Development of Cyber Threat Intelligence System in a SOC Environment for Real Time Environment

Aperame Varatharaj
Department of Information Systems Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
apesakthi@gmail.com

Prabath Lakmal Rupasinghe
Senior Lecturer: Department of Information Systems Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
lakmal.r@sliit.lk

Chethana Liyanapathirana
Assistant Lecturer: Department of Information Technology
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
chethana.l@sliit.lk

***Abstract*—Now a days, Information Communication Technology (ICT) plays an important role in the world. In IT, Cyber Security holds a vast place. Cyber Threat Intelligence (CTI) leads the significant place within Cyber Security, as many Cyber Threats need to be faced every day by a particular organization. Security Operation Center (SOC) helps to monitor and analyze an organization's security position in Real Time. This paper proposes about the Cyber Threat Intelligence framework in a SOC Environment in Real Time. The proposed framework contains of three layers, which are built above Security Onion. The Layer 1 comprises of input data from online and offline sources. In Layer 2, implemented two components namely Filter data and Cut down data, which receive the data from Layer 1. Finally, in Layer 3 delivers a detailed report. As the input for the Layer 1, Financial Datasets is used. These Financial Datasets, which helps in order to detect the Financial Frauds. Machine Learning is used to train the model. By implementing CTI System in an organization, it helps to gain predictive output regarding the upcoming threats. Also, it helps to ensure the reputation of an organization by establishing trust between the users. Helps to increase the number of customers to an organization. The above are the advantages gained by a particular organization by having a CTI System.**

Keywords*—Information Communication Technology . Cyber Security . Cyber Threat Intelligence . Security Operation Center . Security Onion . Elastic search-Logstash-Kibana . Machine Learning

I. INTRODUCTION

Now a days, ICT (Information Communication Technology) holds an important role in the revolution of the globe. Due to this situation, the evolution of the technologies accompanies a huge space, which helps to build the "Global Village". As a result, the people can come to know all the matters like Weather Forecast, Lifestyle, Science and Technology, Health and Medicine, Telecommunication, Agriculture, Education, Architecture, Aeronautical as well as Research Activities which are the greatest contributions of ICT. With the combination of Inventions and Technologies, it leads a way to make room for Cyber Threats. Therefore, Data and Information are very much important for the development. In this, some data can be malicious and some are not. So, it is the duty of humans to correctly identify the malicious as well as needed data. It is very significant to know to separate some malicious evidences and not. For this process the Cyber Threat Intelligence (CTI), takes an important role for the development.

Cyber Threat Intelligence (CTI) is data together and analysed by an organization so that to recognise a cyber-threat's intentions and attack behaviours. But, most of the organizations today are mainly focused only on the elementary applications, like firewalls, incorporating intelligence boost with prevailing IPS (Intrusion Prevention System) and SIEMs (Security Information and Event Management System). With the help of CTI System, the organizations can come to identify the opponent's subsequent activities. Thus, the company can energetically can defence themselves for future attacks.

In this paper, I have proposed a comprehensive system, with threat intelligence in a SOC (Security Operation Centre) environment, for Real Time. The SOC plays the major important role, in most of the organizations by providing services like managing the vulnerability, threat intelligence, digital investigation and collection and analysis of data. The data collection to the real time environment is mainly done through open source and non-security-focused sources. Then, the collected data enters the SOC and examined with the help of threat intelligence and finally inform to the end-users as a report whether there are any risks and threats available or not.

In this paper, I have got Financial Dataset to detect the Fraud activities. Also, I have proposed cyber threat intelligence framework with three layers. Layer 1 is responsible in order to input data from online and offline sources. In Layer 2, implemented two components namely Filter data and Cut down data, which receives the data from Layer 1. Finally, in Layer 3 delivers a detailed report with the help of Elastic search-Logstash-Kibana (ELK) stack [2]. These three layers are built above the platform, which is known as Security Onion. In order to identify the malicious behaviour, and to predict the behaviour of malicious accounts, support vector machine (SVM) is used [1]. Moreover, to train the Machine Learning models, in order to detect the anomalies, many different types of algorithms are used.

