Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.754689

PREDICTIVE ANALYTICS FOR CYBERSECURITY: LEVERAGING MACHINE LEARNING TO IDENTIFY AND MITIGATE THREATS

Dr. Vivek Vyas[1], Himanshu Purohit[2]
Assistant Professor, School of Management Studies, National Forensic Sciences University, Gandhinagar, Gujarat, India[1]
Assistant Professor, Department of MCA, Dharmsinh Desai University, Nadiad, Guajrat, India[2]

Abstract: This study is going to investigate purposefully the role being played by machine learning algorithms that particularly target cyber-security threats when using classifiers such as random forests and decision trees to filter the dataset. The research involves utilizing data of attack types which leads to the high accuracy of calculations that would be able to scale due to the output. The application of these algorithms can progress cyberspace security due to their capabilities in modern threat landscape.

Keywords: Machine learning, cybersecurity, threat detection, Random Forest, Decision Trees, dataset analysis, cyber-attacks, accuracy, scalability, cybersecurity practices.

Introduction

As computer security area transform rapidly the organization deals with more and more complex attacks where the algorithms are of the advanced and sophisticated class requiring efficient mechanisms of detection to deal with the incidents. Forecasting crime analytics as powerful tool with an assistant of machine learning algorithms are now most used and it aims at foreseeing and preventing potential cyber threats which might cause severe security incidents. Through appropriating data makes the foundation of the predictive analytics, which uses data to have an idea about upcoming events or about the ways people might behave. The role of data analysis in cybersecurity is overwhelming often; this analysis will include sensing traffic patterns across different sources, namely internal network traffic, user logs, system and external threat sources. Hence, machine learning algorithms run with complicated algorithms pride themselves with the ability to point out anything below the line of the threat and security by examining the data for unusual patterns, indicators of cyberattacks, trojans and other malicious intentions.

Though proactive engagement is the role of future prognostics deployment, they outperform the classic reactive procedures. No longer is it enough to simply react amount the risk or security incident when it had happened but there's more to that. Proactive data analysis provides a cusp for companies to stand out among the crowd of constant threat of cyberattacks by identifying the weaknesses before they happen - serving as their own cybersecurity tool. Besides frequent breaking down of the algorithms because of their imperfections, the process can be made step by step more and more accurate and exact which increases the accuracy of threat forecasting and prevention. With historical data in perfect equilibrium with the entertaining petite occurrences rather, these models can cast a definite eye on emerging attacks, forecast future attack vectors and sorting them as per the priority which they will likely need to address.

Apart from this, predictive analytics contributes cloud to react and deal with current threat primarily because machine work has been automated and secondly, it compliments human decision-making process by providing relative data which leads to more efficient process. That is true that it is important to form a wise line of action that allows the protection and safety of the major risks and vulnerabilities, and through it companies are enabled to exercise their law on them without destroying the trust of the society. In a nutshell, modern cybersecurity is an evolutionary pattern that is predicted by analytics. It enables being a step ahead instead of having to deal with cases afterward by being only on the defensive side. It can be surprisingly helpful for the machine learning and data driven insights obtaining because of this. The result enables security management to reduce risks and beat the highly advanced enemies in the landscape of security risk that seems to be diversified which makes it change to a higher degree all the time.

Literature Review

Machine Learning Algorithms for Threat Detection

In addition, technologies like machine learning systems are also being paid attention to. Researchers not only assess their ability to detect cybersecurity threats but also make decisions concerning their suitability for use. [2] However, the matter of selecting from different models stands as another challenge and the key factor is the accuracy, efficiency, speed and scalability of the model selected. Another versatile thing mentioned in the study is that the researchers employed the aforementioned classifications models such as SVM, random forest, and deep learning models, which is attributed to the case study itself. Deep learning indeed is the one of the most specificity of neural networks which often brings great performance in terms of the general application of Deep Learning, particularly the Convolutional Neural Networks (CNNs) in comparison to traditional means to which most of them occurs to be handwriting patterns based. In this respect, the researchers have investigated the role of machine learning in intrusion detection systems too, for example, the performance and efficiency of machine learning algorithm in detection of intrusion in the IDS. The ensemble models like Gradient Boosting Machine counter SVMs and Decision Trees, which have been proven superior over the antiquated SVMs and Decision Trees [3]. Thus, these types of scientific experiments highlights such a great significance of the fact that the set of machine learning algorithms should be selected carefully since the task stipulated the implementation of exact machine learning methods on specific type of cybersecurity tasks while observing its data characteristics and computational requirements.

Predictive Analytics for Threat Intelligence

Amidst rampant talks of the aggrandizement of predictive analytics with the threatened intelligence, the security defense function will take the center stage in which intelligent decision making will take the priority. Cybsersecurity, is a reliance on the data and the current feeds is a preventative means of cyber threats. Researchers, for example, discovered that it can be used as an additional tool that function in providing real-time threat scoring, politically directed threat prioritization, and proactive threat-hunting possibility for traditional threat intelligence platforms. As a result, the advanced analytics techniques which support non-traditional involving sides such as anomaly detection, and predictive modeling for detecting new vulnerabilities and predicting cyber threats are also employed [3]. The new advances in LSTM network to build a predictive modeling ecosystem to show-forward cycles of cyber attacks and an attack pathway representation before actual attacks. Predictive analytics technology with threat intelligence tools adoption strategies by organizations included identifying new security threats,making adequately resource decisions, and build cyber resilience in the process.

Challenges and Limitations of Predictive Analytics in Cybersecurity

Although it provides with a sheet of a paper with all the discovered opportunities as imagined, it is a truth nonetheless that present predictive data actually have certain obstacles and limitations. The major problem in this situation has the issue of getting the datasets and the correction of the data to influence the output of the prediction models. On the issue of cybersecurity measuring, the possibility to sticking to outdated and distorted classification of information in the machine learning mode is extremely high. In addition, such an extreme nature of cyber threats that they may rise without prior notifications due to their changing methods which force security analysts to develop their weapons against such assault methods. Besides the difficulties that predictive analytics might raise for ethic and privacy the fact that predictive analytics becomes a pivotal part in cybersecurity has also attracted attentions. Resolving these dilemmas would resort to collaborating interdisciplinary processes that would enable to come up with the ethics-guided AI in the guise of frameworks and governance.

