Detecting Security Vulnerabilities using Machine Learning Models

Rohit Vishwakarma, Sreekanth R Gunishetty, Shamanth R Rao, Anil M S
Co-author: Vinay Joshi
Dept. Computer Science and Engineering, PES University
Bengaluru, India

***Abstract*—Machine Learning and Artificial Intelligence have established their own footprints in various domains and its applications. Machine Learning techniques are used in different areas of the Cybersecurity field such as Network Traffic Analysis, Detection of Attacks and Anomalies. All Cybersecurity tasks can be divided into five categories such as prediction, prevention, detection, response and monitoring.**

We are building a cybersecurity tool that runs a machine learning model to combat cyber breaches and frauds. Machine Learning Models include Logistic Regression, Isolation Forest, Decision Trees, etc for detecting frauds and anomalies such as breach as terms of use, Identity theft, Self Delivery fraud. Fraud detection includes monitoring the behavior of populations of users in order to estimate, detect, or avoid undesirable behavior. Our cybersecurity tool is meant to ease the work of a human by detecting anomalies faster by analyzing the data and logs and training on them.

I. Introduction

Machine Learning and Artificial Intelligence stand out as the most trending keywords of today's world of technology. Machine Learning Techniques are used in various fields and domains such as Image and Speech recognition, Traffic prediction, Product recommendations, Self-driving cars, Email Spam and Malware Filtering, etc.

All Cybersecurity tasks can be divided into five categories such as prediction, prevention, detection, response and monitoring. The Technical layers in the dimensions of network, endpoint, application, user and process. Machine Learning Models can be used in Network Traffic Analysis of all traffic at each layer and detect attacks and anomalies. Machine Learning can be used as regression to predict next system calls and classification to divide programs into malware, spyware, ransomware.

In these various applications of ML in the Cybersecurity field, We are building a cybersecurity tool that runs Machine learning models to combat cyber breaches and frauds. This cybersecurity tool is meant to ease the work of a human by detecting anomalies faster by analyzing the data and logs and training on them. For example, if a human needs to go through 1000 logs or account activities per hour, this machine learning tool should be able to reduce that to 200-100 logs/activities. Thus it reduces the amount of data and also will be faster in threat response than humans.

The Usage of web logs adds a remarkable innovative approach as very few research works are done using Our method. Machine Learning Models include Logistic Regression, Isolation Forest, Decision Trees, etc for detecting frauds and anomalies such as breach as terms of use,Identity theft, Self Delivery fraud. Fraud detection includes monitoring the behavior of populations of users in order to estimate, detect, or avoid undesirable behavior.

This project is about implementing machine learning models in the field of cybersecurity. We are going to train the machine learning model on the user activities on the website. This tracks their previous activities and predicts their future activities and any deviation to this will trigger alarm. So a cyber analyst doesn't need to go through piles of logs and data everyday, but only a select few high priority ones. The machine learning model trains everyday and reports cases to cyber analysts who then report if it is an actual case or not. Based on the outcome of this re-trains for the next day and it will be prepared for future attacks.

Many popular tools are on the market right now relating to this such as patternex, chronicle security and many more. They analyze petabytes of data for cyber breaches which are humanly impossible. We are implementing and enhancing a model created from MIT's AI square and Chronicle security. It is a supervised machine learning model for cybersecurity. The supervised machine learning model is supervised by a cyber analyst who can recognise anomalies from the given set of logs sent to him/her by the model.

II. PROBLEM STATEMENT

Our Problem statement is finding the security vulnerabilities and cyber frauds such as fraud transactions, account takeover etc at a large scale that is humanly impossible. So Our domain for the Capstone project is Cybersecurity along with Machine learning.

A typical example of how this works is, a human needs to monitor thousands of logs per day of a website to check for anomalies. Our project tries to minimise this by machine learning all the logs and looking for anomalies and reports few suspected logs to the human for inspection. A thousand logs for inspection is reduced to a few hundredS or less for inspection. This reduces response time as well as the amount of logs to check.

A. Purpose

The Purpose of the project is to develop a cybersecurity tool to combat breaches in a network or a large website. They train and analyze a large set of data, looking for anomalies and breaches. They report those back to analysts. So an analyst need not go through a large set of data. This tool reduces the amount of data a cyber analyst needs to go through.

B. Benefit

The Benefit of the project finds to be Instead of the tiresome work of looking at the same but large data which will be inefficient for a human, a machine can go through these at ease and report only those which require special attention.

C. Limitations

The Limitations of the approach is the accuracy of the model and the type of threats it can handle. A machine is still a machine. It still has the accuracy limit. So it may not report all anomalies. The type of threats depends on the complexity of the cyber attack. Humans can invent new types of attacks and the model has to learn it first before reporting about this. So an attack not learnt by the model leaves the system vulnerable.