The rest of the paper is organized as follows: Section 2 describes the Literature Survey. Section 3 propose a system model design. Section 4 of this paper describes the methodology that is being implemented. Section 5 offers a general introduction about tools and techniques used for the development of cyber threat intelligence. Section 6 presents results and findings which are based on essential parameters needed for cyber threat intelligence framework. The conclusion and the future work of the whole paper is presented in Section 7.

II. Literature Survey

I performed a Literature survey on existing platforms with nearly similar capabilities and functionalities for my research project. Some of the eminent researches are reviewed here.

[1] Along with the ever growing computer network applications, cyber threat is becoming exaggerated as the IoTs' (Internet of Things) trending, opens up new dimensions for the cyber attackers. Novel interdependence of the nodes between SIoT (Social Internet of Things) and IoT lays out more chances or pathways for the network attackers. This paper discusses about the inter relationship between the users and the SIoT users, proposing SIoT accounts/ malevolent conducts assumption based on threat intelligence malicious accounts of SIoT gains the malevolent account conduct in social networks. The malicious behaviour predictions model of SIoT was designed to six modules and the institution of these six key modules of the SIoT malicious behaviour prediction models/ working principals follows as: data pre-processing, SVM (Support Vector Machine) trainer and classifier construction. Through SVM classifier, related threat intelligence based data is gathered to identify the malicious behaviours in the exact mode, in the SIoT environment. 16,000 experimental samples were used and collected data translated into numerical format which were used for training the SVM classifier. Considering the features of traditional social networks/ account malicious behaviour and SIoT account features, analysing and building of the feature vector of malicious behaviour on SIoT account is done. Though SVM capacitated the solving of the short comings done by other machine learning, room for improvement in classification accuracy does exist. This paper is relevant to my project because it talks about data inputs.

[2] New reports show that malware attacks have improved and increased in recent years. From ransomware, adware, botnets and such malware has grown into attacking hospital systems and control systems of Supervisory Control and Data Acquisition (SCADA). Currently assessing the effectiveness of CTI feeds are not possible in the capacity of the available CTI (Cyber Threat Intelligence) systems. New CTI framework made of three layers has been germinated in this paper. Layer 1 is the input layer. This layer collects all the data together and categorizes them into malevolent and non-malevolent data and stores them. Layer 2 supervises the characteristics of the threats given from layer 1. Layer 3 consists of threat management, reports of the steps and the defence against them. The methodology used in this layered framework is configuration, data collection, pre-processing, classification and filtering, and report generation. This paper is relevant as it proposes a framework which can distinguish malicious and non-malicious data from within given datasets.

[3] Presently, APT (Advanced Persistent Threat) attacks have become a major concern as cyber-attacks/ speedy growth surpasses the defence mechanism. To act on the defence promptly, active defence system is gradually formed. It ignites the immerging of a lot of security data. With this trend, the network threat intelligence has taken the form of a hotspot in the field. Based on the deep analysis of CIF (Collective Intelligence Framework), this paper suggests a system to classify valuable information amongst the magnanimous network threat intelligence as CIF is capable of joining established malevolent information together from various sources and connect process to create new threat intelligence. Data processing method was designed under data pre-processing, collecting Observable Collections based on content extraction and evaluation system provider. This paper talks about each module and data setting of CIF, based on magnanimous source of threat intelligence created by CIF. This effort was to find a basis for the real time analysis of the suggested threat intelligence. As the time of extracting data is limited, it's not complete in analysing the popular time dimension change.