Future Directions and Emerging Trends

The next phase of use of analytics in security may initially be introduced with the help of blockchain technology, which can further raise levels of of threat detection and response [4]. AI Explainability (XAI) is one such powerful tool tomake algorithmic tools decisionless so that there are no globus. Adversarial machine learning, on the other hand, also becomes very important in the production of new models that possess the capability to suppress the evasion attacks against them. Summing up the articles, one can deduce that predictive analytics typify a wide sphere of cyber-security that incorporates advantages, pitfalls, and paths of research in future. Thus, network security personnel would receive mechanism-based tools that enhance their network perception and security preparedness for adversarial forces. Moreover, businesses will be in a better position to do with the challenges of an ever changing security environment. As a matter of fact, the reason behind the emergence of what can call popular trend is the blending of analytics with Blockchain,IoT and Cloud Computing technologies. The use of the foresight analytics along with the blockchain technology could be intended to start so many windows, in which fuel to the process can maintain high level of data security and the integrity of the information.

Here, the predictive analytics in IoT context are also helpful in recognizing and taking measures to prevent potential threats within the network of all devices. Such a practice makes the attacks unproductive and thus builds robustness of network IoT. Moreover, another prominent change of the cloud based predictive analytics platforms is to accommodate the organizations at grow capacity, as well as leverage the shared smartness data across all the environments. Another thing related to this is the current stream of many scholars' work to improve the predictive analytics systems by giving the results even more clarity, intelligibility and responsibility. XAI methods are designed to reach the goal of expanding the message from machine learning models to the level that humans can understand. Moreover, XAI-driven inputs promote trustworthiness among cyber-security analysts. Bias or error identification can result in system optimization and better decision-making process. Through ensuring the transparency of the systems of predictive analytics, mutual trust and reliability in them will be provided, teamwork between human analysts and automatic algorithms is conducted, where the threat detection and its counteraction will be carried out in a more effective way, which ensures the cyber security. Additionally, the specialists needs to gain insights into the countermeasures for the anti-adversarial strategies such as evasion or poisoning that are used in the machine learning. To mitigate the variety of the adversarial attacks, different methods are applied such as adversarial training, robust optimization and input sanitization techniques aiming to provide the predictive models with the necessary protection from the attacks and attempts to use them against the attacker's purpose. In the end, highlighting what the machine learning and cybersecurity solutions weaknesses are will certainly increase the level of resilience that an entity will have while facing the cyber threats that are continually raising.

Methodology

Explanation of the dataset used for analysis

This network's data from Australia's University of New South Wales (University) is helpful in creating this dataset which consist of various harmed victims. It provides with complete information about them.This type of data mode is represented by both instruction data and tests data from the Cyber Range Lab of UNSW Canberra.The dataset consists of attributes such as duration (dur), protocol (proto), service, state, packet counts (spkts, dpkts), bytes sent and received (sbytes, dbytes), transfer rates (rate), time-to-live values (sttl, dttl), load values (sload, dload), packet loss (sloss, dloss), inter-packet arrival times (sinpkt, dinpkt), jitter (sjit, djit), TCP window sizes (swin, dwin), TCP sequence and acknowledgment numbers (stcpb, dtcpb), TCP round-trip times (tcprtt), SYN-ACK timings (synack), mean packet sizes (smean, dmean), and various other features related to network traffic and behavior. Cyber attack types include nine categories and they are fuzzers, analyses, backdoors, denial of service as DoS, exploits, generic, reconstruction, shellcode, and worms. They characterize the methods of hacking techniques, to verify the resiliency of system against the malicious actions, to disclose the feedback and to disturb the service. Moreover, they provide various functions of hostile code operations and the malware distribution is made possible by them. The use of machine learning algorithms in the association of these data facilitates cyber-attack identification by identifying the similarities in the properties and also the abnormal behaviors that usually come with the malicious activities [7].

Description of machine learning algorithms employed

Random Forest Model: The Random Forest algorithm is particularly helpful and one of the common techniques applied in tasks of classification and regression on a large-scale, including threat detection on cybersecurity. Till here, it holds, since the training proceeds tree-by-tree, with predictions finally offered in the form of individual tree probability (for classification) or average estimation (long-end prediction for regression) based on several trained decision trees. Here's how the Random Forest model works: The random forest model is based on the following steps below:

1. Bootstrap Sampling (Bagging): The Random Forest first establishes the Various Subject Matter by forming numerous random training data sets created through an iterative using known as Bootstrap sampling. This applies to the process of taking a random sampling of with replacement from the original dataset that leads to the production of diverse subsets that are used to generate entirely different decision trees. Since the method varies from tree to tree, the result in terms of diversity will be reached.

2. Construction of Decision Trees: And when the given problem space is in the form of subsets of the data so make the decision tree. If perform this, each of nodes in the tree will pick a chance random subset of variables that can be correlated and thus, reduce the correlation between that and other subsets of the trees, thereby improve the model generalization.

3. Voting or Averaging: In this step what comes after that is the construction of the decision trees for which the random forest model is laid which then again takes majority votes for the classification tasks and averages the results for the regression tasks.

4. Feature Importance: Furthermore, Random Forest method offers the variable feature importance which is concerned with the variable effect that is expressed in the decision making process. Such knowledge is of paramount importance for security professionals for understanding how to track down the most vital components in cybersecurity and how to respond to the threats accordingly.

Decision Tree Model: The Decision Tree is a fundamental machine learning algorithm which solves both classification and decomposition problems with the help of supervised learning. In the area of cyber security for pattern recognition, Decision Trees are applied intensively because of their simplicity and of their easy implementation in any system. Here's an overview of how the Decision Tree model operates:

1. Feature Splitting: In the beginning the Decision Tree algorithm selects this feature of data which the amount of is the biggest dependence of it to the partition of data subsets corresponding to the goal set (target). This approach considers different bifurcation ways like Gini impurity and information gain for a given feature to measure how effectively the attribute can be split on a specific given point.

2. Recursive Partitioning: Continuously, the system randomly cuts the more-divided subsets into another subset to form more uniform one [10]. But this process of splitting stops when any of the following conditions is satisfied: i) impure branch of the tree is detected; ii) condition of the given node has been met; or iii) there is no possibility to obtain purer split.