III. LITERATURE REVIEW

This section presents a substantial survey on various research works and existing fraud detection methods that help to create,inform and reform our ideas on the implementation of the model.

A. Survey of Fraud Detections[1]

This paper presents a survey of various techniques used in Credit card fraud detection, Telecommunication fraud detection and Computer intrusion detection with the goal of providing a comprehensive review of different techniques to detect frauds.

1) Credit Card Fraud Detection: Their Research work divides Credit Card Fraud into two types such as Offline Fraud and Online Fraud. Offline Fraud can be done by using a stolen physical card which can be locked by the card issuer before it can be used in fraudulent manner, whereas Online Fraud can be committed using various means such as web,online shopping or cardholder-not-present. To detect the chances of frauds using credit card, they have surveyed on various machine learning techniques: -

- Outlier Detection
  An Outlier is an observation among other observations in a data found to be suspicious which was generated by a different mechanism. Unsupervised Learning approach does not consider any prior knowledge about fraudulent and non-fraudulent transactions which models a baseline distribution to detect the observation shows greater departure from the norm. Supervised Learning models differentiate the transactions so that the new observation can be assigned to classes.
- Neural Networks
  Neural Networks will be trained using the past transaction data of a particular set of customers which makes the network process the current spending patterns to detect the possible anomalies. In Neural Networks, Each node contains a weighted connection to several other nodes in adjacent nodes.

2) Computer Intrusion Detection : Intrusion can be referred to as the possibility of a deliberate unauthorized attempt to access information, manipulate information or render a system unreliable or unusable. Intrusion can be two ways such as misuse intrusions and anomaly intrusions.

- Misuse Intrusions Misuse Detection tries to recognise the attacks of previously observed intrusions in the pattern or a signature form and directly monitor for the occurrence of these patterns. It includes various approaches such as model based reasoning,state transition analysis,keystroke dynamics monitoring,etc.
- Anomaly Detections Anomaly Detection detects the large deviations from the user profiles to possible intrusions. The Detection methods can be statistical approaches, predictive pattern generation and neural networks.

3) Telecommunication Fraud: Telecommunication Frauds contain Superimposed and Subscription frauds where Superimposed frauds happen from using a service without having a necessary authority detected by the appearance of unknown calls on a bill. Subscription frauds occur from receiving a subscription to a service with false identity with no intention of paying.

- Visualization Methods Visualization techniques rely on human pattern recognition to detect anomalies and are provided with close-to-real-time data feeds.
- Rule Based Approaches Rule based approaches provide efficient user profiles containing explicit information, where fraud criteria are referred to as rules.

B. A Novel Web Fraud Detection Technique using Association Rule Mining [2]

In this paper, They introduce a new architecture for web fraud detection using Apriori algorithm for association rule mining and phishing tank databases in web advertising networks.

The Architecture includes various elements in their proposed model such as

- Clients
- ISPs
- Web Access log
- Data Transformation
- Database Generation
- Association Rule Mining
- PhishTank Database
- Match Extracted Pattern

The Proposed Model is used for learning with the system and approximating the fraud cases which provides a platform for testing using web logs. The PhishTank Database is used for finding a pattern of fraud URL using sample log files to

create the frequent pattern. The web logs file is modified with the fraud URLs randomly selected and a part of the URL is produced in the system.

Their proposed algorithm for fraud detection works in three phases:

- Phase I Input: Log table L Output: Transformed log table S
- Phase II Input: Log table S, URL, Time and Minimum support Output: Frequent patterns with support
- Phase III Input: Frequent Patterns Output: Frauds or not

Evaluation parameters and results of the model are demonstrated in terms of

- Accuracy
- Error rate
- Memory Consumed
- Search Time

C. Credit Card Fraud Detection Using Machine Learning Models and Collating Machine Learning Models [3]

This paper focuses on application of Data Mining on finance fraud detection in databases to automate analysis of huge volumes of complex data. Their work checks the performance of various machine learning models such as Decision Tree, Random Forest, SVM and Logistic Regression on highly skewed credit card fraud data.

Credit card datasets are rarely available, highly imbalanced and skewed. Their research work includes the type of sampling approach used, selection of variables and detection techniques used. Logistic Regression SVM Model Decision Tree Random Forest

The Architecture Design contains a data cleanup and validation which includes removal of redundancy, filling spaces in columns, conversion into necessary classes. The data can be divided into two parts such as training and test dataset. K Fold cross validation is done in which the original sample is randomly partitioned into k equal sized subsamples. The subsamples are used as training data for various ML Models.