[4] The modern threat landscape proves that attackers are becoming bigger and better and it's hard to stop and defend against these attacks. One of the reasons for this might be, the non-existent knowledge of CTI (Cyber Threat Intelligence) within the community and quality and procedures used in threat information sharing. To improve the intelligence on CTI systems, first data on the most frequent keyword searches relating to CTIs and threats were analysed. Second, comparing English written data on these topics and assessing them. Third is to make sure only appropriate sources were used for assessments, the papers discovered from the search processes were analysed against different criteria. By analysing all these literature data, it can be concluded that there is no definite definition for CTI. However, some key features of CTI can be obtained from existing definitions. One is context, which permits security analysts to assess threats and help them generate a defence plan. Another one is element. CTI consists of three main elements which are relevant, time lines and actionable. These elements also help security analysts gain relevant data and process them in a short amount of time so they can gain dynamic knowledge in order to make a suitable decision. This paper is relevant to my project because it offers information on security against attacks which can be done in a relatively short amount of time.

[5] Cyber threat hunting is a process to identify threats from alerts created by IDS (Intrusion Detection Systems) in united networks it protects important assets. Honeypots are one of the examples of cyber threat hunting, especially HoneyC. Honeypots are more resourceful than any other technologies as it delivers more sources to assess for recognition of cyber-attacks. As honeypots are carriers of large amount of data, it's difficult to analyse them. In order to analyse such massive datasets, elastic search technology is used. Cyber threat analysis is the main factor to threat hunting. Information of threats can be obtained by Honeypots and assessed in order to understand threats about to happen. Such analysis has shown that attacks are not zero-day and they are consistent and have sequences. To identify these sequences ELK (Elasticsearch, Logstash and Kibana) has been introduced. The ELK stack helps to showcase data, generate visualization and a dashboard for data of any size. The ELK is useful because it can read any amount of data as it contains elastic search and it can operate like any other database. This paper is very relevant to my project as it gives me a way to compare sequences of data and how to prevent threats before they occur.

[6] Currently available security controls are not enough to defend against the attacks which are evolving constantly made by the ever growing knowledge of the attackers. Cyber security's main resource is CTI (Cyber Threat Intelligence). CTI helps enhance the already existing security controls by data gained from multiple feeds. Machine learning domains are using the trending deep learning algorithms nowadays. To compare the diligence of a Convolutional Neural Network

against conventional methods like SVM (Support Vector Machine) performances of several machine learning ways in identifying the places of hacker forum posts which may consist of useful CTI are examined against each other. These experiments used GPU as it's cheap and high performing. Through this experiment, it's proven that existing classifiers like SVM acts just as well as CNN (Convolutional Neural Networks).

This paper is relevant to my project because it showcases the methods used in machine learning and its components.

III. System Model Design

The Datasets which I used in this paper are mainly Financed data. Then the collected data goes inside the Security Operation Center (SOC) where all the process get monitored. The main advantage of SOC is analysing, detecting and responding to the cybersecurity incidents.

The SOC provide services like Data collection and analysis, Threat Intelligence, Digital Investigation and Vulnerability Management. In this, one of the services that I am going to implement is Cyber Threat Intelligence (CTI) System.

Towards identifying the malicious behaviour of threats, they are separated as Behaviour Data Acquisition Module, Threat Intelligence Collection Module, Data pre-processing Module, Support Vector Machine (SVM) Training Module, Support Vector Machine (SVM) Classification Module and Behaviour Prediction Result Output module. The block diagram of the module is shown in Fig. 1.

- The Behaviour Data Acquisition Module – It is responsible in analysing the behaviour of data, whether is it spam / not spam, the current level of approval, the change of public affiliation and the occurrence of content inspection.
- Threat Intelligence Collection Module – It is in charge, to collect the threat from the SOC Environment in Real Time Environment.
- Data Pre-processing Module – In this, the collected data from these two modules (Namely Behaviour Data Acquisition Module, Threat Intelligence Collection Module) are constant with the behavioural features, but the data structure and the format are not the same. So, in order to classify the data format in a unified manner this data pre-processing module is used.
- Filter the Collected Data - The collected data get filtered with the application software, which is developed on the platform known as Security Onion.
- Dependency Checker – It helps to solve the interoperability issues between the threat sharing peers.
- SVM Training Module – This module, gives the classified information. Each dataset is trained and a skilled classifier is used for the harmful behaviour prediction.
- SVM Classification Module – It is a basic portion of the entire system, where it classifies the malicious behaviour datasets.
- Behaviour Prediction Result Output Module – It is in charge for to output the harmful behaviour prediction results.
- Finally, the report is generated with the help of Kibana software.

