PRESENTED BY: JUSTIN JOY
COMPUTER SCIENCE AND MATHEMATICS
ADELPHI UNIVERSITY, NEW YORK
CAPSTONE ADVISOR: DR. SUKUN LI
SPRING 2024

Preventing Phishing Attacks

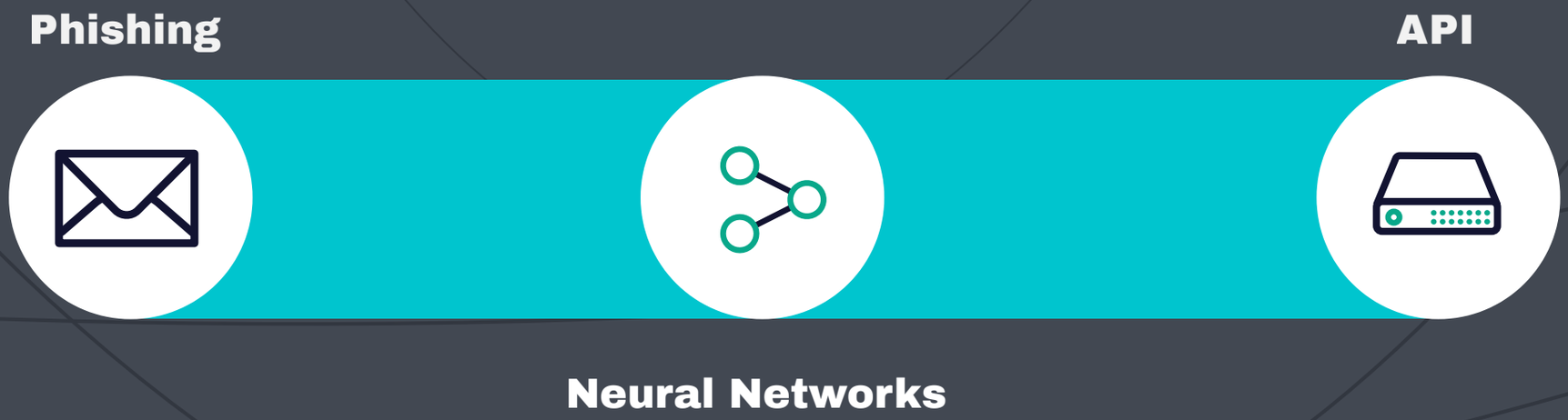
Using Deep Learning

INTRODUCTION

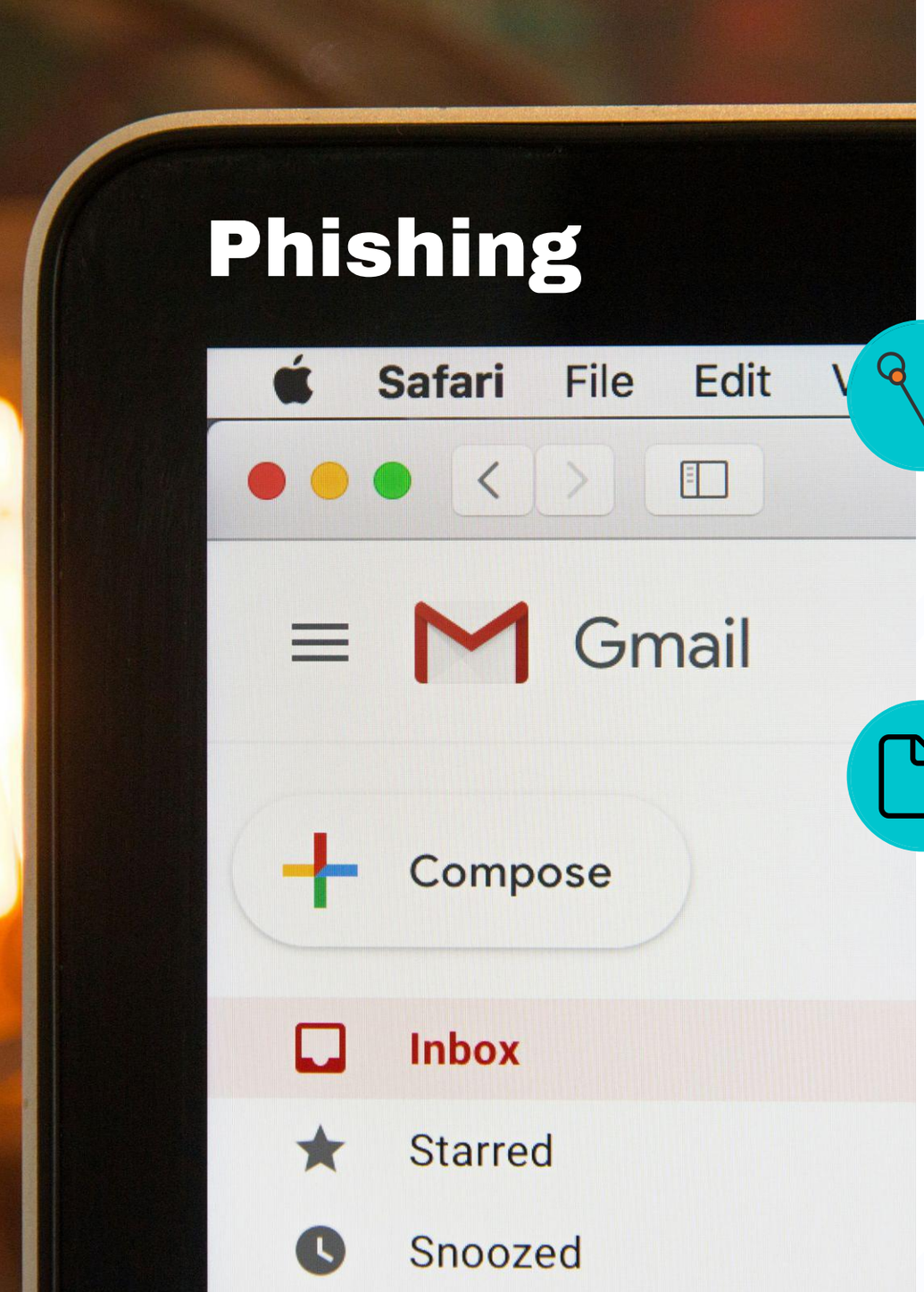
- **Objective:** To create a program that will monitor an email inbox and detect phishing emails using Artificial Intelligence (AI) and other analytics.

