iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** IV **Month of publication:** April 2017

DOI: http://doi.org/10.22214/ijraset.2017.4069

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Prediction Framework for Fault Resolution in Networks

Sushma. M[1], Niroop. R. S[2], Varsha. K. R[3]
[1,2,3]Department of CSE, R. V. College of Engineering, Bengaluru, Karnataka, India

Abstract: ***The continued increase in the size and complexity of modern networks has led to a commensurate increase in the size of their logs. To ensure high availability and correct operation of networks, it is essential that failures be detected promptly and quickly, their causes have to be diagnosed and remedial actions taken. Router logs are an invaluable resource to systems administrators during fault resolution. A lot of time spent in fault resolution is in sifting through large volumes of information, which includes router logs, to find the root cause of the problem. Therefore, the ability to analyse log files automatically and accurately will lead to significant savings in the time and cost of downtime events for any organization. The automatic search, analysis and prediction of errors using router logs is the primary motivation for the work carried out in this project. Different supervised machine learning techniques are compared and it is shown that a prediction model framework using RandomForest Algorithm can be used for automated fault detection as it has more accuracy and efficiency.***
Keywords: Log Analysis, Supervised Machine Learning, KNN, Random Forest, SVM, Logistic Regression

I. INTRODUCTION

The existing heterogeneous networks have high availability requirements, and failure in these networks can lead to loss of revenue, customer dissatisfaction, and may even have legal consequences. During the last decade, data centres and computer networks have grown significantly in processing power, size, and complexity. As a result, organizations commonly have to handle many gigabytes of log data on a daily basis. In order to ease the management of log data, we can make use of data mining and machine learning methods for discovering event patterns from event logs. This knowledge can be used for many different purposes like the development of event correlation rules, detection of system faults and network anomalies, visualization of relevant event patterns.

The traditional approaches for troubleshooting relies on the knowledge and experience of domain experts to figure out the solutions manually and is very inefficient. An automated tool to monitor, manage and maintain a heterogeneous system of routers is necessary. Mining of router log files by using a large amount of log data to prevent occurrence of errors or alert the system by predicting the possible errors that can occur.

The objective of this project is to pre-process the router log data for automated analysis and then automatically find frequent errors from historic monitoring data by using supervised machine learning models. In particular, we propose to mine router log files for network management by acquiring the needed knowledge automatically from a large amount of historical log data. Then, we compare the efficiency and accuracy of the implemented learning models to provide the best learning and prediction model for fault diagnosis. Minimal possible assumptions about the router log data are made so that this analysis framework can be widely used.

II.LITERATURE SURVEY

“LogCluster - A Data Clustering and Pattern Mining Algorithm for Event Logs “[2] about Event log Clustering – SCLT and it dealt with large amounts of textual log data without well-defined structure. However, this paper only deals with linear clustering and the entire process is not automated.

“Proactive Failure Detection Learning Generation Patterns of Large-scale Network Logs” [4] deals with concepts like Feature Extraction, Log Template. The algorithm used automatically learns the relationship between critical failures and log messages without using any previous knowledge. Limitations include automatic update of the features and the model not implemented.

“Spatio-temporal Factorization of Log Data for Understanding Network Events” [5] deals with Syslog – SNMP. The algorithm processes network logs, including syslogs and alarm messages reported by the NMS for network anomalies. The algorithm cannot handle complexity of long messages.

“Fault Diagnosis in Enterprise Software Systems Using Discrete Monitoring Data” [6] gives a model for diagnosing recurrent faults. Challenges of this thesis include comprehensive, low-cost approach to process log files and this model assumes complete reliability and controllability of the model.

The paper “Process mining of event logs in auditing: opportunities and challenges” [7] deals with Event logs, ERP Databases and

Process Mining. This algorithm deals with various formats of event logs and matching of long messages. This model cannot recognize anomalies or frauds that cannot be captured by analysing input data.