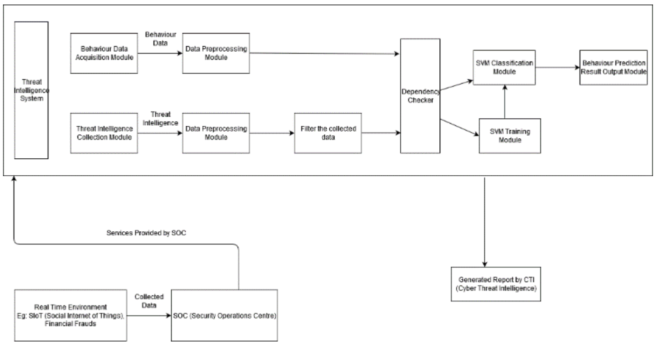

Threat Intelligence System
Behaviour Data Acquisition Module
Behaviour Data
Data Preprocessing Module
Threat Intelligence Collection Module
Threat Intelligence
Data Preprocessing Module
Filter the collected data
Dependency Checker
SVM Classification Module
SVM Training Module
Behaviour Prediction Result Output Module
Services Provided by SOC
Generated Report by CTI (Cyber Threat Intelligence)
Real Time Environment Eg: SIoT (Social Internet of Things), Financial Frauds
Collected Data
SOC (Security Operations Centre)

Fig. 1. System Model Design

IV. Methodology

For this development of the proposed system, a free and open source Linux distribution platform, Security Onion is used in order to identify the Intrusion Detection. Then, the simulated data/ data from outside, are input into an open source application software which is used to filter the data and as a next step, it helps to cut down the unnecessary data from outside. The risks/ threats help organization to collaborate and helps to share the information seamlessly, which are related to privacy management. The Financial datasets which are needed as input for this proposed system are gained from Kaggle. After the initial filtering and per processing, machine learning base classification models will be used to predict the threat level which are associated to the defined system. These will be developed with the help of the Deep Learning Algorithm, which is based in Machine Learning.

This proposed system, comprises in a layered approach. The results gained as the output of each layer feeds to the next layer. The methodology used is divided into four steps.

Step 1: Configuration: Sets of hardware as well as the software is used for each layer.
Step 2: Data collection: This is an individual module, where it doesn't depend with any other modules. Mainly the datasets are belonging to Financial Datasets. The Financial Datasets are mainly derived from Kaggle.
Step 3: Filter the data and Cut down the unnecessary data from outside. Step 3 is mainly based on the Machine Learning Algorithms such as Support Vector Machine (SVM), Logistic Regression, Random Forest Classifier etc. The datasets are divided as train and test dataset respectively, in order to train the Machine Learning model to achieve highest accuracy rate.
Step 4: Report generation: In this final step, the report is generated in real time with alerts, notifications, types of malware detected as graphical representation for the ease of users. Kibana is used for the generation of reports.

A. Data Acquisition by Financial Datasets for the Cyber Threat Intelligence (CTI) System

a. Obtain the Dataset

Obtaining the Financial Dataset is the major important part in my research. Because, this process helps to get the accurate Machine Learning model. The Financial Dataset was gained from Kaggle. In order to detect the fraud detection activities, this dataset greatly helped. More the amount of Datasets gained, the Machine Learning model accuracy rate will be increased.

The Financial Dataset which I gained from Kaggle, consists of 31 features, 28 of which have been anonymized and are labelled from V1 to V28. The remaining 3 features are the time, amount of the transaction as well as whether that transaction was fraudulent or not. Furthermore, there are no missing values in the dataset. The Financial Dataset, which I obtained contains 284,807 transactions.