- **Task:** Experiment between different AI models and with the combination of AI and third-party APIs, generate a risk assessment report for the user.

Background Information



Phishing



What is a phishing email?

Phishing emails are deceptive emails containing malicious links to fake websites to gather a user's personal information and data.

How can you tell if an email is a phishing email?

It's very difficult to detect a phishing email, however there are some key patterns found in most phishing emails. For example, generic greetings, words that relate to urgency, insecure links, spelling mistakes, etc.

User Definition

- According to Forbes, New York is the 5th most affected stated by phishing scams.
- According to statista, Bulk phishing was the most widespread phishing scam and financial institutions were targeted the most.

ENTERPRISE USAGE

- The goal is to make a tool that companies and businesses can deploy, so that their emails can be monitored, and phishing attempts can be detected quickly.
- Employee emails would be continuously monitored by AI and if a possible phishing email is detected, a report will be sent to their email.

INDIVIDUAL USAGE

- Phishing attacks can affect many individuals lives as well, so this tool would be beneficial for non-enterprise use as well.
- Individual Users can use this program the same way, to monitor their own emails to protect them from phishing emails.

Artificial Neural Networks

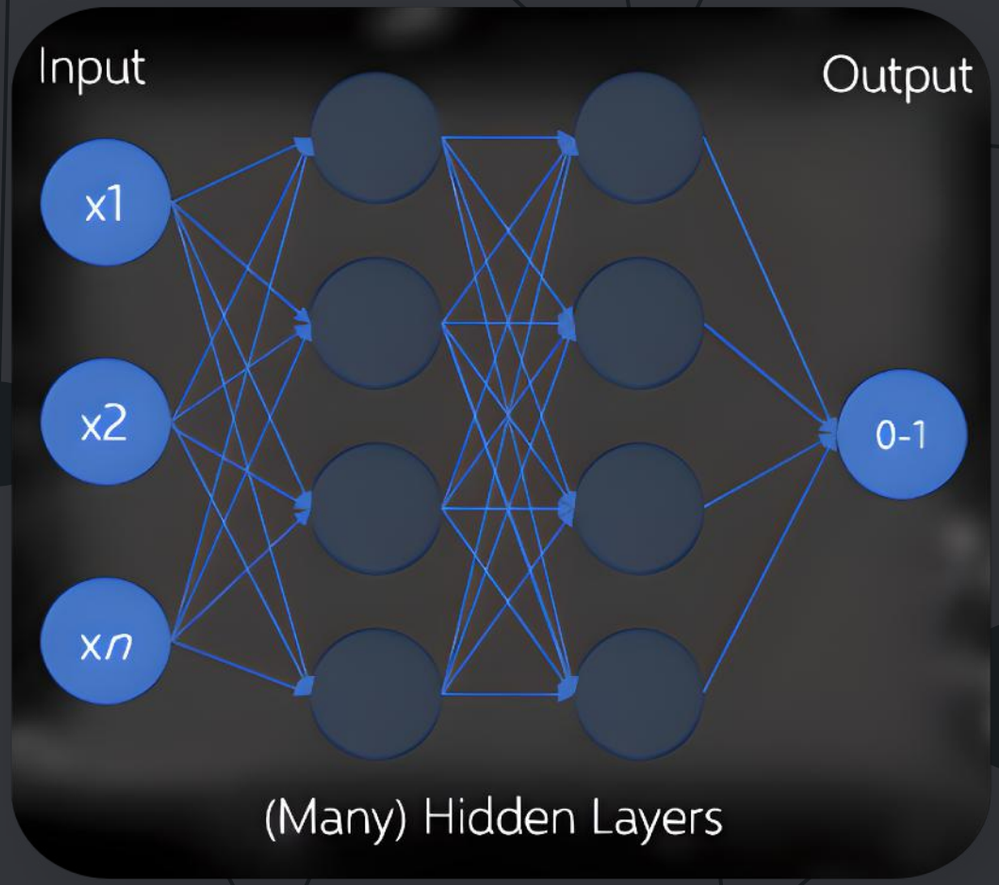
A COMPUTER MODELED AFTER THE HUMAN BRAIN