III. METHODOLOGY

We extract the required log files from different nodes in the network. Then, pre-process and clean the raw router log data to obtain the required useful logs in csv format. Errors are searched using rule based classifier and regex matching. These errors are used to train the prediction model using a supervised learning model. Different algorithms - k-Nearest Neighbour, Support Vector Machine, Random Forest and Logistic Regression - are compared and the best model in terms of accuracy and efficiency is chosen.

The classification performed on an input error is accurate only if the error type has been seen previously. Otherwise, the prediction accuracy is decreased.

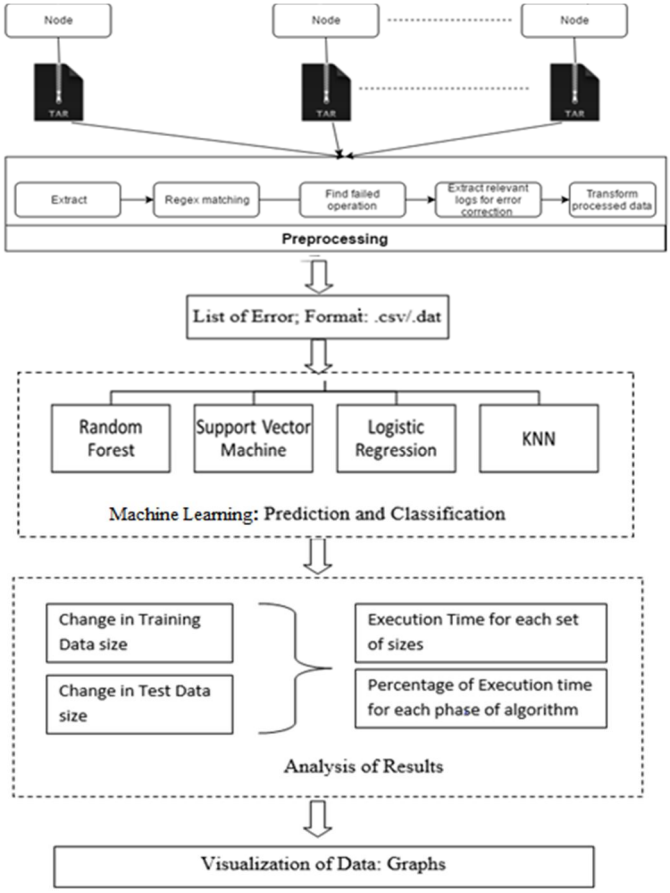


Fig. 1. High level System Design

IV. IMPLEMENTATION

A. *Pre-Processing of Logs*

Using Python 2.7.11 programming language, we parse the raw log file and store required information in dictionary. We dump these dictionaries into memory using pickle library and can be retrieved when required. We search the dictionaries to find errors.

Algorithm: Pre-process Log files

Input: Router log tar file

Output: .xls file containing each event and timestamp indicating errors.

1) Uncompress the input log file

2) For each node in log director

3) Extract each type of .gz file to different trace files.

4) For each event in log:

5) Validate event and date format using Regex matching
6) Convert date to timestamp
7) Store each event and its timestamp in dictionary
8) Dump the dictionary into memory using pickle.
9) Repeat Step 3 and 4 for all nodes.
10) Create list of operations and find the last failed operation.
11) For each value in dictionary
12) Match value with predefined error pattern using Regex
13) Repeat Step 7 for all nodes.
14) Export the event indicating error to .xls format

B. Machine Learning Models

At first the raw log data is converted to vectors using vectorizer function i.e. measuring impact each word in the log for predicting the error class. This vectored input acts as input for the following three algorithms, which are implemented in Python 2.7.11.

1) *K-Nearest Neighbour:*
a) Determine parameter k(number of nearest neighbours).
b) Calculate the distance between the query instance and all the training samples.
c) Determine the nearest neighbour based on the k-th minimum distance.
d) Get the classification of the nearest neighbour and use simple majority to predict the class of test instance.
2) *Support Vector Machine:*
a) Assume a hyperplane.
b) Evaluate violating points in dataset and modify hyperplane.
c) Repeat until dataset has no violating points wrt hyperplane.
d) Classify points based on its location wrt hyperplane.
3) *Random Forest Classifier:*
a) Choose the number of training points, N and the number of features in the training data, D.
b) Choose L to be the number of individual models in the ensemble.
c) For each individual model l, choose dl (dl < D) to be the number of input variables for l. It is common to have only one value of dl for all the individual models.
d) For each individual model l, create a training set by choosing dl features from D without replacement and train the model.