The Results of the models found to be 97.7

D. A Dynamic Rule Creation Based Anomaly Detection Method for Identifying Security Breaches in Log Records [4]

In this paper, they have proposed a method for anomaly detection in log files, based on data mining techniques for dynamic rule creation which supports parallel processing, provides distributed storage and distributed processing of data using Hadoop Framework. The Three major detection methods used for monitoring malicious activities:

- Scanning based on searching for pre-defined strings in files
- Activity monitoring monitors a file execution and its behaviour
- Integrity checking creates a cryptographic checksum for chosen files and periodically checks for integrity changes

The Data Mining involves major steps to gain knowledge discovery in the databases Data cleaning Data Manipulation Data Transformation Data Mining Pattern Evaluation Knowledge Interpretation

Data Mining is semi automatic knowledge extraction which contains majorly two algorithms:- Classification It is the process of data mapping to previously definedcategorical data. Clustering It is the method of mapping the data into groups. Parallel processing involves MapReduce used for processing large datasets by using both Map and Reduce functions with the generation of key-value pairs and joining all the values with the same key. Static Rule-based Correlation is based on static rules,specified before the analysis. Dynamic Rule Creation solves the problem with continual breach patterns by searching for unknown potential breaches. The Detection Strategies includes a record analysis which are based on selection criteria such as Anomaly Detection,MisuseDetection and Hybrid Detection

Their Institute uses six most important report categories that are collected and analyses in order to identify possible breaches:-

- Network Activity Reports
- Resources Access Reports
- Malware Activity Reports
- Authentication and Activity Reports
- System and Data change Reports
- Failure and Critical Error Reports

IV. PROPOSED METHODOLOGY

In this section, We are going to describe the way we have gathered the data preprocessed for the project. Understanding how the various approaches are used to obtain desired output which explains the technical details of the project.

A. Data Interpretation and Pre-Processing

- Breach in terms of use:
  - Data generation: Every ecommerce site has a limit to buy the quantity of products by a single user. This is usually done to prevent mass buying and selling by a fraud retailer. So to detect this is to detect the users who are buying a product in excess quantity(i.e. more than the average of the quantity bought by genuine users by some excess amount). To generate this data, we create a few users who start buying in excess quantities, which is more than the average of quantities for the same product by a few multiples of standard deviation.
  - Data preprocessing: Since there are many unwanted columns such as zip code, url, http code, cardnum etc. we remove these columns. We also have a few rows that are not ‘orders’, so they don’t have items or quantity. These rows need to be removed. We also convert quantity and price to numeric.
- Self Delivery Fraud Detection:
  - Data generation: This is a fraud where a merchant who sells items on the ecommerce site themselves commit fraud. This is done by creating a few fraud

accounts where they have the same pincodes. Then they can keep ordering in large quantities of the products sold by the merchant. This gives the merchant a chance to rate his products every time he orders. So he/she can increase the ratings of the products sold by him. To generate this data, we first create a few fraud accounts which have similar pincodes, usually one or two pin codes are used for all accounts. Then the fraud accounts order only the items sold by the merchant and give 5 star ratings. This way the merchant increases ratings for his own products and thus gives fake reviews.

- Data preprocessing: We need to look at the parameters that make a user a fraudulent user by his transactions. To perform the fraud, a merchant creates a user who performs large quantities of transactions on products belonging to a specific merchant. He also orders the product to the same pincode so that he can restore it to his inventory back. The user also gives 5 star ratings to the product to increase the ratings. So the parameters we need to look at are:
  - % of total ratings by this account: The percentage of ratings given by this account to that of total percentage of product rating.
  - % of delivery to the pincode of the user to that of all the delivered pin codes for that product.
  - % of transactions for this product to that total no of products bought by the user.
  - % of 5 star rating: Ratings to the specific product to that of total ratings given by the user.
  - % of 4 star rating: % of 4 star rating for the product to that of total ratings given by the user.
  - % of 1 star rating: % of 1 star rating for the product to that of total ratings given by the user.

- Credit Card Fraud Detection
  - Data generation: Credit card frauds are done by either stealing a credit card that was already being used by a user before or by getting a stolen credit card to perform a large number of transactions on the account. So the data generation step involves reusing the credit card that was already used by an account or hacking into an account and then performing a large number of transactions on the account.
  - Data pre processing: To detect if the credit card entered is already in use, we make a list of credit cards in use currently, so we know if someone enters the already in use credit card. To detect the large number of fraudulent transactions on an account, we go for a statistical formula known as z-score. Z-score defines how much standard deviations away is the point from the mean. So no matter what the mean is and where the point lies, all points get reduced to a singular metric of score known as z-score. So we define the limit to a credit card based on the z-score of transactions performed on the card.