All the following parts have been done with the help of the Machine Learning Algorithms. Jupyter Notebook is used for to train the Machine Learning Model. Also, Python scripts are used in order to develop the Machine Learning model.

b. Data Pre-processing Phase

As all the analysts are anonymized, I decided to focus on non-analysts predictors. Namely, time and amount of the transaction, which are under the Exploratory Data Analysis (EDA). In this transactions, 99.83% of the transactions are non-fraudulent, as well as 0.17% are fraudulent. The following Fig. 2 proves that.

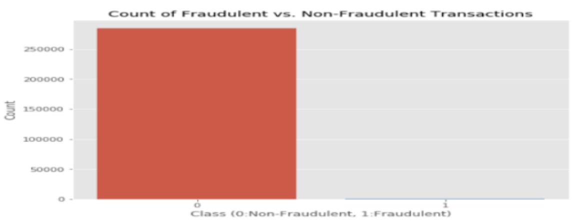


Fig. 2. Fraudulent vs. Non-Fraudulent Transactions

c. Creating a Training Set

To achieve the high performance, random under-sampling method is used in order to train the dataset with the balanced class distribution, which helps to detect the fraudulent transactions.

In order to create the balance training dataset, took all the fraudulent transactions and counted them. Then, randomly selected the same number of non-fraudulent transactions and joined both of them. Then, I again visualized the difference and the Fig. 3 shows the class distributions. Also, in this all the valid transactions are considered as 0 and Fraudulent transaction are considered as 1.

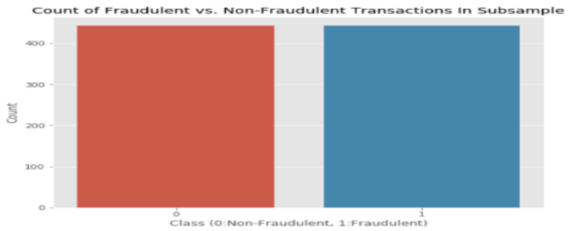


Fig. 3. Class Distributions

d. Outlier Detection and Removal

Box and Whisker plot provides about all the transactions and 1.5 times the Inter-Quartile Range (IQR) are normally considered to be outliers. But, when removing all the transactions tend to decrease the training data size. So, I decided to mainly focus on extreme outliers which are situated outside of 2.5 times the Inter-Quartile Range (IQR). The following Fig. 4 depicts the Box and Whisker plot with high negative correlation, as well as the Fig. 5 depicts with high positive correlation.

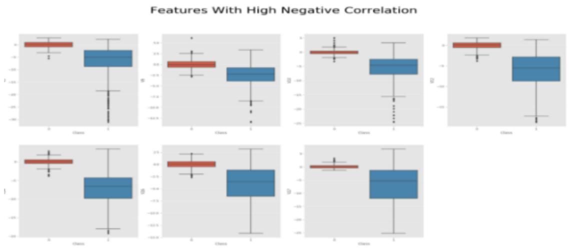


Fig. 4. Negative Correlation

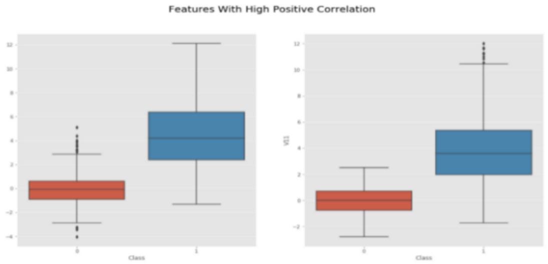


Fig. 5. Positive Correlation

e. Dimensionality Reduction with T-distributed Stochastic Neighbor Embedding (t-SNE)

T-distributed Stochastic Neighbor Embedding (t-SNE) is a machine learning algorithm which is used for visualization. It is used to embed high-dimensional data for visualization with the low-dimensional space of two or three dimensions. t-SNE plots often seems to display as clusters. So, I plotted a scatter plot to show the fraudulent and non-fraudulent transactions, which is shown in Fig. 6.

