

**School of Science and Engineering**

**Title**

**Combined Capstone Internships Report**

**March 2024**

**Mouradi Asmae**

**Supervised by:**

**Xxxxx**

**and**

**Dr. Lamiae Bouanane, Al Akhawayn University**

**Student conduct and copyright**

The student agree that:

1. Permission has been obtained for any third party content (eg Data, illustrations, photographs, charts or maps).
2. The results described in this report have not previously been published

**Student’s name and signature: Asmae Mouradi**

Title

Internship Report

**Student Statement:**

As the author of this combined capstone plan, I, Asmae Mouradi, assert that the framework outlined herein is authentically derived from my professional judgment. I affirm that the research, coding, data collection, and interpretation of results have been conducted with utmost professionalism and ethical consideration. I have properly acknowledged and cited the contributions of others in line with established academic standards.

Asmae Mouradi

Student’s name

Approved by the Supervisor

Dr. Lamiae Bouanane

# ACKNOWLEDGMENT

Before everything, I would like to thank God for all the opportunities he has offered me, and for the strength he provides, enabling me to move forward and become the person I am today despite my weaknesses.

My journey at UM6P has been a truly enriching opportunity. I extend my heartfelt gratitude to the entire team for their efforts in creating a welcoming atmosphere and making each day of my professional experience a joy.

I want to thank my supervisor, Dr. Lamiae Bouanane, for her guidance and support throughout this capstone project.

To my siblings, I take this opportunity to thank you for always supporting me and pushing me forward by believing in me. The generosity you each show has had a great impact on my achievements.

Last but not least, I am grateful for the two people who illuminate my path day after day with their continuous guidance, love, and care. My dearest parents. Thank you for everything you have done to me from my childhood until now, and for what the future holds.

**CONTENT**

[ACKNOWLEDGMENT 2](#_Toc178542357)

[LIST OF FIGURES 5](#_Toc178542358)

[LIST OF TABLES 6](#_Toc178542359)

[ABSTRACT (ENGLISH) 7](#_Toc178542360)

[RESUME (FRENCH) 7](#_Toc178542361)

[1. INTRODUCTION 8](#_Toc178542362)

[2. INTERNSHIP BACKGROUND 8](#_Toc178542363)

[3. PROBLEM STATEMENT 9](#_Toc178542364)

[3.1 Requirement Engineering 10](#_Toc178542365)

[a. Functional Requirements 10](#_Toc178542366)

[b. Non-Functional Requirements 11](#_Toc178542367)

[4. PROJECT SPECIFICATIONS. 11](#_Toc178542368)

[5. STEEPLE ANALYSIS 11](#_Toc178542369)

[6. INTERNSHIP PLAN 12](#_Toc178542370)

[7. LITERATURE REVIEW 13](#_Toc178542371)

[8. ENGINERRING STANDARD 13](#_Toc178542372)

[9. METHODOLOGY & CAPSTONE DESIGN 13](#_Toc178542373)

[9.1 Technology Enablers 13](#_Toc178542374)

[9.2 System Architecture 13](#_Toc178542375)

[9.3 Models and approaches 14](#_Toc178542376)

[9.3.1. Random Forest Classifier: 14](#_Toc178542377)

[9.3.2. XGB: 14](#_Toc178542378)

[9.3.3. Decision Tree: 14](#_Toc178542379)

[9.3.3. Logistic Regression: 14](#_Toc178542380)

[9.3.4. Naïve Bayes 14](#_Toc178542381)

[9.3.5. SVM 14](#_Toc178542382)

[9.4. Implementation 15](#_Toc178542383)

[8.4.1. Data Collection 15](#_Toc178542384)

[9.3.2. Data preparation 16](#_Toc178542385)

[9.4.3. Experiment tracking and versioning 16](#_Toc178542386)

[10. DATA PRESENTATION 16](#_Toc178542387)

[11.SIMULATIONS, RESULTS AND INTERPRETATION 16](#_Toc178542388)

[12.LEARNING STRATEGIES 16](#_Toc178542389)

[13. CONCLUSION AND FUTURE WORK 16](#_Toc178542390)

[REFERENCES 16](#_Toc178542391)

[APPENDIX A: CODE 16](#_Toc178542392)

[APPENDIX B: 16](#_Toc178542393)

# 

# LIST OF FIGURES

[Figure 1 : xxxxxxxx x](#_Toc133255142)

# LIST OF TABLES

[Table 1 : xxxxxxxx x](#_Toc133255199)

**ABSTRACT (ENGLISH)**

During my internship at Cyber Lab, I develop a machine learning model that can detect network attacks using SIEM (Security Information and Event Management) platform, which refers to creating a predictive model using machine learning techniques that can identify malicious activities or security threats within a network. This solution avoids incorrect alerts such as the false positive (FP), which can potentially lead to unneeded investigation that will consume time of the SOC team away from malicious threats. This approach of including artificial intelligent and machine learning is different from the traditional way the company was using, in which they were detecting the attacks through static rules sets and threat intel artifacts. Furthermore, I was able to deploy a middleware that will integrate with the lab architecture to access live network telemetry and report model predictions to the lab SIEM. This project brought about considerable time and cost efficiencies, while also improving accuracy, optimizing operations, and supporting better decision-making within the organization.

