National College of Ireland

Efficient log analysis using advanced detection and filtering techniques

MSc in Cybersecurity

Suyash Sharma
Student ID: X17170681

School of Computing
National College of Ireland

Supervisor: Ross Spelman

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Suyash Sharma
Student ID:	X17170681
Programme:	MSc in Cybersecurity
Year:	2019
Module:	
Supervisor:	Ross Spelman
Submission Due Date:	12/08/2019
Project Title:	Efficient log analysis using advanced detection and filtering techniques
Word Count:	5933
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	9th August 2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Efficient log analysis using advanced detection and filtering techniques

Suyash Sharma
X17170681

Abstract

Digital crimes are increasing exponentially and people with possession of even a simple digital device, can facilitate a cyber attack. As the adoption of digital technologies and devices grows, it will be of utmost importance for digital investigators to develop a well thought strategy to analyze the raw binary data obtained from digital media. Log analysis provides useful way for alerting, monitoring, security and compliance, auditing , incident response and forensic investigations. The research aims at developing an efficient log parsing module based on machine learning to detect anomalies in data. Isolation forest algorithm and outlier detection methods are employed in the log parsing module for finding anomalous transaction for this case. This approach increases fraud detection rates and minimizes false alarms rates. Accuracy of Isolation Forest was found to be highly efficient and thus can be used in digital investigations to detect fraud transactions and anomalies.

Keywords: digital investigations,log parser, super timeline analysis , machine learning , Isolation Forest, ,Local Outlier Factor, visualization, information security

1 Introduction

With advances in development of electronic devices and digital storage devices, there is a challenge for digital investigators to process events and data and events during analysis. The methods of analyzing timeline gives an in-depth summary of events that took place during the time of the incident. The process of evaluating and analysing the artifacts(data) to obtain key pieces of information without the utilization of effective tools is a huge and time consuming task. One of the effective tools for obtaining timelines is Log2timeline that provides a framework for analyzing logs and developing timelines. Different log files can be parsed that are present on the suspects system and a series of timeline relating to the events produced by the system is obtained which can be logged and analyzed by digital investigators during analysis Inglot et al. (2012). The directories and data structures is also parsed automatically without manual intervention by this tool. Log2timeline supports numerous input formats but one of the main drawbacks is that it produces a lot of output data while analyzing logs which are nonessential during an analysis. With an increase in amount of inputs, the corresponding number of associated events increase in parallel, thus parsing of data and analyzing logs becomes complex. Therefore an effective log parsing module is needed to overcome these drawbacks Esposito and Peterson (2013).

How efficiently can a Machine learning based log parsing module be used to analyze digital artifacts(data) with low levels of false positives.
The research aims to develop a log parsing module for the Log2timeline tool that can be utilized for gathering evidence and reduce the Time to evidence Inglot and Liu (2014) factor which is crucial. Major issue faced in forensics is the enormous amount of data arising from disk images, memory dumps and traces left on the network that needs to be analyzed. The key aspects of any computer security tool areInglot et al. (2012):-

- Minimal intervention by the investigators.
- The amount of false alerts generated should be minimum and in this research, machine learning methods are used to efficiently detect anomalies.

Large volumes of data is involved in analyzing logs. Even Startups generate gigabytes of data every day and large companies like Facebook log terabytes of log data daily.(During 2009, Facebook already started logging nearly 25TB on a daily basis)[1]. The way to analyze massive amounts of data is by leveraging machine learning in interpreting log data in an automated manner. Investigations pertaining to logs of chat records needs reviewing thousands of messages to gather evidence and evaluating them individually is very time consuming. In this research an log parsing module for anomaly detection is developed using the Isolation Forest algorithm. Anomaly detection based on logs Lin et al. (2016) has gained much importance in the industry and academia as well. For standalone systems, developers need to check logs or script rules for detecting anomalies manually. The goal is to identify anomalies present in log files using a module developed using machine learning that is trained to detect malicious events. We have implemented and reviewed anomaly detection methods such as supervised methods i.e SVM and two unsupervised methods i.e Isolation Forest and Outlier detection. The results are evaluated based on precision (percentage of the number of real anomalies that are found to be correct), recall(percentage of the number of actual anomalies that are identified), and efficiency (the amount of time processing over various log volumes).Though data is limited, the results from the findings can be used as a baseline in future development.

This research aims analyze the below mentioned points:-

- An overview of commonly-used methods of detecting anomalies built on the basis of automated log analysis.
- An evaluation is made benchmarking the efficiency and effectiveness of the current detection methods of anomaly.

2 Literature Review

This section highlights an overview of the literature's presenting applications on which machine learning can be implemented in the field of log analysis which are beneficial for digital investigations are mentioned. Machine learning is utilized on a large scale in the field of digital investigations and behavioural forensics to analyze enormous data for detecting anomalous behaviour. It enables digital investigators to scan diverse data sets for modelling, profiling and predicting. The advantages include early detection of

[1]https://www.sumologic.com/blog/machine-learning-log-analysis/

signatures relating to intrusion attempts, hacking and fraud activities. The applications of machine learning, analyzing logs and the role of timeline analysis are listed in this section which aids in digital analysis.

2.1 Fraud Detection

A study was conducted by Awoyemi et al. (2017) in which they utilized Machine Learning strategies using K-nearest neighbour, logistic regression and nave bayes for digital investigations using the fraud credit card transactions dataset. The dataset consisted of European cardholders information containing 284,807 transactions. A hybrid technique of oversampling and under sampling of the dataset was carried out. The three methods were conducted on the pre-processed information. The work was executed in Python. Parameters such as exactness, accuracy and Matthews relationship coefficient were assessed, taking into consideration the rate of classification and performance. The classifiers k-nearest means, logistic regression and nave bayes classifiers showed as accuracy of 97.69%, 54.86% and 97.92% respectively. The relative outcomes demonstrate that k-closest neighbor performs superior to nave bayes and the technique of logistic regression. The Three classifiers corresponding to various Machine learning methods (Logistic Regression, Nave Bayes and K-closest neighbors) are prepared on genuine of Mastercard exchange information and their executions on Visa extortion location assessed and other metrics relevant for analysis. The very imbalanced dataset is inspected in a hybrid approach where the positive class is oversampled and the negative class under-sampled, accomplishing distributions of two datasets. The efficiency of the three classifiers are analyzed on the two sets of information utilizing Matthews Correlation coefficient measurements, rate of balanced classification, precision and accuracy.