4) *Logistic Regression:*
a) Vectorised input is used as input to the sigmoid function
b) Cost factor is calculated using the sigmoid function.
c) $J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$
d) Gradient descent is used to arrive at the optimum value of J(θ).
e) Classification is done based on the value of J(θ).

V. EXPERIMENTAL ANALYSIS AND RESULTS

A. Experimental Dataset

The initial experimental dataset used is a collection of tar.gz files. Individual zip file is named in accordance with the timestamp of router logs. Around 15 number of such files makeup the initial dataset.

After pre-processing, we obtain a .csv file, which contains three columns Timestamp, Log, Error class. Pre-processed dataset has 7356 entries. Out of 7356 entries different proportions of dataset is used as train dataset and test dataset for error class prediction using the different machine learning techniques

B. Evaluation Metrics

The Evaluation metrics are the key to understanding how the prediction model performs when applied to a test dataset. The following metrics are used

1) Accuracy A key metric to start with is the overall prediction accuracy. It is the fraction of instances that are correctly classified.
2) Classification error number of samples incorrectly classified (false positives plus false negatives) and is evaluated by the formula

$$E_i = \frac{f}{n} \cdot 100$$

C. Performance Analysis

Through performance analysis of the above mentioned evaluation metrics such as accuracy and classification error, we try to prove the correctness of the chosen algorithms and the algorithm better suited for our system.

1) *K-Nearest Neighbour:* This learning model takes more time in testing the data. As evident from the table below as the test data size increases the time taken for testing increases. In terms of accuracy, this algorithm provides 87% accuracy, which is least among all the algorithms used in this project.

Table I. Execution time of k-nn

Test case	Train data size	Test data size	Training Time	Test Time	Total Time
Test case-1	6856	500	4.665	3.282	7.947
Test case-2	6356	1000	4.198	6.08	10.278
Test case-3	5856	1500	3.648	8.456	12.105
Test case-4	5356	2000	3.505	10.416	13.921
Test case-5	4856	2500	3.149	12.527	15.676
Test case-6	4356	3000	2.882	13.712	16.594

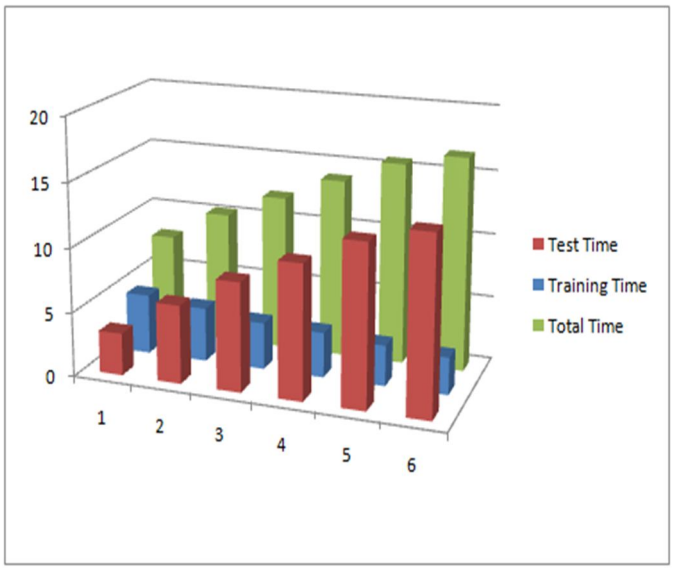


Fig. 2. Graph representing execution time of k-NN

2) *Support Vector Machine:* The time taken for each phase of the process of prediction is tabulated. This is repeated for increasing training data sizes and it is observed that the total time increases as expected. An important observation that can be made here, is maximum amount of time is spent during matrix creation step. The algorithm shows an accuracy of 94.82% but requires a lot of time for larger data sizes.

