Microsoft Cybersecurity Incident Prediction Final Report for DSCI 632

David W. Blankenship Jr.

dwb65@drexel.edu

Table of Contents

[Abstract 3](#_Toc191938350)

[Introduction 3](#_Toc191938351)

[Motivation 3](#_Toc191938352)

[Application 3](#_Toc191938353)

[Infrastructure 3](#_Toc191938354)

[Dataset Background 3](#_Toc191938355)

[Dataset Acquisition 3](#_Toc191938356)

[Dataset Basic Characteristics 3](#_Toc191938357)

[Data Preparation & Exploratory Data Analysis 4](#_Toc191938358)

[Data Preparation 4](#_Toc191938359)

[Exploratory Data Analysis 4](#_Toc191938360)

[Missing Data Analysis 4](#_Toc191938361)

[Univariate Analysis 4](#_Toc191938362)

[Time Feature Analysis 4](#_Toc191938363)

[Mitre Techniques Analysis 4](#_Toc191938364)

[Model Development & Evaluation 4](#_Toc191938365)

[Pipeline Development 4](#_Toc191938366)

[Model Development 4](#_Toc191938367)

[Model Training 4](#_Toc191938368)

[Model Evaluation 4](#_Toc191938369)

[Conclusion & Future Scope 4](#_Toc191938370)

[Conclusion 4](#_Toc191938371)

[Future Scope 4](#_Toc191938372)

# Abstract

# Introduction

## Motivation

The motivation for this project is the increasing relevance of cybersecurity. [point to examples] Goal for this project is to assist in cybersecurity research

## Project Structure

Maybe talk about the models and overall structure of this project.

As this project is meant for DSCI 632 Applications of Cloud Computing the project is meant largely to be done using PySpark .

## Infrastructure

For this project I or we[?][][][] used separate tools for the Exploratory Data Analysis (EDA) and the Model portions. For the EDA we or I[?][][] used a Google Colab Notebook. For the model development, training, and evaluation we used a [][]][][][][]. The Notebooks for the project can be found at my GitHub page linked here: <https://github.com/General-Cow/> [][][insert full path]].

# Dataset Background

## Dataset Acquisition

This dataset is publicly available on Kaggle at the following reference [Kaggle location][][]. Additionally, the dataset is associated with a research paper [] that dives into a deeper description of the dataset, potential use cases, and the [][][]. Discuss the results.

## Dataset Basic Characteristics

The dataset comes presplit by the authors into a train and test dataset. The split is a 70:30 split and is stratified by the authors according to stratified based on IncidentGrade (our target), OrgId, and DetectorId. The train dataset is 2.3 GB and has 9516837 rows of data and 45 columns: 44 features and 1 target. The test dataset is 1 GB and has 4147992 rows of data and 46 columns. The extra column in the test dataset is a Usage column and is dropped later as it does not exist in the train dataset. The table describing the features of the dataset from the original paper [] is recreated below without alteration.

|  |  |
| --- | --- |
| Feature | Description |
| Id | Unique ID for each OrgId-IncidentId pair |
| OrgId | Organization identifier |
| IncidentId | Organizationally unique incident identifier |
| AlertId | Unique identifier for an alert |
| Timestamp | Time the alert was created |
| DetectorId | Unique ID for the alert generating detector |
| AlertTitle | Title of the alert |
| Category | Category of the alert |
| MitreTechniques | MITRE ATT&CK techniques involved in alert |
| IncidentGrade | SOC grade assigned to the incident |
| ActionGrouped | SOC alert remediation action (high level) |
| ActionGranular | SOC alert remediation action (fine-grain) |
| EntityType | Type of entity involved in the alert |
| EvidenceRole | Role of the evidence in the investigation |
| DeviceId | Additional metadata on evidence role in alert |
| Sha256 | Unique identifier for the device |
| IpAddress | Name of the device |
| Url | SHA-256 hash of the file |
| AccountSid | IP address involved |
| AccountUpn | URL involved |
| AccountObjectId | On-premises account identifier |
| AccountName | Email account identifier |
| DeviceName | Entra ID account identifier |
| NetworkMessageId | Name of the on-premises account |
| EmailClusterId | Org-level identifier for email message |
| RegistryKey | Unique identifier for the email cluster |
| RegistryValueName | Registry key involved |
| RegistryValueData | Name of the registry value |
| ApplicationId | Data of the registry value |
| ApplicationName | Unique identifier for the application |
| OAuthApplicationId | Name of the application |
| ThreatFamily | OAuth application identifier |
| FileName | Malware family associated with a file |
| FolderPath | Name of the file |
| ResourceIdName | Path of the file folder |
| ResourceType | Name of the Azure resource |
| Roles | Type of Azure resource |
| OSFamily | Family of the operating system |
| OSVersion | Version of the operating system |
| AntispamDirection | Direction of the antispam filter |
| SuspicionLevel | Level of suspicion |
| LastVerdict | Final verdict of threat analysis |
| CountryCode | Country code evidence appears in |
| State | State of evidence appears in |
| City | City evidence appears in |

