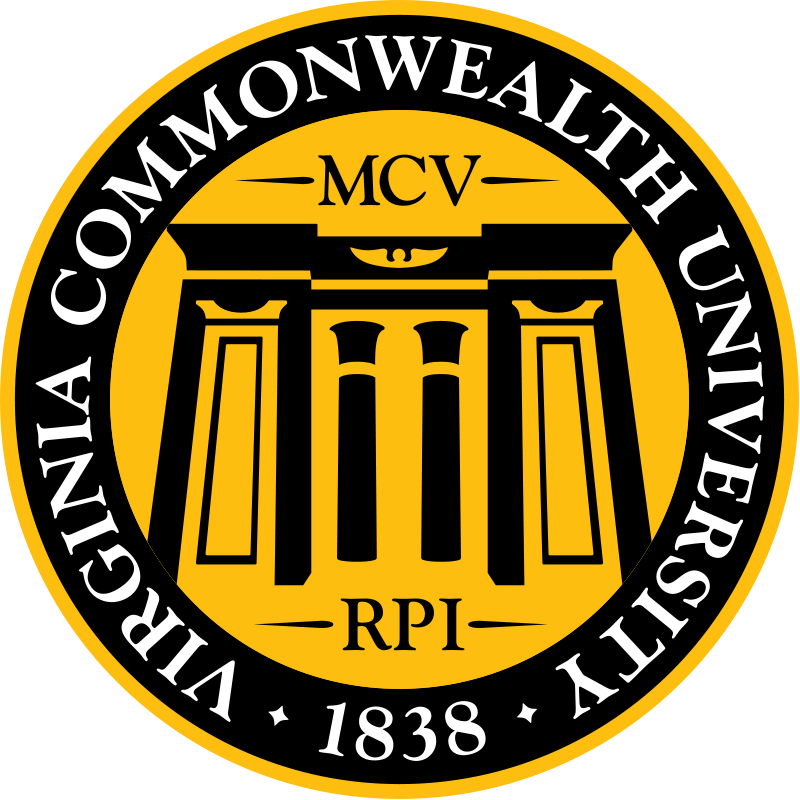
**EMAIL ANOMALY DETECTION**

**CAPSTONE SENIOR DESIGN 2019 - 2020**



### 

### **PREPARED BY**

Bharat Venkat

Jeremy Bailey

Liam Fernando

Mohammad Malik

**TABLE OF CONTENTS**

**Design Process 4**

Current State / Description of tool 4

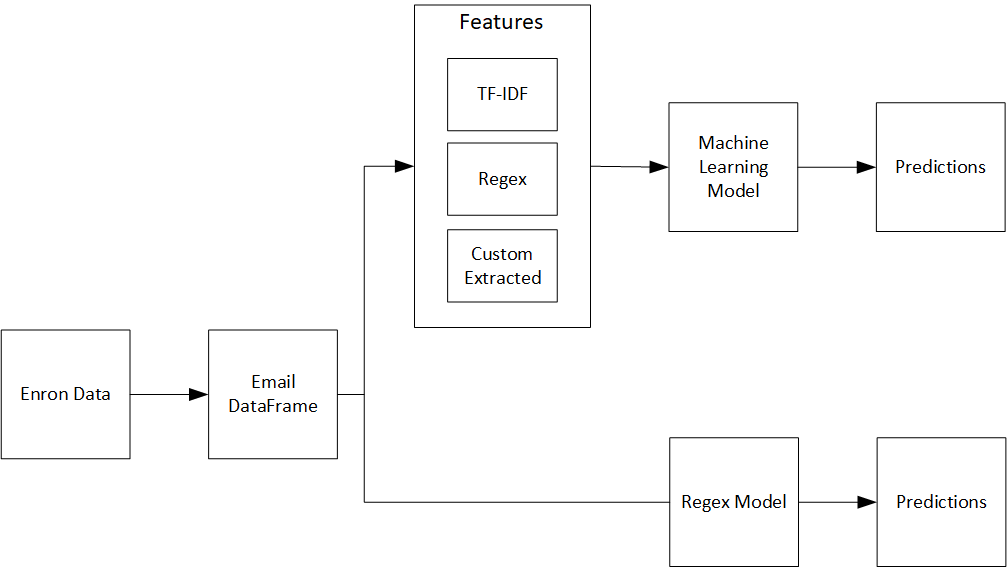
Future Product 4

Functionality 4

**Conclusion 5**

**DESIGN PROCESS**

## Architecture Diagram



## Current State/Description of Tool

The purpose of this tool is to detect the exfiltration of sensitive information from emails and email attachments. The tool will be able to scan the body of an email for any personal identifiable information as well and known sensitive material. If any data is identified it will flag it.

Features the tool offers.

* Able to determine the feel of the email where it can be positive or negative.
* Ability to identify a large list of known personally identifiable information
* Determine if code is being exfiltrated.
* Check any type of attachment and scan the attachment for data.
* Analyzing images sent in attachments for sensitive information.
* Determining the feel of an email weather it be positive or negative

The tool also provides numerous analytics about the emails. Information like the length of the email and number of characters.

