**Phishing Email Detection Project**

**User Manual**

Chien-Huan Kao, Ozgun Ozkan

1. **Introduction**

Welcome to the Phishing Email Detection User Manual. Our system provides real-time detection capabilities to identify phishing emails and promptly alerts managers. Additionally, it is equipped to efficiently handle large volumes of data. Please follow this manual's guidance to install our system step-by-step.

1. **System Requirements**

Hardware Requirements:

1. Minimum 4 Core CPU
2. Minimum16 GB memory
3. 10 GB of available disk space

Software Requirements:

1. Operating System: Windows 10, macOS, or Linux
2. Python 3.x
3. Hadoop 3.3.6
4. Spark 3.5.0

Recommend Using:

* AWS EMR

1. **Installation**
   1. Download the Phishing Email Detection software from our GitHub. <https://github.com/georgeKao6856/CSCI6444-phishing_detection_big_data_project>
   2. In AWS EMR, Create a cluster.
   3. According to the machine’s IP address, upload Makefile, phishing\_consumer.py, phishing\_detection\_model\_visual.py, phishing\_listener.py, phishing\_producer.py, and Phishing\_Email.csv to the master node.
   4. Connect to the master node and open a terminal.
   5. Run the command ‘make’ in the terminal, it will install every necessary dependencies and start the Kafka service.
2. **Getting Started**
3. Training the model:

Run the command ’spark-submit phishing\_detection\_model\_visual.py’

1. Start the phishing email listener:

Run the command ‘phishing\_listener.py’

1. Start the Kafka consumer:

Run the command ‘spark-submit phishing\_consumer.py’

1. Start the Kafka producer:

Run the command ‘python phishing\_producer.py’

1. **Output**

* testData.csv – The data in this file is for simulating the new email that comes into our system.
* phishing\_email.txt – This file contains all the phishing emails our system detected.
* confusion\_matrix.png – This diagram displays true positives, true negatives, false positives and false negatives.
* pie-chart-phishing.png – This diagram displays the percentage of phishing and non-phishing email in the dataset.
* word\_cloud\_dataset.png – This diagram displays the most frequent words found in all phishing emails in the dataset.
* word\_cloud\_model.png – This diagram displays the most frequent words found in all phishing emails that found by our system.
* cpu\_utilization.jpg – This diagram displays the CPU utilization when you train the model.