"Study and Design a Framework for Abnormality Analysis of Stream Data"

[1]**Pranita P. Bavaskar,** *PG Student, Department of Computer Science and Engineering, SOCSE, Sandip University, Nasik, India*
[2]**Onkar Kemker,** *Dean of Department of Computer Science and Engineering, SOCSE, Sandip University, Nasik, India*
[3]**Aditya Kumar Sinha,** *Associate Director, HOD-ACTS, CDAC, Pune, India*

---***---

Abstract: In current world there are various mechanism for log analysis. The different types of tools and methods are used for log analysis. There are different purposes associated with log analysis like security, system tracking, performance improvement of system, abnormality analysis. In this research, we develop and analyse framework that monitor students activity during online examination. Our goal is to detect cheating by looking into logs using machine learning and ELK stack. This stack is applied for elastic search and is powerful open-source search and analysis engine used for full-text search and for analyzing logs and metrics. Logstash is an open-source tool which is part of ELK stack used to ingest and transforms log and events. In our design we set frame for exam logs to check student behaviour and if any hall ticket log goes out of this framework like if he/she try to inject pen-drive or access internet during exam that will be detected.

Keywords — Beats, Elasticsearch, Kibana, Logs, Logstash, Machine Learning.

1. INTRODUCTION

In the present digital world online exams are almost every where like for banking exams, government exams, college exams with this change we also worry about cheating in online exams so we have to detect online exam cheating so we can detect cheating by log files there are many tools that are used in different scenarios in collecting the logs and analysing those logs to detect the malicious activity, there are also many commercial tools to give the same more accurately. The tools using in this Architecture is a combination of tools including Elasticsearch, Logstash and beats and Kibana from the ELK Environment. The main reason why many use the commercial tools is it don't need much knowledge on the internal working of the tool while using those tools everything is automated in these tools from the collection of the logs to the generating the report to the client. In this treat growing world there much demand for the security operation centers. By using this architecture many can built an environment with less effort and more accuracy.

2. METHODS AND TECHNOLOGY

A. The need for log analysis

Logs provide track of how our system is behaving. However, the content and format of the logs changes among different services, among different components of the same system. For online exam cheating detection we collect the logs of each computer in the exam hall. In this log file each activity of each student is recorded and by tracking that logs we can find the behaviour of student like he/she inject the PD or try to access the some black listed web sites. Basically to find he/she try to cheat or not by looking into the log files.

Log files have its own format so we have to convert into our format to import in python program so we use ELK stack.

B. Introduction to ELK Stack

ELK stack use as a complete log analysis solution, and ELK stack is role of each of the open source components of the stack, namely, Elasticsearch, Logstash, and Kibana. Also, it briefly explains the key features of each of the components.

C. What is ELK Stack?

The ELK platform is a complete log analytic solution, built on a combination of three open source tools—Elasticsearch, Logstash, and Kibana. It tries to address all the problems and challenges related to logs. ELK utilizes the open source stack of Elasticsearch for searching and analyzing data; Logstash for centralized logging management, which includes shipping and forwarding the logs from multiple servers and Kibana for powerful and human understandable data visualizations. ELK stack is currently maintained and actively supported by the company called Elastic

Let's look at a brief overview of each of these systems:

- Elasticsearch
- Logstash
- Kibana

1) Elasticsearch:

Elasticsearch is a distributed open supply search engine primarily based on Apache Lucene, and released under an Apache 2.0 license (because of this that it may be downloaded, used, and changed freed from fee). It provides horizontal scalability, reliability, and multitenant capability for actual-time search. Elasticsearch capabilities are available through JSON over a RESTful API. The looking abilities are sponsored via a schema-much less Apache Lucene Engine, which permits it to dynamically index facts without understanding the shape beforehand. Elasticsearch is able to gain speedy search responses as it makes use of indexing to search over the text.