**Keywords**: Machine learning model, SIEM (Security Information and Event Management (, False Positive, True Negative

**RESUME (FRENCH)**

Lors de mon stage au Cyber Lab, j'ai développé un modèle d'apprentissage automatique capable de détecter les attaques réseau à l'aide de la plateforme SIEM (Security Information and Event Management), qui consiste à créer un modèle prédictif utilisant des techniques d'apprentissage automatique pour identifier les activités malveillantes ou les menaces de sécurité au sein d'un réseau. Cette solution évite les alertes incorrectes telles que les faux positifs (FP), qui peuvent potentiellement conduire à des investigations inutiles consommant le temps de l'équipe SOC au détriment des menaces malveillantes. Cette approche incluant l'intelligence artificielle et l'apprentissage automatique diffère de la méthode traditionnelle utilisée par l'entreprise, où les attaques étaient détectées à travers des ensembles de règles statiques et des artefacts d'intelligence de menaces. De plus, j'ai pu déployer un intergiciel qui s'intégrera à l'architecture du laboratoire pour accéder à la télémétrie réseau en direct et rapporter les prédictions du modèle au SIEM du laboratoire. Ce projet a permis de réaliser des économies de temps et de coûts considérables, tout en améliorant la précision, en optimisant les opérations et en soutenant une meilleure prise de décision au sein de l'organisation.

**Mots clés**: Modèle d'Apprentissage Automatique, SIEM (Gestion des informations et des événements de sécurité), Faux Positif, Vrai Négatif

# INTRODUCTION

As cyber threats continue to move forward and become more sophisticated, the traditional security measures are no longer sufficient to protect networks and sensitive data. In today’s landscape, network is becoming very sensitive many attackers are trying to steel data through different types of attacks. Due to the complexity of cyberattacks, traditional detection systems like Security Information and Event Management (SIEM) platforms, suffer to keep high accuracy. SIEM systems collect, aggregate, and analyze log data from various sources within an organization's IT infrastructure to detect suspicious activities and potential security threats. They frequently result in False Positive (FP) and True Negative (TN), which can cause to unseen threats or unneeded alerts.

Machine learning and artificial intelligence present powerful tools in order to enhance the detection and response to attacks. SIEM platforms rely on a combination of static and dynamic rule sets to monitor network activity and detect malicious behaviors. Static rules trigger alerts when predefined patterns, such as a single IP address attempting to reach multiple addresses, are detected, while dynamic rules use visual representations, such as graphs, to identify unusual network spikes or anomalies. However, these systems often fall short when faced with sophisticated attacks, leading to either misclassification (False Positives) or missed detection (True Negatives).

A clear illustration of these evolving cyber threads is the attack on Morocco by the iranian hacker group Lyceum in November 2021. This advance operation targeted internet service peroviders and telecomunication, in Israel, Tunisia, and Saudi Arabia in addition to Morocco. Thoses incidents give a demonstation of the advancement of network attack strategies to infiltrate and compromise digital infrastructure. Such attacks not only threaten national security but also have the potential to disrupt economic stability and citizen privacy. The fallout from these attacks highlights the critical need for robust and accurate network attack detection system.

Our goal is to reduce the frequency of False Positives and True Negatives, ultimately improving the overall reliability of SIEM systems in identifying and responding to cyber threats.

This capstone project will document the journey of our research. Starting from finding of a network attack detection dataset to the training of different model until the integration and deployment of our model inside the SIEM of a cyber security lab. Throughout this project, we desire to create a helpful tool that can serve as a valuable good for network attack detection.

In summary, this research will document the development and implementation of AI models designed to augment existing SIEM detection mechanisms. By refining the accuracy of threat detection, this project seeks to create a more robust defense against evolving network attacks, enhancing the security posture of organizations.

# INTERNSHIP BACKGROUND

UM6P, Cyber Lab is a cyber security lab, locating in Benguerir, having a security operation center (SOC) responsible to prevent, detect and investigate attacks that happen on the network. The primary goal of the company for cyber security education is to detect malicious activities that might harm their network which can lead to data leakage. The company has more than 50 employees some of them in the cyber lab and others working with OCP.

My ambitions to intern at UM6P Cyber Lab is to learn in the field of cyber security and artificial intelligent. Furthermore, the Lab is working on detection of malicious activities using static and dynamic methods, which make it a good fit for me to enhance my understanding in cyber security field and how can AI further enhance it. This internship will allow me to deepen my understanding of cybersecurity and artificial intelligence, fields that I am passionate about. I am particularly excited. This opportunity will open doors to me to work on real projects, collaborate with experienced teams, and further develop my expertise in artificial intelligent.

# PROBLEM STATEMENT

The challenge of network attacks remains a critical concern in cybersecurity, posing significant threats to data security, organizational operations, and user privacy across global networks. These cyber threats increase due to a combination of factors, including inadequate security measures, lack of real-time threat detection, and the increasing sophistication of attack methodologies employed by cybercriminals.



SIEM platforms are increasingly used in network attack detection. They accomplish this by employing static rule sets known to be predefined conditions that trigger alerts and dynamic rule sets which are adaptive conditions that adjust to network changes. These platforms provide dashboards where specific thresholds can be set to monitor network activity.

Additionally, SIEM platforms enhance their detection capabilities by processing threat intelligence data, such as Indicators of Compromise (IOCs), they contribute to provide the cybersecurity team with crucial knowledge after a data breach or any malicious event occurs.

Most of the time SIEM system face problems with accuracy, mainly because of the huge log data that they process. Large volume of data lead significant noise, such as false positives that is time consuming and true negative which are not appropriately flagged as malicious event. Those problems may impact negatively the performance of the SOC (Security Operation Center) responsible for investigation by focusing on false events instead of solving on actual threats.



These accuracy issues often arise from three main factors:

1. **Limitations of Standard Detection Rules:** SIEM systems often rely on broad, predefined static rules to detect potential threats. While these rules can identify known attack patterns, they struggle to detect new or unknown threats. Even when incorporating static rules derived from Indicators of Compromise (IOCs), the SIEM may miss detecting attacks.
2. **Insufficient Contextual Information:** SIEM systems can lack detailed contextual information about the activities they monitor. Without understanding the normal behavior patterns, the SIEM cannot accurately flag anomalies. This deficiency can lead to false positives or missed detections of actual threats.
3. **Inadequate Data Correlation:** SIEM platforms may fail to effectively correlate related security events across different logs and data sources. So, failing to link related events can allow complex attacks to go unnoticed. This poor data correlation can result attacks remaining undetected, as the system doesn't connect the dots between seemingly isolated events.

After thorough analysis, we decided to focus our efforts on integrating machine learning model into our SIEM platform. This choice was driven by two main considerations: firstly, machine learning models demonstrated high accuracy after fitting them with the test dataset. making it the most consistent choice for accurate detection of network attacks. Secondly, the ease of integration of these models into the existing SIEM platform for live detection ensures that they can be seamlessly implemented without disrupting ongoing operations or requiring significant system modifications.

We precisely developed and evaluate five distinct machine learning models: Random Forest, XGB, Logistic Regression, CatBoost, each one is responsible to detect network attacks using binary and multi-classification. Our vision is to develop a machine learning models that will be integrated in the SIEM to improve the accuracy of our system and save considerable amount of time for the SOC team members. Which is a form of signature based detection. Instead of relying on static, predefined rules as in the classical signature based detection, the machine learning models learn to identify patterns from data, based on features present in the training data.

This approach will supplement analysts with additional data and insights to make better judgment calls (Gouveia & Correia, 2020).

# PROJECT SPECIFICATIONS

## 4.1 Requirement Engineering

### Functional Requirements

* **Data Collection:** 
  + The system shall be able to automatically collect real-time network activity data from various network devices, including logs, performance metrics, event notifications, and configuration details.
* **Data Processing:**
* The system shall be able to preprocess collected network data, which includes cleaning, normalizing, and preparing it for analysis.
* The system shall be able to extract relevant features from the processed network data that are indicative of network security events.
* **Model Training:**
* The system shall be able to develop and train machine learning models using historical network data to detect patterns and anomalies indicative of potential security threats.
* The system shall be able to continuously optimize the model based on new network data and feedback to enhance detection accuracy.
* **Threat Detection:**
  + The system shall be able to detect threats in real-time, analyzing incoming network data to identify suspicious activities or anomalies.
  + The system shall be able to specifically identify anomalies that deviate from normal network behavior and may signify security incidents.
* **Integration and Reporting:** 
  + The system shall be able to integrate seamlessly with the existing SIEM system, ensuring proper data input and export of findings.