- Artificial Neural Networks are inspired by the human brain.

Consists of:

- Nodes or "neurons" are organized in layers.
- The model on the right is a simple feed forward model:
  - **input layer** = receives the input features from the dataset. Each neuron in this layer represents a feature of the input data.
  - **hidden layers** = where most of the computation takes place in an artificial neural network.
    - Each neuron receives inputs from the neurons in the previous layer.
    - The inputs are multiplied by the weights, and biases are added.
    - The result then goes through an activation function, which introduces non-linearity into the model, thus allowing the model to learn complex patters.
  - **output layer** = a final prediction from the model

- The goal is to separate the data into smaller pieces that can be reassembled by the model in a way that can help it understand or make predictions, like how the human brain would piece together information.



Sourced from Stanford University

LSTM and Transformer Models

FOR THIS STUDY, AN LSTM MODEL AND TWO TRANSFORMER MODELS (BERT/GPT) WILL BE USED

Long Short-Term Memory (LSTM)

Long Short-Term Memory Network (LSTM) is a type of deep learning, sequential neural network designed to retain information over extended periods. This is a special type of Recurrent Neural Network (RNN) which cannot retain previous information.

Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT), is a natural language processing (NLP) model developed by Google, designed to understand the context of words in text by considering both the words before and after each word, enabling it to understand a semantic meaning.

Generative Pretrained Transformer (GPT)

Generative Pretrained Transformer (GPT), is a model that can perform NLP tasks, developed by OpenAI. These models are also pretrained on a large dataset. This model is designed to predict the next word based on the previous words, making it excellent at generating text.

Methodology and Results

THE METHODS USED TO CONDUCT THIS STUDY AND THE OUTCOMES

Datasets

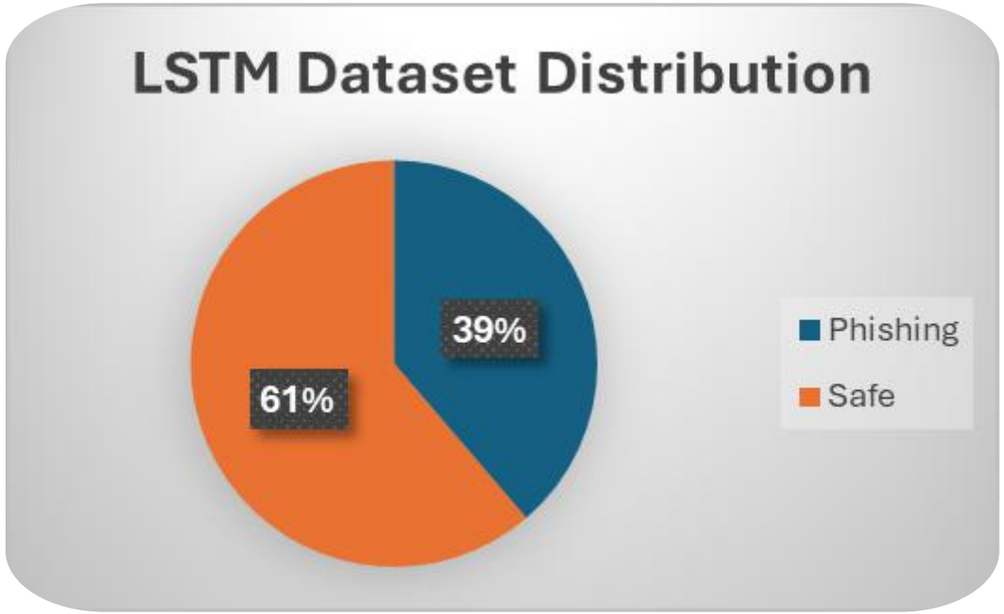
Datasets were sourced from Kaggle, UC Berkley and Generated Using ChatGPT

LSTM Model

LARGE DATASET

This Dataset consisted of 18,211 emails.

