National College of Ireland

Securing Data with User and Entity Behaviour Analysis (UEBA) Approach Using Machine Learning Models

MSc Research Project
Cyber Security

Sumeet
Student ID: x19236123

School of Computing
National College of Ireland

Supervisor: Dr. Imran Khan

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sumeet
Student ID:	x19236123
Programme:	Cyber Security
Year:	2021
Module:	MSc Research Project
Supervisor:	Dr. Imran Khan
Submission Due Date:	16/12/2021
Project Title:	Securing Data with User and Entity Behaviour Analysis (UEBA) Approach Using Machine Learning Models
Word Count:	5066
Page Count:	17

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:	Sumeet
Date:	31st January 2022

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):	

Securing Data with User and Entity Behaviour Analysis (UEBA) Approach Using Machine Learning Models

Sumeet
x19236123

Abstract

This project introduces User Entity and Behavior Analysis (UEBA) implemented using machine learning classification models to monitor the activities performed by the users like mouseover and mouseout and alert strange behavior for rectification or threat mitigation. The logs are generated based on these event activities, a baseline is then created for the usual and unusual events and detection is made by training the model and evaluating the results obtained from machine learning algorithms like XGBoost and Random Forest classification methods. User behavior anomalies are used to train the machine learning models and the results are used to prevent internal breaches and attacks that can compromise critical data. This project will find its application in organizations using critical and sensitive data to prevent leakage or exposure to vulnerabilities. The dataset for this project has been picked from Mendeley Data and Python libraries are used for this implementation. The accuracy for the implementation is observed to be 89.8% and 90% respectively for Random Forest and XGBoost.

Keywords: UEBA, Data Loss Prevention (DLP), Security Information and Event Management (SIEM), Exploratory Data Analysis (EDA).

1 Introduction

1.1 Motivation and Project Background

It has always been a concern that we are not ready for the attacks that happen around the globe. There can be many reasons behind this. Lack of knowledge, carelessness, unable to set the priorities based on the threat levels, etc. are some of the factors.

User Entity and Behaviour Analysis (UEBA) was earlier known as only User Behaviour Analysis (UBA). UEBA is a part of cybersecurity solutions that use the latest innovative technologies like machine learning and artificial intelligence to detect abnormal and malicious users, machines, and all other entities in the environment networks. Traditional tools available fail to extract the correlation rules or patterns of the attack because they involve multiple systems in an organization with data sources spread across. Since UEBA uses machine learning technologies, these attacks are easily identified and hence its importance. UEBA has three major dimensions: Use cases, data sources, and analytics as shown in Figure 1. UEBA solutions gather behavior information from users or other entities in the network to be monitored, detected, and alerted which forms the

use cases. The data source for UEBA ingested from data lakes or any other repositories are not deployed directly in the environment which might alert the hackers. UEBA solutions are smart enough this way. The analytics part of UEBA solutions uses machine learning models or any other smart technologies to detect anomalies with threat signatures and other rules. Advanced analytics should feature correlation-based analytics that is available in traditional Security Information and Event Systems (SIEMs).

UEBA use cases include malicious insiders, compromised insiders. Incident prioritization, entity analytics (IoT), Data Loss Prevention (DLP), and Data Leak Prevention. Data sources include active directory from authentication systems, VPNs and proxies from access systems, configuration and management databases, Human resource data, anti-malware, and antivirus systems, network traffic management and threat feeds. The true capability of UEBA is the Holistic analysis across organizations, IT systems, and data sources to monitor relevant data from a specific user or entity. Major advantages of UEBA over other systems include aggregation where the manual effort of analysts to review huge event floods and combine them to detect a threat becomes obscure. Through these solutions, false positives are reduced since alerts are only created after multiple signs of unusual behavior saving time for analysts. Traditional correlation rules might not apply for all users but UEBA is a smarter solution that generated a context-sensitive baseline for each group of users.

UEBA also uses these models to assess the threat level, creating a risk score that can help guide the appropriate response. Increasingly, UEBA uses machine learning to identify normal behavior and alert to risky deviations that suggest insider threats, lateral movement, compromised accounts, and attacks.

The data used for processing can be picked from general repositories, data warehouses, or through Security Information and Event Management (SIEM) which is a source of various kinds of data like log data, packet capture data along with security monitoring systems. UEBA uses cross-organizational data from security systems collected and stored in SIEM. This is the reason for this popular combination. Threat signatures, machine learning, and statistical models detect anomalies forming the analytics component.

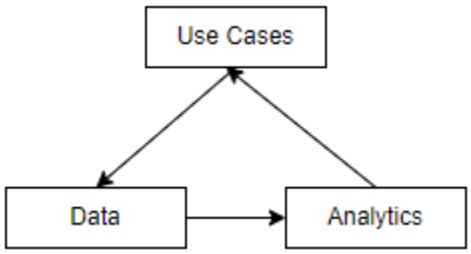

Figure 1: UEBA Layout

All possible threats even from the inside can be monitored using UEBA and not just tracking devices and events. Advanced analytics can also be used for robustness along with traditional and SIEM correlation-based analytics. Security teams usually go through all the alerts manually to catch real threats. UEBA can automate this for better efficiency and reduced error rates to prioritize genuine threats in a better way than usual. The transition from UBA to UEBA is better highlighted because of more accurate threat detection by correlating the behavior of other entities with that of the user.[1]

[1] https://www.fireeye.com/products/helix/what-is-ueba.html

1.2 Research Question and Objectives

Research Question: How accurately can a Machine Learning Model analyze and detect the malicious events to be used as a proactive measure?

- Pre-process the UEBA dataset chosen with numerical methods to select relevant user activities.
- Implement XG Boosting and Random Forest machine learning algorithms.
- Evaluate the results obtained using accuracy to check the malicious events.
- Compare the results of the models concerning performance and accuracy.

This report will highlight the research work made to these questions. All the information like data collection, tools required, pre-processing, model training is updated. The design of the project along with the implementation and evaluation is discussed with the help of plots.

2 Related Work

2.1 User and Entity Behaviour Analysis in Securing Data

Datta et al. (2021) proposed an efficient UEBA model that involves machine learning techniques. They tried to examine four various unregulated algorithms that they considered to be opposite to the general way of machine learning methods like K-Means and Agglomerative Clustering algorithms for insider threats detection. They performed the implementations and concluded that their model was more efficient will less time taken. They pointed out the UEBA tools like Forcepoint, Exabeam, Fortinet and many more that are used by different industrial companies.

The paper by Yousef and Jazzar (2021) is trying to focus on the effectiveness of UEBA from insider attacks with the help of machine learning. Based on that, many solutions are coming to the market. How this solution can act as a proactive measure is also highlighted in the later part. The advantages like less time to respond, automated threat detection, false positive reduction are highlighted. Disadvantages like dealing with user privileges, knowledge of insiders, etc. are also highlighted.

Khaliq et al. (2020) brought us to notice the values techniques and methodologies that are used in UEBA. There are many approaches like role-based detection, user and entity mapping, risk score calculation and user profiling techniques. They tried to set up their top model design so that it can generate relevant UEBA solutions. They also highlighted that open source communities are not able to provide a better solution to any threat activity in terms of UEBA.

S and Babu (2020) highlighted more on the activities performed by the users and the logs and events generated by these activities. With the help of statistical analysis and machine learning methods, they highlighted how these log events can be utilized to understand the patterns This paper highlighted how we can use this in the detection of anomalies so that the investigation can be performed. The paper also focused on user behaviors and the analysis is performed on these user behaviors to check the advantages, highlights and their usage. There were many aspects compared and analyzed like IP address, username, usage date and usage time. Then after this, it was analyzed for about

one-quarter duration. Scripting, mining and raw data processing before performing any machine learning model.

Das et al. (2017) bought a new concept of implementing UEBA using the Louvain method that can categorize to different peer groups to perform analysis. Based on the no. of employees and their features, they were able to extract the details and provide a proper description for them. So, they conducted their test into two parts. They used Niara's data to prove their robustness, scalability and simplicity. They had their data sourced at Lightweight Directory Access Protocol (LDAP). In their final step, they performed node related algorithm to get a bipartite graph. In this, we can observe how a model can work with the help of node reassignment and this way based on the feedback given, human action can be derived.

Salitin and Zolait (2018) published a paper that tried to evaluate by securing the behavior of the users that leads to the successful prevention of network attacks and attacks that are not yet performed or observed in the case of zero-day attacks. They used surveys and literature reviews of high-profile people and top vendors. These interviews and surveys are then used to check on the efficiency of the results that lie on behavior analysis. It also highlighted the advantages and the disadvantages of UEBA in case of network security that can occur at any uncertain time leading to a zero-day attack. The signature-based overview was highlighted overall manner. About the limitations, they mentioned that it is not efficient to low and slow attacks that are prevailing for a long time. Also, as per Machine learning, it is difficult to check for privileged users and the insider knowledgeable.

Shashanka et al. (2016) focussed on a high technique platform they built so that it can carry on the threat measures and based on that, there are many use cases setup that will help streamline the complete process. This will ultimately lead to the tracking of all the user and human behavior to check if any alerts are generated against those use cases. Any anomalous activities observed will throw an alert that is derived from Singular Values Decomposition (SVD). All the implementation of the machine learning model can be seen in this paper.

2.2 Big Data (UEBA) utilization with Machine Learning Algorithms

Khanna (2021) wrote a paper where he gave insights about the Computer Vision User and Entity Behaviour Analysis (CVUEBA). It is a kind of insider threat detection mechanism that is formed to gain benefits from Computer Vision techniques. VGG-19, MobileNetV2 and ResNet-50 are some models compared here. The writer has also provided details about the Feed-Forward Dense Neural Network architecture (DNN) with ReLU activations. Based on this thought, he wanted to check the performance of the CVUEBA.

Naik et al. (2021) provide an overview of the artificial intelligence and machine learning technologies used in the field of cybersecurity. They are reviewing the application which can, later on, help us in fighting cyberattacks. The effect of conditionally implementing distributed artificial techniques and easily defined artificial intelligence methodologies are highlighted. Hence, based on that whatever problems arise in the complete setup is discussed here along with its possible solutions.

Rashid and Miri (2021) mention how UEBA can be outsourced. The negative point, in this case, is privacy can be compromised as this will cause sharing of data to third parties. They implemented noise to every data that were published in case of anomaly detections

carried out. They highlighted the advantages of using their proposed scheme that includes no leakage of personal data and UEBA activities will be carried out instantly. Secondly, training to their models to avoid data exposure. They used One-Class Support Vector Machines (OCSVM), etc. and they used the Mahalanobis distance metric for statistically evaluation of their results. To calculate performance evaluation, they used Area Under the Receiver Operating Characteristic Curve (AUROC).

Amraoui and Zouari (2020) presented a framework in their paper that can secure Smart Home Systems (SHSs) which is fully automated. When these applications work automatically, they make sure only legitimate actions are to be performed, here controlled commands are used to perform and govern the action. This kind of regular activity is performed to control the Smart App with the help of the One-Class Support Vector Machine (OCSVM) for each installed SmartApp. Only those that have regular patterns are considered as correct ones and the rest others are considered as doubtful ones which will be blocked. They had AD as the basis of the framework with which they were able to experiment.

Habeeb et al. (2019) mainly focussed on the real-time network anomalies and detections, as well as checked the critical problems associated with the machine learning algorithms. At the beginning of their paper, it explained taxonomies and contexts of real-time big data processes and anomalous data. It also highlighted how machine learning was implemented to handle big data. Later on, they tried to find the possible issues faced while handling such big data processing. Real-time data is constantly received from external sources and defined time intervals that can have their speed rates. Redundancy, parameters selection, nature of input data, data visualization, etc. are some of its challenges.

2.3 Machine learning decision on Random Forest and XGBoost

Chen et al. (2021) mentioned how important we have a random forest in daily activities. This can be used to perform the risk analysis and found the accuracy rate to be 86 %. In the case of large-scale group activities, Random Forest plays a vital role in assessing the risks and predicting the accuracy for the same. It has also used deep learning which helps to understand the importance of feature learning.

Jiang et al. (2021) conducted an emergency department survey to check and recognize patients that have high risks so that resources can be allocated to them accordingly. Based on the provided suspected cardiovascular disease information, four different machine learning models were trained. It highlighted how XGBoost working was out of the box and provided almost exact pulse rates, blood pressure, oxygen saturation, etc. and much other information had significant results compared to other models. It helped them to categorize which patients are of high risk and low risk.

Xu and Yin (2021) performed Random Forest and we found factor analysis used by the authors to reduce the dimension of students used in physical education. We were able to find the improved performances of the students. The experiments that were performed showed 88.51% accuracy. The quality of good physical education classes can also be observed by the tests performed.

Ravichandran et al. (2020) presented information about the quality of fairness of XGBoost as a model. They highlight how regularising this fairness will help us remove the correlation of any kind of sensitive data and its actual values which will ultimately help us in the fairness of the model in comparison to other machine learning models.

Wang and Lu (2020) presented a new model that supports the Intrusion Detection System (IDS) in the case of IoT devices. They placed an IP server camera and then placed many IoT attacks on the IP camera which here acts like a victim. They used XGBoost-LSTM slacking model to check the numerical testing. They were able to achieve a score of 0.983 in real-world which was an amazing result.

Bentéjac et al. (2019) highlighted the techniques that can be used in the case of XGBoost that can be counted based on training speed, setup of parameter and generalized performance among the random forest, gradient boosting and XGBoost has been discussed. It also talks about the ensembles of classifiers.

Probst and Boulesteix (2017) mentioned about four goals that they performed in their paper. Chances of finding non-monotonous to be less and explained under which cases it can shrink, cannot be performed as done for Brier score and logarithmic loss, the limit of the extent to the problem in case of using a large number of datasets and arguing in favor of T settings till classical error messages. In their report, they tried to conclude that more trees are better in any way.

Chen and Guestrin (2016) talks about how XGBoost is considered as an end-to-end tree-boosting kind of system. This system is generally attained to achieve the state-in-the-art and also includes many machine learning challenges and quests. This is a system that is all about scattered awareness for a variety of data and is possibly used in understanding the tree method. This method can solve many real-time problems even without fewer resources.

2.4 Summary

Based on the above papers reviewed, we can observe that checks on anomalous behavior have always been a prime concern and organizations are trying to implement this with the SIEM solution managed in their organization. To distinguish legitimate and benign threat activities, machine learning methods of Random Forest and eXtreme Gradient Boosting can be used which can be used as an efficient approach. The focus will be on developing a model based on the dataset that can be used as a use case in a larger picture with which we can use this concept as a proactive measure in any organization.

3 Research Methodology

Knowledge Discovery in Databases (KDD) is the methodology used for this research since every process widens the choices of the user to get the best results from it. KDD uses multiple processes iteratively to extract meaningful data from a huge data source. This method includes trivial steps like data collection, data transformation and feature extraction, model implementation and result evaluation which is explained in detail concerning the question and dataset chosen for this project. Figure 2 explains the KDD process stages as implemented. Major implementation of KDD related to cyber security is in fraud detection, threat modeling and many others to reduce vulnerabilities. This method doesn't involve business implementation and hence is considered as a research-based methodology.

There were many studies and research work carried out before the selection of the dataset. While choosing the dataset, an abundant quantity of data is mentioned both row and column-wise. User and Entity Behaviour Analysis is mainly dealt with human behavior and the logs generated by the machines used. UEBA is the rising secured

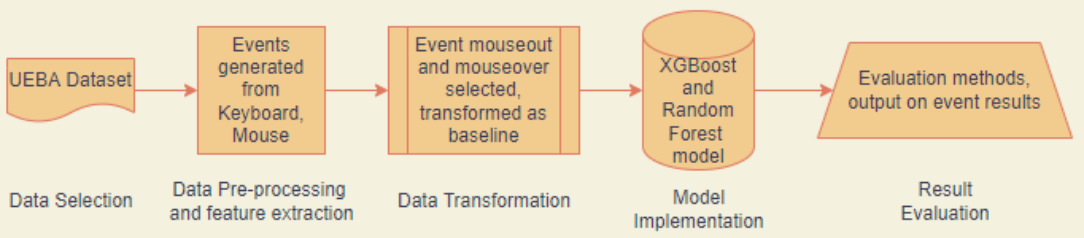

Figure 2: KDD Methodology

method companies are adopting into many SIEM solutions. UEBA related datasets were found from many platforms like Kaggle, FacebookAI, Google AI, etc.

3.1 Data Selection

The first step in KDD is to select a dataset that suits the objective of this research. This data is then transformed into a feature of importance or variables used for the implementation. The dataset apt for this project has been taken from open source Mendeley Data[2] and does not involve ethical concerns. Privacy terms and conditions are read and carefully understood before the data is used or shared. The data includes the behavioral dynamics collected from a web chat application with 142691 records. 113471 for keystroke dynamics and 29220 for mouse dynamics. This data has been gathered from almost 11 users with a mean of 28524 $\pm$ 18541 records has been obtained per user. The behavioral dynamics include 9 features like _id to uniquely identify each record, cursor, timestamp of generation, xpos and ypos for the position of the mouse of the user, key for keyboard event behavior, user for user identity and session_id. Independent variables considered for this project will be xpos and ypos and the dependent variable is the event feature used to detect behavior anomalies.

3.2 Data Pre-processing and Feature Extraction

This process in KDD involves the processing of the raw dataset into the information of interest which can be used in the implementation directly. Since this is a classification of events, mouseout and mouseover events have been chosen out of all the events available. The data has been trimmed for whitespace, null and NA values. Since the model training doesn't include string data, timestamp, user, _id and session_id have not been considered directly but at the end of the implementation, these data are referred to infer the identity of the malicious user or an attacker. Considering the above insights, the implementation-ready dataset will consist of 4 features and 7915 records of behavior dynamics.

3.3 Data Transformation

This step of KDD is to transform the data as per the requirements of the model. Machine learning models can only use numerical values and string values cannot be used for training. To keep the implementation simple, we have considered 1 for mouseover and 0 for mouseout. Correlation between variables is a measure that shows how the variables

[2] https://data.mendeley.com/datasets/f78jsh6zp9/2/files/ed39f7ca-e77d-44b2-9df3-39bbf9e0f4fb

are related linearly. Since XGBoost and Random Forest suffer from handling correlated variables and to improve the accuracy of the detection, the correlation between the variables is checked by drawing heatmaps and pair plots. Once this is complete, the data is now ready to train the models with user behavioral patterns.

3.4 Machine Learning Models

Machine learning models were chosen since they are robust and reliable. Machine learning models used for this behavioral event detection of the UEBA dataset are XGBoost and Random Forest algorithms. Below is the justification for why the above methods were used.

XGBoost is an efficient and flexible algorithm under the gradient boosting framework that uses parallel tree boosting and was built for the sole purpose of performance and computational speed. Boosting algorithms works by adjusting the weights obtained from the previous classification. This algorithm is designed to handle the missing data as well. This is the most famous algorithm in machine learning due to its ability to produce highly accurate results. Since the data here is non-continuous and is likely to be sparse, this algorithm is the best choice to be implemented. This algorithm is suited when the number of features is less than the records and is a mixture of numerical and categorical variables.

Random Forest is a classification method that constructs multiple decision trees at training time and the output is the class that is selected at the end of most trees. This is a technique that combines many classifiers to conclude even if it's a complex problem. This is a simple method to understand and trace with performance as an added benefit. This method automates missing data, works well for numerical and categorical data solving overfitting problems as well.

3.5 Result Evaluation

Once the model is created, results are evaluated based on Precision, Recall, f1-score and Accuracy.[3].

Precision refers to the ratio of relevant counts with the retrieved counts. It will point out only the events which are required based on the requirement

$$\text{Precision} = \frac{|\overline{Relevant\ Data}| \cap |\overline{Retrieved\ Data}|}{|\overline{Retrieved\ Data}|} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Figure 3: Precision

Recall refers to the ratio of relevant counts that are successfully retrieved. This can gather events that are only relevant to the detection and analysis.

F1-score is the ratio of the harmonic mean of the precision and the recall. When we combine the values of precision and recall, we will get f1-score.

[3] https://en.wikipedia.org/wiki/Precision_and_recall

$$\text{Recall} = \frac{|Relevant\ Data| \cap |Retrieved\ Data|}{|Relevant\ Data|} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Figure 4: Recall

$$\text{F1-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Figure 5: F-1 Score

Accuracy refers to the ratio of True events and all possible events (True events and False events). This will finally give the output of true events, that is events that are malicious and non-malicious. There is no such case as mid-malicious.

$$\text{Accuracy} = \frac{True\ Events}{True\ Events + False\ Events} = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative}$$

Figure 6: Accuracy

The results of precision and recall can help to understand which values are positive and which ones are negative. This is very important to understand the filtration of events based on their severity of infection.

4 Design Specification

4.1 Architecture

Figure below shows the architecture of the entire project. This can be classified into four major stages.[4] The first step is the data preparation where is collected from Mendeley data, the entities like mouse events are filtered and their behaviors are filtered. Data is pre-processed and transformed to the relevant format. Behaviour Profiling consists of the baseline that is either created or assumed in this case. Here, mousemove events were categorized as 1 and mousemove events were as 0. Events associated with mousemove will be considered as True positive and that of mousewheel will be considered as False positive. When all the above three are set up, anomaly detection will come into place. As the approach is to find out only mousemove events, so this event is treated here as an anomaly. The mousewheel event is treated as a benign event. Here, after implementation confusion matrix is generated with the results. With this detection, predictions were made to find out the list of true positive and false-positive events.

[4] https://www.semanticscholar.org/paper/User-and-entity-behavior-analytics-for-enterprise-Shashank a0356ec411bc7fc716a37e79fc85902a9e03378e/figure/2

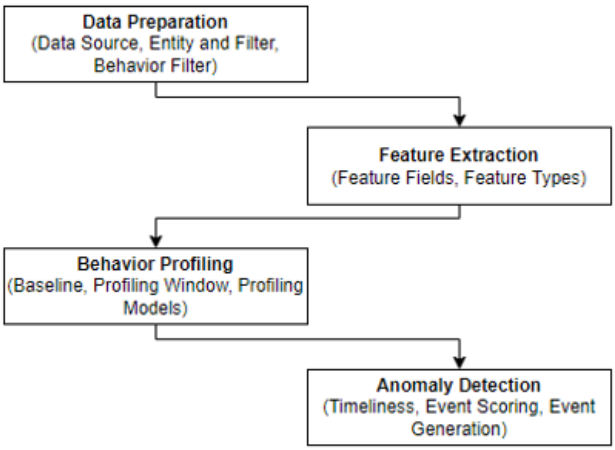

Figure 7: UEBA Architecture

4.2 Model Implementation

To find the malicious events in the environments based on the log events, the below basic process can be used. Since we are not connected with the live environment, a dataset with UEBA events is used to check the working of the model. The complete report shows the detailed performance of the model as depicted in below figure. There are a few advantages and disadvantages that are also highlighted.

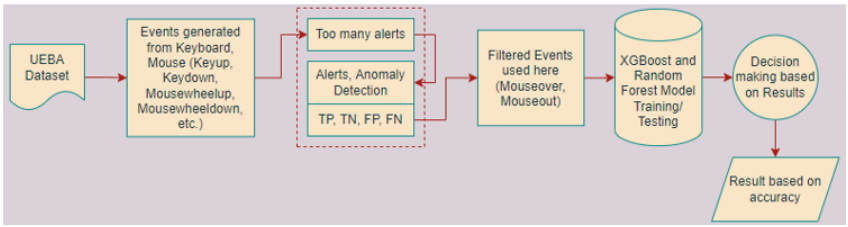

Figure 8: Process Flow

5 Implementation

5.1 Data Pre-processing

Exploratory Data Analysis (EDA) was taken into consideration after we have clean data with us. EDA is used to find patterns, anomalies, or any relationships to pass the relevant information. Figure 5 shows a pair plot used to check the distribution of both the single variables and the multiple variables, i.e. the link of the relation between two variables. Here, the Seaborn library pairplot function is called which passed out the dataframe.[5] The histogram which is on the diagonals conveys the distribution of the single variable while the scatterplots are present on the upper and the lower half of the triangle formed.

[5] `https://towardsdatascience.com/visualizing-data-with-pair-plots-in-python-f228cf529166`

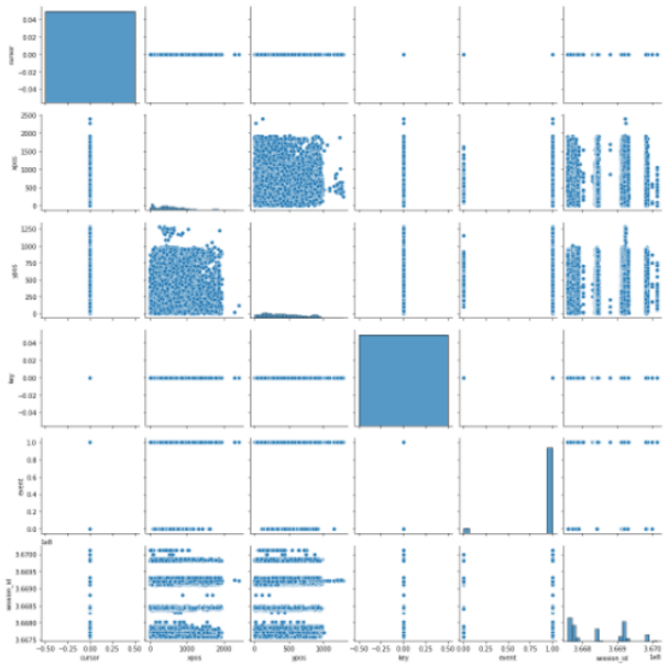

Figure 9: Pair plot for the dataset

Below figure shows a heatmap used to plot complex data figures so that they can be measured as a whole dataset at once. The values in the heatmap below are very low and are indicated by blue. This shows that the variables are not correlated.

Figure 10: Heatmap for variables

5.2 Model Implementation

The pre-processed dataset has been divided into train and test data in 65% and 35% respectively. The model is trained with train data and then the test data is used to classify the event behavior. The classified data is compared with the previously populated

data to check for accuracy. Figure 11 shows the XGBoost model fitting parameters with the train data and Figure 12 shows the same for the random forest.

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, enable_categorical=False,
              gamma=0, gpu_id=-1, importance_type=None,
              interaction_constraints='', learning_rate=0.300000012,
              max_delta_step=0, max_depth=6, min_child_weight=1, missing=nan,
              monotone_constraints='()', n_estimators=100, n_jobs=4,
              num_parallel_tree=1, predictor='auto', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
              tree_method='exact', validate_parameters=1, verbosity=None)
```

Figure 11: XG Boost model fitting

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

Figure 12: Random Forest Classifier Model Fitting

5.3 Feature Importance

Below figure shows the feature importance plot for variables. It is used to point out the feature selection in the dataset. From Figures 13 and 14, we can conclude that results mainly depend on two factors, xpos and ypos. These values help to get the accuracy more precise. These attributes are compared with each other and explicitly provide the calculation of the result.[6]

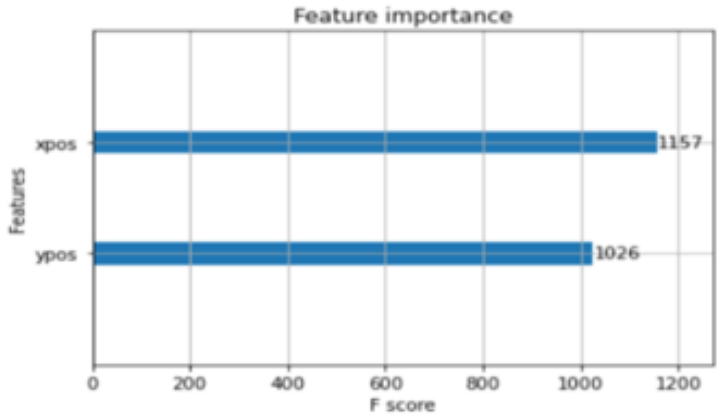

Figure 13: Feature Importance for XG Boost

[6] https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-pyth

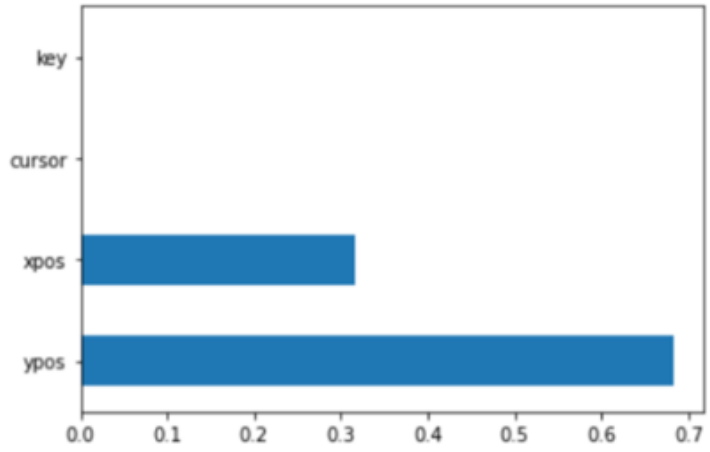

Figure 14: Feature Importance for Random Forest

6 Evaluation

After the implementation of the model, plots were generated to find out the outcome of the model implemented. Figure 15 shows the confusion matrix and classification report for XG Boost. The confusion matrix shows the summary of correct and incorrect predictions. Classification report for XGBoost shows the accuracy as 91%, precision as 88%, recall value 82% and fi-score as 85%. Here, it gives a clear decision over the analysis for any individual events. Malicious events will be considered fully malicious and non-malicious events. There is no such mid-malicious category in this case.

```
------CONFUSION MATRIX------
[[2116   68]
 [ 192  395]]
------CLASSIFICATION REPORT------
              precision    recall  f1-score   support

           0       0.92      0.97      0.94      2184
           1       0.85      0.67      0.75       587

    accuracy                           0.91      2771
   macro avg       0.88      0.82      0.85      2771
weighted avg       0.90      0.91      0.90      2771
```

Figure 15: Confusion matrix and classification report for XGBoost

Figure 16 shows the confusion matrix for the random forest with total positive and false negative consisting of major counts which are considered to be a good result. The classification report gives the accuracy as 90%, a precision score of 82%, recall value 87% and f1-score value of 84%. With this, accuracy based on the checks whether the events are malicious or non-malicious is fully justified as it shows high accuracy to check if any events are malicious or non-malicious.

```
array([[2101,  190],
       [  83,  397]], dtype=int64)
```

```
print(classification_report(predForest,y_test))
```

	precision	recall	f1-score	support
0	0.96	0.92	0.94	2291
1	0.68	0.83	0.74	480
accuracy			0.90	2771
macro avg	0.82	0.87	0.84	2771
weighted avg	0.91	0.90	0.91	2771

Figure 16: Confusion matrix and Classification report for Random Forest

6.1 Discussion

As per the above evaluation, the confusion matrix for Precise, recall, f1-score, and accuracy is generated. When we see the precision gives higher values than recall, this means the algorithm that is used here will be providing us more relevant information than irrelevant ones. The values we received for the f1-score are higher than recall and precise. This helps us to understand if the dataset has been completely used with the help of a given model.

Anomalies can be effectively detected based on the behavior profiles and observations can be made. Mouseover and mouseout events when taken to consideration, the higher values associated with events can be derived during XGBoost for Precision and F1-score and Random Forest in case of Recall. There is a slight difference between the accuracies of XGBoost and Random Forest. Based on the 65-35 train and test percentage and the above outcomes, we can still conclude that when it comes to User Behaviour Detection and Analysis, XGBoost will generate higher accuracy results. This way, there are fewer chances to miss any malicious behavior events that can harm the environment.

Data Loss Prevention can be achieved if it is assured that the detections made with the model are highly accurate and after the observations of the results, this conclusion can be achieved. This can avoid huge time loss in the case of DDoS, where there is a bombardment of attack attempts and can impact the data of an organization. UEBA model created with the help of Machine Learning Algorithm can act here as a proactive measure and this situation can be avoided.

Table 1: Results Comparision

Model	Precision (%)	Recall (%)	F-1 Score (%)	Accuracy (%)
XGBoost	88	82	85	91
Random Forest	82	87	84	90

7 Conclusion and Future Work

The main objective of the project was to provide an answer to the research question How accurately can a Machine Learning Model analyses and detect the malicious and non-malicious events which can be used as a proactive measure? To find the potential and performance of the model, a Classification report, confusion matrix was used to perform the evaluation. Precision, Recall, F1-score, accuracy were plotted to discuss the efficiency of the model. Since there is a lack of time that made us work on the limited datasets with limited information. Hence, the prediction is based on those datasets. In the case of the future, it can be performed with better hardware and software, larger datasets, and more number events.

Through this project, the research question that if machine learning models like Random Forest and XG Boost can successfully analyze and detect the malicious events from the UEBA dataset used here has been answered with high accuracy results of 90% and 91% respectively. Artificial Intelligence and Machine Learning can be an efficient way to implement UEBA with high-achieving goals that can help to prevent random cyberattacks around the globe. Once these datasets, or logs, are analyzed and implemented with UEBA fundamentals, the environment can behave as a proactive shield that can help us from attacks that are available outside as well as inside the network. Future work in this regard will be to use a real-time dataset containing a more and wide range of features in this emerging hacking world outside and use machine learning as a proactive measure that may be used in developing antivirus or antimalware software.

8 Acknowledgement

My special thanks of gratitute to my supervisor Dr. Imran Khan who gave me an opportunity to complete this idea into execution. Also his guidance has been very helpful in completing the artifacts in a right way.

References

Amraoui, N. and Zouari, B. (2020). Behavior-based anomaly detection for securing smart home systems automation, *2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1–6.

Bentéjac, C., Csörgo, A. and Martínez-Muñoz, G. (2019). A comparative analysis of xgboost, *CoRR* **abs/1911.01914**.
URL: *http://arxiv.org/abs/1911.01914*

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system, *CoRR* **abs/1603.02754**.
URL: *http://arxiv.org/abs/1603.02754*

Chen, Y., Zheng, W., Li, W. and Huang, Y. (2021). Large group activity security risk assessment and risk early warning based on random forest algorithm, *Pattern Recognition Letters* **144**: 1–5.
URL: *https://www.sciencedirect.com/science/article/pii/S0167865521000192*

Das, A., Shen, M.-Y. and Wang, J. (2017). Modeling user communities for identifying security risks in an organization, *2017 IEEE International Conference on Big Data (Big Data)*, pp. 4481–4486.

Datta, J., Dasgupta, R., Dasgupta, S. and Reddy, K. R. (2021). Real-time threat detection in ueba using unsupervised learning algorithms, *2021 5th International Conference on Electronics, Materials Engineering Nano-Technology (IEMENTech)*, pp. 1–6.

Habeeb, R. A. A., Nasaruddin, F., Gani, A., Hashem, I. A. T., Ahmed, E. and Imran, M. (2019). Real-time big data processing for anomaly detection: A survey, *International Journal of Information Management* **45**: 289–307.
URL: *https://www.sciencedirect.com/science/article/pii/S0268401218301658*

Jiang, H., Mao, H., Lu, H., Lin, P., Garry, W., Lu, H., Yang, G., Rainer, T. H. and Chen, X. (2021). Machine learning-based models to support decision-making in emergency department triage for patients with suspected cardiovascular disease, *International Journal of Medical Informatics* **145**: 104326.
URL: *https://www.sciencedirect.com/science/article/pii/S1386505620313873*

Khaliq, S., Abideen Tariq, Z. U. and Masood, A. (2020). Role of user and entity behavior analytics in detecting insider attacks, *2020 International Conference on Cyber Warfare and Security (ICCWS)*, pp. 1–6.

Khanna, S. (2021). Computer vision user entity behavior analytics, *CoRR* **abs/2111.13176**.
URL: *https://arxiv.org/abs/2111.13176*

Naik, B., Mehta, A., Yagnik, H. and Shah, M. (2021). The impacts of artificial intelligence techniques in augmentation of cybersecurity: a comprehensive review, *Complex & Intelligent Systems* .
URL: *https://doi.org/10.1007/s40747-021-00494-8*

Probst, P. and Boulesteix, A.-L. (2017). To tune or not to tune the number of trees in random forest?, *Journal of Machine Learning Research* **18**.

Rashid, F. and Miri, A. (2021). User and event behavior analytics on differentially private data for anomaly detection, *2021 7th IEEE Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pp. 81–86.

Ravichandran, S., Khurana, D., Venkatesh, B. and Edakunni, N. U. (2020). Fairxgboost: Fairness-aware classification in xgboost, *CoRR* **abs/2009.01442**.
URL: *https://arxiv.org/abs/2009.01442*

S, R. and Babu, S. (2020). Detecting anomalies in users – an ueba approach, *Proceedings of the International Conference on Industrial Engineering and Operations Management (IEOM) 2020*.
URL: *http://www.ieomsociety.org/ieom2020/papers/632.pdf*

Salitin, M. A. and Zolait, A. H. (2018). The role of user entity behavior analytics to detect network attacks in real time, *2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pp. 1–5.

Shashanka, M., Shen, M.-Y. and Wang, J. (2016). User and entity behavior analytics for enterprise security, *2016 IEEE International Conference on Big Data (Big Data)*, pp. 1867–1874.

Wang, X. and Lu, X. (2020). A host-based anomaly detection framework using xgboost and lstm for iot devices, *Wireless Communications and Mobile Computing* **2020**: 1–13.

Xu, Q. and Yin, J. (2021). Application of random forest algorithm in physical education, *Scientific Programming* **2021**: 1–10.

Yousef, R. and Jazzar, M. (2021). Measuring the effectiveness of user and entity behavior analytics for the prevention of insider threats, *Xi'an Jianzhu Keji Daxue Xuebao/Journal of Xi'an University of Architecture & Technology* **XIII**: 175–181.