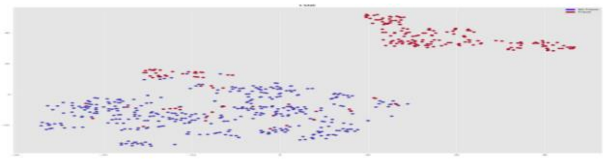


Fig. 6. t-SNE plots

f. Classification of Algorithms with Machine Learning

In order to test the performance of the algorithms, I split the data as 80/20 for train-test. To avoid the over fitting, I used the resampling technique known as k-fold cross-validation. Then, repeated the same process for every single fold and finally averaged the resulting predictions.

In order to get the best output from the given data, I spot-checked with most of the popular classification algorithms. They are as follows.

- Logistic Regression (LR)
- Linear Discriminant Analysis (LDA)
- K-Nearest Neighbors (KNN)
- Classification and Regression Tree (CART)
- Support Vector Machine (SVM)
- Random Forest Classifier (RF)

The following Fig. 7 shows the results gained in this spot-checking.

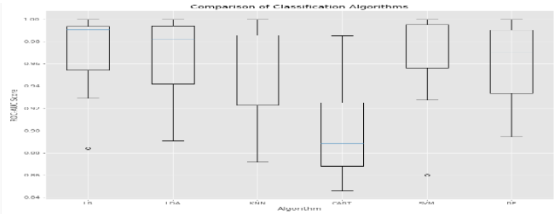
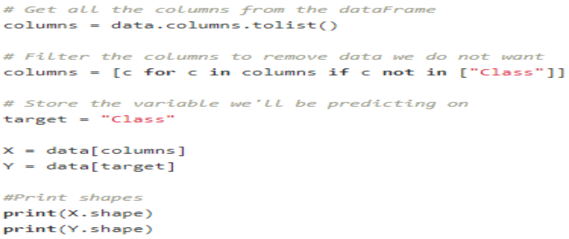


Fig. 7. Comparison of Classification Algorithms

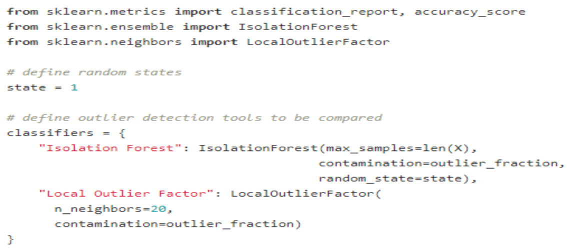
Also, for the Fraud Detection purposes I implemented two components. Namely, Filter Data and Cut down data. So, these above mentioned two components also implemented above the platform named as Security Onion. The following Fig. 8 shows the Data filtering with the help of the Machine Learning Algorithms.



```
# Get all the columns from the dataFrame
columns = data.columns.tolist()

# Filter the columns to remove data we do not want
columns = [c for c in columns if c not in ["Class"]]

# Store the variable we'll be predicting on
target = "Class"

X = data[columns]
Y = data[target]

#Print shapes
print(X.shape)
print(Y.shape)
```

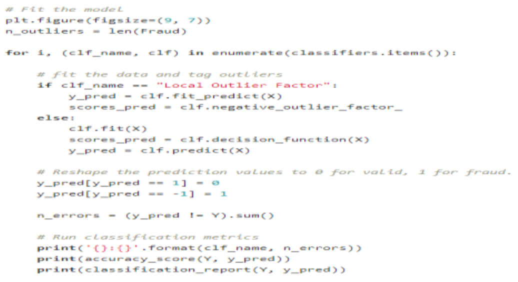
Fig. 8. Filter the Data

The following Fig. 9 and Fig. 10 shows the Cut down the unnecessary data from the gained Dataset, with Machine Learning algorithms like Isolation Forest and Local Outlier Factor (LOF) Algorithm.



```
from sklearn.metrics import classification_report, accuracy_score
from sklearn.ensemble import IsolationForest
from sklearn.neighbors import LocalOutlierFactor

# define random states
state = 1

# define outlier detection tools to be compared
classifiers = {
    "Isolation Forest": IsolationForest(max_samples=len(X),
                                        contamination=outlier_fraction,
                                        random_state=state),
    "Local Outlier Factor": LocalOutlierFactor(
      n_neighbors=20,
      contamination=outlier_fraction)
}
```