2.2 Log analysis

Log analysis is used to enhance capabilities of software systems in a wide variety of aspects such as failure diagnostic, verification of program, performance detection and detecting anomalies. Methods of log analysis consists of log mining and parsing. Log parsing methods such as LogSig, LKE, IPLOM and SLCT were evaluated by which system source code was not required. An offline method of log parsing was developed that made use of linear processing of space and time. He et al. (2016) developed a technique of log parsing using system source code for log parsing. For the purpose of mining logs, methods such as detecting anomalies using PCA dimensionality in which a matrix generated from logs is provided as input. System logs were used for describing run time behavior of system. In all the mentioned research works, log analysis was utilized for problem solving and in our research anomaly detection based on logs is developed.

2.3 Insider threat detection

A machine learning module for identifying insider threats based on big data was developed by Mayhew et al. (2015). The developed module known as Behavior based access control(BBAC) to determine an actors trustworthiness i.e. usage and behavior patterns and documents. The BBAC module could analyse malicious behavior from network connections, text messages from chat and email messages, HTTP requests. The

prototype combined big data batch processing of batches to train classifiers. The currently deployed solutions of monitoring relies mostly on detection based on predefined signatures, the examples include Snort and Host based security system (HBSS). These provide protection against previous known attacks but targets which are more sophisticated such as unknown signatures and zero-day attacks are not handled. The delays in assessment of anomalies and identifying actors can prove costly. Large amount of audit data collected in the form of server logs can aid in decision making and time to evidence is a crucial factor. The BBAC design module had sensors for collecting network flows, audit records, application level content such as chat and email messages, wiki pages and PDF documents. Large variety of real time data was fed using a feature extraction process and was trained using appropriate data sets. It parsed through raw data and information was computed to provide a clear representation. Unsupervised learning method such as K-Means++ was used to cluster group actors and supervised learning such as support vector machine. The BBAC module strategically used a combination of unsupervised and supervised techniques to classify and scale entities such as actors, hosts and documents of suspicious/anomalous behavior. The results showed promising results with the module classifying URL's as malicious with a 99.6% accuracy rate and valid URLs being classified as malicious at 1.0% rate.

2.4 Parsing of Textual and E-mail documents

Documents containing text is one of the useful sources of evidence during digital evidence. According to a recent study done [2] a staggering 4 billion accounts related to email is present clearly highlighting a high possibility for obtaining evidence. Usage of classification algorithms provided significant results. Tang et al. (2016) presented a framework for the process of text based learning and presented techniques for identifying text by using classifiers. The framework utlized machine learning modules and analysis of text to identify whether a text message is valid or not. Classification algorithm is used which is trained to different messages of text. The concept of classification concept is utilized in developing the machine learning log parser module. Crimes originating from Emails such as phishing and other methods have gained prominence and based on the keyword construction technique present in header fields, the logs obtained can used for digital investigations.

2.5 Parsing Network traffic data logs

Analyzing the Network traffic logs can be valuable for digital investigations but has a few technical challenges includes:
(1) The results of the network log consists of significant amount of false positives.
(2) New Sophisticated complex techniques are being used to bypass network intrusion detection systems.
Duan et al. (2015) proposed a framework for grouping similar attacks signatures obtained from the network traces. Analysts were alerted and helped in identification of imminent attacks by observing processes using clustering algorithm such as Hidden Naive Bayes (HNB). This facilitated states the ability of machine learning in digital investigations as the usage of machine learning showed a detection rate of nearly 90%. Botnet attacks are causing a serious risk and a major financial damage since they target the popular IoT

[2]Email Statistics Report 2019-2023 -http://www.radicati.com

devices which are being used extensively. Standard network forensic tools cannot detect new forms of botnet. Koroniotis et al. (2018) utilized network flow identifiers that enabled detecting botnets using machine learning techniques for analyzing networks. Network flow identifiers along with relevant UNSW-NB15 dataset effectively helped in botnet detection with low false positives.

2.6 Analysis of Data and Event logs

Digital Investigations involves analyzing large amount of data which is a critical issue. The FBI released statistics which stated that an average online crime case is nearly 500 GB [3]. These data are obtained from sources such as images of disk, network traces and memory dumps etc. The timeline is developed using activities of file system log files, registry files, registry entries etc. A classification method proposed by la Hoz et al. (2015) made use of Self-Organizing Maps (SOMs)- a neural machine learning based model with statistical techniques that parsed the data logs for anomalous network behaviour.

2.7 Classification of file fragment

The type of the file is determined by analyzing the file fragments which are an important aspect of digital investigations. File reconstruction based on content during the process of file parsing is essential as they are stored in a fragmented form in the memory or on the disk. A classification method based on the data type was proposed Zheng et al. (2015) wherein the fragment that needs to be classified is supplied as a data type instead of file type. By frequency distribution, entropy are extracted and a classifier is built on training the data set and Support Vector Machine(SVM) algorithm which determines the data fragments. The results displayed a 88.58% accuracy.

2.8 Automation of report and timeline-file based analysis

The tool Log:Mole Eichelberger (n.d.) automatically processes anomaly files obtained from the CSV export files of log2timeline. The different modules are analyzed and results are combined to reduce the time to analyze in the linux OS platform as well as in windows. While analyzing samples of malware and detection of URL's which were malicious, showed inconsistent results and these were the limitations of the tool. In our proposed research the machine learning module can be developed and to be more efficient in detecting anomalies. The module can be integrated with IOC tool such as Redline to accurately identify malware.

2.9 Attribution based on web history

The browser histories semantic properties can be integrated for digital analysis was suggested by Pretorius et al. (2017). Cosine, Jaccard and other similarity functions were used to identify the probability of a anomalies with respect to browser history. GrammarViz3.0 tool was used for semantic character extraction of sessions called as signatures that were compared further among users for identifying signatures. This method can be utilized on the log parser for training the machine learning algorithm.

[3]FBI.RCFL Program Annual report for Fiscal Year 2017

2.10 Time Line Analysis

Gathering a timeline of events is a useful aspect in the process of digital investigation. The log entries obtained from different events are the sources for the timelines . A timestamp provides the start time of an application in which data from the system was accessed. A detailed event timeline is needed for filtering the huge amount of log events that are generated. Research related to creation of timeline is listed in this section. An important aspect of obtaining information about event timeline that provides a better overall understanding is Visualization. An anomaly present which differs from the standard profile of the system can be detected during the process of digital investigation. Visualization aids in analyzing timelines since the entries contain timeline of logs with enormous capacity of data.

2.10.1 System files Analysis

The present forensic tools reconstructs events of disk image and access to information about the metadata and contents regarding it is obtained. Metadata contains information about the creation time, modified time and last access time. Many events relating to a file is translated sequentially into a timeline by tools such as Sleuth kit. In the timeline based on metadata, the limitation was that the time information was not considered that were present while analyzing contents of file Hargreaves and Patterson (2012).