B. ML Modelling

Breach in terms of use To Detect the Outliers from the large datasets which contain the transaction details. The Breach of Terms of Use Fraud majorly depends on the two attributes namely, Product id and Quantity which the user has bought. We have used several models to detect the outliers from the dataset

Cluster Based Local Outlier Factor (CBLOF) CBLOF is a meaningful algorithm which provides importance to local data behavior. It can be used as a measure for identifying the physical significance of an outlier. But, It detects the outliers in a very good manner but it frames some of the inliers as fraudulent transactions.

Machine Learning model	Outliers	Inliers
Cluster Based Local Outlier Factor (CBLOF)	274	27181
Histogram-Based Outlier Detection (HBOD)	270	27185
K- Nearest Neighbours	172	27282
Isolation Forest	271	27184

Histogram-Based Outlier Detection (HBOS) A Histogram-based outlier detection (HBOS) algorithm is presented, which scores records in linear time. It assumes independence of the features making it much faster than multivariate approaches at the cost of less precision.

KNN

KNN works on a principle assuming every data point falling in near to each other is falling in the same class. In other words it classifies a new data point based on similarity. It detects the outliers during irregular conditions and the pattern of detection doesn't match with Our Dataset fields and attributes.

Isolation Forest

Isolation forest works on the principle of the decision tree algorithm. It isolates the outliers by randomly selecting a feature from the given set of features and then randomly selecting a split value between the maximum and minimum values of the selected feature. It gives us the best fraud detection model for Breach in terms of use.

Self Delivery Fraud Detection

Decision Trees

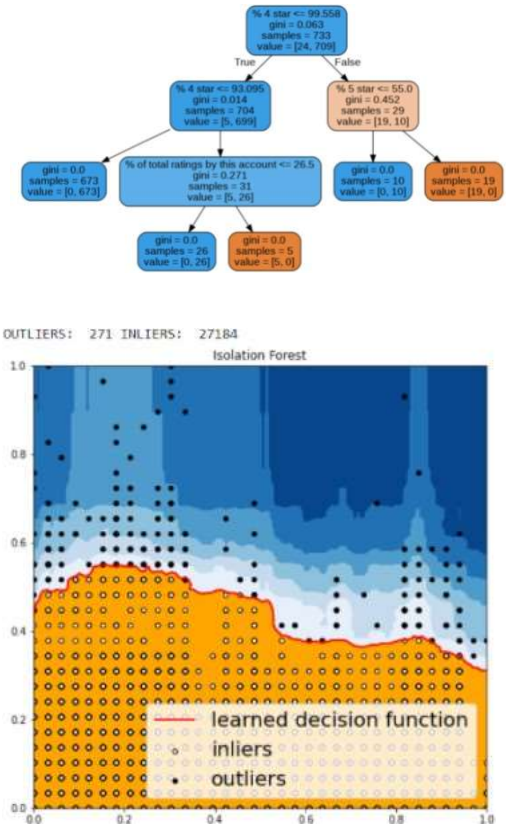
Decision Tree is a model which classifies based on multiple covariates or for developing prediction algorithms. It classifies Our Dataset population into branch-like segments that construct an inverted tree with a root node,internal nodes and leaf nodes.

Random Forest

Random Forest is a supervised learning algorithm which adds additional randomness to the model,while growing the trees. It searches for the best feature among a random subset of features instead of fetching for the most important feature while splitting a node.

Credit Card Fraud Detection

As We need the Outlier Detection model to detect the Credit card frauds in the transactions. We tried to use the Isolation forest model to detect the higher Z Scores transactions. The Model provides us with an accuracy of 87-89



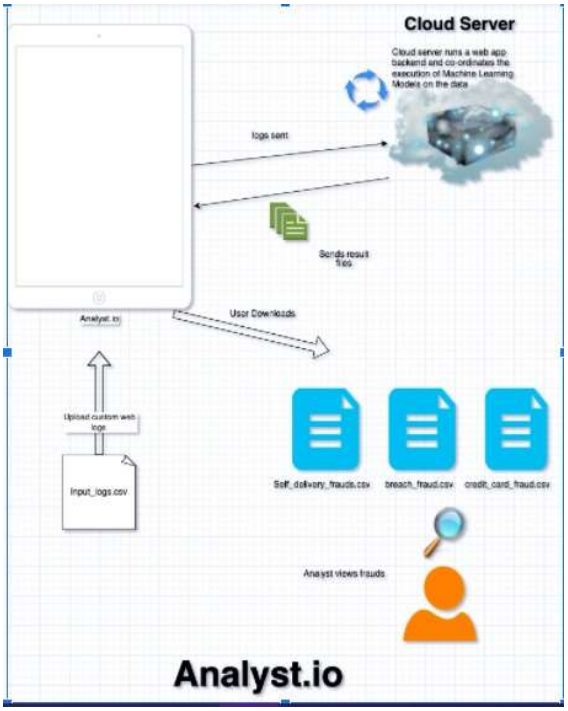


Fig. 1. **Backend Design**

User Interface and Backend Design

User Interface

We have built a responsive and modern looking web application with a stunning landing page which guides and excites the user or analyst about the advantages which the user may get using the product. It has a sleek and classy professional UI. The design is kept as user friendly as possible without doing an compromise on the look and feel.

The design has the following features.

- Landing page.
- About.
- Learn more.
- How to page.
- Input and Output page.

User Interface is built using the following

- Html
- Cascading Style Sheets (CSS)
- JavaScript (Client side)
- Bootstrap 4
- Pixabay pictures.

V. SYSTEM REQUIREMENTS SPECIFICATION

A. 1. Product Perspective

1) Product Features:

- It allows the cyber analysts to take threats on actions faster than traditional human analysts.
- Good, advanced tools can take actions on themselves even without human interaction.
- Self learning tool. A new kind of threat can be easily learnable by these models.
- These tools can be installed as a side tool of a network with not much overhead.
- All the actions performed by this tool are visible to analysts. Analysts can also set alarms when large anomalies are detected. This can work 24*7. So analysts just need to go through actions performed by it to see for any changes.

2) Operating Environment: The Operating environment depends on where it has been deployed. This can be deployed on a large corporate network. In this case, this can be implemented in network firewalls. If this is deployed on a large website, this can be implemented in firewalls or in databases. This can be implemented on any operating system. This can also be implemented on databases.

3) General Constraints, Assumptions and Dependencies:

- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units:

"Wb/m^2" or "webers per square meter", not "webers/m^2". Spell out units when they appear in text: ". . . a few henries", not ". . . a few H".
- Use a zero before decimal points: "0.25", not ".25". Use "cm^3", not "cc".)

4) Risks:

- Since this may require good hardware, limiting the hardware might cause the efficiency of the system to go down even below what humans can do.
- This cybersecurity tool scans user info. So, it is always risky what this might do to the data.

B. Functional Requirements

- The tool requires data with anomalies and should be supervised by analysts
- The data with a known number of anomalies or breaches should be tested against the tool for accuracy.
- Null or invalid inputs can be processed more or can be ignored. Even during production, these things should be tested.
- The tool can include many different kinds of machine learning models.
- The output should be one of the few inputs with anomalies stating the anomalies in it.
- Machine Learning models use various algorithms such as Logistic Regression, Decision Trees, Isolation Forest,etc using Python Libraries.

C. External Interface Requirements

User Interfaces

We are generating our own dataset with our custom sample e-commerce website. We convert the logs generated from the website into a dataset. We will try to detect the following frauds using unsupervised models and supervised models. Model Predictions and Accuracy can be improved by feeding an input labelling from analysts and more data logs.

Software Requirements

- Cybersecurity related tool Burp Suite: Tools for monitoring network.
- Web Technologies Front end : HTML , CSS , Javascript, Bootstrap 5, templating engines like Jinja2 , Express.js Back end : Python , Javascript with their respective frameworks (Flask , Node.js,)
- Databases : Mongoose ,ORMs like SQLalchemy with suitable databases.
- Machine Learning Models Unsupervised model such as K-means clustering, KNN,etc for more noticable and general issues such as sudden increase in activities whereas Supervised model like Logistic Regression, Decision Trees,etc for complex issues noticeable only by cyber analysts such as fraudulent transactions.

D. Non-Functional Requirements

Performance Requirement

We require that as soon as anomalies are found, it should be either put into queue if low level vulnerability for later review, else should be immediately alerted to the analyst.

Safety Requirements

- The tools should be transparent on what data it is taking or what action it has performed.
- If it detects anomalies in databases, it should take action on data that are of least importance as in personal info. Other data should be put into review by analysts.
- All the actions should be put into a log file.
- Few data can be excluded from reviewing like personal info or important data from databases.

Security Requirements

- Since it may go through all the data especially in databases, analysts should be careful not to give it all the sensitive data.
- Some actions might cause inconvenience to real users.

E. Other Requirements

- Easy management and deployment
- The software must remain resilient in the face of attacks.
- Usability, Reliability, supportability, quick response in case of an attack.
- The software must be able to adapt to any kind of new attacks and basic adaptability.
- The software is expected to give good performance with fast response to the application about the attacks.

VI. SYSTEM DESIGN

A. Design Considerations

1) Design Goals: This is a relatively newer concept. Many startups like harvest.ai got acquired by giants like Amazon for improvising in this field. So our design goal is to implement this model on a large scale website and test for efficiency.

- The system will be trained to implement both a supervised model and unsupervised model. Unsupervised model is well enough for more noticable and general issues such as account takeover and sudden increase in activities. Supervised model is for complex issues noticeable only by cyber analysts such as fraudulent transactions.
- We are going to see training on millions of logs(depending on the dataset). So our model should be efficient enough to detect attacks inside those records in minutes.
- Since we train it everyday depending on the attacks detected, we need good computing capacity. There is a separation between web servers and the model. The model works independent of the server. The server sends data asynchronously to the model.

2) Types of Frauds are to be detected:

- **Breach as terms of use**: Users buying some products in bulk possibly for resale. For example, a household doesn't require a washing machine or fridge more than one/two

in quantities. So, if a user buys these products in bulk, it is a fraud.

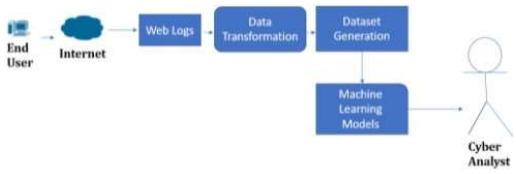
Detection: The product the user brought should be compared with other users who bought the same products. If the majority of users don't buy the products in higher quantities, it is fraud. This is similar to taking the average of quantities of all the users who bought the same product and comparing it to the users who bought it new.

- **Self-delivery fraud**: Merchants ordering and delivering for themselves and giving the orders good ratings to improve their merchant rating.

  Detection: If Many orders are for the same pincode and if all the orders received give 5 stars for the order, and if many newly created accounts order from the same merchant for the same pincode delivery, then there may be a case of fraud. The definition of many orders is that orders should have at least 20-30
- **Identity theft**: Hackers use stolen credit/debit cards for making orders for different addresses than the original user's address.

  Pre-condition: Hackers might have stolen the info from the user physically or from phishing. The original user should have previously entered his card info in the site and must have stored his delivery address in the account.

  Detection: If we detect that a new user is adding an already added card info of some other user or if the recent login to an account is trying to change address and order as many items as possible in a short span of time, there can be a case of fraud. Definition of a short span of time is if the user was ordering on avg 12 items or 5000 worth of goods in a week/month and if there is a double in this count now, there may be a case of fraud. [5]



Event Driven Architecture

3) Architecture Choices: The System follows an Event-driven architecture. The set limit for each account by the model triggers when the user activity in that account exceeds the limit. So our model structure is independent of the structure of the web server model. Web server asynchronously sends the data to our machine learning model and our model processes and reports the anomalies to the cyber analyst. The structure of the model is as follows:

- An outside web server sends data to the model. The model consists of a powerful server that trains the data both supervised and unsupervised.
- The human analyst reads through the report from the model and trains it by reporting false positives and positives.
- The model can either store the data with itself or send it back to the web server for reporting.

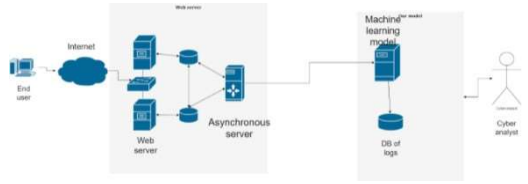
4) Constraints, Assumptions, Requirements and Dependencies: We assume that web owners are willing to take risks of sending their user activity data to us complying to laws of the country. We report whatever is given and all the activity of the model can be seen in logs.

We are training a machine learning model, so it needs good computing requirements. The model's accuracy depends on the amount of data web owners are willing to share. A good machine learning model is data hungry.

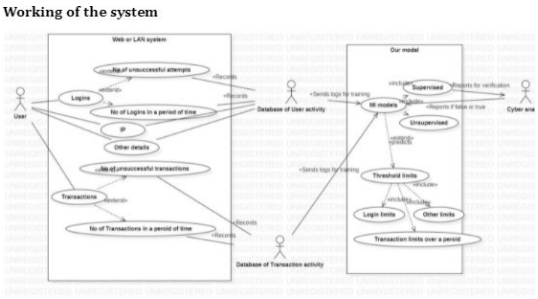
- **Interoperability requirements**: The requirement for this is that web servers should send the data from machine generated logs, json, csv or any machine generated,automated logs.
- **Interface/protocol requirements**: There are no requirements for interface or for the display of the output as the requirements for such lies in the hands of the web owners who take this service.
- **Data repository and distribution requirements**: The model takes and stores the data given by the web server. But the model resides in the server controlled by the website owners and is processed locally. So no distribution requirement.
- **Efficiency**: The efficiency of the model depends on the machine learning model implemented and the amount of data it gets. It also depends on the proficiency of the human analyst.
- **Performance**: The performance of the model is the time required for the model to train and report the anomalies. The model should be able to report within minutes of seeing the attack in the dataset of millions of logs. The train requirement is, the model needs to be trained at the end of the day for the next day, so, less requirement.
- **End-user requirement**: The end user here is not the clients of the web server. Instead are the web owners themselves. So there are no requirements for the end user here.
- **Availability of Resources**: The model can be trained in a separate powerful machine. It requires good computing capacity.
- **Hardware or software environment**: Hardware requirement is as mentioned above. Software requirement is any linux based system and access to the internet.
- **Deployment**: Deployment might require intervention of humans and maynot be just an installer. There needs to be a check for the type and format of the data. The model might require change in the format of data it receives. It also requires to be trained initially for a few days before being deployed.

B. High Level System Design

The customer of the website accesses the web server which in turn stores data related to user activity. The data or logs are sent to an asynchronous server in the web server which sends it to the cybersecurity model. The model stores it in a file



server. The data is then trained and verified by cyber analysts from the file server.



Above is the overall working of the system.

A user is a guest visitor until he logs in. A guest visitor after successful login becomes a user. The number of unsuccessful and successful logins is recorded. They are sent for training for the model. It is individual user based training. It sets the limits based on the usage on the account.

A user can perform as many transactions as he can. The model sets the limits based on previous account activities. Too much increase in transactions sets the trigger for the model.

The model can train based on the input by the human analyst. After it has trained, it sets the limits for individual user accounts based on previous activities. It does both supervised as well as unsupervised training.

Interface

There are no interfaces required since it doesn't face any users of the website. It just needs to be reported to a human analyst. The output can be of any format comfortable to the analyst such as json, or the link to the page for the user account monitoring page.

Hardware requirement

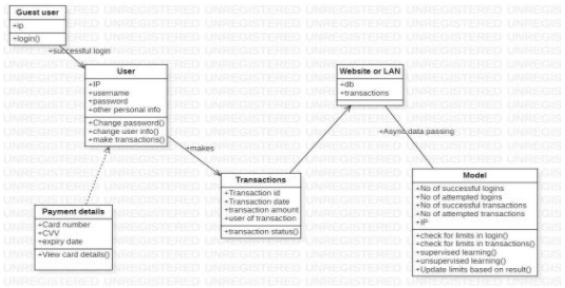
Since this trains a machine learning model, the system will require a high throughput machine. This can be kept separate to the web server since data can be passed to this system from the web server.

C. Design Description

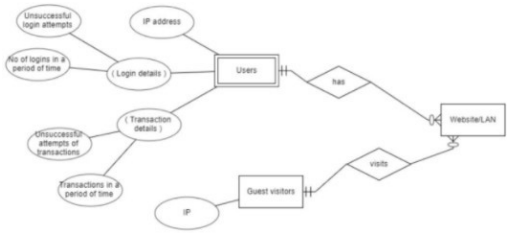
1) Master Class Diagram: Before describing the relation between the model and the web server, the relation between the website and the user needs to be understood before deploying.

A guest user becomes a user after logging in. The number of times logged in and changes in the 2 factor authentication data and related login data is stored and sent to the model.

Same rule applies for transaction data for the account. All the related data is sent for the model for analysis.



The model then sends the result to the analyst for analysis. The result from analysis obtained is again trained by the model.



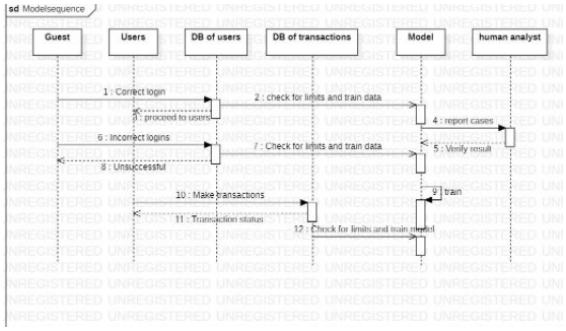
2) ER Diagram / Swimlane Diagram / State Diagram (include as appropriate): ER diagram above showing typical structure of a web server system. These are the parameters we are going to train on the AI model.

#	Entity	Name	Definition	Type
		ENTITIES		
1.	Users	Website accounts	The visitors to the site with the account in the site.	Guest users and logged users
2.	Website/LAN	The website	The website and it's associated database.	website,server or local network
#	**Attribute**	**Name**	**Definition**	**Type (size)**
		DATA ELEMENTS		
1.	Login related details	Login data such as login times and 2-factor auth	Has login data for training on account activity	login data and 2-factor data
2.	Transaction related data	Transaction data	Transaction data for fraud detection	Successful,unsuccessful transactions

Sequence Diagram

The Sequence diagram of how the system is going to report to the human analyst. Whenever a user logins or performs the transactions, the details are sent to the model in the asynchronous manner, so the web server performs with no lag.

3) User Interface Diagrams and External Interfaces: Since the model gives output to a cyber analyst and not to end users, there is no graphical interface. It can produce output in the form of a csv,log or json file. The outputs can be customised according to need.



Since it is not for the end user, It is a standalone system without any external interfaces.

4) Packaging and Deployment Diagram : Deployment requires implementation of machine learning models first. Since it is out of scope of this document, there is no deployment diagram.

5) Tools and modules: Decision trees,hidden markov model for machine learning models. Any backend technologies like Node.js, Flask can be used for testing. Burp-suite is used for monitoring networks.

6) Design Details:

- **Novelty** : This is a newer concept since machine learning is being applied to the cyber security field. The cyber security field was previously dominated by human resources with fewer tools and softwares for help.
- **Innovativeness** : Many newer softwares such as harvest.ai are being acquired by software giants such as amazon. This shows it is a new field with lots of opportunities.
- **Interoperability** : Here the intended software being developed works in conjunction with the web backend. It monitors traffic and scans logs. So it is a side package for existing web servers.
- **Performance**: We are yet to have an idea of performance for this since it is a new platform. But we are going to see millions of logs being trained by the model in an hour.
- **Security**: Since it monitors traffic and user activity, we are going to be transparent about what it scans and what it is going to report to cyber analysts. They can see what this model can see.
- **Reliability**: The idea of reliability depends on the efficiency of the model after training it. It's efficiency will increase on time. So reliability depends.
- **Maintainability**: This requires cyber analysts to report for false positives or true cases. So it needs to be maintained daily.
- **Portability**: This is easily portable. But the model might need re-training based on change in platform or data.
- **Legacy to modernization**: Legacy systems are the manual tools and softwares cyber analysts use. This automates that.This can be reused on many systems. But may require training on newer data.
- **Application compatibility**: This may be only compatible with common web server systems such as linux systems. Further may be dependent on packages such as tensorflow or pytorch.
- **Resource utilization**: This may be heavily CPU/GPU intensive. This requires training and so will use a lot of resources almost every day.

VII. Conclusion and Future Works

The Project is about Detecting vulnerabilities and frauds in online e-commerce or transaction related websites. We have done research in the domain of Application of Machine Learning in the field of Cybersecurity frauds.. We narrowed down Our Project to detect three types of frauds namely,

- **Breach of terms of use**: Buying items in excess possibly for resale.
- **Self-delivery fraud**: Merchants buying for themselves for good ratings.
- **Identity theft**: Stolen cards used for fraud purchases.

The Project uses data analysis such as outlier detection and machine learning techniques to detect and analyse a large dataset of logs. We have implemented three major models to detect those frauds

- The **First model** can be implemented by outlier detection of user activities over a period of time. The model thus trained by this is again trained when it gets a fresh set of data from the website on the next day or another week.
- The **Second model** detects merchants ordering for themselves and giving high ratings. The plan to detect this is by looking at the deliveries and rating for the different products from the same merchant. All high ratings and same delivery locations raise suspicion.
- The **Third model** detects card frauds where stolen cards are used for an unusually high number of purchases.

Our Goal is to reduce the amount of logs a human needs to go through to detect anomalies in the user accounts. We have combined all these models and made a system with the dashboard and trained the newly available data with the model. So The model becomes more and more accurate overtime. Thus developed models can adapt to many different sites as it can keep on training and can become more and more accurate overtime.

As future work,we could look at extending number of frauds detected in Our Website Analyst.io and present more insights on the data representation and visualizations to enable the users to grasp the data distribution in a more efficient and convenient manner.Looking at other vulnerabilities like card testing,friendly fraud, account takeover fraud etc.Integration with other log management tools can be supported in the future.

References

[1] Kou, Yufeng, et al. "Survey of fraud detection techniques." IEEE International Conference on Networking, Sensing and Control, 2004. Vol. 2. IEEE, 2004

[2] Tripathi, Diwakar, Bhawana Nigam, and Damodar Reddy Edla. "A novel web fraud detection technique using association rule mining." Procedia computer science 115 (2017): 274-281.

[3] Campus, Kattankulathur. ”Credit card fraud detection using machine learning models and collating machine learning models.” International Journal of Pure and Applied Mathematics 118.20 (2018): 825-838.
[4] Breier, Jakub, and Jana Brani šová. ”Anomaly detection from log files using data mining techniques.” Information Science and Applications. Springer, Berlin, Heidelberg, 2015. 449-457.
[5] Maniraj, S., et al. ”Credit card fraud detection using machine learning and data science.” International Journal of Engineering Research and 8.09 (2019).