Table II. Execution time of svm

Test case	Train data size	Create matrix	Training Time	Test Time	Total Time
Test case-1	1000	5.637695	0.466797	0.52929	6.64844
Test case-2	2000	8.533204	0.583006	0.50195	9.63281
Test case-3	3000	11.81738	0.751953	0.50585	13.0898
Test case-4	4000	15.35938	0.90625	0.50976	16.791
Test case-5	5000	16.41309	1.292969	0.49609	18.2168
Test case-6	6000	21.10254	1.245117	0.50585	22.8682

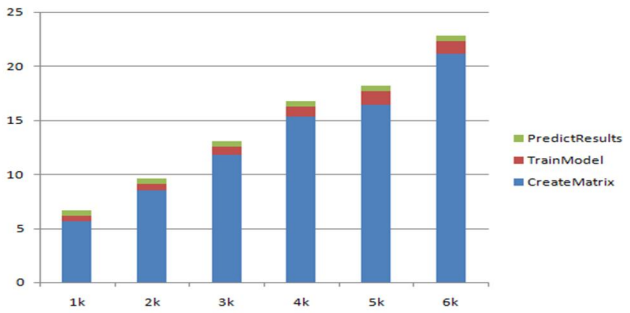


Fig.3. Graph representing execution time of SVM

3) *Random Forest Classifier:* This learning model takes more training time when compared to test time. It uses eager learning method and test time is considerably small. It has an accuracy of 99%.

Table III. Execution time of random forest classifier

Test case	Train data size	Test data size	Training Time	Test Time	Total Time
Test case-1	6856	500	5.888	0.425	6.314
Test case-2	6356	1000	5.002	0.7596	5.762
Test case-3	5856	1500	4.486	1.014	5.501
Test case-4	5356	2000	4.287	1.388	5.675
Test case-5	4856	2500	3.82	1.898	5.718
Test case-6	4356	3000	3.691	1.64	5.331

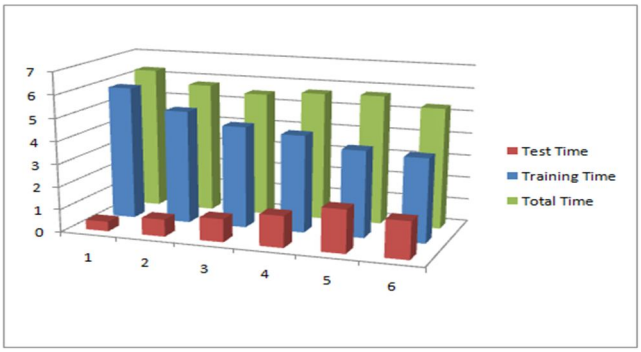


Fig.4. Graph representing execution time of Random Forest Classifier

4) Logistic Regression: This learning model takes more testing time when compared to training time when compared to random forest classifier. Training time decreases noticeably as the training data size decreases. This model also provides an accuracy of 99%.

Table IV. Execution time of logistic regression

Test case	Train data size	Test data size	Training Time	Test Time	Total Time
Test case-1	6856	500	3.6	0.441	4.046
Test case-2	6356	1000	3.402	0.938	4.341
Test case-3	5856	1500	2.934	1.219	4.154
Test case-4	5356	2000	2.865	1.659	4.524
Test case-5	4856	2500	2.613	2.069	4.682
Test case-6	4356	3000	2.093	2.738	4.832