- 7,055 were phising emails
- 11,156 were safe emails



Datasets

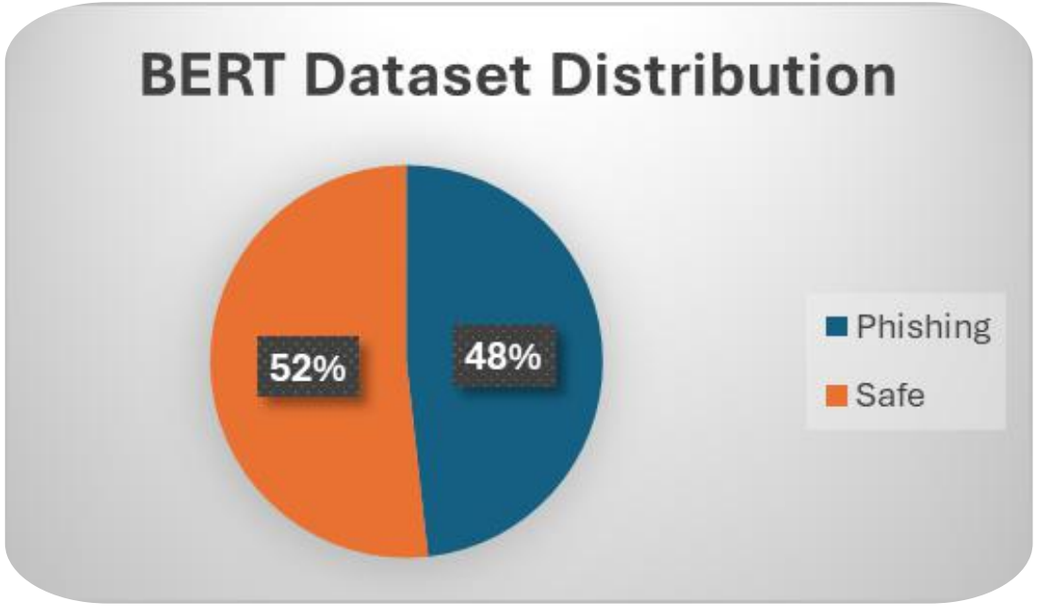
Datasets were sourced from Kaggle, UC Berkley and Generated Using ChatGPT

BERT Model

SMALL DATASET

This Dataset consisted of 199 emails.

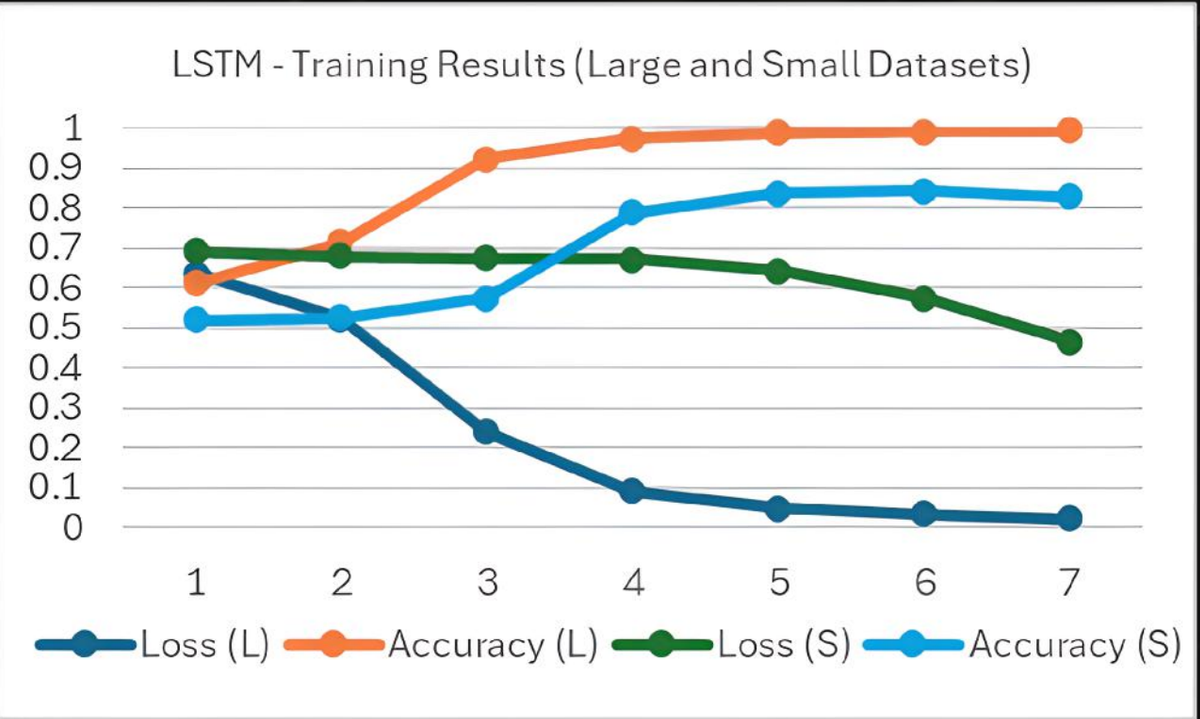
- 96 were phising emails
- 103 were safe emails



LSTM Training (Large vs. Small)

LSTM Training Results (Large and Small Datasets)

	Loss (L)	Accuracy (L)	Loss (S)	Accuracy (S)
1	0.6379	61.21%	0.6925	51.94%
2	0.5225	71.58%	0.68	52.71%
3	0.2396	92.49%	0.6751	57.36%
4	0.0901	97.56%	0.6722	79.07%
5	0.0473	98.86%	0.6428	83.72%
6	0.0338	99.22%	0.5746	84.50%
7	0.0216	99.53%	0.4657	82.95%
avg.	0.227543	88.64%	0.62899	70.32%

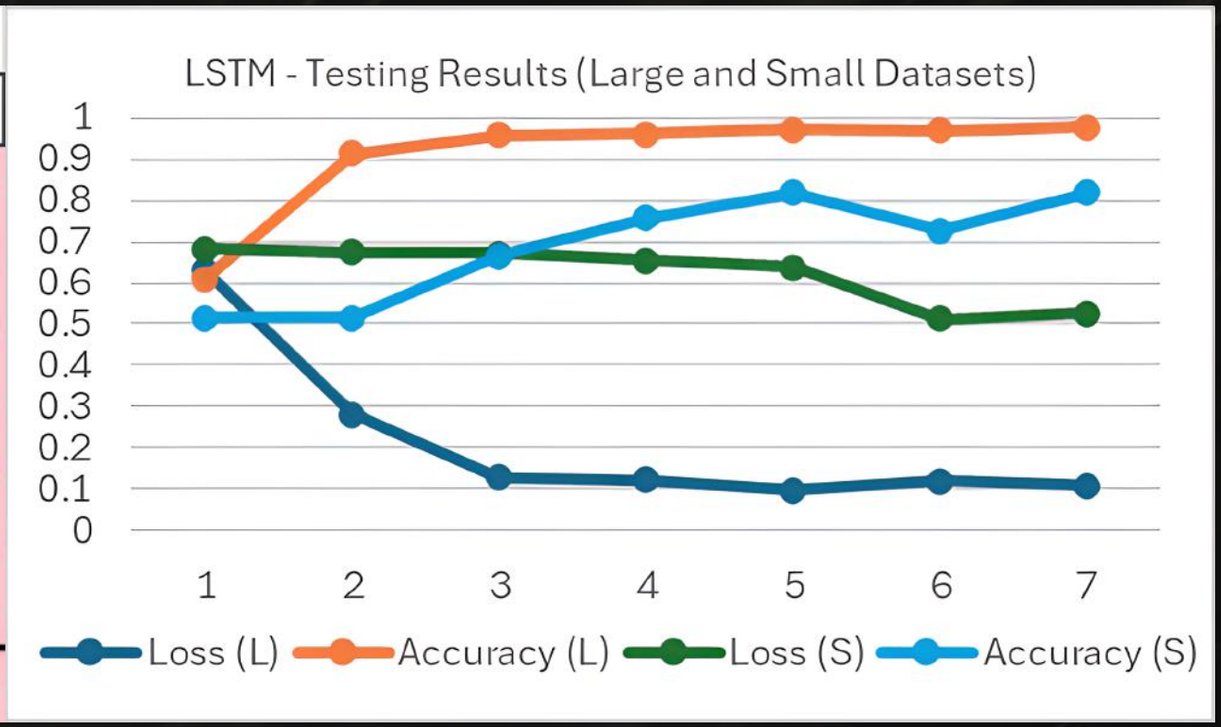


The LSTM model performed better on the larger dataset. The accuracy increases to 99% and the loss decreases to 2%. Whereas, the smaller dataset only peaked 82% accuracy and had a relatively high loss at 47%.

LSTM Testing (Large vs. Small)

LSTM Testing Results (Large and Small Datasets)

	Loss (L)	Accuracy (L)	Loss (S)	Accuracy (S)
1	0.6288	61.07%	0.6822	51.52%
2	0.2807	91.26%	0.6731	51.52%
3	0.1274	95.66%	0.6723	66.67%
4	0.1216	96.02%	0.6532	75.76%
5	0.098	97.23%	0.6378	81.82%
6	0.1212	96.81%	0.5128	72.73%
7	0.1098	97.61%	0.5267	81.82%
avg.	0.2125	90.81%	0.62259	68.83%

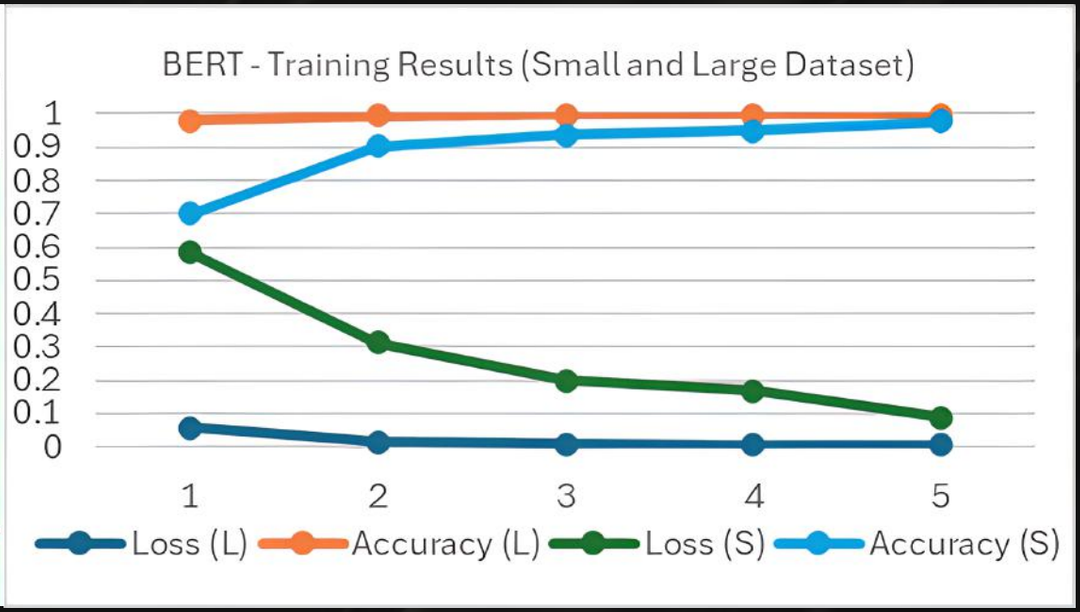


The LSTM model's testing results also show improvements in the larger dataset. Although the accuracy slightly dips down and loss slightly creeps up towards epoch 6, it quickly stabilizes at the end, showing that the model may not be overfitting

BERT Training (Large vs. Small)

BERT Training Results (Large and Small Datasets)

	Loss (L)	Accuracy (L)	Loss (S)	Accuracy (S)
1	0.0568	98.02%	0.5826	70.09%
2	0.0145	99.54%	0.3138	90.28%
3	0.008	99.76%	0.1988	93.61%
4	0.005	99.83%	0.169	95.00%
5	0.0058	99.81%	0.0886	97.78%
avg.	0.01802	99.39%	0.27056	89.35%

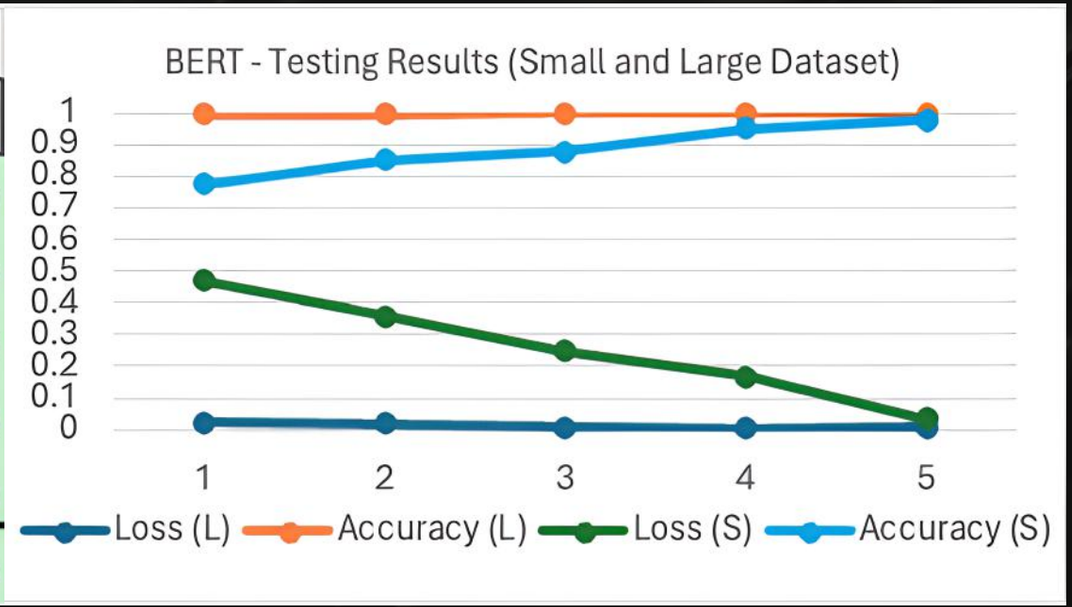


The BERT model's training results also show a steady increase in accuracy as well as decrease in loss.
The model that was trained on the large dataset showed extremely high accuracy however performed poorly in real life tests.