2.10.2 Timelines providing file timestamps

The Cyber Forensic Time lab tool improved the timeline analysis of metadata of files by obtaining times of file system from Windows Registry files, FAT, MBOX archives, NTFS, EXIF data, link files and files of windows registry Hargreaves and Patterson (2012). Further developments of the tool was suggested such as searching predefined signatures automatically of anomalous activities in the obtained timeline effectively. Log2timeline parses through files and directories and facilitates scanning Gudhjónsson (2010). Effective input modules provides an effective timelines of file systems. Gathering artifacts of activities related to same events produces a 'super event' in an output format of type SQLite.

2.10.3 Digital analysis of browsers

A framework was presented by Dija S et al. (2017) in which browsers were divided into two parts namely the process of acquisition and the other being analysis of the suspects machine during digital analysis. The useful features such as bookmarking, filtering, timeline analysis and exploring capabilities were provided by this framework.

2.10.4 Visualisation

Tools are available both open source and paid for viewing the timelines of system events visually. The tools Zeitline , facilitates importing file system events from Sleuth kit and other sources to be imported on the GUI based toolHargreaves and Patterson (2012). Screening and finding for events is also faciltated by this tool. Aftertime [4] , a java based application facilitates generation of timeline by displaying visuals in the form of histograms.

[4]https://www.holmes.nl/NFIlabs/Aftertime/index.html

3 Methodology

The dataset used for this research consists of credit cards details of European citizens made during the month of September 2013. It consists of 492 frauds in a total of 284,807 transactions which occurred in a span of two days. The positive class(frauds) account for only 0.172% since the dataset is largely unbalanced. Due to PCA transformation the input variables consists of numerical data. Citing confidential issues, the background information and other original features are not provided in the data. The principal components gathered with PCA are features such as V1 to V28 and features such as 'Amount' and 'Time'. Time consists of the number of seconds occurred between the first transaction and the subsequent transactions. The 'Amount' feature is the transactional amount which can be used for dependent cost-sensitive learning. The response variable is the feature 'Class' and it takes the values 1 and 0 for representing fraud and legal respectively.

	# Time Number of seconds elapsed between this transaction and the first transaction in the dataset	# V1 may be result of a PCA Dimensionality reduction to protect user identities and sensitive features(v1-v28)	# V2	# V3	# V4
	0 173k	-56.4 2.45	-72.7 22.1	-48.3 9.38	-5.68 16.9
1	0	-1.3598071336738	-0.0727811733098497	2.53634673796914	1.37815522427443
2	0	1.19185711131486	0.26615071205963	0.16648011335321	0.448154078460911
3	1	-1.35835406159823	-1.34016307473609	1.77320934263119	0.379779593034328
4	1	-0.966271711572087	-0.185226008082898	1.79299333957872	-0.863291275036453
5	2	-1.15823309349523	0.877736754848451	1.548717846511	0.403033933955121
6	2	-0.425965884412454	0.960523044882985	1.14110934232219	-0.168252079760302
7	4	1.22965763450793	0.141003507049326	0.0453707735899449	1.20261273673594
8	7	-0.644269442348146	1.41796354547385	1.0743803763556	-0.492199018495015
9	7	-0.89428608220282	0.286157196276544	-0.113192212729871	-0.271526130088604
10	9	-0.33826175242575	1.11959337641566	1.04436655157316	-0.222187276738296

Figure 1: Dataset of the credit card transactions

A framework was developed using random forest as a classifier by Xuan et al. (2018). Usage of Decision tree models in data mining due to flexibility and simplicity in manipulating various types of data attributes. Single tree model is suitable for only certain datasets and easy to over-fit. Ensemble overcomes these drawbacks by combining groups of individual decisions and compared to single classifiers are more accurate. Random forest being an ensemble method is an association of multiple tree predictors in which the trees depends on random independent datasets. The capability of random forest is impacted by strength of individual trees and correlation among the different trees. Random forest perform better when the single tree is strong and there is less correlation among trees. Random forest is robust to outliers and noise. Two kinds of random forests were introduced namely random forest I and random forest II that differ in the base classifiers. The performance of these two random forests were examined using credit card transactions of real-life B2C dataset. Random forest provides good results on smaller data sets but problem exists related to imbalanced data. There is scope for improving the algorithm such as the voting mechanisms which assumes each of the base classifiers

to be of equal weight but there may be cases some might be important than others. It was highlighted the usage of a more improved algorithm such as Isolation Forest.

Digital investigators need to understand what the data is trying to indicate to determine anomalies. Due to large number of metrics being involved while analyzing logs is time consuming to be monitored manually and traditional BI is not sufficient. Automated real time machine learning anomaly detection methods can be used to great effect when working at this scale. There are mainly two categories of machine learning methods: supervised and unsupervised. In this section we highlight the supervised and unsupervised methods used in anomaly detection and the implemented log parser module which makes use Isolation Forest algorithm. We have also compared the results obtained from Isolation Forest with the widely used Support Vector machine algorithm.

A fraud detection method using KNN algorithm and outlier detection was proposed Malini and Pushpa (2017) to develop solution for detecting fraud. They devised an approach to reduce the false alarm rates and increase detection rate of fraud. The method developed could be implemented on fraud detection system of credit cards, to prevent fraudulent transactions. In outlier detection method, unsupervised learning method is utilized for detection of fraud as it provides new insights about the observed data. Unsupervised data does not require labeling of data or knowledge about fraudulent transactions in prior. Training is not needed to distinguish between an illegal and valid transaction. It follows a normal behavioural pattern to determine any unusual activity. The downside of supervised methods is that it requires the model to be trained with both fraudulent and non-fraudulent use cases before deploying the model in a live scenario. Unusual behaviors is detected only after the training is performed on the system. This is the major advantage of using unsupervised method over supervised data as it does not need to be trained for distinguishing between an illegal and valid transaction. Credit card scam has become a major concern these days. To ensure safety of monetary transaction systems in an effective way, structuring an organized and precise credit card scam detection system is essential. Over sampling and extracting the principal direction of data, the KNN method was used to determine the anomaly of the target instance which suited detecting fraud with the limitation of memory. It was found that less computational requirements and memory were utilized by Outlier detection methods that helped detecting credit card fraud. Outlier detection was found to be fast and efficient on large datasets. Compared to other known methods of detecting anomaly, experimental results proved that KNN method is efficient and accurate.

