MICROSOFT

CLASSIFYING CYBERSECURITY INCIDENTS WITH MACHINE LEARNING

Builds a machine learning model to classify Cyber-security incidents into TP, BP, and FP.

Automates triage for SOCs, improving efficiency and reducing manual effort.

Submitted By,

MIRTHU BAASHINI B

Data Science Student

Batch – MDE95

GUVI GEEK NETWORKS

ABSTRACT

This project addresses the challenge of automating the triage of Cyber-security incidents to improve the efficiency of Security Operation Centers (SOCs). Using Microsoft's GUIDE dataset, which contains labeled incident data, a machine learning model is developed to classify incidents into True Positive (TP), Benign Positive (BP), and False Positive (FP) categories. Due to the dataset's large size (9.5 million records), a sample of **1.5 million records for training** and **1.5 million records for testing** was selected for processing.

After applying preprocessing steps, including dropping unnecessary columns, high-cardinality reduction, and label encoding, the training dataset was balanced using SMOTE to handle class imbalance. Multiple models, such as Decision Tree, Logistic Regression, Random Forest, and XGBoost, were evaluated. Among these, **Random Forest emerged as the best-performing model**, achieving a macro-F1 score of **0.67** on the test set, demonstrating balanced precision and recall.

The project highlights the use of machine learning to automate SOC workflows by reducing false positives, enhancing threat prioritization, and providing reliable predictions. The final model offers scalability and practical implementation, making it suitable for real-world deployment in Cyber-security environments. By automating repetitive tasks and providing accurate classification, this solution enables SOC analysts to focus on addressing critical incidents, strengthening enterprise security operations.

INTRODUCTION

The increasing complexity and volume of Cyber-security threats have made it essential for organizations to adopt automated systems for incident management. Security Operation Centers (SOCs) play a critical role in safeguarding enterprise environments by triaging and responding to incidents. However, SOC analysts often face the challenge of manually handling thousands of alerts daily, many of which turn out to be false positives. This not only wastes valuable time but also increases the risk of missing critical threats. This project aims to build a machine learning model to **automate the triage process**, reduce manual effort, and improve SOC efficiency.

Using Microsoft's GUIDE dataset, which contains evidence, alert, and incident-level data, this project demonstrates how machine learning can be leveraged to solve this challenge. The dataset includes a mix of categorical and numerical features, requiring extensive preprocessing and class imbalance handling. A sample of **1.5 million records was used for training**, while **1.5 million records were reserved for testing**. By evaluating multiple models, including Random Forest and XGBoost, the project identifies the best-performing model for real-world deployment. The outcome is a scalable solution that enables faster response times, allowing SOC analysts to focus on genuine threats.

PROBLEM STATEMENT

Cyber-security incidents generate an overwhelming number of alerts daily, making accurate triaging essential to ensure an effective response. Misclassification of these alerts can have serious consequences:

- **False Positives (FP)**: Waste valuable analyst time and resources.
- **False Negatives**: Lead to missed genuine threats, leaving organizations vulnerable.
- **True Positives (TP)**: Require proper identification to prioritize critical incidents effectively.

To address these challenges, this project aims to develop a machine learning model that:

1. **Accurately classifies Cyber-security incidents** into True Positive (TP), Benign Positive (BP), or False Positive (FP) categories.
2. **Enhances the efficiency of SOCs** by automating the triage process and prioritizing critical alerts.
3. **Reduces manual effort** while maintaining high accuracy and reliability in incident classification.

DATASET OVERVIEW

The dataset provided by Microsoft contains hierarchical information across three levels: **Evidence**, **Alert**, and **Incident**, offering a comprehensive view of each cyber-security event. It consists of a mix of **categorical and numerical features** that capture key details about incidents, including **OrgId, IncidentId, Category, DetectorId**, and **Timestamp**. The target variable, **IncidentGrade**, classifies incidents into **True Positive (TP)**, **Benign Positive (BP)**, and **False Positive (FP)** categories. This classification helps prioritize genuine threats while filtering out false positives and non-critical incidents.

Due to the dataset's large size of **9.5 million records**, a sample of **1.5 million rows** was selected for training, with another **1.5 million rows** for testing. The dataset posed several challenges, including **class imbalance** and **high-cardinality categorical features**. Some columns, such as **IpAddress, AccountName, and AlertTitle**, contained thousands of unique values, requiring dimensionality reduction techniques to improve model performance. To address this, high-cardinality features were reduced by retaining the top 3 most frequent values and labeling the rest as **"Others."**

The dataset also contained missing values in several columns, with some exceeding 50% missing data. Such columns were dropped to maintain data quality. Additionally, the **Timestamp** column was transformed to extract the **Date** for temporal analysis, although it was later dropped as it didn’t contribute significantly to the model’s performance. Overall, the dataset required extensive preprocessing to ensure balanced class representation, manageable dimensionality, and improved predictive accuracy.