Fig. 9. Define the outliers



```
# Fit the model
plt.figure(figsize=(9, 7))
n_outliers = len(Fraud)

for i, (clf_name, clf) in enumerate(classifiers.items()):

    # fit the data and tag outliers
    if clf_name == "Local Outlier Factor":
        y_pred = clf.fit_predict(X)
        scores_pred = clf.negative_outlier_factor_
    else:
        clf.fit(X)
        scores_pred = clf.decision_function(X)
        y_pred = clf.predict(X)

    # Reshape the prediction values to 0 for valid, 1 for fraud.
    y_pred[y_pred == 1] = 0
    y_pred[y_pred == -1] = 1

    n_errors = (y_pred != Y).sum()

    # Run classification metrics
    print('{}:{}'.format(clf_name, n_errors))
    print(accuracy_score(Y, y_pred))
    print(classification_report(Y, y_pred))
```

Fig. 10. Cut down the unnecessary data

g. Report Generation

This is the final step. As the result, the gained Cyber threats for an organization are shown to the end user as the graphical representation as reports. So, I generated the report with the help of the software, Elastic search-Logstash-Kibana (ELK) stack. This report is generated in Real Time in order to produce with alerts, notifications and types of malware detected for the end-users.

Elasticsearch is a search and analytics engine. Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch. Kibana lets users visualize data with charts and graphs in Elasticsearch. Elastic search-Logstash-Kibana (ELK) allow users to take the data as any format, and to search, analyze and finally helps to visualize the data in real time.

The following Fig. 11 and Fig. 12 shows the interface of Kibana [2].

Fig. 11. Kibana Dashboard I

Fig. 12. Kibana Dashboard II for Real Time

V. TOOLS IN CYBER THREAT INTELLIGENCE

From organization and security professionals now a days the Cyber Threat Intelligence (CTI) system holds an important place. Without the threat intelligence tools, the threat data becomes uncontrollable. Due to this presently there are lots of tools available for Cyber Threat Intelligence (CTI).

The Oracle VM Virtual Box plays the role of Virtual Machine. Inside the Oracle VM Virtual Box the platform named as Security Onion is installed.

The Security Onion is a free and open source Linux Distribution, which is used as the platform in order to implement the Cyber Threat Intelligence (CTI) system.

Elastic search-Logstash-Kibana (ELK) stack is used as the application software, to detect threat detection, visibility and incident response for the real time. Elasticsearch is a search and analytics engine. Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch. Kibana lets users visualize data with charts and graphs in Elasticsearch. Elastic search-Logstash-Kibana (ELK) allow users to take the data as any format, and to search, analyze and finally helps to visualize the data in real time.

Kibana Dashboard provides wide visibility and it enables interactive threat hunting. This also helps to generate report in order to articulate to the end users about the malicious behaviour of threats in real time.

Machine Learning Algorithms are used to train the Machine Learning Models in order to detect the anomalies.

In Machine Learning, Shallow as well as Deep Learning methods are used for the detection of anomalies. Also, to Filter the data and to Cut Down the unnecessary data from outside, for the implementations the Machine Learning models are being used. Jupyter Notebook is used for to train the Machine Learning models. Python Scripts are used to develop the Machine Learning Models.

VI. RESULTS & FINDINGS

To get the optimized trained model, the Machine Learning models are tuned with multiple Machine Learning Algorithms. Further to train the Machine Learning Models, Shallow and Deep Learning models are also used. The table (I) shows the accuracy level obtained for the Machine Learning models.

