Progress Evaluation: Milestone 3

Project: *The Phisherman - AI-Powered Phishing Email Detector*

Team Members:
Curtis Jones - *cjones2022@my.fit.edu*
Elton Batista - *ebatista2022@my.fit.edu*
Jordan Chesley - *jchesley2022@my.fit.edu*

Faculty Advisor: Khaled Slhoub - *kslhoub@fit.edu*

Client: Khaled Slhoub - *College of Engineering and Science: Department of Electrical Engineering and Computer Science*

Milestone 3 Task Matrix

Task	*Completion*	*Jordan*	*Elton*	*Curtis*	*To Do*
Jira Setup	*100%*	*Implement epics and user stories into Jira, refine backlog*			*Completed*
AI model testing	*100%*	*Parse and evaluate collected data to choose model*	*Extensively test LLM models with datasets*	*Retrain Models using Google Colab for more accurate testing, extensively test non LLM models with datasets*	*Completed*
AI model training	*100%*	*Collect more quality datasets and combine into one large dataset for testing*	*Research different ways to train the selected model (either from scratch or using a different library) and do performance/accuracy testing*	*Train and test selected model on larger dataset*	*Completed, continuously train model based on changes in tools we use*

Web Interface Construction	*100%*	*Implement OAuth API to feed user's emails into web application*	*Format user's emails into web application*	*Develop AI Model Processing service*	*Rudimentary web application with dashboard and OAuth login complete, remaining work involves UI and refinement of the web app through remaining Milestones*

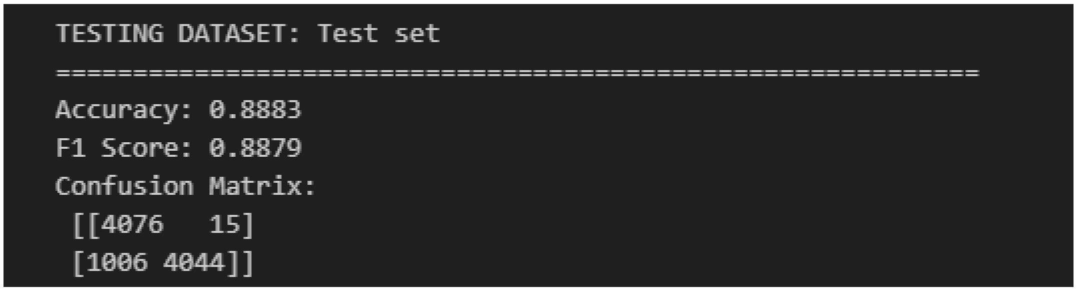
Milestone 2 Task Discussion:

Jira Setup

We have successfully integrated Jira into our workflow. With the next semester of the project, we will be able to implement our testing procedures through milestones 4-6 in Jira.

AI Model Training and Testing

We utilized additional resources to provide us with better hardware for training and testing. This involves a GPU that provided us with significantly more power, leading to better performance times simulating an ideal deployment environment while making training times incredibly faster. With help from a PHD student, we were able to refine our training methodologies, including more widely used libraries and more refined testing procedures, including better statistical analysis. This allows us to test the various functions we have implemented or have yet to implement without interference from major discrepancies in the model. As of now, our web app's artificial intelligence functionality is powered by a trained RoBERTa model. Pictured below is the accuracy and performance of our model. (Note: This test was done on a weaker computer than the model was originally trained on, therefore the performance result is lower than it would be when deployed on a proper server.) The largest issue with our implementation of the model so far is these newer emails. While in testing, as you'll see below as well, the model has a tendency to give false negatives on promotional emails.



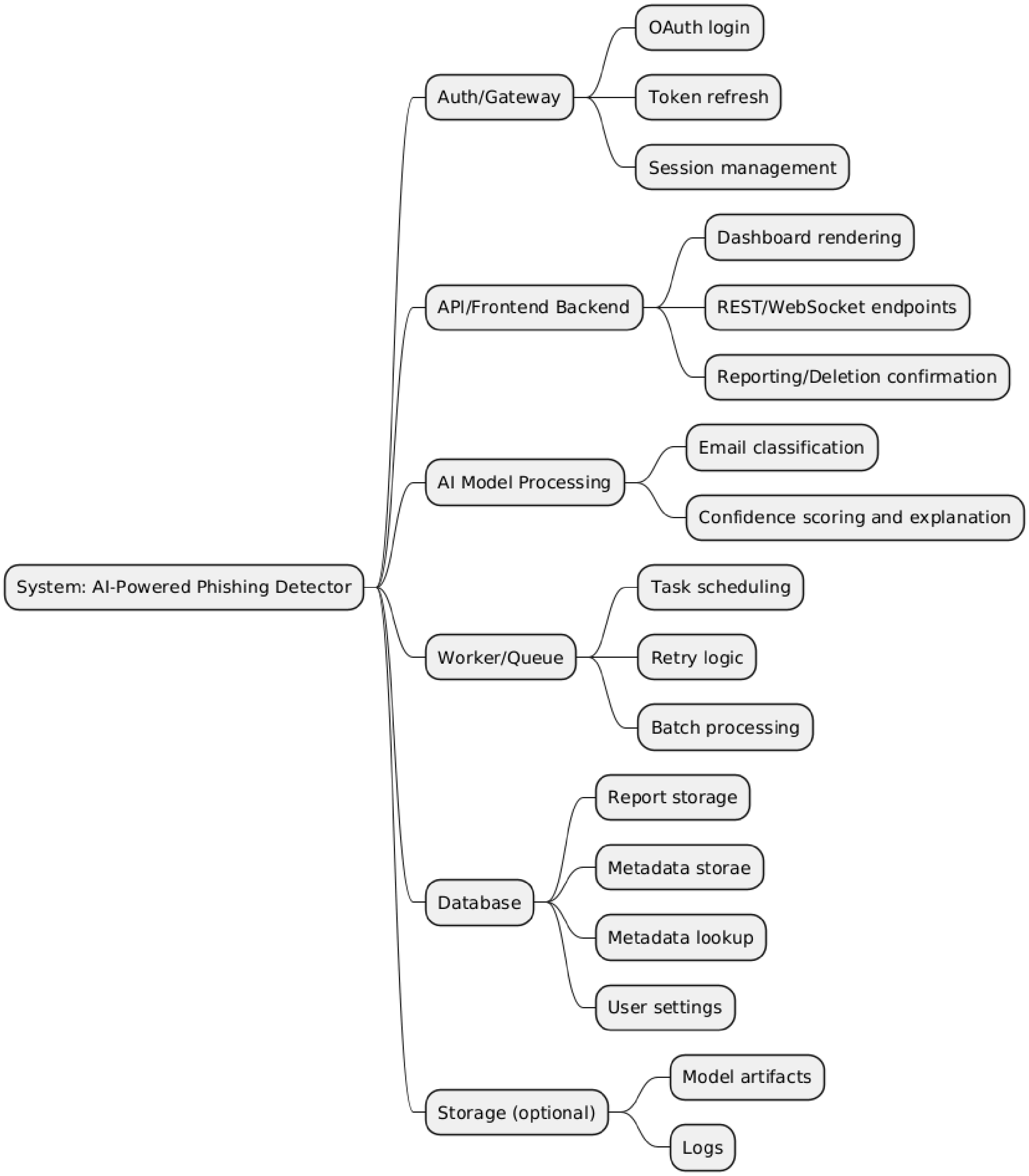
```
TESTING DATASET: Test set
============================================================
Accuracy: 0.8883
F1 Score: 0.8879
Confusion Matrix:
 [[4076   15]
 [1006 4044]]
```

Web Interface Construction

A rudimentary web interface has been constructed, including the base functionalities for our web app. This includes our implementation of OAuth as the login page, the first implementation of a functional dashboard, and our housing for the artificial intelligence model. This provides us a baseline moving forward to work off of. Services yet to be implemented include Worker/Queue for bulk processing and the database. As of now, the web app satisfies the following Function Executions as defined in our design document:

User logs in with OAuth
Emails fetched via Email Provider API
API/Frontend shows results to User

Decomposition Tree



Functionality with some or full implementation:
OAuth login
Session management
Dashboard rendering

REST/WebSocket endpoints
Email classification
Confidence scoring

Demo Web App (Note: the web app was not running on the best hardware; a server hosted on better hardware could take as little as 10 seconds to finish classification):
Youtube Link

Discussion of the contribution of each team member to the current Milestone

Curtis Jones

- Developed the AI Model Processing service
- Trained and integrated first implementation of our AI model
- Researched additional methods for collecting datasets
- Coordinated with outside resources to improve methodologies

Elton Batista

- Developed scripts for data collection, focusing on accuracy and performance.
- Integrated the model into a web browser for accessibility and usability.
- Researched and tested methods to improve demo security.
- Investigated Gmail API tools to match project requirements.
- Created a backlog of epics and user stories based on existing documentation to guide project planning.