3.1 Supervised Learning

In supervised learning, the system tries to find out from the previous examples that are given. However, in unsupervised learning, the system tries to search out the patterns directly from the instance given. Thus if the dataset is labeled it comes under the downside of supervised learning, if the dataset is untagged then it is a case of unsupervised learning. Rushin et al. (2017) made a study using supervised machine learning algorithms like Random Forest, SVM, Decision Tree etc for detecting fraud and made a comparison with their accuracy, Recall, F1-Score, Precision etc. These algorithms were evaluated using real world credit card data to determine valid /fraud transaction. A supervised method basically helps to determine the labels based on past transactions. However these do not recognize patterns that has not been recognised in the past. For this research we have used unsupervised techniques based on the study by Carminati et al. (2015) to

determine the class of the transactions.

3.1.1 Support Vector Machine

Support Vector Machines (SVMs) utilizes classification algorithm similar to logistic regression. SVMs are supervised learning models with associated learning algorithms that analyze information used for classification and multivariate analysis. A Support Vector Machine (SVM) may be a discriminative classifier formally outlined by a separating hyper plane. In other words, given information which are labelled (supervised learning), the algorithmic program outputs associate degree optimum hyperplane that categorizes new examples. SVM has been used extensively for traditional techniques for prevention such as passwords, identification systems and PINs. SVM performs well for identifying fingerprints, facial recognition etc. Gyamfi and Abdulai (2018) proposed a method to tackle fraud by using Spark with SVM techniques.

3.2 Unsupervised Learning

Unsupervised Learning is a class of Machine Learning methods to find the patterns in data. The data given to unsupervised algorithmic program aren't tagged, which suggests solely the input variables(X) are given with no corresponding output variables. In unsupervised learning, the algorithms are left to themselves to find fascinating structures within the information.

3.2.1 Local Outlier Factor

The local deviation of the sample density to it nearest neighbours is calculated by the Local Outlier Factor. Since it is local the expected or anomaly rating is dependent on how the sample is isolated with regards to the neighbourhood surrounding it. The number of neighbors are considered, is chosen based on two important factors:
a) greater than the minimum number of sample a group has to hold, so that other sample can be local outliers comparable to this group.
b) smaller than the maximum number of nearby sample that can possibly be local outliers.
The Local Outlier Factor (LOF) rule is associated with an unattended anomaly detection technique that computes the native density deviation of a given information with relevance to its neighbors Gyamfi and Abdulai (2018). It considers outliers of the samples that have a considerably lower density than their neighbors. this instance shows a way to use LOF for outlier detection that is that the default use case of this figure in Scikit-learn. Note that once LOF is employed for outlier detection prediction, decision_function and score_samples strategies are used.

3.2.2 Isolation Forest

Isolation Forests is one of the newly researched algorithms for detecting anomalies. This algorithm relies on the factor that anomalies are data points which are different and few. Relying on these properties, the anomalies are likely to be detected using a mechanism called isolation. This method is useful and compared to existing methods are fundamentally different. The usage of isolation as an efficient and effective way to find anomalies compared to the extensively used method based on density and distance measures. The

algorithm consists of low linear time complexity and a low memory requirement. It helps in developing a model which performs accurately having small number of tress and also using limited sub samples of a fixed size irrespective of size of the data set. Each point of the data is isolated and split into either inliers or outliers by the algorithm. Huge data sets with several dimensions can be analysed by this algorithm. The Isolation Forest algorithm isolates the observations by choosing a feature randomly and then selecting a value for split between the maximum and minimum values of the feature which are selected. Isolating anomaly observations is simple as only a few conditions are needed to distinguish those cases from normal observations. However, isolating normal observations require more conditions. Therefore, calculation of an anomaly score can be done on the basis of number of conditions required to classify a given observation. The algorithm functions by creating the isolation trees first, or random decision trees. Then, based on the path length to isolate the observation the score is calculated. The tree structure represents the separation that takes place recursively, the amount of split needed to partition the sample should be similar to the length of the path from the node present at the root node to the ending node. This path length, averaged over a forest of such random trees, is estimation of normality and our conclusion function. Random separation produces attentively shorter paths for variation. Hence, if shorter path lengths for a particular sample is produced by a forest of random trees, they have a high probability to be anomalies. The Isolation Forest algorithm consists of following process:- Forest, Isolation Tree and Evaluation (Path Length) The Forest step facilitates in clustering i.e grouping, while isolation tree reduce the isolated transaction and finally evaluation stage is responsible for execution of algorithms and produces graphical output, making use of data visualization techniques like histograms, correlation matrix etc. The evaluation or path length step takes into consideration the length of the path and states the procedures of Isolation Forest Algorithm and the Local Outlier Factor. So, following the above mentioned stages we develop a solution for our project.

4 Design specification and Implementation

The Log2Timeline tool facilitates digital investigations by creating super timelines that provides information about the systems events and log files. The timeline of the entire system is analyzed and outputs log information into a single, comprehensive timeline. Log2timeline outputs a file with all artifacts present in a system, along with their respective timestamps containing information about the user's activity. The forensic analyst outputs the results of log2timeline onto a database for querying, generates reports to capture key pieces of information in a sequentially. Log2timeline's functionality of collecting artifacts into a single timeline of collected artifacts and helps digital investigators in obtaining details about an incident. The current Log2timeline tool consists of many log parsing modules for analyzing different applications. Existing Log2timeline tool produces a lot of backlogsDebinski et al. (2019) and there is no machine learning log parsing module available for this tool as far as our knowledge is concerned. In this proposal a log parsing module which utilizes machine learning is developed. This module is fed into the log2timeline system.

The proposed log parsing module consists of anomaly detection algorithms such as Local outlier factor and Isolation forest algorithm. It comprises of two parts, the first deals with testing the outliers and a score is given. The second part deals with isolating the

anomalies and separating it from the general group. The first algorithm is sent with the dataset consisting of 56,000 odd transactions which determines the outlier factor based on 28 parameters. The transactions are observed with respect to various parameters and given a score of either a 0 or 1 where 0 states that it is a fraudulent transaction and 1 determines that it is a valid transaction. The transaction is proceeded next with the Isolation Forest algorithm in which the anomalous/bad values are isolated and is segregated into different groups. These techniques are then alerted and intimated to the user. The software used to develop the log parser is Jupyter notebook which is a sub product of Anaconda. Jupyter notebook is operated online and is locally hosted onto the cloud. The initial step is to first import the Various packages such as Scipy, MatplotLib, NumPy, Pandas and Seaborn are imported initially. Pandas are used for importing data, manipulating and inserting data onto the canvas. Pandas are vital in our research as it is facilitates importing the datasets and 56,000 transactions into the Jupyter notebook environment. The NumPy library facilitates mathematical calculations in the dataset. Every mathematical transactions are done with the help NumPy. Scipy library facilitates scientific calculations such as sin and cos operations in the same dataset, Matplotlib is a mathematical graph plotter which is used for the purpose of data visualization. Different types of data visualization that can be done using matplotlib are Histograms, Double matrix squares, Anomaly graphs etc. Seaborn is a matplotlib based library which aids visualization of python data. An interactive interface is provided that displays informative graphical statistics. It is mainly utilized for plotting scatter diagrams.