METHODOLOGY

1. Data Preprocessing:

- Dropped columns with more than **50% missing data** and handled remaining missing values.
- **Extracted date-related features** from the Timestamp column for analysis and later dropped the column as it did not contribute to the classification task.
- Applied **label encoding** to categorical variables to convert them into numerical formats.
- Reduced high-cardinality features by retaining the top 3 frequent values and labeling the rest as "Others".

2. Handling Class Imbalance:

- Used **SMOTE (Synthetic Minority Over-sampling Technique)** to balance the training dataset. The original distribution was skewed, with TP incidents dominating. After SMOTE, all classes had equal representation.

3. Model Selection:

Evaluated four machine learning models using the preprocessed training data:

- **Decision Tree:** Used as a baseline model to establish initial performance benchmarks and test basic decision-making rules.
- **Logistic Regression:** Tested for linear separability of the data, though it struggled due to the complex relationships between features.

- **Random Forest:** Selected for its robustness, ability to handle high-cardinality categorical features, and better generalization on unseen data.
- **XG-Boost:** Evaluated for its advanced tree-based learning capabilities and efficient handling of imbalanced data with strong predictive performance.

4. Model Evaluation:

- The models were evaluated on the test set using **macro-F1 score, precision, and recall** to ensure balanced performance across all classes.
- **Random Forest emerged as the best model**, achieving strong performance with balanced precision and recall.

RESULTS AND DISCUSSION

The Random Forest model emerged as the best-performing classifier among the four models evaluated, including Decision Tree, Logistic Regression, and XG-Boost. After balancing the training dataset using SMOTE, the model achieved perfect scores during cross-validation, indicating strong predictive capability. When evaluated on the unseen test dataset, which consisted of the remaining 500,000 rows, the Random Forest model achieved the following results:

- **Macro-F1 Score:** 0.67
- **Precision:** 0.71
- **Recall:** 0.66

These metrics reflect the model's ability to generalize well across all three classes—True Positive (TP), Benign Positive (BP), and False Positive (FP).

Comparison Across Models

- **Decision Tree:** Moderate performance with a macro-F1 score of **0.51**, highlighting its limitations in handling complex decision boundaries.
- **Logistic Regression:** Macro-F1 score of **0.20**, showing that the dataset's complexity cannot be captured effectively by this model.
- **XG-Boost:** Competitive performance with a macro-F1 score of **0.63**, but its computational overhead made Random Forest a more practical choice.

Feature Importance Analysis

Random Forest’s feature importance scores highlighted several key predictors that significantly influenced the model’s classification decisions. Among the most important features were:

1. **Incident Grade:** The target variable showed strong correlation with certain attributes like Category and DetectorId.
2. **Category:** Provided context for the type of threat and helped differentiate between TP, BP, and FP classifications.
3. **Date (derived from Timestamp):** Temporal patterns in incidents contributed to the model’s ability to classify recurring patterns of activity.

These insights into feature importance offer valuable information for SOC analysts, as they can help prioritize alerts based on key attributes.

Confusion Matrix Insights

The confusion matrix revealed minor misclassifications between BP and FP categories. This is likely due to overlapping feature distributions between these two classes. However, the misclassification rates were minimal, and the model's overall performance remained consistent.

Business Impact

The results demonstrate that the Random Forest model can significantly reduce the workload for SOC analysts by automating the triage process. By accurately classifying incidents, the model helps organizations:

- Focus resources on true threats (TP).
- Minimize time wasted on false positives (FP).
- Reduce the risk of missing genuine threats (FN).

This project shows how machine learning can transform SOC workflows, enabling faster and more accurate responses to Cyber-security incidents while improving operational efficiency.

CONCLUSION

This project demonstrates the successful application of machine learning to automate Cyber-security incident classification. The **Random Forest model** provides a reliable and scalable solution for automating the triage process in SOCs. The project highlights the importance of preprocessing and data balancing techniques such as SMOTE. Feature engineering, such as reducing high-cardinality features and encoding, contributed significantly to the model's performance.

By deploying this model in real-world SOC environments, organizations can **focus on genuine threats**, **reduce the risk of false positives**, and **improve response times**. Continuous updates with new data will ensure the model's adaptability to emerging Cyber-security threats, strengthening enterprise security operations.

REFERENCE

https://docs.google.com/document/d/1YJX_AxRIX9JsYlnJNMCsMiOToa_kmFmfU-ZcmA89xBA/edit?tab=t.0

DECLARATION

I declare that, this Documentation is prepared by Mirthu Baashini B, Data Science Student at Guvi Geek Network, Batch MDE95.