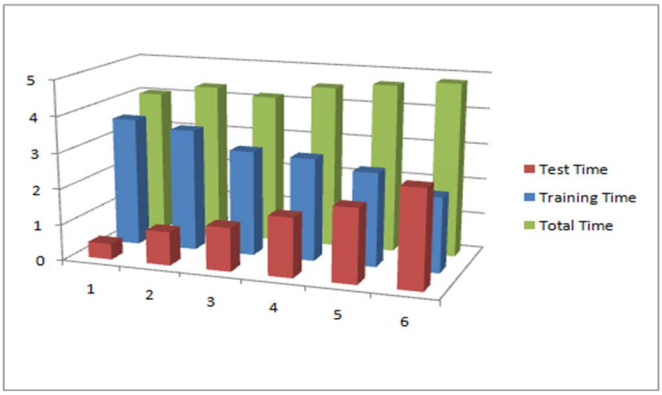


Fig.5. Graph representing execution time of Logistic Regression

D. Inference from the Results

Among the different prediction models used, Random Forest classifier and Logistic Regression provide the same accuracy. But, Logistic Regression is much faster than Random Forest. All four models are compared side by side to choose the best algorithm with higher accuracy and lower error rate. It is seen that Logistic Regression algorithm has 99% accuracy for the test data set used and takes the least time. Hence, Logistic Regression Algorithm is used to build the prediction model.

VI. CONCLUSION AND FUTURE WORK

We have a built a prediction model for predicting errors in networks using the different supervised learning algorithms. Various aspects of the four algorithms to predict the error in the logs and tested for different cases. It is seen that Logistic regression performs the best and can be used to build the automatic analysis and prediction tool.

A. Limitations

a) Few predictions of logs are false positive.
b) As supervised learning model is used, training of model is compulsory.
c) Any effects of recovery actions are not considered.
d) The model cannot learn if the log pattern is completely changed.

B. Future Work

1) Build a centralised system to execute this tool in routers so that error-detection is done in real-time.
2) Analyse the false positive logs for abnormalities.
3) Implementing online learning by which the system can improve itself by getting feedbacks.
4) Side Effects of Recovery and Probe Actions

REFERENCES

[1] MuhammetMacit, EmrullahDelibaş, BahtiyarKaranlık, Alperenİnal, TevfikAytekin, "Real time distributed analysis of MPLS network logs for anomaly detection", Network Operations and Management Symposium (NOMS), 2016.

[2] RistoVaarandi, MaunoPihelgas, "LogCluster - A Data Clustering and Pattern Mining Algorithm for Event Logs ", 11th International Conference on Network and Service Management (CNSM), 2015.

[3] Feng Cheng, Andrey Sapegin, Marian Gawron , Christoph Meinel "Analyzing Boundary Device Logs on the In-Memory Platform", IEEE 17th International Conference on High Performance Computing and Communications (HPCC), 2015.

[4] Tatsuaki Kimura , Akio Watanabe, Tsuyoshi Toyono , Keisuke Ishibashi, "Proactive Failure Detection Learning Generation Patterns of Large-scale Network Logs", 11th International Conference on Network and Service Management (CNSM), 2015.

[5] Rui Ren, Xiaoyu Fu, Jianfeng Zhan, Wei Zhou, Zhen Jia, Gang Lu, "LogMaster: Mining Event Correlations in Logs of Large-scale Cluster Systems", IEEE 31st Symposium on Reliable Distributed Systems (SRDS), 2012.

[6] Thomas Reidemeister,"Fault Diagnosis in Enterprise Software Systems Using Discrete Monitoring Data", Doctoral Thesis, University of Waterloo, 2012.