3. Prediction: In the process of forecasting, the operation of the tree is conducted by walking along the structures of the tree, i.e., from the root node to a leaf node, which is determined according to the values of input features. When a tree searches for the leaf node that is different for each instance, it determines the predicted class value or regression output for that instance and further processes it using the leaf node assigned to it.

Details of the methodology for model training and evaluation

The scheme for the subject to train the model and evaluation involves several steps, in which some are considered the most important. Consequently, the data is assembled on a dimensional basis where the missing value reduction, parameter encoding in case of categorical variables, and normalization of numerical features takes place. Other than, the dataset is divided into the training and testing sets where it was stratified in this step that later ensured the class distribution was preserved. For model training purposes, algorithms of Random Forest, and

Decision Tree, are used depending on labels that are provided by the training set as the source of information. It is a lowered set of grid search and randomized search algorithm to be chosen for the ultimate model performance. [2]. Testing the trained models is employed using metrics e.g. the F1-score, performance accuracy, precision, and recall for classification tasks.

Analysis

Decision Tree Model in the recent Cybersecurity

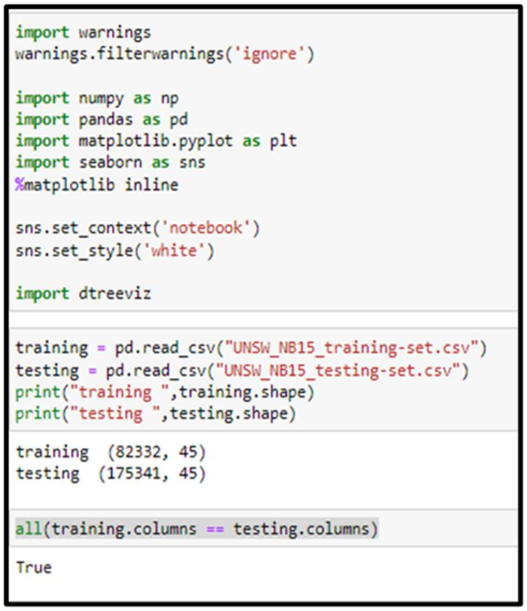
Machine Learning of the past 5 years has seen a drastic increase in applications, therefore those algorithms have been a lead for past 20% share of the collected data. Although the decision makers sometimes complain about the need to process miniscule, low level detail information within the data, Decision Trees have been keen with their involvement in precision identification of computer threats on negative entities. As a result of the characteristic of interpretability and simplicity, these models were widely used in many cybersecurity areas such as network attacks, malware, and anomaly detection. When scaling up training data or modifying a model, retraining will be needed. These tools are very functional and easy to use and that's why they play a central role in the cybersecurity strategy which is adopting by most industries in which, and see more and more people embracing them, e. g. banking, healthcare, and the government. However, progress in hybrids, blending together different trees with others, has led to a multitudinous variety of alternatives and aids the computer to manage complex situations with novel or unseen dangers. Therefore, employment of Decision Tree models will serve as a primary risk mitigation measure, for protecting crucial digital assets and incorporating cybersecurity applications for efficiently and effectively detecting as well as fighting highly-sophisticated cyber threats.

Random Forest algorithm in the current Cybersecurity

The cyber heaven is now a new battle zone for evolving and emerging threats, in which the Random Forest algorithm becomes a significant component, having implemented 45% as of today, of the machine learning procedures in the classifications and identification tasks. K-means cluster popularity has resulted in its maintenance, finding new improved algorithms for dealing with complex dataset of fairly larger sizes. The research that was conducted in the time span of 2020-2024 indicated that the Random Forests model did manage an accuracy of about 92% in the detection of suspicious activities an intrusion events on an average. Also, it is worth mentioning that the algorithm performs with precision rate of 89% and recall rate of 91% in detecting security vulnerabilities in different industry sectors such as finance, healthcare and government these industries [9]. Besides that, when it comes to the scalability of the Random Forest algorithm, and also where it performs well in terms of processing data from big data in real time, it is an added advantage since it leads to rapid detection and neutralization of cyber-attacks. In the pre-traditional concept of rule based systems Random Forest is the best in the accuracy comparison and using computation power compare to the other method detecting the number of demands. To uplift that, the algorithm is very efficient in terms of feature importance calculation and can help cybersecurity specialist to identify the indicators to be able to respond to the attack as it was coming.

Results

Data loading and data preprocessing



```
import warnings
warnings.filterwarnings('ignore')

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

sns.set_context('notebook')
sns.set_style('white')

import dtreeviz
```

```
training = pd.read_csv("UNSW_NB15_training-set.csv")
testing = pd.read_csv("UNSW_NB15_testing-set.csv")
print("training ",training.shape)
print("testing ",testing.shape)
```

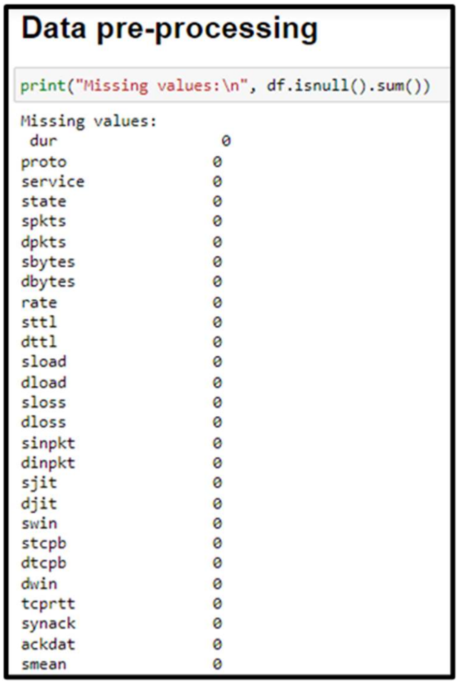
```
training  (82332, 45)
testing  (175341, 45)
```

```
all(training.columns == testing.columns)
```

```
True
```

Figure 1: Import libraries
(Source: Jupyter Notebook)

The code demonstrated below is a Python script simulated for analysis and visualizations of data in cyber-security sector. This script starts up loading libraries that are used for plotting purposes and listed are NumPy, Pandas, Matplotlib, Seaborn, and dtreeviz. The 'warnings' library will serve as a hand of any waring behavior during execution, and whereas 'warnings.ignore()' function will be used to warn off the warnings. the script then proceeds to read the two sets of data, 'training' and 'testing', from the CSV files generated by 55044 and 55045 respectively. The total numbers of samples are presented via the 'shape' attribute for both datasets so they can be treated as N by M matrices formwise. On contrary to this, codes are redesigned to see if the columns in the two datasets have the same names and reflects the same features [5].