BERT Testing (Large vs. Small)

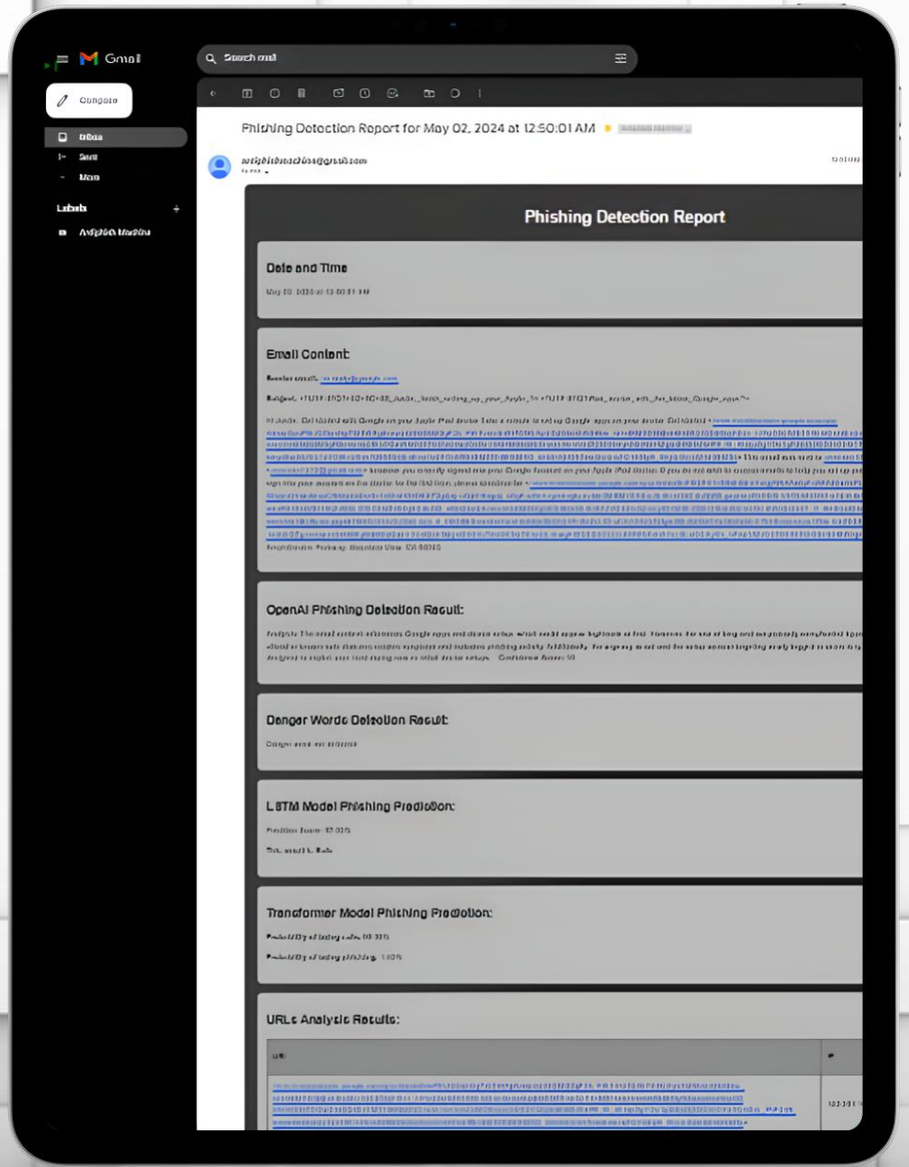
BERT Testing Results (Large and Small Datasets)

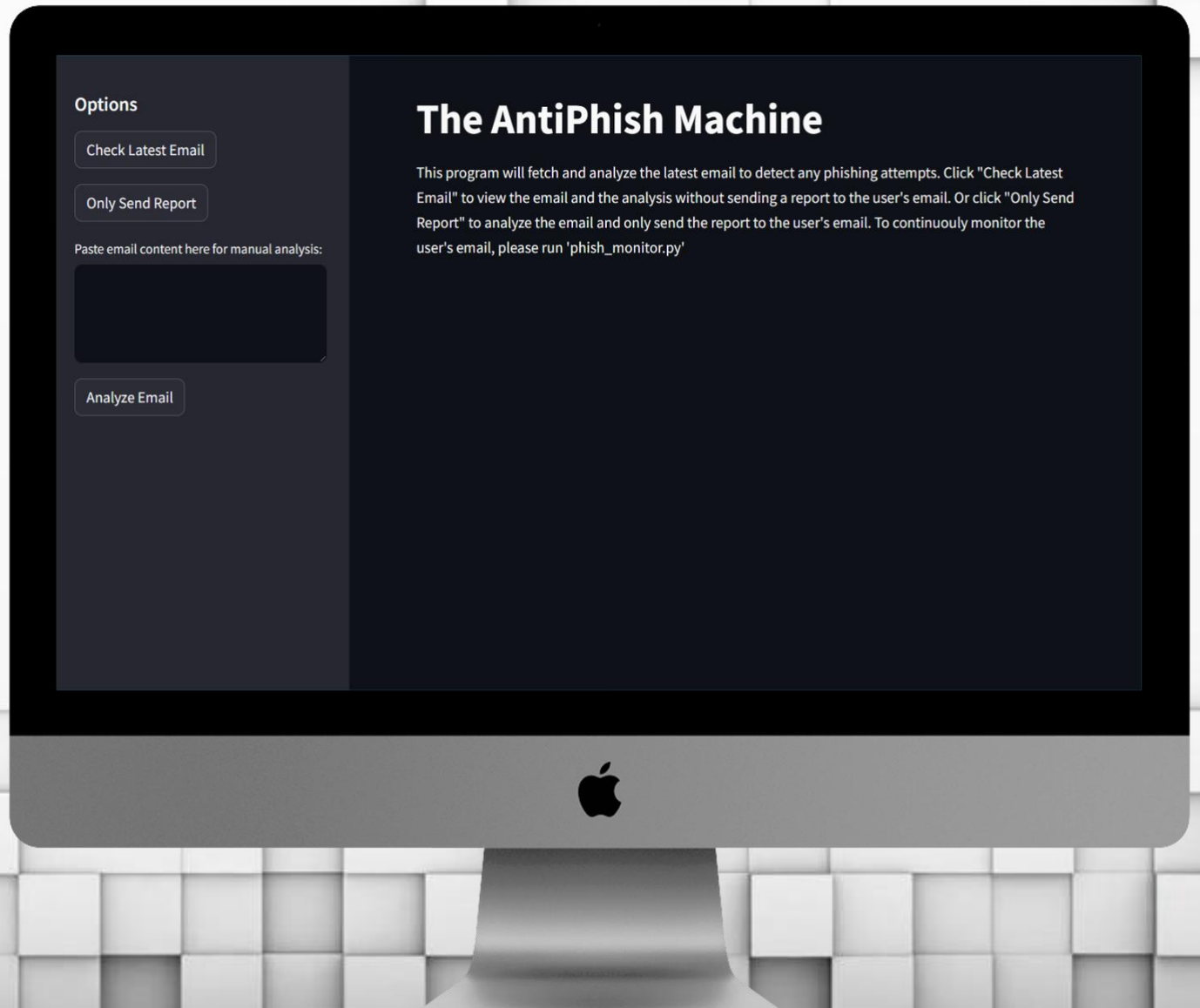
	Loss (L)	Accuracy (L)	Loss (S)	Accuracy (S)
1	0.0211	99.31%	0.4702	77.50%
2	0.017	99.45%	0.3593	85.00%
3	0.0055	99.75%	0.2469	87.50%
4	0.0044	99.89%	0.166	95.00%
5	0.0081	99.84%	0.0328	97.50%
avg.	0.01122	99.65%	0.25504	88.50%



The BERT model's testing results show that the loss is gradually dropping and the accuracy grows. This shows that the model is most likely not overfitting.
The model trained on the large dataset had almost perfect accuracy on the test set, while the model trained on the smaller dataset had an 88.5% accuracy.

Demonstration





QUESTIONS?

Email: justinjoy@mail.adelphi.edu

Challenges

What are some challenges faced during this study?

QUALITY DATASETS

The quality of the dataset was lacking in some areas. Although the data was cleaned before the model was learning it, it's impossible to see if every single data is relevant to the problem.

SUBTLE/VERY REALISTIC PHISHING EMAILS

Some emails may look way too realistic, and this is definitely a possibility. In cases like this, the URL analysis and email analysis would be the best tools to help determine if the email is a phishing attempt or not.

MODEL FINETUNING

The models needed to be finetuned quite heavily. In this case, by finetuning the model, I mean that the model itself needed some adjustments, like "Dropout" rate.

API (OPENAI GPT) RELIABILITY

Sometimes the OpenAI assistant does not give a response, which in this version of the development, causes the program to crash at times.