Elasticsearch is used by many big companies, such as GitHub, SoundCloud, FourSquare, Netflix, and many others. Some of the use cases are as follows:

- **Wikipedia**: This uses Elasticsearch to provide a full text search, and provide functionalities, such as *search-as-you-type*, and *did-you-mean* suggestions.
- **The Guardian**: This uses Elasticsearch to process 40 million documents per day, provide real-time analytics of site-traffic across the organization, and help understand audience engagement better.
- **StumbleUpon**: This uses Elasticsearch to power intelligent searches across its platform and provide great recommendations to millions of customers.
- **SoundCloud**: This uses Elasticsearch to provide real-time search capabilities for millions of users across geographies.
- **GitHub**: This uses Elasticsearch to index over 8 million code repositories, and index multiple events across the platform, hence providing real-time search capabilities across it.

2) Logstash:

Logstash is a data pipeline that helps collect, parse, and analyze a large variety of structured and unstructured data and events generated across various systems. It provides plugins to connect to various types of input sources and platforms, and is designed to efficiently process logs, events, and unstructured data sources distribution into a variety of outputs with the use of its output plugins, namely file, stdout (as output on console running Logstash), or Elasticsearch.

It has the following key features:

- **Centralized data processing**: Logstash helps build a data pipeline that can centralize data processing. With the use of a variety of plugins for input and output, it can convert a lot of different input sources to a single common format.
- **Support for custom log formats**: Logs written by different applications often have particular formats specific to the application. Logstash helps parse and process custom formats on a large scale. It provides support to write your own filters for tokenization and also provides ready-to-use filters.
- **Plugin development**: Custom plugins can be developed and published, and there is a large variety of custom developed plugins already available.

3) Kibana:

Kibana is an open source Apache 2.0 licensed data visualization platform that helps in visualizing any kind of structured and unstructured data stored in Elasticsearch indexes. Kibana is entirely written in HTML and JavaScript. It uses the powerful search and indexing capabilities of Elasticsearch exposed through its RESTful API to display powerful graphics for the end users. From basic business intelligence to real-time debugging, Kibana plays its role through exposing data through beautiful histograms, geomaps, pie charts, graphs, tables, and so on. Kibana makes it easy to understand large volumes of data. Its simple browser-based interface enables you to quickly create and share dynamic dashboards that display changes to Elasticsearch queries in real time.

Some of the key features of Kibana are as follows:

- It provides flexible analytics and a visualization platform for business intelligence.
- It provides real-time analysis, summarization, charting, and debugging capabilities.
- It provides an intuitive and user friendly interface, which is highly customizable through some drag and drop features and alignments as and when needed. It allows saving the dashboard, and managing more than one dashboard. Dashboards can be easily shared and embedded within different systems.
- It allows sharing snapshots of logs that you have already searched through, and isolates multiple problem transactions.

There Are 3 modules :