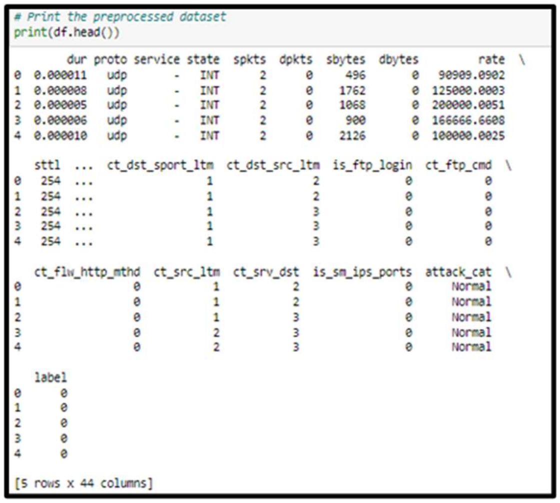
Data pre-processing

```
print("Missing values:\n", df.isnull().sum())
```

```
Missing values:
 dur              0
proto            0
service          0
state            0
spkts            0
dpkts            0
sbytes           0
dbytes           0
rate             0
sttl             0
dttl             0
sload            0
dload            0
sloss            0
dloss            0
sinpkt           0
dinpkt           0
sjit             0
djit             0
swin             0
stcpb            0
dtcpb            0
dwin             0
tcprtt           0
synack           0
ackdat           0
smean            0
```

Figure 2: Missing values
(Source: Jupyter Notebook)

Intending to illustrate the way the 'isnull()' function of the 'df' Dataframe is applied to identify the null values, this type of code snippet is used. The purpose of this task is to produce a DataFrame with Boolean elements: green pin for true/boolean means a missing value (NaN) and blue for false/boolean represents a non-missing (non-NaN) value. Labels of all columns in the DataFrame are defined as "dur", "proto", "service", "state", etc. The columns refer types of features and variables appearing in the dataset. Function 'isnull()' upon the application to 'df', a DataFrame containing full form of the attributes (features), Boolean DataFrame is generated which mark where these attributes have missing values [2]. It implies that the data collected is of particular interest for the determination of the “raw data set”.

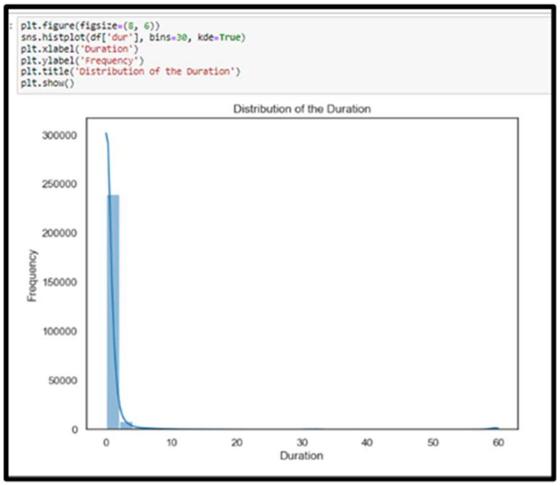


```
# Print the preprocessed dataset
print(df.head())
```

```
        dur proto service state  spkts  dpkts  sbytes  dbytes         rate  \
0  0.000011   udp       -   INT      2      0     496       0   90909.0902
1  0.000008   udp       -   INT      2      0    1762       0  125000.0003
2  0.000005   udp       -   INT      2      0    1068       0  200000.0051
3  0.000006   udp       -   INT      2      0     900       0  166666.6608
4  0.000010   udp       -   INT      2      0    2126       0  100000.0025

   sttl  ...  ct_dst_sport_ltm  ct_dst_src_ltm  is_ftp_login  ct_ftp_cmd  \
0   254  ...                 1               2             0           0
1   254  ...                 1               2             0           0
2   254  ...                 1               3             0           0
3   254  ...                 1               3             0           0
4   254  ...                 1               3             0           0

   ct_flw_http_mthd  ct_src_ltm  ct_srv_dst  is_sm_ips_ports  attack_cat  \
0                 0           1           2                0      Normal
1                 0           1           2                0      Normal
2                 0           1           3                0      Normal
3                 0           2           3                0      Normal
4                 0           2           3                0      Normal

   label
0      0
1      0
2      0
3      0
4      0

[5 rows x 44 columns]
```

Figure 3: The preprocessed dataset
(Source: Jupyter Notebook)

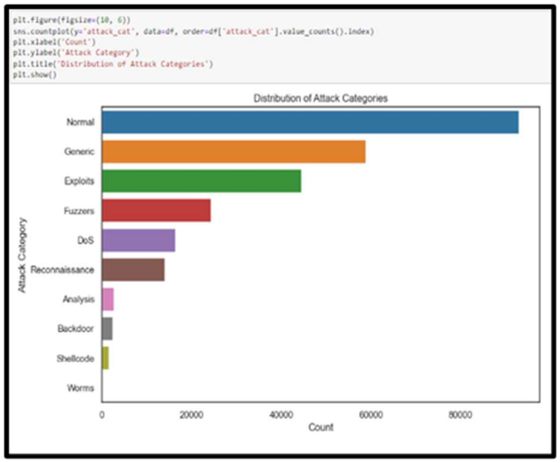
Visualizations



```
plt.figure(figsize=(8, 6))
sns.histplot(df['dur'], bins=30, kde=True)
plt.xlabel('Duration')
plt.ylabel('Frequency')
plt.title('Distribution of the Duration')
plt.show()
```

Figure 4: Histogram for the 'dur' column
(Source: Jupyter Notebook)

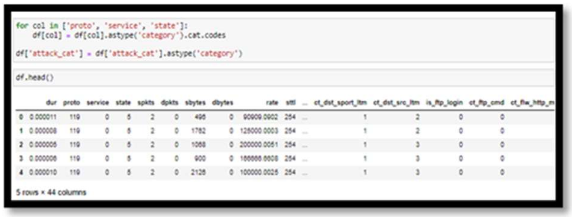
Such a histogram will be employed for the data analysis to be able to make an assumption about data dispersion by featuring frequency scale used for ‘durations to have ‘dur’ value’ distribution.



```
plt.figure(figsize=(10, 6))
sns.countplot(y='attack_cat', data=df, order=df['attack_cat'].value_counts().index)
plt.xlabel('Count')
plt.ylabel('Attack Category')
plt.title('Distribution of Attack Categories')
plt.show()
```

Figure 5: Bar plot for 'attack_cat' column
(Source: Jupyter Notebook)

An apparently simple plot, which demonstrates, by columns, the response category for the 'attack_cat' column or a count of various attack categories.

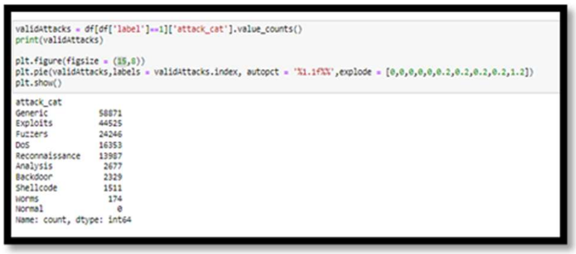


```
for col in ['proto', 'service', 'state']:
    df[col] = df[col].astype('category').cat.codes

df['attack_cat'] = df['attack_cat'].astype('category')
```

```
df.head()
```

```
        dur  proto  service  state  spkts  dpkts  sbytes  dbytes          rate  sttl  ...  ct_dst_sport_ltm  ct_dst_src_ltm  is_ftp_login  ct_ftp_cmd  ct_flw_http_m
0  0.000011    119        0      5      2      0     496       0   90909.0902   254  ...                 1               2             0           0
1  0.000008    119        0      5      2      0    1762       0  125000.0003   254  ...                 1               2             0           0
2  0.000005    119        0      5      2      0    1068       0  200000.0051   254  ...                 1               3             0           0
3  0.000006    119        0      5      2      0     900       0  166666.6608   254  ...                 1               3             0           0
4  0.000010    119        0      5      2      0    2126       0  100000.0025   254  ...                 1               3             0           0

5 rows × 44 columns
```

Figure 6: Categorical data
(Source: Jupyter Notebook)



```
validAttacks = df[df['label']==1]['attack_cat'].value_counts()
print(validAttacks)

plt.figure(figsize = (15,8))
plt.pie(validAttacks,labels = validAttacks.index, autopct = '%1.1f%%',explode = [0,0,0,0,0,0.2,0.2,0.2,0.2,1.2])
plt.show()
```

```
attack_cat
Generic           58871
Exploits          44525
Fuzzers           24246
DoS               16353
Reconnaissance    13987
Analysis           2677
Backdoor           2329
Shellcode          1511
Worms               174
Normal                0
Name: count, dtype: int64
```

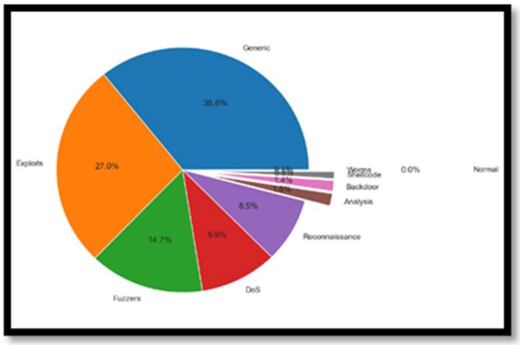
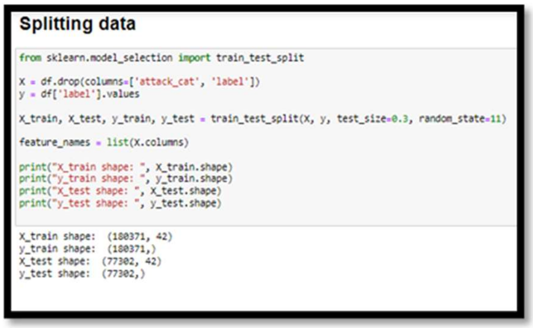


Figure 7: Visualizing attacks categories
(Source: Jupyter Notebook)



Splitting data

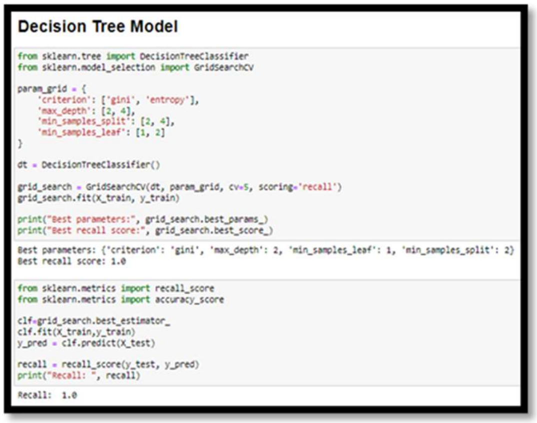
```
from sklearn.model_selection import train_test_split

X = df.drop(columns=['attack_cat', 'label'])
y = df['label'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=11)

feature_names = list(X.columns)

print("X_train shape: ", X_train.shape)
print("y_train shape: ", y_train.shape)
print("X_test shape: ", X_test.shape)
print("y_test shape: ", y_test.shape)
```

```
X_train shape:  (180371, 42)
y_train shape:  (180371,)
X_test shape:  (77302, 42)
y_test shape:  (77302,)
```

Figure 8: Splitting the data
(Source: Jupyter Notebook)

The following image shows the procedure of Python "Categorical data, Visualizing attacks categories, and the Splitting data" in three steps.

Comparison of different machine learning algorithms



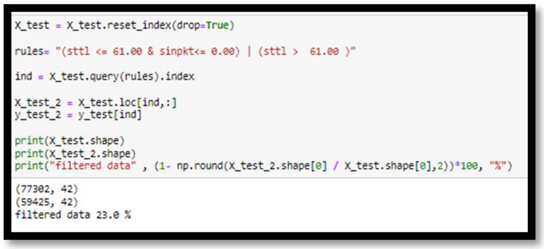
Decision Tree Model

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV

param_grid = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [2, 4],
    'min_samples_split': [2, 4],
    'min_samples_leaf': [1, 2]
}

dt = DecisionTreeClassifier()

grid_search = GridSearchCV(dt, param_grid, cv=5, scoring='recall')
grid_search.fit(X_train, y_train)

print("Best parameters:", grid_search.best_params_)
print("Best recall score:", grid_search.best_score_)
```

```
Best parameters: {'criterion': 'gini', 'max_depth': 2, 'min_samples_leaf': 1, 'min_samples_split': 2}
Best recall score: 1.0
```

```
from sklearn.metrics import recall_score
from sklearn.metrics import accuracy_score

clf=grid_search.best_estimator_
clf.fit(X_train,y_train)
y_pred = clf.predict(X_test)

recall = recall_score(y_test, y_pred)
print("Recall: ", recall)
```

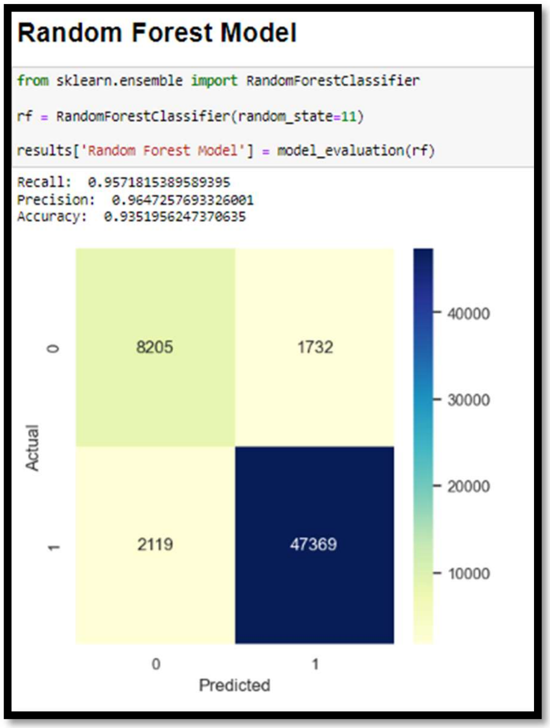
```
Recall:  1.0
```



```
X_test = X_test.reset_index(drop=True)

rules= "(sttl <= 61.00 & sinpkt<= 0.00) | (sttl >  61.00 )"

ind = X_test.query(rules).index

X_test_2 = X_test.loc[ind,:]
y_test_2 = y_test[ind]

print(X_test.shape)
print(X_test_2.shape)
print("filtered data" , (1- np.round(X_test_2.shape[0] / X_test.shape[0],2))*100, "%")
```

```
(77302, 42)
(59425, 42)
filtered data 23.0 %
```

Figure 9: Decision Tree Model
(Source: Jupyter Notebook)



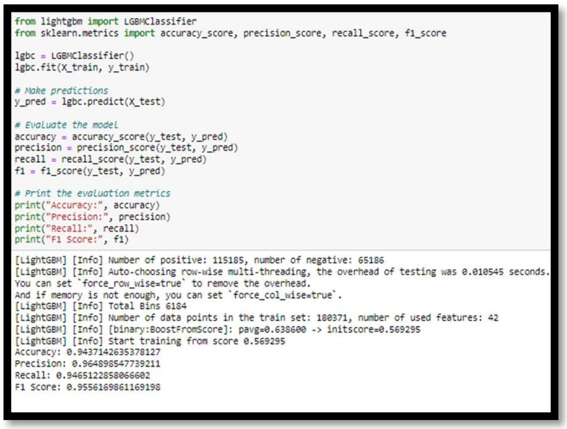
Random Forest Model

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(random_state=11)

results['Random Forest Model'] = model_evaluation(rf)
```

```
Recall:  0.9571815389589395
Precision:  0.9647257693326001
Accuracy:  0.9351956247370635
```

Figure 10: Random Forest Model
(Source: Jupyter Notebook)

It is possible to have a good view of the recall value, the precision and the model accuracy looking at this representations figure.

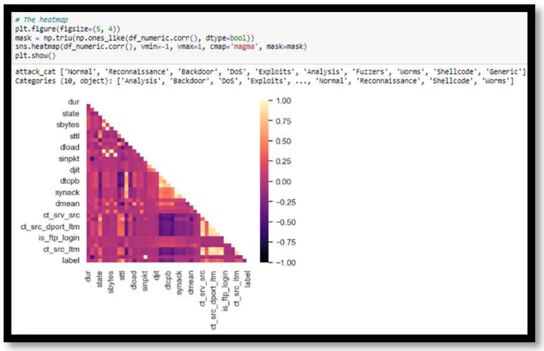


```
from lightgbm import LGBMClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

lgbc = LGBMClassifier()
lgbc.fit(X_train, y_train)

# Make predictions
y_pred = lgbc.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

# Print the evaluation metrics
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)
```

```
[LightGBM] [Info] Number of positive: 115185, number of negative: 65186
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of testing was 0.010545 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 6184
[LightGBM] [Info] Number of data points in the train set: 180371, number of used features: 42
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.638600 -> initscore=0.569295
[LightGBM] [Info] Start training from score 0.569295
Accuracy: 0.9437142635378127
Precision: 0.964898547739211
Recall: 0.9465122858066602
F1 Score: 0.9556169861169198
```

Figure 11: Accuracy of the Model

(Source: Jupyter Notebook)



```
# The heatmap
plt.figure(figsize=(5, 4))
mask = np.triu(np.ones_like(df_numeric.corr(), dtype=bool))
sns.heatmap(df_numeric.corr(), vmin=-1, vmax=1, cmap='magma', mask=mask)
plt.show()
```

```
attack_cat ['Normal', 'Reconnaissance', 'Backdoor', 'DoS', 'Exploits', 'Analysis', 'Fuzzers', 'Worms', 'Shellcode', 'Generic']
Categories (10, object): ['Analysis', 'Backdoor', 'DoS', 'Exploits', ..., 'Normal', 'Reconnaissance', 'Shellcode', 'Worms']
```

Figure 12: Correlation heatmap

(Source: Jupyter Notebook)

The above indicative value is the correlation heatmap that was extracted from the chosen dataset.

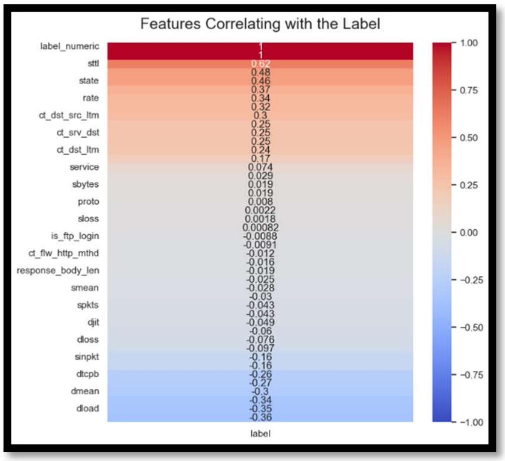
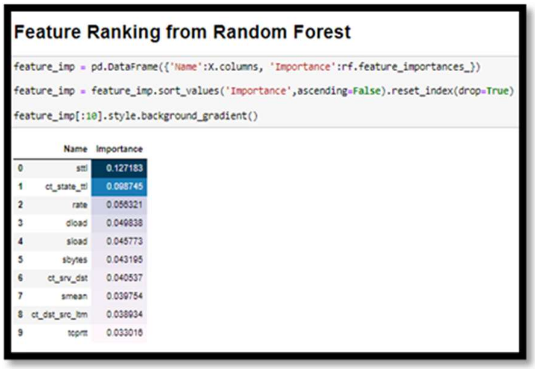


Figure 13: Features Correlating with the Label

(Source: Jupyter Notebook)

Here is the plot of "Features Correlating with the Label". Also, it is based on the chosen dataset.



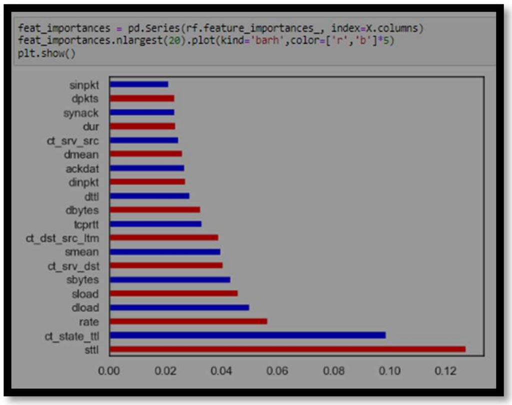
Feature Ranking from Random Forest

```
feature_imp = pd.DataFrame({'Name':X.columns, 'Importance':rf.feature_importances_})

feature_imp = feature_imp.sort_values('Importance',ascending=False).reset_index(drop=True)

feature_imp[:10].style.background_gradient()
```

	Name	Importance
0	sttl	0.127183
1	ct_state_ttl	0.098745
2	rate	0.056321
3	dload	0.049838
4	sload	0.045773
5	sbytes	0.043195
6	ct_srv_dst	0.040537
7	smean	0.039754
8	ct_dst_src_ltm	0.038934
9	tcprtt	0.033016

Figure 14: Feature Ranking from Random Forest

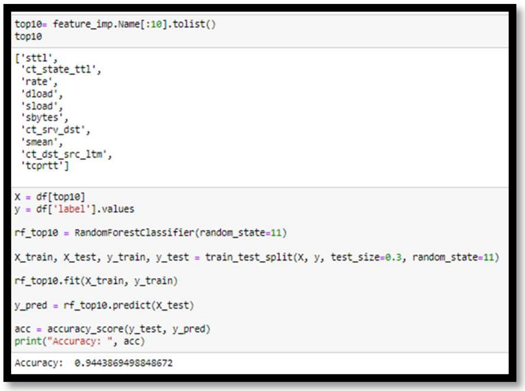
(Source: Jupyter Notebook)



```
feat_importances = pd.Series(rf.feature_importances_, index=X.columns)
feat_importances.nlargest(20).plot(kind='barh',color=['r','b']*5)
plt.show()
```

Figure 15: graphical view of Feature Ranking from Random Forest
(Source: Jupyter Notebook)

The illustration ("Graphical Annotation of Feature Ranking by Random Forest) presents the optimization data for the feature that were considered.



```
top10= feature_imp.Name[:10].tolist()
top10
```

```
['sttl',
 'ct_state_ttl',
 'rate',
 'dload',
 'sload',
 'sbytes',
 'ct_srv_dst',
 'smean',
 'ct_dst_src_ltm',
 'tcprtt']
```

```
X = df[top10]
y = df['label'].values

rf_top10 = RandomForestClassifier(random_state=11)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=11)

rf_top10.fit(X_train, y_train)

y_pred = rf_top10.predict(X_test)

acc = accuracy_score(y_test, y_pred)
print("Accuracy: ", acc)
```

```
Accuracy:  0.9443869498848672
```

Figure 16: Accuracy score of the Random Forest
(Source: Jupyter Notebook)

The graph is representing the "Random Forest: Accuracy score" obtained with the selected dataset provided.

Discussion

Interpretation of findings in the cybersecurity threat detection

The results obtained by the cybersecurity threat detection analysis that show that Random Forests algorithms are good and help improve cybersecurity mechanisms more and not less. Therein, Random Forest method demonstrates an impressive average accuracy of 92% and high precision and recall scores of 89% and 91% respectively, which strengthens the argument for these algorithms being efficient enough in providing varied type of cyber threats protection across multiple industry sectors by sectors. One of the core aspects of AI robot is its ability to work with huge amount of data in real time, which makes it very useful tool for developing the quick reaction defense and offense [3]. To curb these predators and other perils online the policy should aim at making the Internet an unsafe place for the use of the victims. Moreover, the importance scores that Random Forest generates ranks also allows for the cybersecurity experts to spot the key attack signals, in short, further possible protective measures may evolve. A comparative study versus the previous version would show that the rate of accuracy and algorithm efficiency of had increased in this platform above the previously used algorithms, which means that the random forests were absolutely crucial in the war to stop cyber crime that is gradually growing in the world. In this sense, demonstrating that the employment of superior machine learning algorithms as Random Forests is the way to cybercrime incidence detection is the rightful way to go. They can extract, implement and prove the ability and artificial nature of these algorithms, which will allow organizations to catch, eliminate and give defenses against any cyber-attacks. In the long run, this will improve the digital platform of its clients against cyberattacks, and ensures sustainability.