4.1 Log parsing module

Log2Timeline is configured with the developed log parsing module created using Jupyter notebook along with Isolation Forest algorithm for training the module on the malicious data sets.

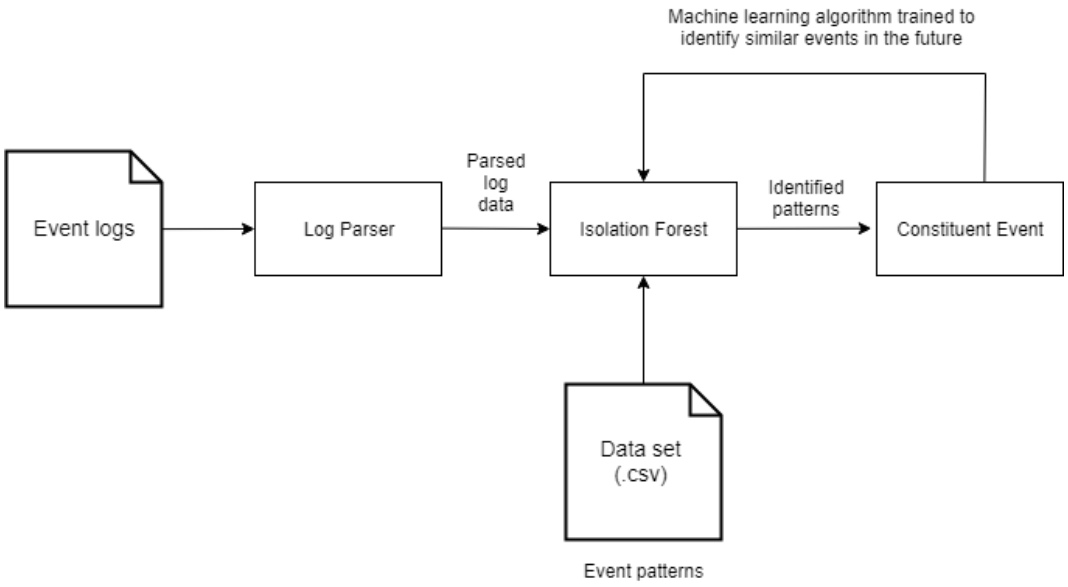

Figure 2: Setup of the Log parsing module

5 Evaluation

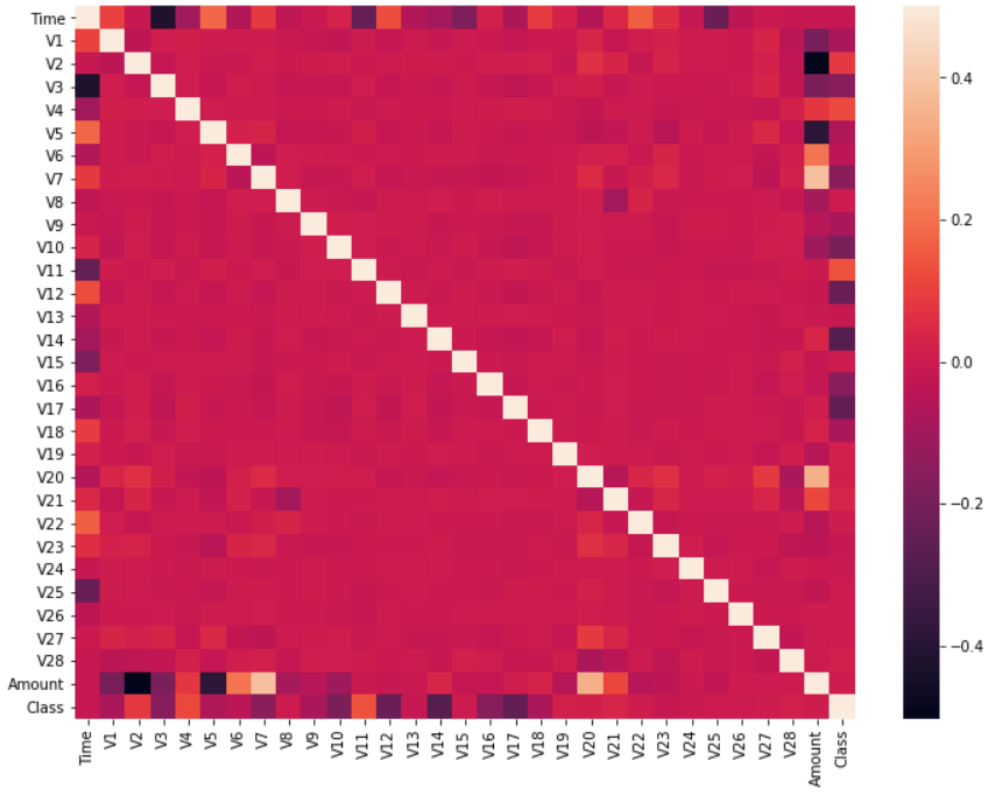

Figure 3: Heat map

The correlation matrix shown in figure 3 can be used to check the correlation of variables between different parameters in the dataset. This facilitates in building the network and fitting our model. The core features are features are displayed and the process of overall classification. It is a figure of pyplot that uses SNS heatmap and Seaborn. This makes the correlation matrix into a visual display. The various pressure points present in the dataset are displayed in the SNS heatmap. It helps to highlight important aspects in the data. The heatmap contains V aspects on both sides of the X and Y axis on a scale that ranges from +0.50 to -0.50. Majority of the transactions are real time and valid, the heat map shows a significance pattern near o points. From the heat map we notice a lot of differences and correlation in 'class' column in all the features of V. The heatmap displays the different correlations present in the classifications in the dataset. In the analysis we have considered only 20% of the credit data 56961 from 284,807 to get a higher accuracy. Histograms are used to visually represent the different anomalies in the individual parameters.