1.Log files

2.ELK stack

3.Machine learning code

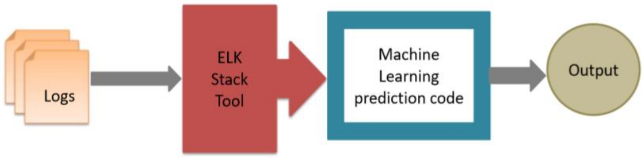


Figure 1: Modules

3. IMPLEMENTATION OF MODULES

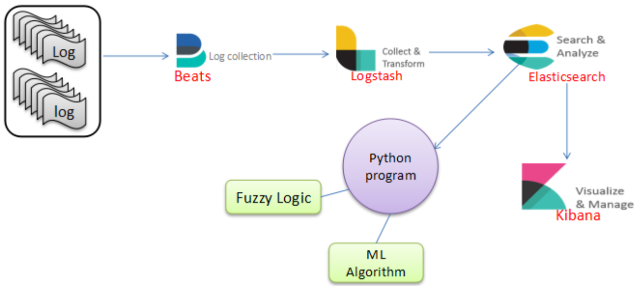


Figure 2: Flow of system

Log files: Here all log files generated from online exam for each student.
File Beats: Here file beat collect all log files from different sources and give to logstash.
Logstash: Logstash transform the collected log files into required format and send to Elasticsearch.
Elasticsearch: Elasticsearch take transformed logs from logstash then search and analyse the logs it generate JSON file we can import that file into python program or simply give to kibana.
Kibana: Kibana is a visualization tool so it accept the output of Elasticsearch and show some visualization according to our requirements.
Machine Learning Algorithms: we can import Elasticsearch output into python then apply some fuzzy logic to set into a format then this fuzzy logic output give to machine learning algorithms to find the anomaly from logs.

1. Support Vector Machine (SVM): It is a supervised mastering set of rules. It is used for each of the regression and classification problems. It has their personal way of imposing as differentiate with the other machine getting to know algorithms. It has potential to address many non-stop and categorical values. I actually have used linear kernel, this kernel is used for enforcing the SVM in python. Linear Kernel may be used as a dot product between two observations. The formula is given as -k(x,xi)=sum(x*xi)
From this above formula, we can say that the output of the 2 vectors say x & xi is the total of the multiplication of every pair of the input values.

2. k-means clustering:It is a strategy for vector quantization, initially from signal preparing, that expects to segment n perceptions into k groups in which every perception has a place with the group with the closest mean (bunch focuses or group centroid), filling in as a model of the group. This outcomes in an apportioning of the information space into Voronoi cells. It is popular for cluster analysis in data mining. k-means clustering minimizes within-cluster variances (squared Euclidean distances), however not customary Euclidean separations, which would be the more troublesome Weber issue: the mean streamlines squared mistakes, though just the geometric middle limits Euclidean separations. For example, better Euclidean arrangements can be discovered utilizing k-medians and k-medoids.

4. Results and Experiments

1. Comparative Result Analysis:

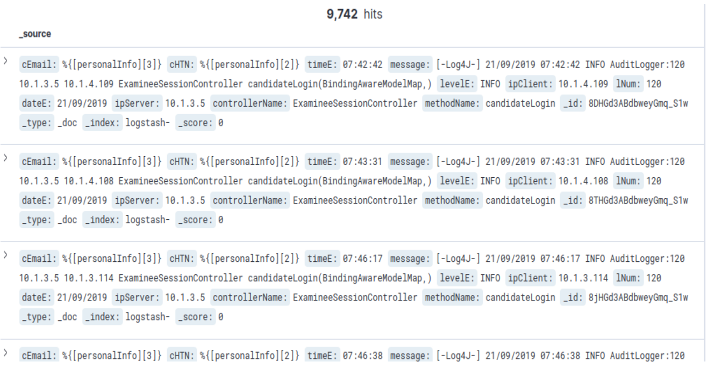
Algorithms	Train Accuracy	Test Accuracy
Kmeans	89.37	89.48
Linear Regression	83.471	85.615
SVM	86.628	87.288

Table 1: Accuracy comparison

So by comparing the accuracy based on the algorithms. So we can conclude that K means algorithm accuracy is good that other algorithms i.e 89%. Linear Regression algorithm have low accuracy than other algorithms and test accuracy of linear regression is better than train accuracy. Support Vector machine algorithm have better test accuracy that its train accuracy.

Output of Different Modules:

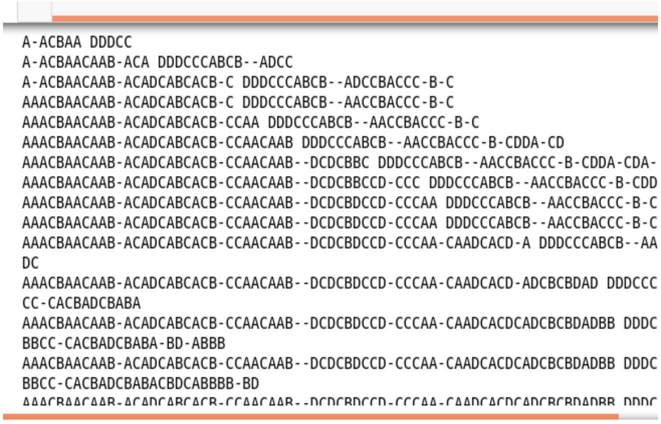
1. Kibana output:



```
9,742 hits
_source
> cEmail: %{[personalInfo][3]} cHTN: %{[personalInfo][2]} timeE: 07:42:42 message: [-Log4J-] 21/09/2019 07:42:42 INFO AuditLogger:120
  10.1.3.5 10.1.4.109 ExamineeSessionController candidateLogin(BindingAwareModelMap,) levelE: INFO ipClient: 10.1.4.109 lNum: 120
  dateE: 21/09/2019 ipServer: 10.1.3.5 controllerName: ExamineeSessionController methodName: candidateLogin _id: 8DHGd3ABdbweyGmq_S1w
  _type: _doc _index: logstash- _score: 0
> cEmail: %{[personalInfo][3]} cHTN: %{[personalInfo][2]} timeE: 07:43:31 message: [-Log4J-] 21/09/2019 07:43:31 INFO AuditLogger:120
  10.1.3.5 10.1.4.108 ExamineeSessionController candidateLogin(BindingAwareModelMap,) levelE: INFO ipClient: 10.1.4.108 lNum: 120
  dateE: 21/09/2019 ipServer: 10.1.3.5 controllerName: ExamineeSessionController methodName: candidateLogin _id: 8THGd3ABdbweyGmq_S1w
  _type: _doc _index: logstash- _score: 0
> cEmail: %{[personalInfo][3]} cHTN: %{[personalInfo][2]} timeE: 07:46:17 message: [-Log4J-] 21/09/2019 07:46:17 INFO AuditLogger:120
  10.1.3.5 10.1.3.114 ExamineeSessionController candidateLogin(BindingAwareModelMap,) levelE: INFO ipClient: 10.1.3.114 lNum: 120
  dateE: 21/09/2019 ipServer: 10.1.3.5 controllerName: ExamineeSessionController methodName: candidateLogin _id: 8jHGd3ABdbweyGmq_S1w
  _type: _doc _index: logstash- _score: 0
> cEmail: %{[personalInfo][3]} cHTN: %{[personalInfo][2]} timeE: 07:46:38 message: [-Log4J-] 21/09/2019 07:46:38 INFO AuditLogger:120
```