-- what the tool is and what features the tool will have.

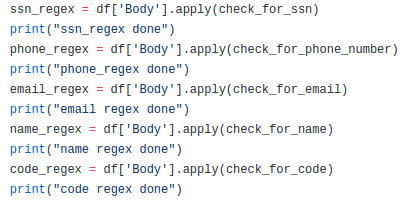
The current state of the project outlines the current capabilities of the tool.

The dataset we are building our model from is the Enron Email dataset. This dataset is widely used because it is a very large repository of real emails. An issue that we have had with this dataset is that it has already been cleansed of sensitive information: what we want to predict with our model.

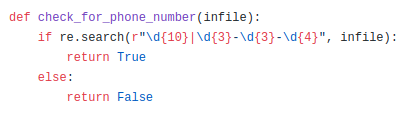
To process the dataset, we’ve parsed each email into a dataframe. Dataframes (both Pandas and PySpark) are easy to work with and process. Sensitive data has been injected (using the Python Faker library) into selected emails. These emails were then labeled as Anomaly emails. We also prescreen the emails before injecting data to see if they already have sensitive content. If they do they are also labeled as Anomaly emails.

Our dataset is balanced (even number of positive and negative samples). The relevant data that we decided to work with is data from the body of the emails (necessary because our injection of data was random: using other info like sender would create false correlations that the model might pick up on).

We have also created a regex model to be able to determine if sensitive information is within our dataset. An example of what type of information is being detected by the regex can be seen below. This model serves as a baseline for the machine learning model.



Each of these types of checks are done on the body of the email. A detailed view on how the regex checks if an email contains a phone number is shown below.



In addition to the rule based model we also created an NLP model to flag emails anomalies or not. We decided to adopt a binary classification approach for simplicity so that we didn’t have to tune our model to perform well at classifying many distinct classes. The features we used in our models are described below.

Email Details

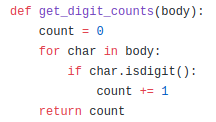
*Length of an email using the built in len() function in python3*



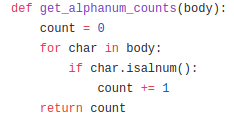
*Number of lines in an email by looking for the count() function in python3*

**

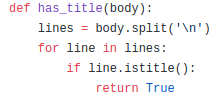
*Number of numeric characters by iterating through each character to see if it is a digit*

**

*Number of alphanumeric characters by iterating through each character and checking with the built in isalnum() function in python3*

**

*If there is a title in an email using the built in istitle() function in python3*



// Template for future implementation.

Regex Based Features

In addition to the aforementioned features we also took elements from the rule based model described earlier, and incorporated each of the regex checks as a feature.

TF-IDF

The last type of feature set we used was Term Frequency - Inverse Document Frequency. TF-IDF is very popular in NLP tasks, and is a measure of how important a given word is to a document in a collection. We thought TF-IDF would be a good feature set to play with because it would give us information on specific words that possibly could indicate some type of exfiltration attempt.

Modelling

PySpark has out of the box support for a number of classification algorithms. We tried Logistic Regression for its simplicity first: it is easily understandable and we thought it would be a good starting point. We then moved on to Support Vector Machine, a more powerful, but black box algorithm. The final algorithm we tried was Gradient Boosted Trees (GBT). GBT is one of the more powerful classical machine learning algorithms, providing good results. This comes at a tradeoff at being very computationally expensive. We were limited in our modeling architectures because the Capital One team wanted to work with PySpark models exclusively. We wanted to explore some deep learning models but PySpark doesn’t have support for these

Evaluation:

Our best performing model was the Gradient Boosted Trees trained on a combination of Regex features and custom extracted features. We evaluated our model using 5 fold cross-validation. We parameterized our model using a grid search. The optimal model had results of:

Test Area Under ROC 0.79991494991495

TP: 78

TN: 553

FP: 19

FN: 107

Sensitivity: 42.16216216216216

Specificity: 96.67832167832168

predictiveACC: 83.35535006605019

MCC: 0.4993564628245596

We think this is a good starting model and aim to improve it in further iterations.

-- Update based on what we have done

need to add what we can do now

## Future Product

The future of the project outlines the features and capabilities that we hope will be added in the future.

We want to have a tool that more accurately classifies sensitive emails than our current model. We also want our tool to be able to work with different types of attachments (pdf, image). Our current model will serve as the backbone for all of this: we will improve it by trying additional feature sets. We are also exploring using different labeling techniques for our emails: hand labeling may be more accurate than the regex based labeling (for preexisting sensitive info) we’ve been using. Reference the the detailed description of the document to see the intended features and design of the tool.

-- what we hope to be able to do. -- updated as next obtainable state.

## Functionality

The Email Anomaly Detection tool is used to detect when sensitive information is being exfiltrated through externally and internally sent emails. It will be designed to work as a stand alone tool as well as a tool that can be added to a set of cyber security related tools a company already uses. The tool should be able to provide all the information a cyber security manager would need while covering large amounts of email data. Because the tool uses machine learning, it will improve over time and new features can be added to better meet the needs of a company into the future.