TABLE I. ACCURACY LEVELS GAINED

No.	Machine Learning Algorithm	Accuracy Level Gained
1	Logistic Regression (LR)	97.0%
2	Linear Discriminant Analysis (LDA)	96.6%
3	K-Nearest Neighbors (KNN)	95.0%
4	Classification and Regression Tree (CART)	90.0%
5	Support Vector Machine (SVM)	97.0%
6	Random Forest Classifier (RF)	96.0%
7	Isolation Forest	99.8%
8	Local Outlier Factor (LOF) Algorithm	99.7%

VII. CONCLUSION & FUTURE WORK

In Information Communication Technology (ICT), Cyber Security (CS) holds a very important place. As the popularity, the Cyber Threat Intelligence (CTI) takes an enormous place. From the beginning of Information Technology (IT), there were many threats developed with evolution. In the Digital world also, these threads are immensely developed.

According to the security researcher’s suggestion, Cyber Threats are having rapid incensement per seconds in the digital world. Due to the rapid growth of threats, CTI is required for a SOC (Security Operation Center) Environment in Real Time. In this paper, I demonstrated CTI system which is made with the help of Machine Learning Algorithms. Furthermore, the dataset I gained are Financial Dataset, which I used to input into the CTI system. Moreover, trained the Machine Learning models with the help of Machine Learning Algorithms in order to implement components like Filter data and Cut down the data. Security Onion is used as the platform in order to implement the components. Finally, the dashboard is generated as the report as shown in Fig. 13.

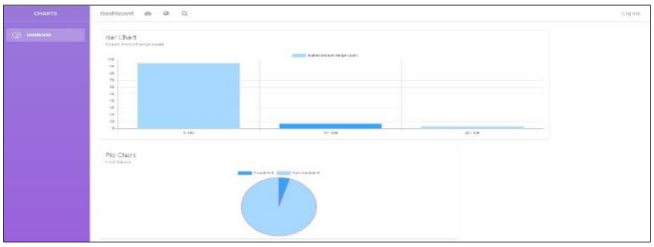


Fig. 13. Dashboard End Results to the end - user

The road ahead includes exploring more about CTI. Specifically, I plan to gather more datasets in the field of Finance in the near future, with more developed and enhanced datasets. With the help of the gained materials in the future, I planned to do research. Furthermore, with the help of this I also planned to give predictive future conclusions regarding Cyber Threats with the help of CTI System. I confirm that this developed CTI System will be more helpful in the present world as well as in future.

REFERENCES

[1] Zhang, H., Yi, Y., Wang, J., Cao, N. and Duan, Q., 2020. *Network Attack Prediction Method Based On Threat Intelligence For Iot*. [online] Springer. Available at: <https://link.springer.com/article/10.1007/s11042-018-7005-2> [Accessed 20 September 2020].

[2] Keim, Y. and Mohapatra, A., 2019. *Cyber Threat Intelligence Framework Using Advanced Malware Forensics*. [online] Springer. Available at: <https://link.springer.com/article/10.1007/s41870-019-00280-3> [Accessed 20 September 2020].

[3] Liu, R., Zhao, Z., Sun, C., Yang, X., Gong, X. and Zhang, J., 2017. *A Research And Analysis Method Of Open Source Threat Intelligence Data*. [online] Springer. Available at: <https://link.springer.com/chapter/10.1007/978-981-10-6385-5_30> [Accessed 20 September 2020].

[4] S, S. and Y, R., 2018. *(PDF) Cyber Threat Intelligence – Issue And Challenges*. [online] ResearchGate. Available at: <https://www.researchgate.net/publication/322939485_Cyber_threat_i ntelligence_-_Issue_and_challenges.> [Accessed 20 September 2020].

[5] Almohannadi, H., Awan, I., Al Hamar, J., Cullen, A., Disso, J. and Armitage, L., 2018. *Cyber Threat Intelligence From Honeypot Data Using Elasticsearch - IEEE Conference Publication*. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/document/8432334.> [Accessed 20 September 2020].

[6] Deliu, I., Leichter, C. and Franke, K., 2017. *Extracting Cyber Threat Intelligence From Hacker Forums: Support Vector Machines Versus Convolutional Neural Networks*. [online] ResearchGate. Available at: <https://www.researchgate.net/publication/322515698_Extracting_cyber_threat_intelligence_from_hacker_forums_Support_vector_machines_versus_convolutional_neural_networks> [Accessed 20 September 2020].