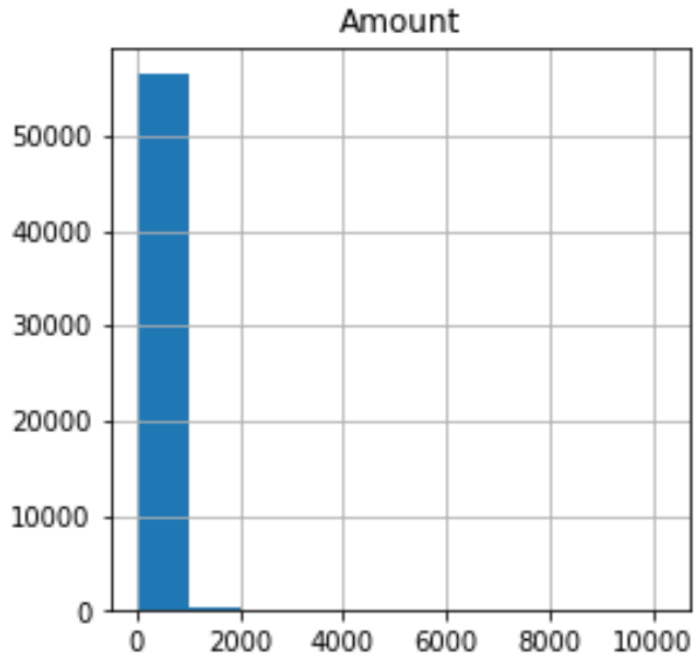

Figure 4: Histogram representation of amount

The above displayed histogram represents the average number of amount transacted by users and the number of valid/fraudulent transactions. 1 represents fraud and 0 represents valid transactions. As seen in the graph, a majority of the transactions are valid and only a small percentage are fraudulent.

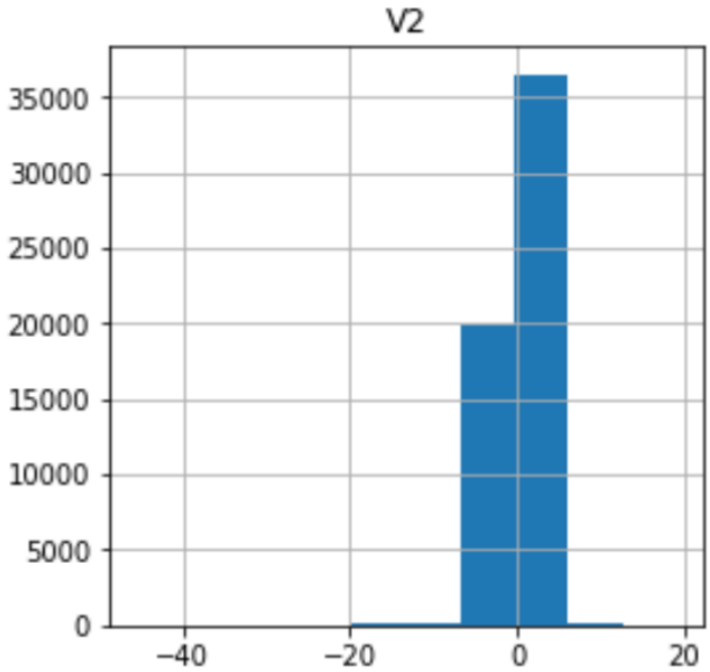

Figure 5: Histogram of Average transaction

The above histogram may represent the average of the transactions and we observe a higher and lower rise based on the transactions made. If the transactions represented are

Unusual transactions , they are displayed by higher or lower variations than the actual average transactions and will produce either a "1" or a "0". The "0" states that there are no transaction errors and are valid and transactions which have "1" or near to "1" are considered a fraudulent transactions.

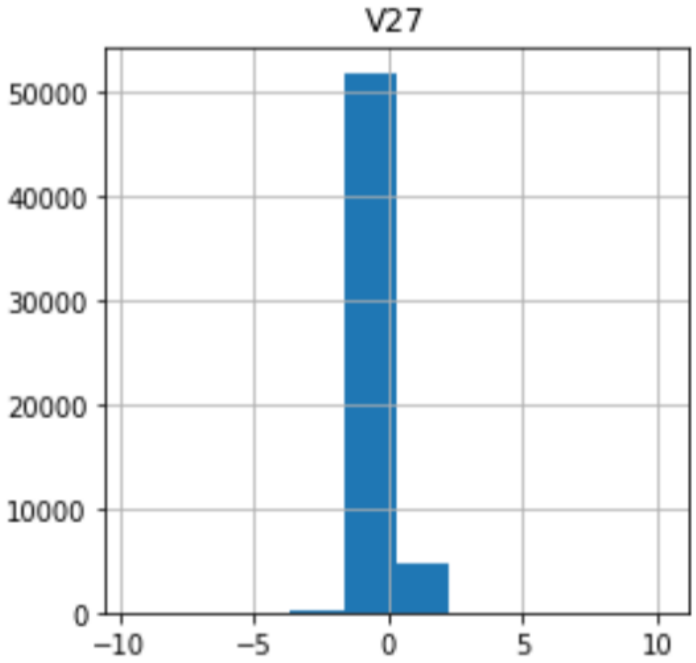

Figure 6: Histogram of location

The above histogram provides details about the various locations of the transactions. If the transaction is from an unusual location which is generally not from the current location, the location is flagged and will alert the user of a possible fraudulent transaction. The value of "0" implies the transaction is within the permitted location perimeter, whereas if the transaction is made from an unusual location it will display "1" which will be flagged and gives a value of "1". This process shows that more than 50000 transactions took place from the usual/original locations and a very small percentage (less than 5%) were made from questionable unusual locations.

Isolation Forest				
Metric	Precision	Recall	f1-score	Support
0	1.00	1.00	1.00	56865
1	0.24	0.24	0.24	96
accuracy			1.00	56961
macro avg	0.62	0.62	0.62	56961
weighted avg	1.00	1.00	1.00	56961

- Total errors detected using Isolation Forest: 147
- Time Elapsed: 27.925991535186768
- 0.9974192868804972

Local Outlier Factor				
Metric	Precision	Recall	f1-score	Support
0	1.00	1.00	1.00	56865
1	0.04	0.04	0.04	96
accuracy			1.00	56961
macro avg	0.52	0.52	0.52	56961
weighted avg	1.00	1.00	1.00	56961

- Total errors detected using Local Outlier Factor: 185
- Time Elapsed: 2.6687846183776855
- 0.9967521637611699

SVM				
Metric	Precision	Recall	f1-score	Support
0	1.00	1.00	1.00	56865
1	0.00	0.00	0.00	96
accuracy			1.00	56961
macro avg	0.50	0.50	0.50	56961
weighted avg	1.00	1.00	1.00	56961

- Total errors detected using SVR: 96
- Time Elapsed: 510.4589068889618
- 0.9983146363301206

The Local Outlier Factor is the fastest algorithm with time elapsed with 2.669 seconds. The isolation Forest is the found to be the most efficient algorithm with a precision being 0.24(24%) for analyzing whether it is a fraudulent transaction or not compared to Local Factor Outlier with 4% and SVM having zero. The SVM is the slowest algorithm with time taken 510.459 seconds.