### Non-Functional Requirements

**• Performance:**

* Quick detection and processing of network threats.
* Efficient reduction of false positives and true negatives.

**• Reliability:**

* Accurate identification of malicious activities across varying network conditions.
* High availability and consistent performance under diverse operational scenarios.

**• Scalability:**

* Capability to manage increasing amounts of network data with the ability to scale without a loss in performance or accuracy.
* Flexibility to accommodate new types of network threats and data sources.

**• Security:**

* Protection of live network network data from unauthorized access and tampering.
* Secure integration with existing SIEM systems.

**• Maintainability:**

* Simplified processes for updating machine learning models and system configurations.
* Comprehensive documentation to support system updates and maintenance.

**• Interoperability:**

* Compatibility with the lab’s existing architecture and various network devices.
* Seamless integration with lab SIEM and other security tools.

# STEEPLE ANALYSIS

In this section, we will explore the STEEPLE analysis, a framework utilized to examine external factors that influence or are influenced by a project.

* **Social:** The project aims to enhance the capabilities of SOC teams, enabling them to focus on more complex security challenges. This can lead to a more skilled and adaptive cybersecurity workforce.
* **Technological:** This project sets a new benchmark for integrating advanced machine learning techniques in network security, potentially inspiring further innovations and applications in cybersecurity. Demonstrates how real-time data analysis can be leveraged to enhance security measures, setting a technological precedent for future security solutions.
* **Economic:** By reducing false positives and true negatives and enabling quicker response to true threats. This project can significantly lower the cost associated with security operations.
* **Environmental:** By optimizing data processing and reducing unnecessary investigations of false alarms, the project contributes to more efficient use of computational and human resources.
* **Political:** The project aligns with national and international cybersecurity policies and regulations, promoting a secure digital environment.
* **Legal:** This project aligns with accurate cybersecurity laws and data protection, adhering to industry standard and regulatory requirements.
* **Ethical:** Ensures the ethical deployment of AI technologies, focusing on transparency, accountability, and the avoidance of bias in machine learning models. By adhering to strict data handling and processing protocols, the project prioritizes the privacy and security of user data.

# INTERNSHIP PLAN

During my combined internship I had the opportunity to succeed a project responsible for the development and integration of a machine learning model responsible for the detection of malicious events that might harm the flow of the network.

* **Objective**:

At the beginning of my internship, the company provided me with six courses on Kaggle to enhance my data science and machine learning skills. These courses covered various topics essential for the project. After completing the Kaggle courses, I began searching for a dataset compatible with the company's lab network. I started working on Jupyter Notebook, creating two separate workspaces for binary classification and multi-classification tasks. In parallel, I utilized MLflow to effectively manage the machine learning lifecycle. I was able to develop and train both classification models. Upon successfully building the models, I generated a Python script that included all the data preprocessing techniques used in the Jupyter Notebook. This step was crucial for preparing the model. Moving forward, I integrated the machine learning models into the company's Security Information and Event Management (SIEM) system. I configured the Python script to send the processed results to the SIEM, enabling real-time detection and alerting of potential threats. After integration, I conducted thorough testing to ensure that the system accurately detected malicious events without generating false positives. In parallel, the team member and I were regularly holding weekly meetings to continuously deepen my understanding in cybersecurity field.

# LITERATURE REVIEW

# ENGINERRING STANDARD

* **ISO/IEC 27001:** This standard for Information Security Management Systems (ISMS) is critical for ensuring the security of the machine learning models and processed data. The project involved integrating machine learning models into the company’s SIEM platform to detect potential network threats. Adhering to ISO/IEC 27001 ensures that all processed data and the telemetry system are secure, with measures in place to protect against unauthorized access and data breaches.
* **IEEE 802.3:** Given that the machine learning models integrated with the live network telemetry, ensuring proper data transmission between systems was essential. Compliance with the IEEE 802.3 standard ensures reliable Ethernet-based communication between the middleware, SIEM platform, and network, facilitating real-time detection and reporting of security threats.
* **NIST Special Publication 800-53:** This set of guidelines provides a comprehensive framework for implementing strong cybersecurity practices, including incident response, system monitoring, and threat detection. By following NIST SP 800-53 standards, the project ensures the implementation of robust security controls, enhancing the detection capabilities of the machine learning models and reducing the risk of false positives.
* **ISO/IEC 25010:** In developing and deploying the machine learning model, the system’s software quality is governed by ISO/IEC 25010 standards. This ensures the solution meets essential criteria for functionality, reliability, performance efficiency, and maintainability, particularly in supporting the detection of network threats with minimal false positives.
* **IEEE 829:** This standard relates to testing the system to ensure accurate threat detection and proper SIEM integration. The adherence to IEEE 829 during testing guarantees that each model's performance is validated, false positives are minimized, and the system operates as expected in real-time network conditions.

# METHODOLOGY & CAPSTONE DESIGN

A multidisciplinary approach is adopted in this project, it aim the integration of advanced machine learning techniques with the SIEM system in order to solve the problem of detection of network attack in order to improve the misclassification of attacks and avoid missing the detection of allerts. The project is structured around the development of a machine learning model able to catch any allert that might occur at the level of the network and the integration and the deployment of this model within a SIEM system.

## 9.1 Technology Enablers

After a deep understanding of the literature review, we have strategically alligned our choice of some technologies to successfully meet both functional and non-functional requirements. Below is a summary of the selected technologies:

* **MLflow**: It’s an open-source platform to manage the ML lifecycle, including

experimentation, reproducibility, and deployment (*MLflow Tracking*, n.d.). It’s used to track experiments, manage the machine learning lifecycle, and document model performance.

* **Machine learning models:** We have opted to use binary and multiclass classification on our datasets. In which we have choosen five models because of there performance: CatBoost, Logistic Regression, Decision Tree, XGBoost, Random Forest for detecting network attacks, each configured with distinct hyperparameters. After performing hyperparameter tuning using grid search algorithm in each of the models we have obtained an accuracy of 94.58%for random forest model an F1 score of 94.12%. 99.85% for XGB with an F1 score of 99.85% and an accuracy of 99.78% for the Decision Tree with an F1 score of 99.79%. The accuracy for the logistic regression was 95.06% with an F1 score of 95.22%. The high percentage of the F1 score indicating a strong balance between precision and recall, and thus a solid performance across different categories. The models were trained using data that originates from a Canadienne institute (*IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, n.d.) a wide number of malliscious and benign traffic were generated. The benign traffic were generated using B-Profile extracts the abstract behavior of 25 users based on the HTTP, HTTPS, FTP, SSH, and email protocols, the malicious attacks through actual traffic in the system. The motivation behind choosing this dataset is for the compatibility of the features with the feature inside the company I am doing my combined internship in.
* **CICflowMeter:** An open-source tool used to extract features from network traffic data, specifically from PCAP files which store captured network traffic. It is widely used in cybersecurity research to generate network flow-based features that help in analyzing and detecting different types of network attacks. A network flow refers to a sequence of packets sent from a source to a destination during a specific time period, and CICFlowMeter helps in analyzing these flows by extracting features such as packet length, flow duration, inter-arrival time, and more. These features are useful for machine learning models that detect and classify network intrusions or other anomalies. CICFlowMeter converts raw network traffic data into structured flow features, which can then be used for further analysis or as input to machine learning algorithms.

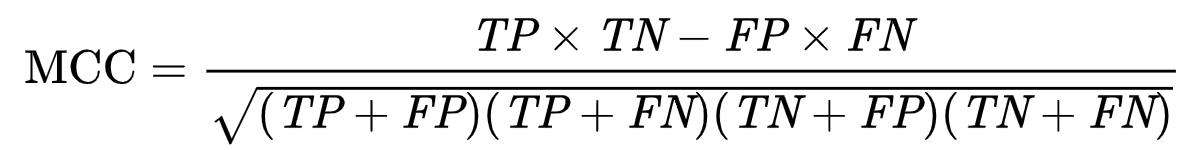
## 9.2 System Design

In this section we will explore a well modular structured architecture by explaining the role of each component and how they interact with each other.

1. CIC-IDS 2017: This dataset was generated by a Canadian institute, where they have developed a CICFlowMeter responsible for extracting features from network traffic data, specifically from PCAP files which store captured network traffic. A necessary cleaning section named the preprocessing phase was needed before moving into the training section. After successfully cleaning the dataset, we were able to train our model and extract the best one out of it.
2. Middleware
   * 1. **CICFlowMeter:** Is a network traffic flow generator distributed by CIC to generate 84 network traffic features. Using this to generate flow data which are features extracted from network traffic. CICFlowMeter is a strong choice as flow data is highly useful for intrusion detection system (IDS) are security tools used to detect malicious activity. CICFlowMeter gather packets and groups them into flows.
     2. **Real time Python-Script:** A file containing code written in python programming language, it is designed to be run as a program. All the preprocessing steps are in the python script as a function.
3. **Jupyter Notebook:** This is a notebook where we did all machine learning steps which involve:
   1. **Data collection:** This phase is responsible for collecting the right data that we want our model to use, ensuring we find the right dataset that is relevant to the problem we are solving.
   2. **Data preprocessing:** In the data preprocessing phase, we focused on cleaning and transforming the dataset to make it suitable for training the model. This involved handling missing values and infinity values, encoding categorical variables, normalizing features, and addressing class imbalance, which is crucial for detecting network attacks.
   3. **Exploratory data analysis (EDA):** We performed various analyses to understand the structure and characteristics of the dataset. This included visualizing the distribution of features, identifying correlations. EDA helped uncover insights about the data and informed decisions on feature selection and engineering for better model performance.
   4. **Machine learning models:**
      1. **Training:** In this phase we have trained using five models of machine learning, below is the explication for each model:
         1. **RandomForest (RF):** RF builds an ensemble of multiple decision trees to improve robustness and accuracy. Each tree is trained on a randomly selected subset of the original dataset, a process known as bootstrap aggregating or bagging, where both the data samples and features are randomly chosen. By combining predictions from all the individual trees, RF mitigates the risk of overfitting and reduces variance, leading to more reliable classification outcomes. The final class decision is determined by majority vote, ensuring that the model's predictions benefit from the collective intelligence of the ensemble rather than any single tree. One key advantage of RF is its low classification error compared to traditional algorithms. Additionally, random forests can save the generated forests for future use, automatically provide accuracy metrics and variable importance, and overcome overfitting problems. During tree construction, randomization is applied to select the best node to split on, with the number of features chosen for splitting each node calculated as √A, where A is the number of attributes in the dataset. This ensemble approach makes Random Forest a powerful and effective classifier for improving accuracy in network attack detection.
         2. **XGBoost:** XGBoost is an ideal model for network intrusion detection, especially for handling large datasets like CIC-IDS2017, which contains 2,827,876 rows and 66 columns. Its ability to achieve superior accuracy in classification tasks comes from gradient boosting, which focuses on minimizing errors by combining weak learners to improve detection accuracy over time. XGBoost's built-in regularization techniques prevent overfitting, ensuring strong generalization to unseen data. It also has the flexibility to deal with various types of data and can be fine-tuned with different hyperparameters to optimize performance. Additionally, it prioritizes important features, making it highly efficient when processing large and complex network traffic.
         3. **CatBoost:** CatBoostClassifier is a gradient-boosting ML algorithm known for its high predictive accuracy and speed. It employs techniques such as ordered boosting and oblivious trees to handle various data types effectively and mitigate overfitting.
         4. **Logistic Regression:** Logistic regression is a statistical method employed to solve binary classification issues by estimating the probability of an observation belonging to one of two classes. Logistic regression builds on the linear regression by employing the logistic function to transform the outcome into a limited range of values between 0 and 1. The logistic regression model is defined by a linear combination of input features, with each feature being assigned a weight and a bias term. It is commonly employed for a broad range of tasks such as anomaly detection
         5. **Decision Tree:** The decision tree structure is like the tree structure but from top to bottom, where the highest node in the tree represents the root. Each internal node represents a test on a feature, each branch indicates the result of the test, and each leaf node indicates a class label.
      2. **K-fold cross validation:** K-fold cross-validation is important because it splits the dataset into K subsets or folds and trains the model K times, each time using a different fold as the validation set and the remaining folds for training. This technique provides a more reliable estimate of a model's performance by reducing the impact of randomness and ensuring that every data point gets used for both training and validation. This helps avoid overfitting and ensures that the model generalizes well to unseen data.
      3. **Testing:** In the testing process, I am using my trained machine learning model to make predictions on the test data. By calculating several key evaluation metrics for evaluating the model's performance. Each metric serves a different purpose:
         1. **Accuracy:** Measures how many predictions are correct by providing a general picture of how often the model is correct.
         2. **Recall:** It measures the proportion of actual positives that are correctly identified. It gives an insight about how many labels did the model correctly identified, out of all the possible positive labels.
         3. **Precision:** It measures how accurate the model's positive predictions are, reflecting its capability to provide relevant outcomes.
         4. **F1-score:** It is valuable when dealing with imbalanced class distribution as it considers both false positives and false negatives.
         5. **Matthews Correlation Coefficient (MCC):** Is a metric that evaluates the quality of binary classifications by taking into account all four elements of the confusion matrix: true positives, true negatives, false positives, and false negatives. The MCC value ranges from -1 means perfect misclassification and 1 means perfect classification, with 0 representing random guessing. MCC can be used for both binary and multi-class classification, and the scikit-learn implementation of MCC supports both types.

A math equations on a white background

Description automatically generated



**Figure X:** Evaluation matrices equations

* + 1. Hyperparameters Tuning

We employed both RandomizedSearchCV and GridSearchCV techniques for hyperparameter tuning to optimize the performance of the machine learning models. We used RandomizedSearchCV, which allows for a more efficient search over a wide range of hyperparameter values by randomly sampling a fixed number of settings from the predefined parameter grid. This approach helps in quickly identifying promising parameter combinations without exhaustively evaluating all possibilities. On the other hand, for GridSearchCV, which performs a huge amount of search over a specified parameter grid, systematically testing all possible combinations of hyperparameters to ensure the best configuration is selected.

* + 1. **Receiver operating characteristic (ROC)** curve are useful for visualizing the performance of classifiers. Smaller values on the x-axis of the plot indicate lower false positives and higher true negatives. Larger values on the y-axis of the plot indicate higher true positives and lower false negatives.
    2. **Area Under the Curve** **(AUC)**, summarizes the model's ability to distinguish between classes over all possible classification thresholds. A higher AUC score indicates a better-performing model.

 **AUC = 1.0**: This indicates a perfect model with no errors in classification.

 **AUC = 0.5**: This indicates a random classifier, one that is no better than flipping a coin.

 **AUC < 0.5**: This indicates a model that is performing worse than random guessing.

* + 1. **Confusion matrix:**

A **confusion matrix** is a performance evaluation tool used in machine learning classification problems. It presents a summary of the prediction results for a model, comparing actual values with predicted ones. We have used confusion matrix for **binary classification**, the confusion matrix consists of four cells: **True Positives (TP)**, **True Negatives (TN)**, **False Positives (FP)**, and **False Negatives (FN)**. These metrics help in calculating important evaluation metrics like accuracy, precision, recall, and F1-score.

As well for **multiclass classification**, the confusion matrix expands to accommodate the multiple classes. Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class. The diagonal elements represent the correct predictions (True Positives for each class), and the off-diagonal elements represent the misclassifications (False Positives and False Negatives between classes).

This, is the github link to view the entire notebook containing the binary and multiclass classification: [Machine Learning Network Attack Detection](https://github.com/asmaemou/Capstone_Notebook.git)

1. **MLflow:** This service functions as a central hub for the machine learning workflow, including tracking of experiments, model versioning. Mlflow is responsible for tracking and managing the end-to-end machine learning lifecycle.
2. Third script Python script: Promoting the models for predictions
3. ELK: It is a group of open-source tools used within many SIEM systems. ELK system stacks Elasticsearch, Logstash, and Kibana in order to create a complete open-source log management system. ELK is a software that can be modified depending on the desire of the user with the goal of creating different platforms. It is one implementation of the SIEM.
4. Hardware:
   1. Switch and Port Span: A hardware containing a port span; used to mirror traffic to the middleware for analysis. The concept of sending copies of the network traffic flows is crucial for real time monitoring without the interaction with the user requests. This step is important to capture a copy of all the network activity in real time, for the purpose of detecting attacks.
      * Each flow has many packets

A diagram of software

Description automatically generated

Fig. 1. System architecture

The dataset under the name CIC-IDS2017 were used in our jupyter notebook. After performing many machine learning model techniques we have saved our model into mlflow. Mlflow is responsible for tracking and managing the end-to-end machine learning lifecycle. Then the python script send the result gathered whether it has found an attack or which type of attack , after that th result is endexed it is sent to the SIEM.

The network data captured through a hardware switch, where a port span is used to mirror traffic from multiple servers to a middleware system. This traffic, consisting of TCP and UDP flows, is processed by the CICFlowMeter, which converts raw network data into formatted flows suitable for analysis. These flows (data) are then passed to a Python script for preprocessing, which includes steps such as feature extraction and data cleaning. Then the python script will concatenate the feature with the predicted results from the model.

## 9.3 Models and approaches

### 9.3.1. Random Forest Classifier:

### 9.3.2. XGB:

### 9.3.3. Decision Tree:

### 9.3.3. Logistic Regression:

### 9.3.4. Naïve Bayes

### 9.3.5. SVM

## 9.4. Implementation

### 8.4.1. Data Collection

A blue rectangular bar graph

Description automatically generated with medium confidence A black background with white text

Description automatically generated

**Fig. 2. Data Distribution**

* **Benign:** It refers to the number of attacks that are not harmful. The reason behind having a very large number of benign attacks is because the Canadian institute have generated the naturalistic benign background traffic using B-profile system which is known to generate a very huge amount of data.
* **Attack:** It refers to the number of attacks that are malicious. They were generated through actual attacks that occurred at the level of their network.

There are heigh types of attacks:

1. PortScan
2. Bot
3. Infiltration
4. Web Attack
   1. XSS
   2. SQL Injection
5. Brute Force :
   1. FTP-Patator
   2. SSH-Patator
6. DoS:
   1. slowloris
   2. Slowhttptest
   3. Hulk
   4. GoldenEye
   5. DDoS
7. Heartbleed

### 9.3.2. Data preparation

The step of data preparation is considered a very crucial step in machine learning lifecycle. For an effective model training a cleaned and a prepared dataset is important to ensure accuracy and efficiency. Uncleaned data can lead to an excessive amount of memory to process, or some models may not handle it at all.

A blue bar graph with white text

Description automatically generated A black background with white text

Description automatically generated

### 

### 9.4.3. Experiment tracking and versioning

# 10. DATA PRESENTATION

# 11.SIMULATIONS, RESULTS AND INTERPRETATION

# 12.LEARNING STRATEGIES

# 13. CONCLUSION AND FUTURE WORK

# REFERENCES

# APPENDIX A: CODE

# APPENDIX B:

**Project Overview:**

In this project, I have started first to understand the problem statement that was assign to me. My goal was to look for a dataset that will be compatible with the network of the company. I was able to find one of the most popular dataset that was generated in 2017 from a Canadienne institute (*Canadian Institute for Cybersecurity*).

I started bybuilding a machine learning model. In this project I have tried at the first stage to use Kaggle since my dataset was extracted from Kaggle then after building two different notebook one for binary classification and the second one for multi-classification I was able to start working locally. I start by building a new notebook and try different methods to see what is the best preprocessing techniques that should be done based on my dataset as an example dropping rows that have duplicate since the instances of that rows is very high it will not impact the occurrence of that feature. After that I move toward adding many algorithms to clean and analyze my dataset such as k-fold cross validation.

After gathering all the plots and being able to come up with an adequate overview of my binary classification notebook I moved forward to implement multi-classification using the same dataset that I used for binary classification. First, I start analyzing how I will be able to distribute my dataset

A screenshot of a graph

Description automatically generated

A screenshot of a graph

Description automatically generated

A close-up of a graph

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**Metrics :**

A math equations on a white background

Description automatically generated

A math equation with black text

Description automatically generated

* TP = True Positives
* TN = True Negatives
* FP = False Positives
* FN = False Negatives
* The numerator (TP \* TN - FP \* FN) represents the difference between the correctly predicted observations and the incorrectly predicted ones.
* The denominator is a normalization term that ensures the coefficient falls between -1 and +1.

**MCC ( The Matthews Correlation Coefficient)  :**

1. is an important metric in machine learning, particularly for binary classification
2. Range: MCC values range from -1 to +1:

* +1 represents a perfect prediction
* 0 represents no better than random prediction
* -1 indicates total disagreement between prediction and observation

A screenshot of a graph

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

<http://localhost:5001>

kill -9 $(lsof -ti:5000,5001) to kill the port

A table of numbers and a black text

Description automatically generated with medium confidence

* In cross validation we have 4 folds for training and 1-fold for validation.
* high standard deviation, indicating less stable performance across different folds
* low standard deviation, indicating consistent performance across different folds
* Since we can notice that for all folds we have the values close to each other.
* Models evaluated:
* Naive Bayes
* Logistic Regression
* Decision Tree
* XGBoost
* Random Forest
* Performance metrics:
* Accuracy scores for each fold
* Mean accuracy across folds
* Standard deviation of accuracy scores
* Model performance comparison:
* XGBoost performs best with the highest mean accuracy (0.9993)
* Random Forest and Decision Tree follow closely (0.9987 and 0.9986 respectively)
* Logistic Regression shows moderate performance (0.8049)
* Naive Bayes has the lowest performance (0.5946)
* Consistency:
* XGBoost, Random Forest, and Decision Tree show very low standard deviations (0.0001), indicating consistent performance across folds
* Logistic Regression also shows consistency (0.0008 std dev)
* Naive Bayes has the highest standard deviation (0.0852), suggesting inconsistent performance
* observations:
* Fold-4 for Naive Bayes shows a significantly higher accuracy (0.7470) compared to other folds, which could indicate an anomaly or a particular data split that favors this model
* The tree-based models (XGBoost, Random Forest, Decision Tree) significantly outperform the other models

--------------------------------------------Multi-classification--------------------------------­

A computer screen shot of text

Description automatically generated

A screen shot of a computer program

Description automatically generated

A graph with blue lines and white text

Description automatically generated