept and which input data will be forgotten. Now, the data collected will be sent as a sequence into an LSTM model. The gates will study the data and identify the data from the sequence that is essential for the prediction. After applying the LSTM algorithm the adequate amount of attributes required in detecting faults are identified.
4.2. Fault Detection

A faulty cloud base web application is developed with the help of a hypervisor which generates the required virtual machine and makes the system a virtual environment. Numerous nodes are present in a cloud environment, to discover which node doesn’t work is a fundamental task to collect faulty data. A cloud master records the number of virtual nodes created. The cloud information system records the task given to the virtual nodes. The data in cloud master and cloud information system is used to easily determine which node is faulty, various techniques like heartbeat tactic, pinging strategy, keepalive alive technique and intrusion strategy helps in effectively collecting the faulty data. A FOCSVM is used in detecting the outliers. Thus, the training data is obtained. There are various types of faults present in cloud systems that occur due to various obstacles like hardware faults, software faults and byzantine faults. Software faults are further classified into network faults, process faults and data faults, to effectively develop a detection and diagnosis model we have to classify the data and train it accordingly. The classification of faults is done using the long short term memory algorithm. Thus, the classified training data is acquired. Logistic regression algorithm is used in detection of faults. Logistics Regression is a statistical model which does not necessarily need too many computational resources.

\[
y_i = \beta_0 + \beta_1 x_i + \varepsilon_i
\]

where,
\[y_i = \text{Dependent Variable}\]
\[\beta_0 = \text{Population Y Intercept}\]
\[\beta_1 = \text{Population Slope Coefficient}\]
\[x_i = \text{Independent Variable}\]
\[\varepsilon_i = \text{Random Error Term}\]
Finding Accuracy and Precision Algorithm

1: Read “test_data.csv” as Test
2: Describe the Test file using describe()
3: Testcat = cattest.apply(encoder. Fit_transform)
4: Test_x = pd.concat([sc_testdf, enc_test], axis=1)
5: If Test_x is true:
   6: Then return “normal”
   7: Else return “anomaly”
8: Selected_features = [i for j, i in feature_maps if j is true]
9: Calculate Accuracy and precision
10: Print Accuracy and precision

By using logistic regression equation (1), the data to be tested is taken as the input to read the file then the test data is described. Using the encoder the test data is transformed. Then checking whether the given data is normal or anomaly. The next step is to select the features extraction and then the using the logistic regression function we are finding the accuracy and precision of the given data. Logistics Regression is applied to the important features extracted with the help of LSTM. The accuracy obtained by applying logistic regression is 0.9747.

5. Results and Discussion

A fault detection model using logistic regression is developed. The important features are obtained when the LSTM algorithm is applied. The output obtained from the data collection and classification phase is used in training this model. Various algorithms are applied to the obtained dataset. Naïve Bayes algorithm, decision tree algorithm, K-neighbors algorithm and logistics algorithm is used to form a comparative analysis. The model accuracy and cross validation mean score is greater for logistic regression. Faults are injected to our model the logistics algorithm will analyze the data and the output will be in either normal/anomaly.

![Figure 3. Result](image.png)
Figure 4. Comparative analyses of accuracy and cross variance mean score

| FEATURES / ALGORITHM | CROSS VALIDATION MEAN | ACCURACY | PRECISION | MACRO AVERAGE | WEIGHTED AVERAGE |
|----------------------|------------------------|----------|-----------|---------------|-----------------|
| NAIVE BAYE ALGORITHM | 0.9071                 | 0.9071   | 0.93      | 0.88          | 0.91            |
| DECISION TREE ALGORITHM | 0.90608             | 0.9067   | 0.96      | 0.94          | 0.93            |
| K-NEIGHBORS ALGORITHM | 0.9314               | 0.945    | 0.95      | 0.93          | 0.91            |
| LOGISTIC REGRESSION  | 0.9649               | 0.9747   | 0.99      | 0.97          | 0.98            |

Table 1. Comparison of results

6. Conclusion

In this study, a fault detection model for cloud systems is developed. Cloud computing has tremendously advanced over the recent years. Many of the internet applications are moving towards cloud computing. Detecting faults is crucial to eradicate the errors that might occur due to unrecognized faults in real time. These faults cause an application failure which affects a large population of users and results in heavy economic loss. In this study, we are using a few strategies and techniques like pinging strategy, heartbeat strategy, intrusion check, keepalive technique to recognize the faults and to collect the data. Outliers are
detected to escalate the accuracy of the model. So, FOCSVM is applied to detect the outliers of the present data. To classify and extract the essential attributes from the collected data we are using the LSTM. This paper presents a Fault Detection approach using logistic regression. Although more study can be done on forward and backward recovery.

References

[1] Khiet B T, Ien V, Nguyen C, Pham T V and Tran H. A fault detection and diagnosis approach for multi-tier application in cloud computing, 2020, Journal of Communications and Networks. 22, 399-414.
[2] Amir J, Abharian Sanaz and Guojun W. Feature Selection and Intrusion Detection in Cloud Environment based on Machine Learning Algorithms, 2020
[3] Bashir M, Kiran M, Irfan-ullah A and Kabiru M. 2016. Optimising Fault Tolerance in Real-Time Cloud Computing IaaS Environment, 2016, 363-70..
[4] Peiyun Z, Sheng S and Mengchu Z. An online fault detection model and strategies based on SVM-grid in clouds, 2018, IEEE/CAA Journal of Automatica Sinica. 5, 445-56.
[5] Mridula D and Gupta neha. 2017 Comparative analysis of fault tolerance models and their challenges in cloud computing. International. 2017 Journal of Engineering & Technology.6. 36.
[6] Guisheng F, Huiqun Y, Liqiong C and Dongmei L. Model Based Byzantine Fault Detection Technique for Cloud Computing. Proceedings, 2012 IEEE Asia-Pacific Services Computing Conference, APSCC 2012. 249-56.
[7] Khoshgoftaar T M, Golawala M and Hulse J V., "An Empirical Study of Learning from Imbalanced Data Using Random Forest," 19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007), Patras, Greece, 2007, 310-17, 46.
[8] Wang T, Zhang W, Wei J and Zhong H, "Fault detection for cloud computing systems with correlation analysis," 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 2015, 652-58
[9] Yang S and Huang H, "Design a Hybrid Flooding Attack Defense Scheme under the Cloud Computing Environment," 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS), Beijing, China, 2019, 41-6
[10] Hazim J, Sarkar and Nurul. Comparison-based system-level fault diagnosis protocols for mobile ad-hoc networks: A survey, 2015 Journal of Network and Computer Applications. 60. 68-81.
[11] Kumaripriti and Kauparmee. A Survey of Fault Tolerance in Cloud Computing. Journal of King Saud University , 2018, Computer and Information Sciences. 10
[12] Sun L D, Hussain H, Farookh , Omar H, Chang and Elizabeth. Cloud service selection: State-of-the-art and future research directions. Journal of Network and Computer Applications, 2014. 45. 134–50.
[13] Pangun P, Marco D, Piergiuseppe, Hyejeon S and Junseong B. Fault Detection and Diagnosis Using Combined Autoencoder and Long Short-Term Memory Network. Sensors, 2019, 19. 4612.