6 Conclusion and Discussion

In our research we have developed a log parser using Isolation Forest algorithm which can used during digital investigation for detecting anomalies and fraud. The Isolation Forest method is relatively new and can be developed for other use cases. We can increase the accuracy rate by increasing the sample size and using other advanced deep learning algorithms. Cyber attacks involving malware, phishing and fraud have become sophisticated with cyber criminals employing new techniques. It is a major concern that detecting fraud, phishing and malware has become more challenging as cyber criminals have resorted to more sophisticated techniques. Tools which utilizes advanced methods for detecting and parsing log data efficiently are needed to combat cyber attacks and aid digital investigations. There is a need to develop tools which uses advanced methods to detect anomalies early that can potentially avoid a possible cyber attack. The usage of advanced outlier detection methods, when compared to existing methods is fundamentally different. Since the proposed log parser which uses Isolation Forests requires low

memory requirement and low linear time-complexity it gives us better tools to improve our detection rates and react faster to new fraud attacks.

Due to time constraints the log parser could not be integrated with the Log2timeline tool. In our future work this can be integrated along with an output module which utilizes ELK Stack for a better log management capabilities. The log2timeline tool consists of the following modules:- a front end/GUI ,shared libraries, input and the output module. To increase threat assessment profile an Indicator Of Compromise (IOC) tool such as Redline will be developed for identifying activities of malicious signatures. Output of Log2Timeline is sent to Elasticsearch, Logstash , Kibana) which is an open source tool that can be used for assisting investigators for analyzing the logs/events in a detailed manner with better visualizations. All the tools used are open source which makes it feasible to implement them in the research. The Output Module can be integrated with the ELK Stack. ELK (Elasticsearch , Logstash, Kibana) Stack[5] can be utilized for log management, analytics and timelining of the system disk image. X-Pack is an open source plugin that can be configured with kibana to aid investigation by efficiently filtering the anomalous events during digital analysis. Kibana allows creating custom dashboards to enable easy analysis of the log2timeline logs. Visualization metrics such as pie charts, horizontal charts, vertical bar charts are provided as shown in Figure 7. Redline is a popular Indicator Of Compromise (IoC) tool for generating a threat and vulnerability profile [6]is used for obtaining a better threat assessment profile. Visualizations are provided for the output generated which contains log file events,web access logs using ELK stack.

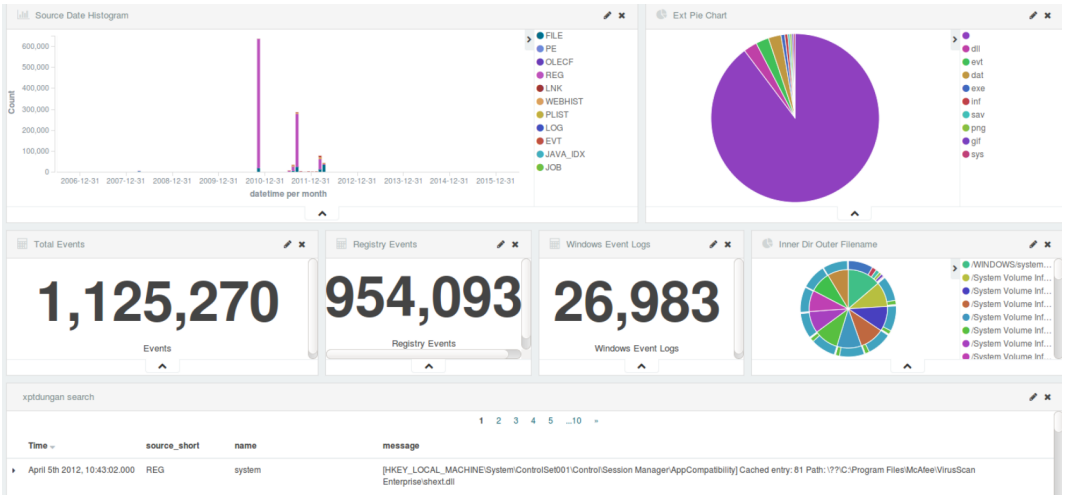

Figure 7: Visualization using Kibana dashboard

The open source framework known as OpenIOC[7] can be utilized for developing efficient indicators for the tool Redline in identifying anomalous events in the registries, web browser history, system files and logs etc. For example the dashboard of Redline can be configured to list the various processes and a Malware Rating Index(MRI) score is assigned to them that helps indicating suspicious artifacts(data).

[5]https://www.elastic.co/elk-stack
[6]https://www.fireeye.com/services/freeware/redline.html
[7]www.openioc.org

The flow chart shown in Figure 8 represents the proposed log parsing module being integrated to log2timeline. File containing logs is the input that is supplied to the system and the developed machine learning log parser is utilized for effectively identifying anomalies present in the system.