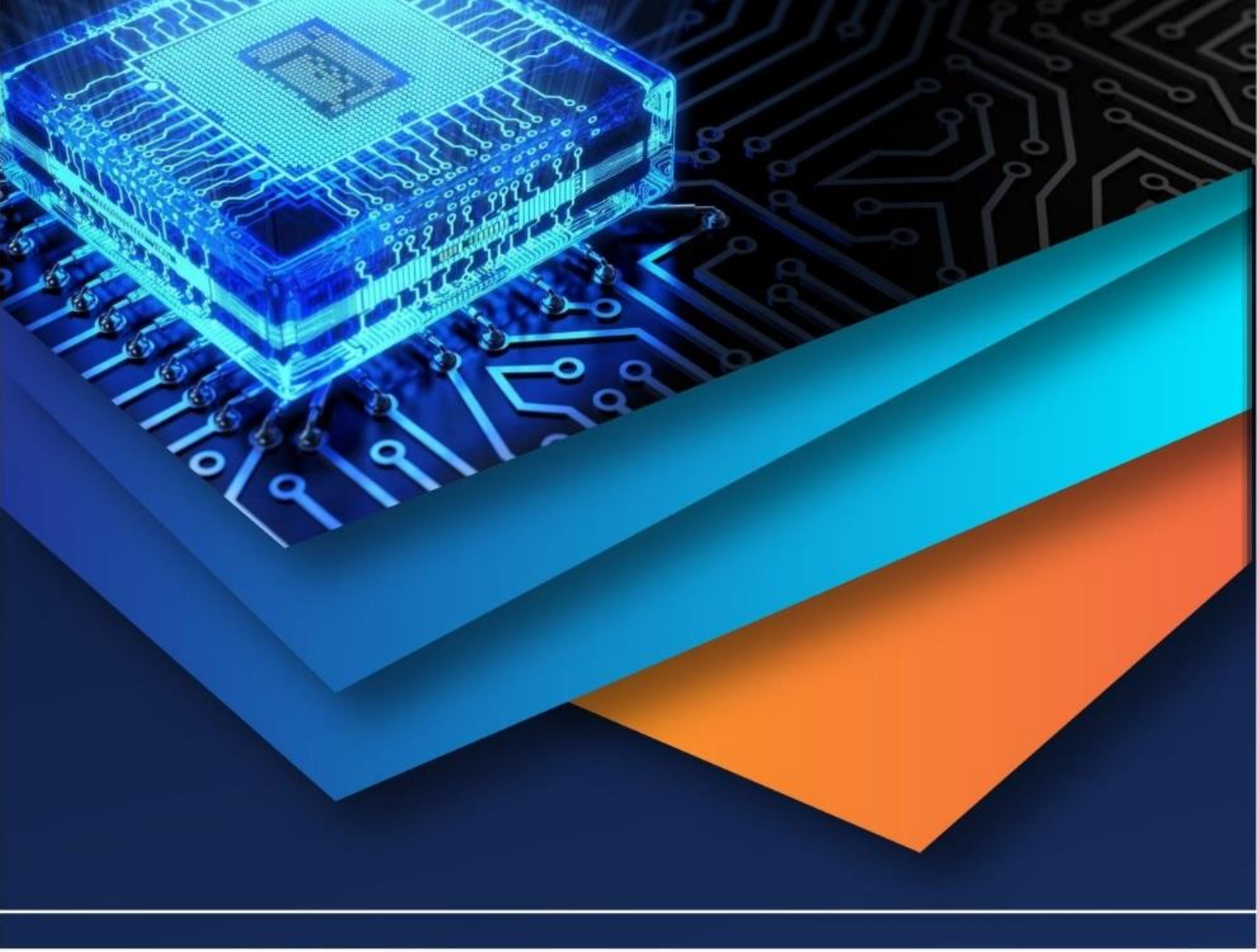
[7] MiekeJans, Michael Alles, Miklos A. Vasarhelyi, "Process Mining of event logs in auditing: Opportunities and Challenges", Sixth International Conference on Extending Database Technology, 2010

[8] Abdallah Ghourabi, Tarek Abbes, Adel Bouhoula, " Data analyzer based on data mining for Honeypot Router", IEEE/ACS International Conference, Computer Systems and Applications (AICCSA), 2010.

[9] Tatsuaki Kimura, Keisuke Ishibashi, Tatsuya Mori, Hiroshi Sawada, Tsuyoshi Toyono, Ken Nishimatsu, Akio Watanabe, Akihiro Shimoda, KoheiShiomoto, "Spatio-temporal Factorization of Log Data for Understanding Network Events", IEEE Conference on Computer Communications(INFOCOM), 2014.

[10] AdetokunboMakanju," Exploring event log analysis with minimum Apriori information", Doctoral Thesis, Dalhousie University, 2012.

[11] Weixi Li, "Automatic Log Analysis using Machine Learning", Uppsala Universitet, 2013.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY