**Automated Security Operations Solutions**

Yuri Mendonca Ribeiro - 2020347

A Report Submitted in Partial Fulfilment

of the requirements for the

Degree of

BSc in Computing in IT (4th year)

Diagram

Description automatically generated with medium confidence

May 2024

Supervisor: Dr. Muhammad Iqbal

The number of words: 3332.00

*In loving memory of my mother, Wyraci Mendonca Ribeiro*

Table of Contents

[Acknowledgements 4](#_Toc166866561)

[Abstract 5](#_Toc166866562)

[Introduction 6](#_Toc166866563)

[Methodology 6](#_Toc166866564)

[Business Understanding 7](#_Toc166866565)

[SOC TOOLS 7](#_Toc166866566)

[Business Analysis – SWOT 8](#_Toc166866567)

[Strengths 8](#_Toc166866568)

[Weaknesses 8](#_Toc166866569)

[Opportunities 8](#_Toc166866570)

[Threats 8](#_Toc166866571)

[Legal/Ethical Considerations 9](#_Toc166866572)

[Machine Learning in Cyber Security 9](#_Toc166866573)

[Data Understanding 10](#_Toc166866574)

[Data Preparation 13](#_Toc166866575)

[Modelling 15](#_Toc166866576)

[Logistic Regression 15](#_Toc166866577)

[Random Forest 15](#_Toc166866578)

[SVM model 16](#_Toc166866579)

[Evaluation 17](#_Toc166866580)

[Deployment 19](#_Toc166866581)

[Conclusion 20](#_Toc166866582)

[References 21](#_Toc166866583)

# Acknowledgements

My Appreciation for the lectures’ support Michael Weiss and Ken Healy. The project supervisor Dr. Muhammad Iqbal, for his invaluable insights and expertise.

I would also like to extend my appreciation to my classmates, Eliable Baliero de Moura, Carolina Franco do Prado, Leisly Pino Duran, Natalia Gomes Fernandes, and Xiaohui Weng, for their constructive feedback and friendship. Their contributions have enriched my learning experience and made this journey all the more memorable.

# Abstract

Developing an automated Security Operation Centre solution (SOC) integrating Wazuh (SIEM & XDR), TheHive, and Shuffle to provide a centralised, cost-effective, and customisable solution for security monitoring and intelligent response.

Using machine learning algorithms and a phishing URL dataset to enhance the intelligent response capabilities within the SOC.

The project will cover open-source tools, business analysis, data analysis, legal ethical considerations about data in cyber security.

# Introduction

This capstone project presents the creation of an open-source, automated SOC solution that integrates Wazuh (SIEM), TheHive (Case Management) and Shuffle (SOAR) based on a cloud environment thus creating a centralised, cost-effective security monitoring and intelligent response.

Furthermore, in order to enhance our security capabilities, this project utilises a phishing URL dataset containing 235,795 instances, along with Python programming and a machine learning (ML) algorithm integrating thus a machine learning model to the SOC.

# Methodology

I will use the Cross Industry Standard Process - CRISP-DM methodology model for this project. Following the CRISP-DM framework we need to break the data mining project into six phases:

* **Business Understanding:** Determining business objectives; Identifying the threats that the SOC aims to address; Understanding the tools, and technologies used in the SOC
* **Data Understanding:** Identifying the relevant data sources for the project and explore and analyse the data to understand its structure, quality and potential limitations.
* **Data Preparation**: Performing data cleansing and normalisation. Selection and extraction of relevant features from the data set for ML modelling; Handling missing values, and imbalanced data, if necessary; Splitting the data into training and testing sets.
* **Modelling:** Choosing the machine learning algorithm for phishing URL detection.
* **Evaluation:** Evaluation of the performance of the models; Interpretation of the model’s result.
* **Deployment:** Creation of a plan for integrating the machine learning model into the SOC’s existing workflows.

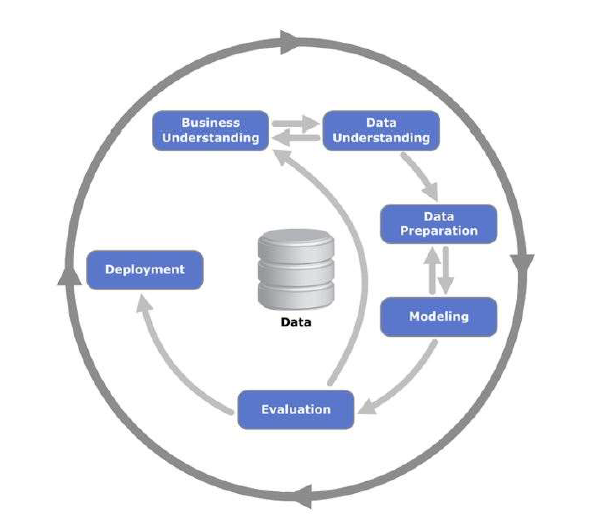


Figure 1- CRISP-DM process diagram

## Business Understanding

In this first phase of the CRISP-DM process, I would like to cover the understanding of the tools and technologies used in the SOC, identify the threats that the SOC aims to address, determine the business objective and explore how machine learning (ML) can be used.

### SOC TOOLS

The Automated Security Operations Solutions project aims to build an open-source security operation centre platform by integrating:

• Wazuh (SIEM – Security Information and Event Management & XDR – Extended Detection and Response).

• TheHive (Case Management) and,

• Shuffle (SOAR – Security Orchestration, Automation and Response).

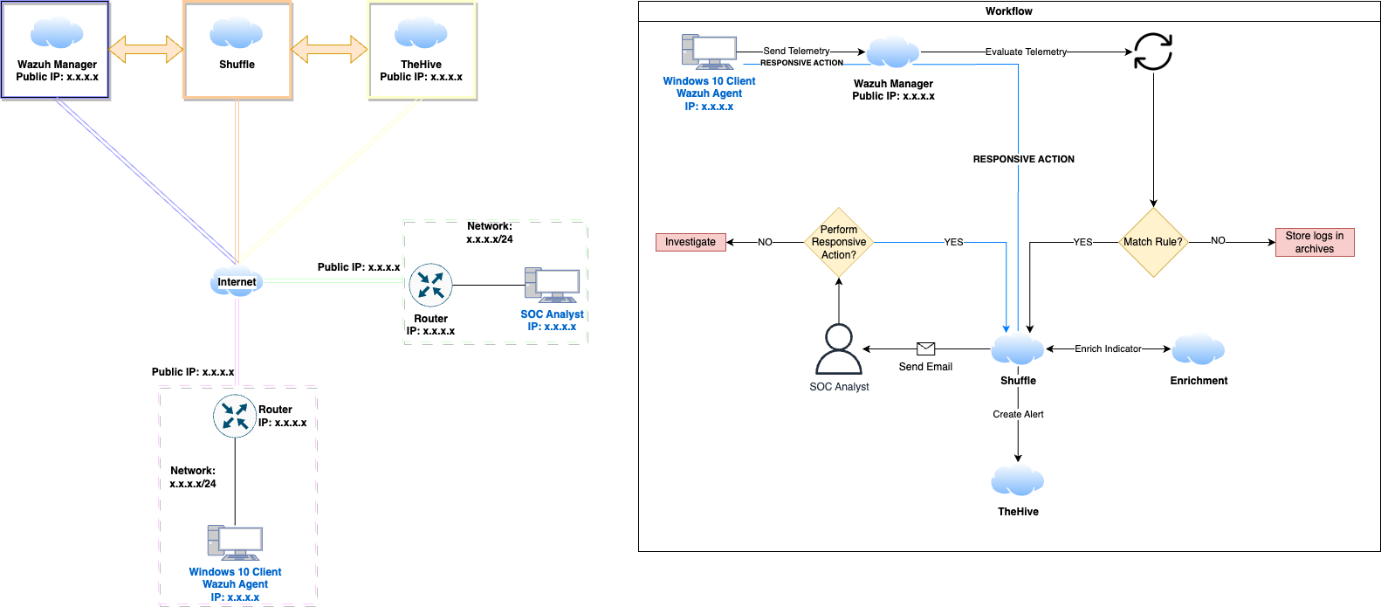


Figure 2- SOC Workflow Diagram

Wazuh is responsible for collecting and analysing security data from endpoints, network devices and cloud workloads (Wazuh, n.d.) while TheHive facilitates threat analysis like malware analysis and incident response workloads (TheHive, n.d.) and Shuffle enables automated security execution and response actions like integrating Wazuh alerts to TheHive and sending emails to the SOC analyst of every security incident event (Shuffler, n.d.).

While Wazuh provides security monitoring capabilities, TheHive can be used to focus on analysing cyber threats and managing security events and Shuffle enables us to build automated playbook workflows for incident response.

One of the advantages of adopting open-source tools is that they avoid vendor lock-in risks as well as provide more flexibility to customise and extend solutions when needed. Also, the lack of licensing fees makes this project cost-effective if we compare it to proprietary SIEM/SOAR stacks.

The choice of hosting the integrated open-source stack on a cloud platform is to take advantage of its predictable billing model, high availability and scalability. Also, locating security data in different available zones worldwide provides a new layer of security.

### Business Analysis – SWOT

Analysing the business potential for an automated security operation solution the S.W.O.T (Strengths, Weaknesses, Opportunities, Threats) framework can be applied as:

### Strengths

* **Cost-benefit**: Using free open-source tools like Wazuh, TheHive and Shuffle we wouldn’t have any cost at the first moment whether we compare to proprietary paid SIEM/SOAR solutions such as Splunk, or Graylog (Graylog, n.d.)
* **Cloud Flexibility**: Hosting our automated SOC solution in a cloud provider gives us the scalability and elasticity that we need to match dynamic security operation needs. (What is cloud scalability?, n.d.)
* **Automation**: Integrating a Security Orchestration, Automation and Response – SOAR tools like Shuffle automation of repetitive security tasks and response actions give us an advantage by replacing manual processes. (Shea, 2024)
* **Customization**: Open-source solutions can be customized to meet further organizational security requirements if needed.

### Weaknesses

* **Implementation**: Integrating multiple open-source tools into a centralized solution can be quite hard and requires high-level technical expertise and skills.
* **Support Problems**: Relying on open-source community support for eventual issues can be a weakness in this business.
* **Skill Requirements**: Need for in-house development skills for deploying, managing and maintaining the open-source stack in order to cut costs.

### Opportunities

* **Cybersecurity Market**: The global cybersecurity market is growing fast every year. According to Fortune Business Insights, the cybersecurity market is projected to grow to $424.97 billion by 2030. (Cyber Security Market, 2023)
* **Security Automation Market**: The global security automation market size was valued at $8.9 billion in 2023. It is expected to reach the mark of $16.7 billion by 2028. One of the reasons is the significant growth of security incidents of phishing emails and ransomware leading companies to automate their process on cybersecurity. (Markets And Markets, 2023)
* **Open-Source Adoption**: Recently the 2024 State of Open-Source Report about the use of open-source software over the year 2023. 33.50% answered – Yes, significantly and 34.07% of the organization's participants answered – Yes. That gives us a quite good opportunity to grow in the sector. (OpenLogic, 2024)

### Threats

* **Established Vendor Competitors**: Large and established cybersecurity vendors such as Splunk, IBM, QRadar, and Graylog dominate the market of SIEM/SOAR.
* **Fast Technology Evolution**: Daily releases and updates along with the fast development of new threats require continuous integration efforts.
* **Talent Shortage**: The cybersecurity workforce is growing; however, the skills shortage remains. That is what is learnt in the ISC2 Cybersecurity Workforce Study by ISC2. Even with the workforce reaching 5.5 million an 8.7 percent increase from 2022, representing 440,000 new jobs still has a huge gap of 4 million qualified professionals. (ISC2, 2022)

The proposition is to provide an integrated, open-source, cloud-hosted security solution platform at a 1/3 of commercial proprietary vendor tools cost as enabling automation, customisation and machine learning features for detection of phishing URLs.

The target customer segments are small and mid-market businesses and education sector.

### Legal/Ethical Considerations

Collecting and processing user data is the most security monitoring activity. It has to adhere to relevant data protection regulations like the General Data Protection Regulation – GDPR in Europe. All user data needs to be addressed through mechanisms like data anonymisation following the guidance of the Data Protection Commission. (Anonymisation and pseudonymisation, n.d.)

Furthermore, there are ethical considerations around monitoring user activities and potential overstep of surveillance. For this particular issue, robust access control and oversight processes have to be implemented.

Ultimately, as a security operation platform, ethical principles like protecting user data from misuse, and disclosure of sensitive information must be enforced.

### Machine Learning in Cyber Security

Machine learning can be used in cyber security for different purposes such as: (Short, 2023)

* Threat Intelligence;
* Anomaly Detection;
* Patch Management;
* Fraud Detection; and
* Phishing Detection.

Phishing is a type of social engineering attack in which the attacker attempts to trick users to steal sensitive information. The common goals of phishing attacks are stealing login credentials, financial information or even installing malware like ransomware. (What is Phishing?, 2023)

For this project, I chose to develop a phishing detection machine learning model to integrate it to the automated security operation centre (SOC).

## Data Understanding

In this second phase of the CRISP-DM process, we must identify the relevant data sources for this project, such as security logs, and network traffic data. Also, I will explore and analyse the data to understand its structure, quality and potential limitations.

After going through numerous datasets of network traffic and security logs I opted to work on an updated phishing URL dataset to demonstrate the capacity of machine learning and detect whether a URL is a phishing website or a legitimate one.

The dataset chosen is the PhiUSIIL Phishing URL Dataset. It is a substantial dataset comprising 134,850 legitimate and 100.945 phishing URLs. (PhiUSIIL Phishing URL (Website), 2024).

The libraries used to load, manage and plot visualisations are the following:

# Libraries to load, manage and plot graphics

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import plotly.express as px

As the dataset is in a CSV format, a simple type of raw that uses commas to separate values I can use the method .read\_csv() for reading and using it into the DataFrame called: phishing\_df:



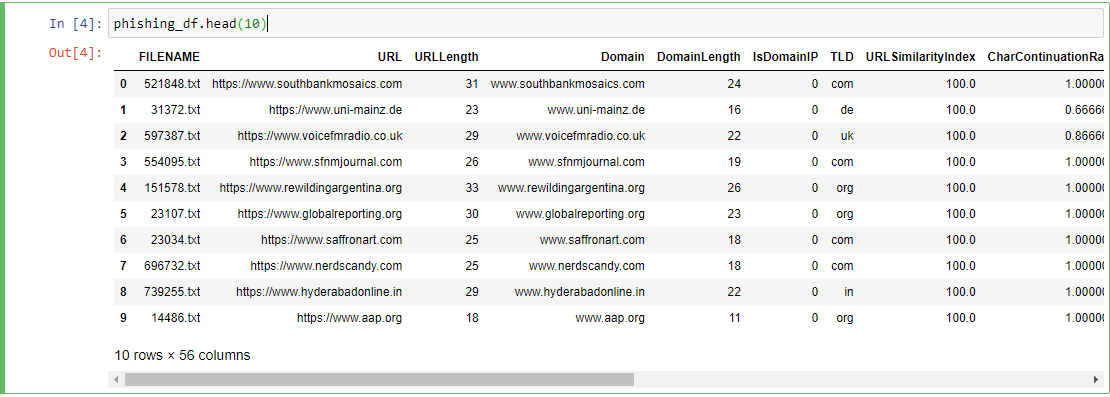
Figure 3- Loading dataset

Using .shape to check the number of rows and columns (235,795 rows and 56 columns):



Figure 4 - Using .Shape

Now I can visualise the first 10 and last 10 records of the dataset using .head(10) and .tail(10):



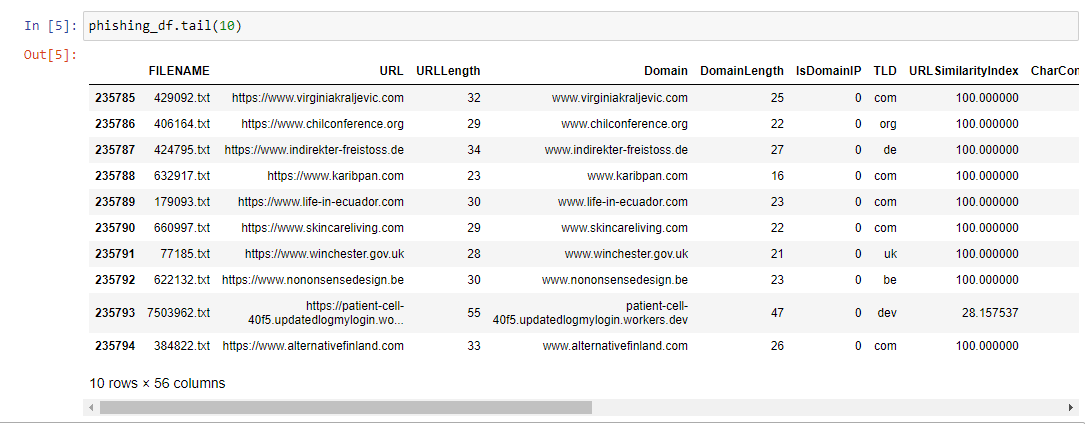


Figure 5 - .tail(10)

Using the method .info() I can check the dataset data types and whether the variables within are quantitative or qualitative:

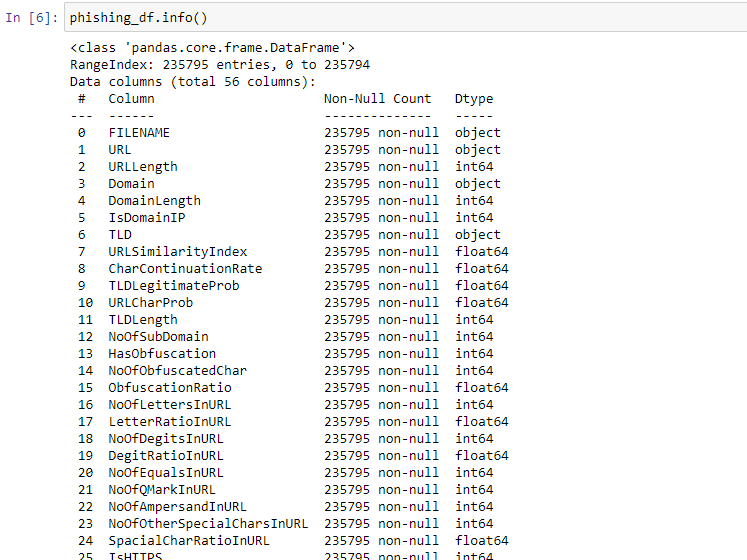


Figure 6- .info() method

With the method .describe() I can extract some information about the quantitative variables ( minimum, maximum, mean and standard deviation):

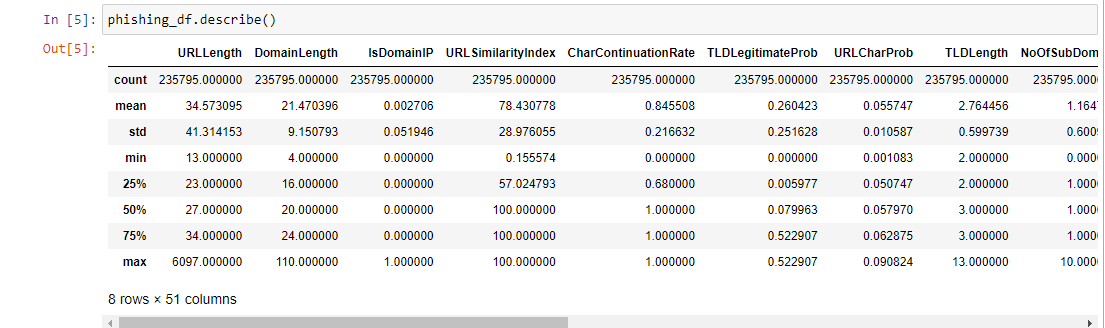


Figure 7- .describe() method

After getting the information about the quantitative variables now I will check the unique values from my label where ‘0’ means legitimate URL and ‘1’ phishing URL:

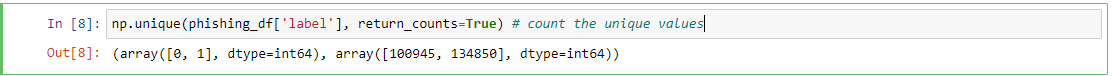


Figure 8- np.unique 'label'

Given the result, there are only two values: 0 and 1 – data type Int64. Where legitimate URLs correspond to 100,945 of the entries while phishing URLs correspond to 134,850 entries in a total of 235.795 registers as observed before using the .shape() method.

Using the countplot() method I can plot a graphic to better visualise the difference between these two values:

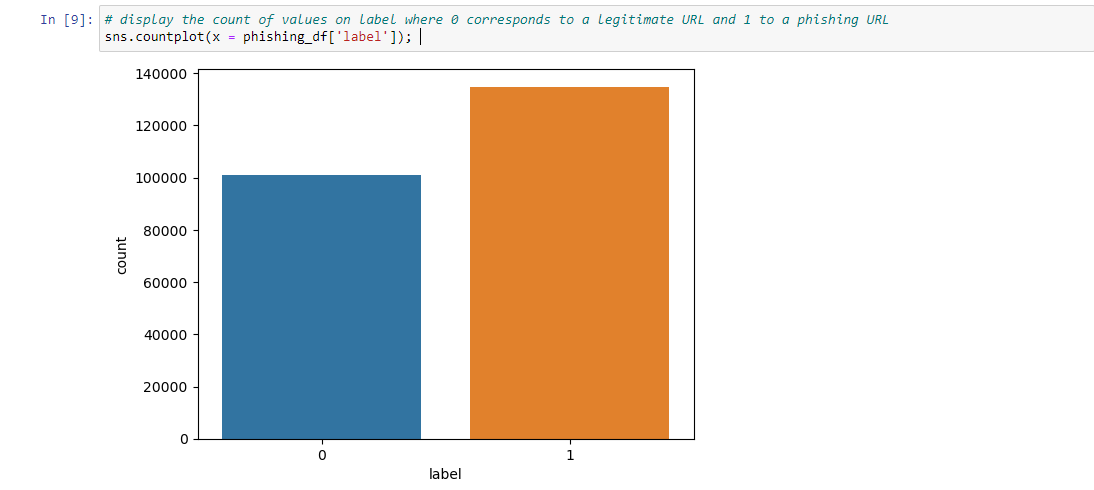


Figure 9- .countplot method

After analysing the columns using the method .info() I have decided not to include the qualitative columns: Filename, URL, Title and Domain leaving only the “TLD” column which will need to go through the normalisation process in the next step of Data Preparation.

## Data Preparation

In this third phase of the CRISP-DM process, the objective is to perform a data-cleansing, handle missing values if there are any and also select and extract the relevant features from the data set to finally split the data for machine learning modelling.

The first step is to check if there are any missing values on the data set using the method .isnull().count:

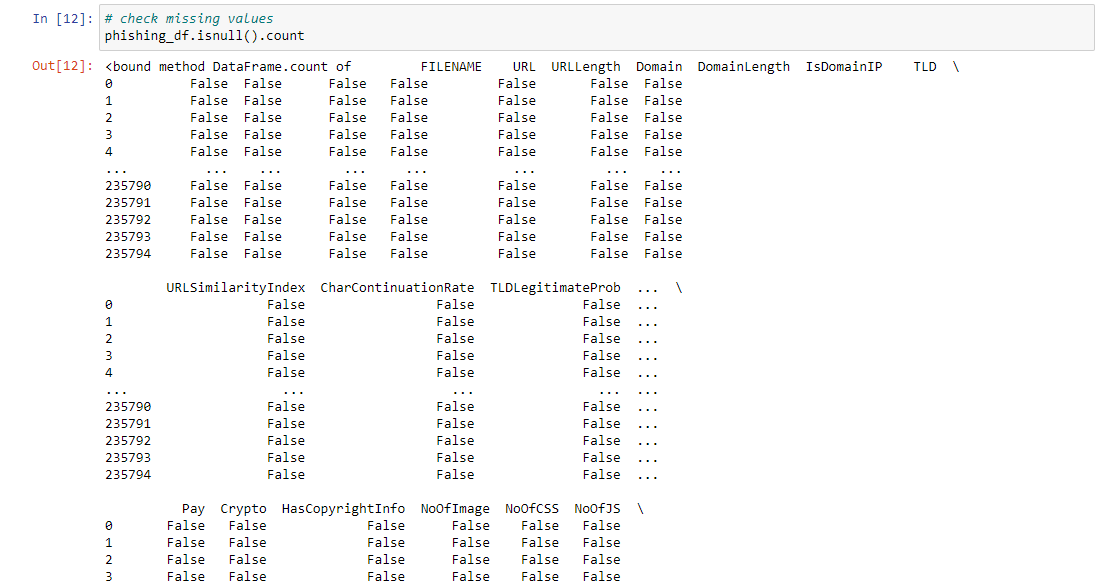
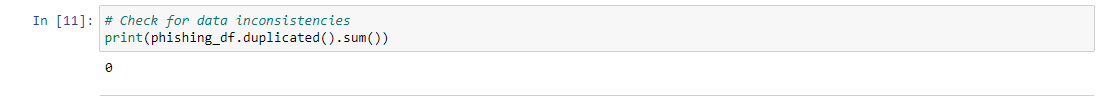


Figure 10 - .isnull().count method

There are no missing values so I can proceed to the next step: Check data inconsistencies using the method .duplicated().sum():



As the dataset does not have any duplicated data, I can then drop the qualitative (categorical) columns: FILENAME, URL, DOMAIN and TITLE:

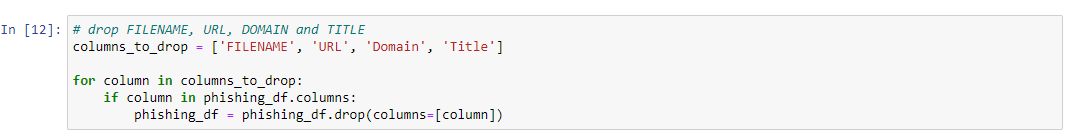


Figure 11- Drop qualitative columns

After dropping the categorical columns that I will not use to train my machine learning model, I can separate features and target value (label):



Figure 12- Separate target variable – label

Now I can use the one-hot-encoding technique on the “TLD” to convert the categorical variable into a format that can be used in machine learning that requires numerical inputs. Using the OneHotEncoder class from sklearn where:

1. Categorical\_col: selects the categorical column, in this case ‘TLD’;
2. Ohe: initiates an instance of the OneHotEncoder class;
3. Temp\_df: creates a temporary DataFrame containing only the ‘TLD’ columns;
4. Encoded\_temp\_df: the fit\_transform method transforms the data into the encoded format as the encoded data is converted to a numpy array using .toarray().
5. Encoded\_temp\_df.columns: Assigns column names to the encoded columns based on the original name (‘TLD’) while the get\_feature\_names\_out method retrives the new column names;
6. Finally, the original ‘TLD’ column is dropped from “X\_df”. Then the encoded columns in “encoded\_temp\_df” are concatenated with the modified “X\_df” resulting a new dataframe contains the encoded “TLD” column.

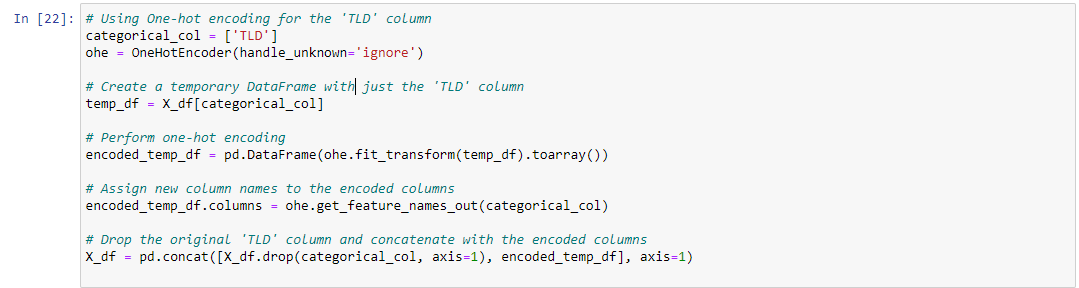


Figure 13 - OneHotEncoder process

After the OneHotEncoder, I can now apply the Standardisation method to my dataframe:

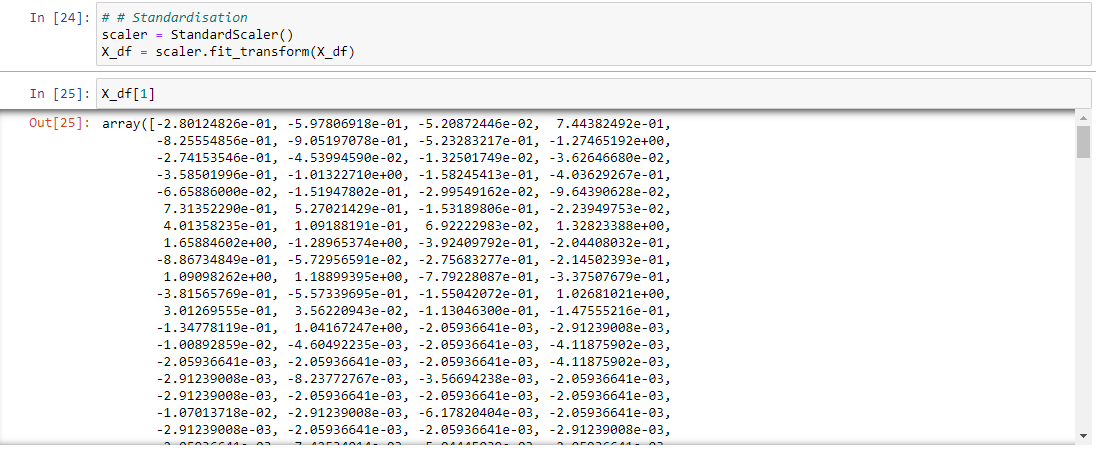


Figure 14 – Standardisation

The last step is to split the data for machine learning modelling. In this case, I have opted to set 15% of my test data which gave me better results avoiding overfitting my training model.

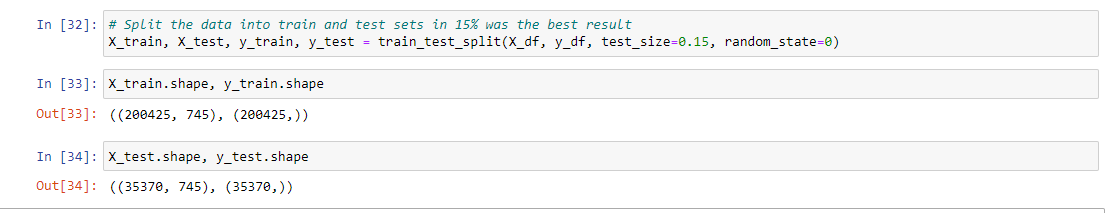


Figure 15- Splitting data

Note that now the dataframe contains 745 columns after the OneHotEncoder.

## Modelling

In the fourth phase of the CRISP-DM process, I have chosen to train three different machine-learning models:

* Logistic Regression
* Random Forest, and
* SVM model

The libraries to use for machine learning:

# Machine learning libraries

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.metrics import confusion\_matrix, ConfusionMatrixDisplay

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.preprocessing import OneHotEncoder

from sklearn.metrics import accuracy\_score, classification\_report

from sklearn.ensemble import RandomForestClassifier

from sklearn.svm import SVC

### Logistic Regression

Logistic regression is a supervised machine learning algorithm used for binary classification tasks. Its goal is to predict the probability that an instance belongs to one of two classes. (Brownlee, Logistic Regression for Machine Learning, 2023)

Some of the key aspects of logistic regression are:

1. It uses the logistic function to transform a linear combination of input features into a probability value between 0 and 1.
2. The logistic function maps any real-valued number to a value within 0 to 1, making it suitable for binary classification tasks.
3. The weights of the logistic regression model are estimated using maximum likelihood

### Random Forest

The Random Forest algorithm is a supervised machine learning algorithm used for both classification and regression tasks. It is an ensemble learning method that builds a multitude of decision trees during training where each tree is built using a random subset of the dataset and a random subset of features. (Donges, n.d.)

The criterion parameter in the Random Forest is used to measure the quality of a split in the decision trees that make up the forest. Basically, the criterion determines how the algorithm will select the best feature to split a node during the tree-building process. (Donges, n.d.)

For this project, I opted to use the “entropy” criterion due to its ability to handle complex relationships and detect new patterns if we compare it to Gini impurity parameter. (Aznar, 2020)

### SVM model

The Support Vector Machine (SVM) is also a supervised machine learning technique used for both classification and regression tasks. The main objective of SVM is to find the optimal hyperplane in an N-dimensional space that can separate the data points into different classes. (Brownlee, Support Vector Machines for Machine Learning, 2020)

SVM has a kernel function that takes low-dimensional input space and transforms it into a higher-dimensional space. The popular kernel functions are Linear, Polynomial, and Radial Function (RBF) kernels. Unfortunately, during the tests, the only kernel able to run it was the Linear kernel.

Below we can see the score results of the models:

Training a Logistic Regression model – Score 99.98%:



Figure 16- Logistic Regression model

Training a Random Forest model – Score 99.99%

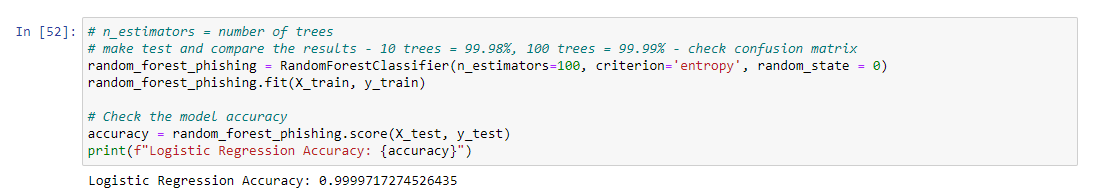


Figure 17- Random Forest model

Training an SVM model – Score 99.98%

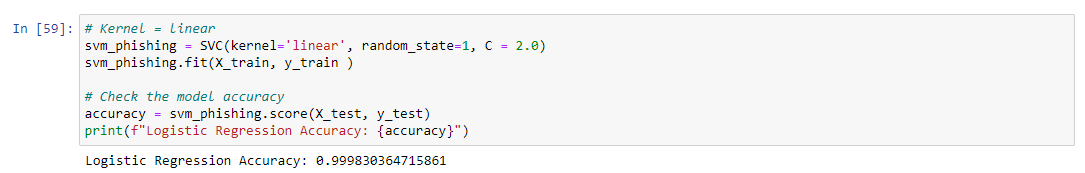


Figure 18 - SVM model

## Evaluation

Evaluation is the fifth phase of CRISP-DM. In this phase, we will interpret the models’ results and evaluate their effectiveness in phishing URL detection.

Even though all models tested had high scores, we can analyse in detail each one accuracy and confusion matrix:

Using the classification\_report function from the sklearn we can generate a detailed report on the performance of a classification model:

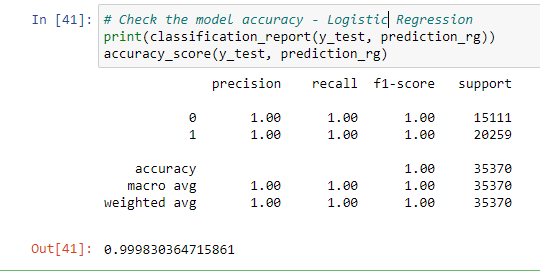


Figure 19 - Logistic Regression Accuracy Report

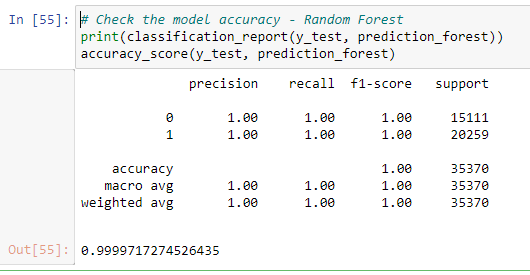


Figure 20- Random Forest Accuracy Report

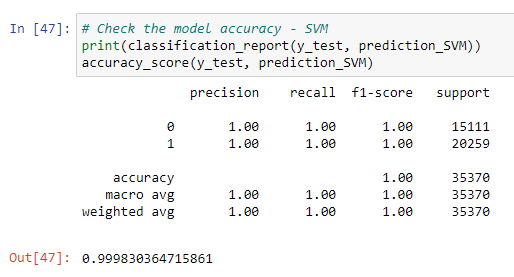
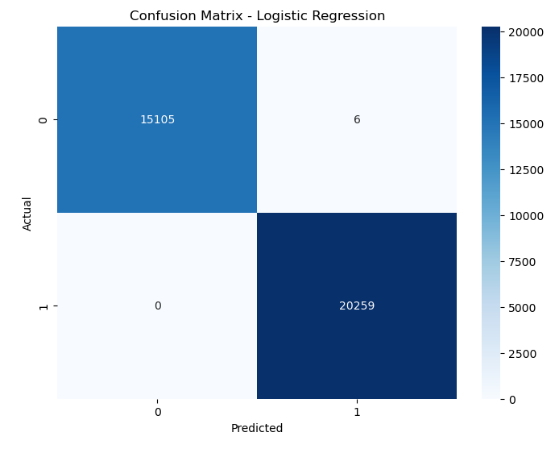
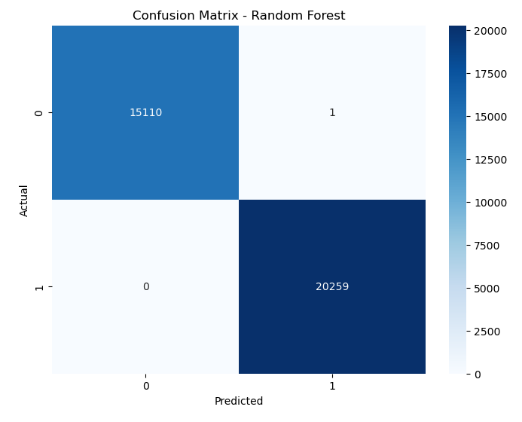


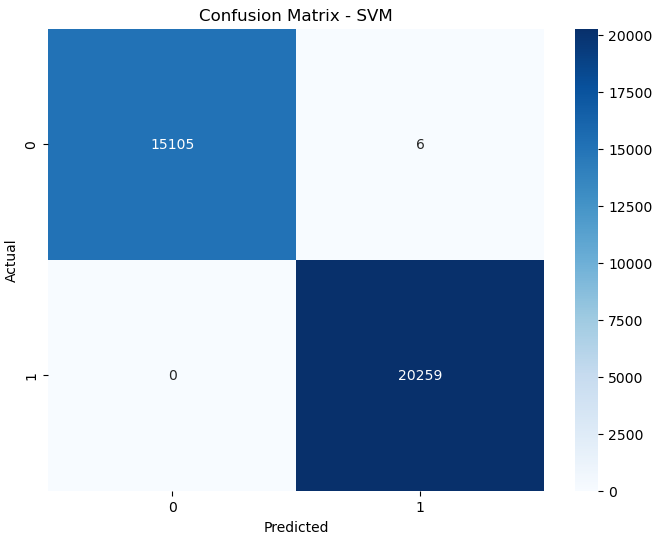
Figure 21- SVM Accuracy Report

Based on the Accuracy report we can notice that the Logistic Regression and the SVM model had the same results. However, Logistic Regression could perform faster than SVM. This happens due to SVM is based on the geometrical properties of the data while Logistic Regression is rooted in a statistical approach. (Geeks for Geeks, 2023)

The Random Forest was the model which better scored, reaching 99.99% accuracy.

Now let’s analyse the results of the Confusion Matrix of each model trained:



The Logistic Regression and SVM had the same results. They wrongly predicted 6 URLs on label 0 (legitimate URL) while they right-predicted all the URLs on label 1 (phishing URL). On the other hand, the Random Forest was able to right-predicted all phishing URLs and wrong only in one on legitimate URLs.

The result obtained with Random Forest is satisfactory to be used on our Automated SOC.

## Deployment

In the final phase of the CRISP-DM process, the goal is to create a plan for integrating the machine learning model for phishing URL detection into the workflow of the Automated Security Operations Centre (SOC) solution.

Given the Random Forest model has achieved an accuracy of 99.99%, will be deployed for real-time detection of phishing URLs. The model can be integrated with the Wazuh SIEM to analyse network traffic.

Whenever a new URL request is made from any monitored device, the URL will be passed through the trained Random Forest model, the model will then classify the URL as either legitimate or a phishing threat. If classified as a phishing threat, an alert can be automatically generated within Wazuh and then passed to TheHive system for further analysis by a security analyst.

For continuous operation, the model needs to be periodically re-trained and updated with new phishing techniques and URLs.

## Conclusion

When I started to develop the SOC project in the summer of 2023, I was not expecting to integrate machine learning into it. The project was developed to increase my skills in building robust cloud systems using open-source tools for cyber security at the same time making a cheap security option for small businesses.

The integration of machine learning for the detection of Phishing URLs showcases the ability to greatly increase the intelligent response capabilities within the SOC. As demonstrated, the Random Forest model reached an impressive 99.99% accuracy making this new feature a key component in this project.

While the proposed solution presents numerous advantages, we cannot forget that the challenges in cyber security are daily and improvements and updates must be carried on regularly. However, the solution offers a promising foundation for further developments whether in phishing detection or malware detection, abnormally network behaviour and so on.

# References

*Anonymisation and pseudonymisation*. (n.d.). Retrieved from Data Protection Comission: https://www.dataprotection.ie/en/dpc-guidance/anonymisation-pseudonymisation

Aznar, P. (2020, December 02). *Decision Trees: Gini vs Entropy*. Retrieved from QuantDare: https://quantdare.com/decision-trees-gini-vs-entropy/

Brownlee, J. (2020, August 15). *Support Vector Machines for Machine Learning*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/support-vector-machines-for-machine-learning/

Brownlee, J. (2023, December 06). *Logistic Regression for Machine Learning*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/logistic-regression-for-machine-learning/

*Cyber Security Market*. (2023, 04 17). Retrieved from Fortune Business Insights: https://www.fortunebusinessinsights.com/press-release/cyber-security-market-9280

Donges, N. (n.d.). *Random Forest: A Complete Guide for Machine Learning*. Retrieved from BuiltIn: https://builtin.com/data-science/random-forest-algorithm

*Geeks for Geeks*. (2023, May 07). Retrieved from Differentiate between Support Vector Machine and Logistic Regression: https://www.geeksforgeeks.org/differentiate-between-support-vector-machine-and-logistic-regression/

*Graylog*. (n.d.). Retrieved from Graylog Pricing: https://graylog.org/pricing/

*ISC2*. (2022). Retrieved from Cybersecurity Workforce Study: https://media.isc2.org/-/media/Project/ISC2/Main/Media/documents/research/ISC2-Cybersecurity-Workforce-Study-2022.pdf?rev=1bb9812a77c74e7c9042c3939678c196

*Markets And Markets*. (2023, 08). Retrieved from Security Automation Market : https://www.marketsandmarkets.com/Market-Reports/security-automation-market-266165.html

*OpenLogic*. (2024). Retrieved from 2024 State of Open Source Report: https://www.openlogic.com/resources/state-of-open-source-report?utm\_source=landingpage&utm\_medium=social&utm\_campaign=OPL-GLB-2024Q1-CON-2024StateofOpenSourceReport&utm\_content=press-release

*PhiUSIIL Phishing URL (Website)*. (2024, March 03). Retrieved from UCI Machine Learning Repository: https://archive.ics.uci.edu/dataset/967/phiusiil+phishing+url+dataset

Shea, S. (2024, February). *SOAR (security orchestration, automation and response)*. Retrieved from TechTarget: https://www.techtarget.com/searchsecurity/definition/SOAR

Short, R. (2023, March 20). *10 Applications of Machine Learning in CyberSecurity*. Retrieved from IT SUPPORT 4U: https://itsupport4u.ie/blog/applications-of-machine-learning-in-cyber-security

*What is cloud scalability?* (n.d.). Retrieved from DigitalOcean: https://www.digitalocean.com/resources/article/cloud-scalability

*What is Phishing?* (2023). Retrieved from Proofpoint: https://www.proofpoint.com/us/threat-reference/phishing

[Figure 1- CRISP-DM process diagram 6](#_Toc166870471)

[Figure 2- SOC Workflow Diagram 7](#_Toc166870472)

[Figure 3- Loading dataset 10](#_Toc166870473)

[Figure 4 - Using .Shape 10](#_Toc166870474)

[Figure 5 - .tail(10) 11](#_Toc166870475)

[Figure 6- .info() method 11](#_Toc166870476)

[Figure 7- .describe() method 12](#_Toc166870477)

[Figure 8- np.unique 'label' 12](#_Toc166870478)

[Figure 9- .countplot method 12](#_Toc166870479)

[Figure 10 - .isnull().count method 13](#_Toc166870480)

[Figure 11- Drop qualitative columns 13](#_Toc166870481)

[Figure 12- Separate target variable – label 14](#_Toc166870482)

[Figure 13 - OneHotEncoder process 14](#_Toc166870483)

[Figure 14 – Standardisation 14](#_Toc166870484)

[Figure 15- Splitting data 15](#_Toc166870485)

[Figure 16- Logistic Regression model 16](#_Toc166870486)

[Figure 17- Random Forest model 16](#_Toc166870487)

[Figure 18 - SVM model 16](#_Toc166870488)

[Figure 19 - Logistic Regression Accuracy Report 17](#_Toc166870489)

[Figure 20- Random Forest Accuracy Report 17](#_Toc166870490)

[Figure 21- SVM Accuracy Report 18](#_Toc166870491)