Table whatever [][][]: Descriptions of columns from original paper [reference]

[make sure the above takes up one page on its own]

Mention privacy aspect

# Data Preparation & Exploratory Data Analysis

Consider changing title to EDA & Feature Engineering

## Data Preparation

## Exploratory Data Analysis

### Missing Data Analysis

I or we[?][] first look at which columns have missing values and find that MitreTechniques, ActionGrouped, ActionGranular, EmailClusterId, ThreatFamily, ResourceType, Roles, AntispamDirection, SuspicionLevel, LastVerdict, and our target IncidentGrade are all missing data. The raw missing count and the percentage of the total missing data are given in TableX[][][] below.

|  |  |  |
| --- | --- | --- |
| Column Name | Missing Count | Missing Percent |
| MitreTechniques | 5468386 | 57.46 |
| ActionGrouped | 9460773 | 99.41 |
| ActionGranular | 9460773 | 99.41 |
| EmailClusterId | 9420025 | 98.98 |
| ThreatFamily | 9441956 | 99.21 |
| ResourceType | 9509762 | 99.93 |
| Roles | 9298686 | 97.71 |
| AntispamDirection | 9339535 | 98.14 |
| SuspicionLevel | 8072708 | 84.83 |
| LastVerdict | 7282572 | 76.52 |
| IncidentGrade | 51340 | 0.54 |

Table description [][][]

I or we[?][][][]] elect to drop the MitreTechniques, ActionGrouped, ActionGranular, EmailClusterId, ThreatFamily, ResourceType, Roles, AntispamDirection, SuspicionLevel, and LastVerdict due to the sheer amount missing. While MitreTechniques has significantly less missing than the rest, it is still a significant majority of the data. We also elect to dive deeper into analysis of the data we do have for this column later. For our target column IncidentGrade, due to the sheer volume of data we already have, we elect to simply drop the null values rather than attempt to impute the target values using something like a KNN imputer. It also avoids the possibility of inserting bias or data leakage.

### Univariate Analysis

Break into analysis of categorical/pseudo-categorical as well as the numericals

For cats show distributions, consider doing OHE for appropriate pseudo-cats

For numericals get the stat description, histograms and maybe boxplots.

Id has 730778

Bivariate Analysis do I want to bother?

### Time Feature Analysis

Each row of data has an associated timestamp for the incident in question. As there may be a time component relevant to identifying cyber[][][]. I or we[?] elect to perform feature engineering on this feature and break the original timestamp column into 8 columns: Year, Month, Day, Hour, Minute, Second, Day of Week, and Week of Year. We analyze the variables as well.

### MITRE Techniques Analysis

MITRE Techniques are [][][]. While we will be dropping the column due to the volume of missing values, inspecting it will be revealing as to the nature of the most common types of attacks seen in the train dataset.

# Model Development & Evaluation

Preamble talking about GCP?

### Pipeline Development

For our model I or we[?][] develop a pipeline for our Pyspark based model. The pipeline has the following features

* No feature scaling required due to our selection of models.
  + Naïve Bayes and Gradient Boosted Trees both do not require scaling.
* We drop the following columns with high percentages of nulls:
  + MitreTechniques, ActionGrouped, ActionGranular, EmailClusterId, ThreatFamily, ResourceType, Roles, AntispamDirection, SuspicionLevel, LastVerdict
  + We also drop the Usage column in the test dataset.
* We drop target nulls as they are a very small portion of our dataset and do not consider it wise to impute the target.
* We develop a custom timestamp transformer to convert our timestamp column into the following new features:
  + Year, Month, Day, Hour, Minute, Second, Day of Week, and Week of Year
* For Category, EntityType and EvidenceRole we use the StringIndexer function to convert the[][][] and the OneHotEncoder to encode these columns as [][][]
  + We also use the string indexer on the target IncidentGrade
* We use the VectorAssembler function to [][][] to get our data into a format that allows us to [][]
* Finally, we create a Pipeline that allows us to easily stage our full dataset and transformations in a manner easy to prepare our data for model training

May need to make another for gbt if hyperparam is different

### Model Development

As a baseline model we elect to use Multinomial Naïve Bayes. [note that feature independence assumption may be bad]

Gradient Boosted Tree Classifier with hyperparam tuning

### Model Training

The Naïve Bayes model is relatively simple to run. We use the baseline modelType for Pyspark which is the multinomial option.

GBT and train results

### Model Evaluation

For our evaluation we use the F1 scores of the classes, the Macro-F1 score, and related measures of the precision and recall. The Macro-F1 score is the average of the different classes F1 scores. These are the recommended metrics from the original authors [reference]. Additionally, we look at the Confusion Matrix as given by the crosstab function in PySpark.

# Conclusion & Future Scope

### Conclusion

### Future Scope

Mention missing data, mitre techniques (desire more of this data, focus on training only subset with mitre, consider UNK imputation and breaking up the different techniques ie T1110 and T1110.003 are dif colums in in the same original), narrow bias of time for the dataset.

# References