Jordan Chesley

- Developed first implementation of Auth/Gateway and API/Frontend Backend service
- Constructed the fundamental structure of our web interface
- Researched the implementation of OAuth to use more email providers, including organizational emails

- Researched methods to address active security concerns that will affect development moving forward

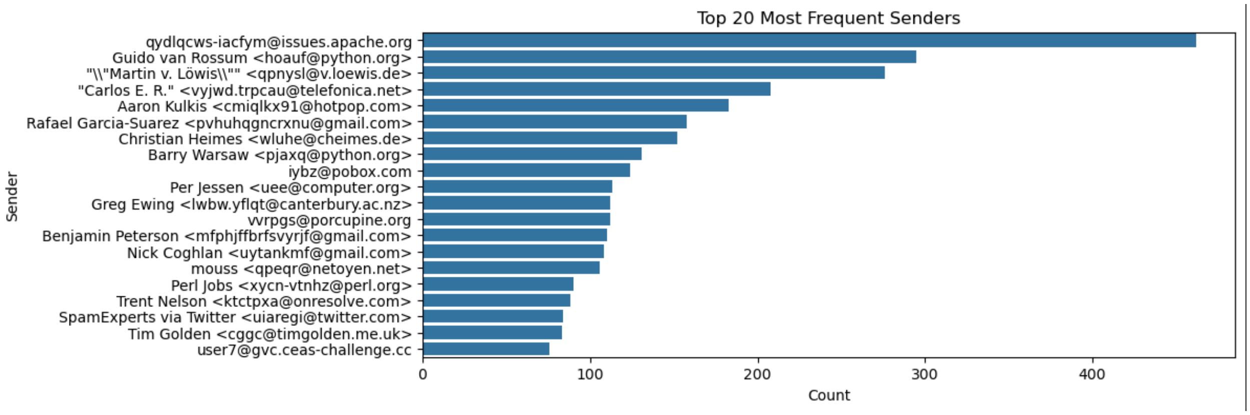
Milestone 4 Task Matrix

Task	*Jordan*	*Elton*	*Curtis*
AI model training	*Format new collected datasets*	*Research additional ways to collect datasets*	*Continuously train models on more datasets for future improvement*
Develop remaining services	*Develop database service*	*Develop storage service*	*Develop worker/queue service*
Develop remaining functions for existing services	*Implement delete/report functionality into existing services*	*Implement functions into the UI*	*Implement statistics / reports functionality through developed database service*
Develop UI elements defined by the Design Document	*Refine dashboard UI to match Design Document*	*Research UI tools*	*Begin construction of Homepage, Loading screen, and statistics page to Match Design Document*

Milestone 3 Task Discussion:

AI Model Training

We plan to introduce new methodologies in collecting datasets for this project. We plan to reach out to cybersecurity students as well as researchers in order to collect more quantitative data. The datasets we have been using currently contain a large amount of repeat emails, as seen in the chart we generated below. While this is beneficial in targeting constant spam in inboxes, more diverse datasets are ideal to cover a larger range of methods scammers use in their phishing emails. We would also like more up to date emails as the nature of these attacks are constantly changing. Also, while we have an acceptable accuracy for the time being, we aim to have above a 90% accuracy rate by the end of the project.



Develop Remaining Services

We have yet to implement the following services into our web interface:
Worker/Queue
Database
Storage

We plan to integrate these services into our web server to provide key functionality for our web app to make it more intuitive for the user.

Develop remaining functions for existing services

Sequence of Function Execution (Example Flow):

User logs in with OAuth
Emails fetched via Email Provider API
Emails queued (Worker/Queue)
Worker dispatches jobs to AI Model Processing
Classification results stored in Database
API/Frontend shows results to User
User confirms deletion/report
Email Provider updated

So far, we have implemented:
User logs in with OAuth
Emails fetched via Email Provider API
API/Frontend shows results to User

The remaining functions must be implemented in accordance with our defined use cases. This includes asynchronous functionality that allows users to see real time updates to their email inbox. In addition too this, we will use our existing functions to build out our statistics page.

Develop UI elements defined by the Design Document

We will research UI tools to allow us to develop our UI elements. The current implementation is extremely rudimentary and inefficient. This involves developing our remaining web pages and updating our dashboard UI to expose less of our design elements to our user through our HTML files.

1. Date(s) of meeting(s) with Client during the current milestone:

   10/6

   10/20

2. Client feedback on the current milestone
   - ... (if Client and Faculty Advisor are the same, write "see Faculty Advisor Feedback below")
   - ...
   - ...
3. Date(s) of meeting(s) with Faculty Advisor during the current milestone: …

   10/6

   10/20

4. Faculty Advisor feedback on each task for the current Milestone
   - Task 1: Satisfied
   - Task 2: Satisfied
   - Task 3: Satisfied
   - Task 4: Satisfied
5. Faculty Advisor Signature: ___**Dr Slhoub**____________ Date: ___**11/23/25**_____