Figure 3: Kibana Output1

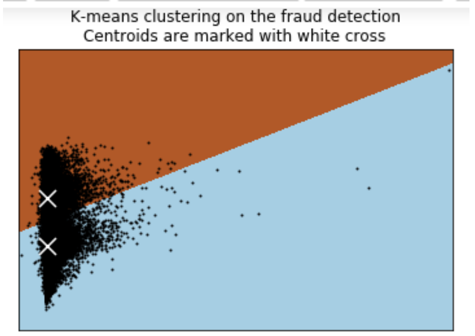
2. Fuzzy Logic:



```
A-ACBAA DDDCC
A-ACBAACAAB-ACA DDDCCCABCB--ADCC
A-ACBAACAAB-ACADCABCACB-C DDDCCCABCB--ADCCBACCC-B-C
AAACBAACAAB-ACADCABCACB-C DDDCCCABCB--AACCBACCC-B-C
AAACBAACAAB-ACADCABCACB-CCAA DDDCCCABCB--AACCBACCC-B-C
AAACBAACAAB-ACADCABCACB-CCAACAAB DDDCCCABCB--AACCBACCC-B-CDDA-CD
AAACBAACAAB-ACADCABCACB-CCAACAAB--DCDCBBC DDDCCCABCB--AACCBACCC-B-CDDA-CDA-
AAACBAACAAB-ACADCABCACB-CCAACAAB--DCDCBBCCD-CCC DDDCCCABCB--AACCBACCC-B-CDD
AAACBAACAAB-ACADCABCACB-CCAACAAB--DCDCBDCCD-CCCAA DDDCCCABCB--AACCBACCC-B-C
AAACBAACAAB-ACADCABCACB-CCAACAAB--DCDCBDCCD-CCCAA DDDCCCABCB--AACCBACCC-B-C
AAACBAACAAB-ACADCABCACB-CCAACAAB--DCDCBDCCD-CCCAA-CAADCACD-A DDDCCCABCB--AA
DC
AAACBAACAAB-ACADCABCACB-CCAACAAB--DCDCBDCCD-CCCAA-CAADCACD-ADCBCBDAD DDDCCC
CC-CACBADCBABA
AAACBAACAAB-ACADCABCACB-CCAACAAB--DCDCBDCCD-CCCAA-CAADCACDCADCBCBDADBB DDDC
BBCC-CACBADCBABA-BD-ABBB
AAACBAACAAB-ACADCABCACB-CCAACAAB--DCDCBDCCD-CCCAA-CAADCACDCADCBCBDADBB DDDC
BBCC-CACBADCBABACBDCABBBB-BD
AAACBAACAAB-ACADCABCACB-CCAACAAB--DCDCBDCCD-CCCAA-CAADCACDCADCBCBDADBB DDDC
```

Figure 4: Fuzzy Logic output

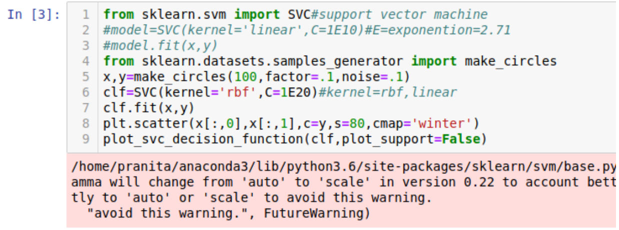
3. Kmeans output:



```
Accuracy 0.8948046006362582
```

Figure 5: k means output

4. SVM output:



```
In [3]:
from sklearn.svm import SVC#support vector machine
#model=SVC(kernel='linear',C=1E10)#E=exponention=2.71
#model.fit(x,y)
from sklearn.datasets.samples_generator import make_circles
x,y=make_circles(100,factor=.1,noise=.1)
clf=SVC(kernel='rbf',C=1E20)#kernel=rbf,linear
clf.fit(x,y)
plt.scatter(x[:,0],x[:,1],c=y,s=80,cmap='winter')
plot_svc_decision_function(clf,plot_support=False)
```

```
/home/pranita/anaconda3/lib/python3.6/site-packages/sklearn/svm/base.py
amma will change from 'auto' to 'scale' in version 0.22 to account bett
tly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
```

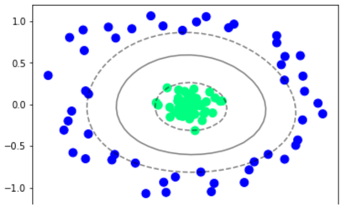
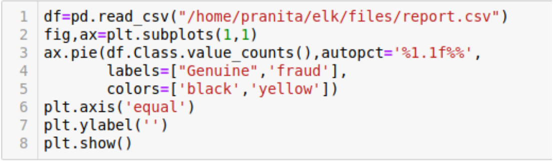


Figure 6: SVM output

5. final output:



```
df=pd.read_csv("/home/pranita/elk/files/report.csv")
fig,ax=plt.subplots(1,1)
ax.pie(df.Class.value_counts(),autopct='%1.1f%%',
        labels=["Genuine",'fraud'],
        colors=['black','yellow'])
plt.axis('equal')
plt.ylabel('')
plt.show()
```

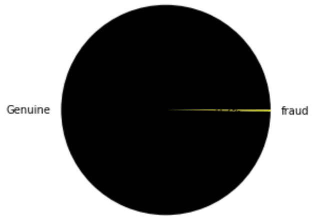


Figure 7: final output

5. CONCLUSION

We can conclude that k means gives better solution than SVM algorithm. There are many tools to analyse log but here we use ELK tool to export file in python. We can use this methods to detect the fraude in online exam or anomaly detection system.