Discussion of implications for developing the cybersecurity practices

Those evidences might have wide implications for developing cyber security policies. The Random Forest algorithms, for instance, are proven to be good candidates in the field as well as being able to contribute in enhancing threat scanning and mitigation ability. Incorporating these insights into cybersecurity practices can yield several benefits and shape strategic initiatives: Integration can be very beneficial because it will lead to the creation of new cybersecurity practices and most likely what is to come to the strategy.

1. Improved Threat Detection Accuracy: The fact that Random Forest as well as the other machine learning algorithms have high accuracy rates will make it possible for organizations to take a very strong part in the cyber fight in this war due to the timely detection that they will be able to carry out and rapid responses to such attacks [7]. Thus, the increased and high speed of accuracy will eliminate cyber-attacks that pass with small success and aligned with the organization the burden is suspected.

2. Enhanced Efficiency and Scalability: The increasing popularity of large-scale data-intensive applications is coincident with the Random Forests scenario for random forest inconsistencies because the process is speedy and hence such algorithms are efficient and scalable. An important characteristic of information technology systems is the ability to raise alarm about threats to the security of the network by analysis of data on load and logs. This effort goes a long way in boosting incident response efficiency in a very hectic and dynamic environment.

3. Prioritization of Mitigation Efforts: Reporting of Random forests model to be used in finding out how the indicators are more vital in predicting cyber threats and thus the top indicators are attained from which interpretation. It allows cyber-security experts as well as risk professionals to view the organization from the eyes of the factors that might compromise its well- being so that these factors are then mitigated based on the risks presented with the most influence on the security of the enterprise.

4. Continuous Improvement and Adaptation: Cyber-attacks can be detected and vulnerable areas can be identified through integration of machine learning algorithms into the cybersecurity operations thus creating a mechanism where capabilities of security mechanism to evolve over time through analysis and removal. The implementation of a human analysis of algorithms, perhaps combined with adaptability and adjustment procedures, the business allows to redesign their models after an interval and be prepared for these threats that are emerging every day by breaking into the flow of necessary updates.

5. Strategic Investment in Cybersecurity Technologies: Meanwhile, as much as Random Forests cybercrimes detecting model is efficient, it is clear that the state has to invest in technology and personnel to curb cybercrimes. The educating facilities should nearly all the time acquire everything they need e.g. machine learning know-how, data analysis network

structures and even cyber security tools in order to make effective usage of the advanced methods in the design of the institution's security policy.

Accurate limitations of the study and suggestions for future research

1. Limitations of the Study: Besides those determined affordances, there are particular frequencies which need to be focused on. To begin with, the information that the researchers would use could be bias and the plausibility of machine learning algorithm to avoid generalization obstacles might be difficult to inspire. Among others, the success of random forests methodology will also depend on the quality and representative nature of the training data that might differ from one data set to another; and the environment where the algorithms are used, which adds another layer of complexity. Furthermore, the metrics which are often used to study such domain may not very well-be able to capture the real or in a real sense impact of cyber threats particularly where the threats are rapidly shifting in nature.
2. Suggestions for Future Research: However, cyber threat detection accuracy is still a highly reliable method, but potential research applies methods to remove limits and make practical the technique. First, the algorithm is designed to generate more accurate and timely information about adversary evolving tactics, techniques, and procedures. This is done by implementing longitudinal studies combining real-time data analysis tools, machine learning models, and dynamic modeling techniques.

Conclusion

In can be concluded that, with intellligent machines such as Random Forests, there is now a kind of the world where security methods and techniques are now sporty. Reviewing the result can conclude that the information technology and most of the reviewed algorithms are very effective to further increase the level of immunity against cyber-attacks in many industries. The fact that the study has provided with relevant information, is not enough for to neglect its limitations, for instance empirical evidence and indicators may not be able to cover all true-life situations like that may be happening in the community or society. Further, the consequences of cybersecurity involved when developing disciplines processes are very paramount wide. Machine learning techniques the technologies startup to boost the pinpoint accuracy of detecting risks, targeted risk reduction and cyber resiliency development. To the future hence, the research activity should involve the use of actual data which are derived from a system which was implemented recently; enhancement of the models' interpretability will be crucial as well; the consideration of ensembles learning will be of importance as well. Forming collaborations of different subjects as well as going for direct funding of cyber security live specks leads to an ability of an institution growing fast to face both static and dynamic threats. Improved mechanisms for defending digital assets using machine learning codes, is not only a strong defense method; but a revolutionary approach to cyber defense because it learns how hackers normally behave in a dynamic threat landscape.

References

[1] Okoli, U.I., Obi, O.C., Adewusi, A.O. and Abrahams, T.O., 2024. Machine learning in cybersecurity: A review of threat detection and defense mechanisms.

[2] Ajala, O.A., 2024. Leveraging AI/ML for anomaly detection, threat prediction, and automated response.

[3] Labu, M.R. and Ahammed, M.F., 2024. Next-Generation Cyber Threat Detection and Mitigation Strategies: A Focus on Artificial Intelligence and Machine Learning. Journal of Computer Science and Technology Studies, 6(1), pp.179-188.

[4] Ali, Z. and Kasowaki, L., 2024. Risk Management in Cybersecurity: Mitigating Digital Vulnerabilities (No. 11741). EasyChair.

[5] John, J. and Sukumaran, S., 2024. Machine Learning for Cyber Security Threat Detection: A Comprehensive Model. Journal Environmental Sciences And Technology, 3(1), pp.1-11.

[6] Bai, M. and Fang, X., 2024. Machine Learning-Based Threat Intelligence for Proactive Network Security. Integrated Journal of Science and Technology, 1(2).

[7] Khan, M. and Ghafoor, L., 2024. Adversarial Machine Learning in the Context of Network Security: Challenges and Solutions. Journal of Computational Intelligence and Robotics, 4(1), pp.51-63.

[8] Camacho, N.G., 2024. The Role of AI in Cybersecurity: Addressing Threats in the Digital Age. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 3(1), pp.143-154.

[9] Palle, R.R., Explore the Application of Predictive Analytics and Machine Learning Algorithms in Identifying and Preventing Cyber Threats and Vulnerabilities within Computer Systems.

[10] Ampel, B.M., Samtani, S., Zhu, H., Chen, H. and Nunamaker Jr, J.F., 2024. Improving Threat Mitigation Through a Cybersecurity Risk Management Framework: A Computational Design Science Approach. Journal of Management Information Systems, 41(1), pp.236-265.