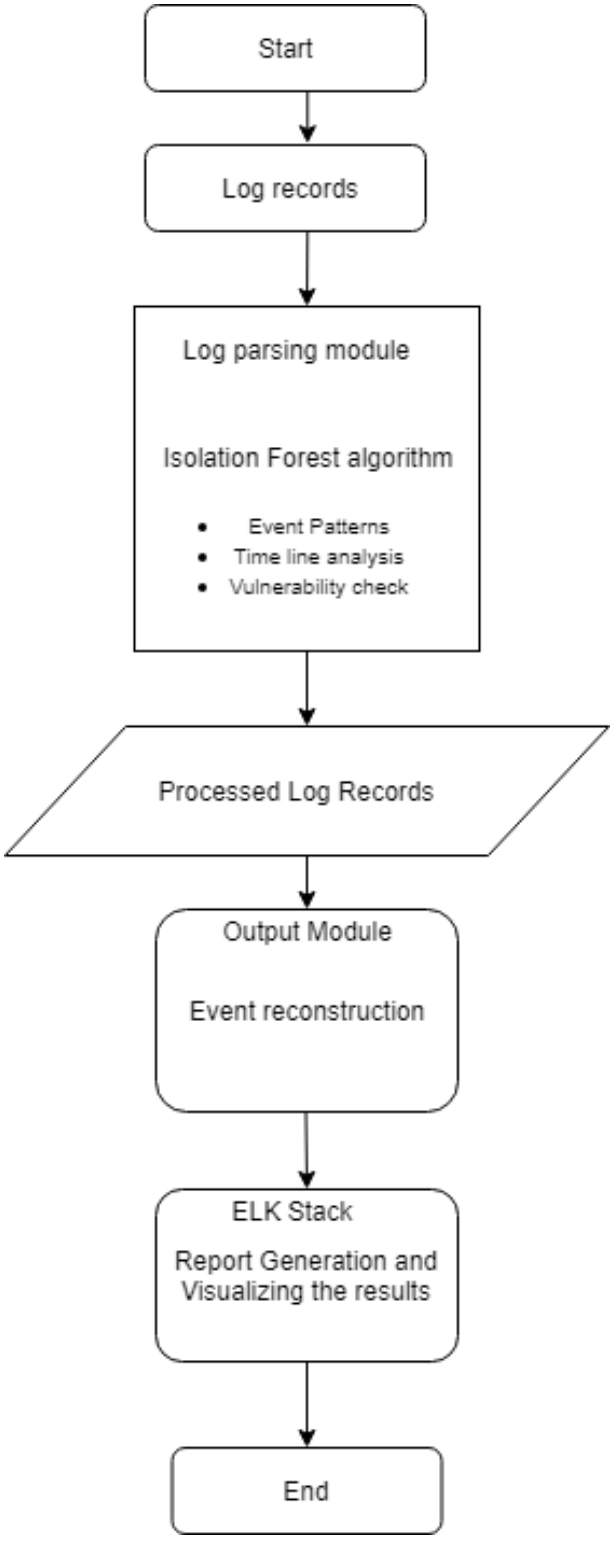

Figure 8: Setup of the proposed Log2timeline

References

Awoyemi, J. O., Adetunmbi, A. O. and Oluwadare, S. A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis, *2017 International Conference on Computing Networking and Informatics (ICCNI)*, pp. 1–9.

Carminati, M., Caron, R., Maggi, F., Epifani, I. and Zanero, S. (2015). Banksealer: A decision support system for online banking fraud analysis and investigation, *Computers Security* **53**: 175 – 186.
URL: *http://www.sciencedirect.com/science/article/pii/S0167404815000437*

Debinski, M., Breitinger, F. and Mohan, P. (2019). Timeline2gui: A log2timeline csv parser and training scenarios, *Digital Investigation* **28**: 34–43.

Duan, Y., Fu, X., Luo, B., Wang, Z., Shi, J. and Du, X. (2015). Detective: Automatically identify and analyze malware processes in forensic scenarios via dlls, *2015 IEEE International Conference on Communications (ICC)*, pp. 5691–5696.

Eichelberger, F. (n.d.). Automation of report and timeline-file based file and url analysis, *SANS Institute: Reading Room - Forensics* .
URL: *https://www.sans.org/reading-room/whitepapers/forensics/paper/34572*

Esposito, S. and Peterson, G. (2013). Creating super timelines in windows investigations, *in* G. Peterson and S. Shenoi (eds), *Advances in Digital Forensics IX*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 135–144.

Gudhjónsson, K. (2010). Mastering the super timeline with log2timeline, *SANS Institute* .

Gyamfi, N. K. and Abdulai, J. (2018). Bank fraud detection using support vector machine, *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 37–41.

Hargreaves, C. and Patterson, J. (2012). An automated timeline reconstruction approach for digital forensic investigations, *Digital Investigation* **9**: S69 – S79. The Proceedings of the Twelfth Annual DFRWS Conference.
URL: *http://www.sciencedirect.com/science/article/pii/S174228761200031X*

He, S., Zhu, J., He, P. and Lyu, M. R. (2016). Experience report: System log analysis for anomaly detection, *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 207–218.

Inglot, B. and Liu, L. (2014). Enhanced timeline analysis for digital forensic investigations, *Information Security Journal: A Global Perspective* **23**(1-2): 32–44.

Inglot, B., Liu, L. and Antonopoulos, N. (2012). A framework for enhanced timeline analysis in digital forensics, *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, IEEE, pp. 253–256.

Koroniotis, N., Moustafa, N., Sitnikova, E. and Slay, J. (2018). Towards developing network forensic mechanism for botnet activities in the iot based on machine learning techniques, *in* J. Hu, I. Khalil, Z. Tari and S. Wen (eds), *Mobile Networks and Management*, Springer International Publishing, Cham, pp. 30–44.

la Hoz, E. D., Hoz, E. D. L., Ortiz, A., Ortega, J. and Prieto, B. (2015). Pca filtering and probabilistic som for network intrusion detection, *Neurocomputing* **164**: 71 – 81.
URL: *http://www.sciencedirect.com/science/article/pii/S0925231215002982*

Lin, Q., Zhang, H., Lou, J.-G., Zhang, Y. and Chen, X. (2016). Log clustering based problem identification for online service systems, *Proceedings of the 38th International Conference on Software Engineering Companion*, ICSE '16, ACM, New York, NY, USA, pp. 102–111.
URL: *http://doi.acm.org/10.1145/2889160.2889232*

Malini, N. and Pushpa, M. (2017). Analysis on credit card fraud identification techniques based on knn and outlier detection, *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, pp. 255–258.

Mayhew, M., Atighetchi, M., Adler, A. and Greenstadt, R. (2015). Use of machine learning in big data analytics for insider threat detection, *MILCOM 2015 - 2015 IEEE Military Communications Conference*, pp. 915–922.

Pretorius, S., Ikuesan, A. R. and Venter, H. S. (2017). Attributing users based on web browser history, *2017 IEEE Conference on Application, Information and Network Security (AINS)*, pp. 69–74.

Rushin, G., Stancil, C., Sun, M., Adams, S. and Beling, P. (2017). Horse race analysis in credit card frauddeep learning, logistic regression, and gradient boosted tree, *2017 Systems and Information Engineering Design Symposium (SIEDS)*, pp. 117–121.

S, D., S, I., A, S. and A, V. J. (2017). A framework for browser forensics in live windows systems, *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pp. 1–5.

Tang, B., Kay, S. and He, H. (2016). Toward optimal feature selection in naive bayes for text categorization, *IEEE Transactions on Knowledge and Data Engineering* **28**(9): 2508–2521.

Xuan, S., Liu, G., Li, Z., Zheng, L., Wang, S. and Jiang, C. (2018). Random forest for credit card fraud detection, *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*, pp. 1–6.

Zheng, N., Wang, J., Wu, T. and Xu, M. (2015). A fragment classification method depending on data type, *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 1948–1953.