REFERENCES

[1] M.Harikanth, Rajarajeswari. "Malicious Event Detection Using ELK Stack through Cyber Threat Intelligence." International Journal of Innovative Technology and Exploring Engineering (IJITEE), (2019).

[2] Sung Jun Son, Youngmi Kwon. "Performance of ELK Stack and Commercial System in Security Log Analysis." 2017 IEEE 13th Malaysia International Conference on Communications (MICC), (2017).

[3] Ibrahim Yahya Mohammed AL-Mahbashi, Dr. M. B. Potdar, Mr. Prashant Chauhan. "Network Security Enhancement through Effective Log Analysis Using ELK." IEEE 2017 International Conference on Computing Methodologies and Communication (ICCMC), (2017).

[4] JIA Zhanpei, SHEN Chao, YI Xiao, CHEN Yufei, YU Tianwen, GUAN Xiaohong. "Big-Data Analysis of Multi-Source Logs for Anomaly Detection on Network-based System." 2017 13th IEEE Conference on Automation Science and Engineering (CASE), (2017).

[5] Siwoon Son, Myeong-Seon Gil, and Yang-Sae Moon, "Anomaly Detection for Big Log Data Using a Hadoop Ecosystem." ©2017 IEEE.

[6] Juan Pablo Gila, Johnny Revecoa, Tzu- Chiang Shena, " Operational logs analysis at ALMA observatory based on ELK stack." Conference paper 2016.

[7] Online “Open Source Search & Analytics” elastic. Sandeep Kumar Dewangan, Shikha Pandey and Toran Verma. "A Distributed Framework for Event Log Analysis using Map Reduce.” International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), (2016).

[8] Jun Bai. "Feasibility Analysis of Big Log Data Real Time Search Based on Hbase and ElasticSearch." International Conference on Natural Computation (ICNC) (2013).

[9] Yan Liu, Wei Pan, Ning Cao, Guangwei Qiao. "System Anomaly Detection in Distributed Systems through MapReduce-Based Log Analysis." International Conference on Advanced Computer Theory and Engineering (ICACTE),(2010).

[10] Qiang FU, Jian-Guang LOU, Yi WANG, Jiang LI. "Execution Anomaly Detection in Distributed Systems through Unstructured Log Analysis." Conference Paper December(2009).

[11] Gaurav Kasliwal. “Cheating Detection in Online Examination ” Spring(5-21-2015).

[12] HarishBabu. Kalidasu, B.PrasannaKumar, Haripriya.P. "A Fraud Detection based Online Test and Behavior Identification Implementing Visualization Techniques." IJCSET(2012).

[13] Razan Bawarith, Dr. Abdullah Basuhail, Dr. Anas Fattouh and Prof. Dr. Shehab Gamalel-Din. "E-exam Cheating Detection System." (IJACSA) International Journal of Advanced Computer Science and Applications, (2017).

[14] Matus Korman, "Behavioral detection of cheating in online exam."(2012).

[15] Júlia Murínová, "Application of log analysis" Brno,(2015).

[16] Mengying Wang, Lele Xu, Lili Guo. "Anomaly Detection of System Logs Based on Natural Language Processing and Deep Learning", International Conference on Frontiers of Signal Processing(2018)

[17] Weixi Li, "Automatic Log Analysis using Machine Learning", Examensarbete 30 hp November(2013).

[18] ShilinHe, Jieming Zhu, Pinjia He, and Michael R.Lyu."Experience Report: System Log Analysis for Anomaly Detection", International Symposium on Software Reliability Engineering(2017).