PROJECT EXHIBITION
INTELLIGENT PHISHING EMAIL DETECTION SYSTEM

Team members (group no. : 170)

Prateek Chaturvedi - 24BCE10168
Simar Kaur - 24BCE10107
Manya Raghuwanshi - 24BCE11428
Harsh Vardhan Singh - 24BCE11183
Aditya Chandra - 24BCE 11458

PROBLEM

PHISHING EMAILS ARE A MAJOR CYBERSECURITY THREAT, FOOLING USERS INTO CLICKING MALICIOUS LINKS OR GIVING AWAY PERSONAL DATA.

REAL-WORLD IMPACT

- USED FOR FRAUD
- IDENTITY THEFT
- CYBERATTACKS

GOALS

BUILD AN LLM MODEL THAT ACCURATELY DETECTS PHISHING EMAILS FROM LEGITIMATE ONES.

DATASET & APPROACH

01

DATASET

We used a public phishing email dataset in CSV format. Each record contains features such as:

- Email content
- Subject line
- Sender address
- URLs and attachments
- Label (Phishing/Legitimate)

02

DATA PROCESSING

- Removed HTML tags and URLs
- Tokenized and cleaned text (lowercasing, stopword removal)
- Tokenized text into subwords using a pre-trained tokenizer (e.g., from Hugging Face Transformers) compatible with our chosen LLM.

03

MODEL TRAINING

- Utilized pre-trained Transformer models (DistilBERT, BERT, RoBERTa, MiniLM) from Hugging Face.
- Fine-tuned these models on our phishing email dataset for the binary classification task (Phishing vs. Legitimate).
- Selected the model with the best balance of accuracy, precision, and computational efficiency for deployment.

04

TOOLS

- Python
- Hugging Face Transformers
- PyTorch / TensorFlow
- Streamlit
- Pandas
- nltk

LIBRARIES USED

1.DATA HANDILNG & PROCESSING: :

- PANDAS
- NUMPY

2.TEXT CLEANING & MACHINE LEARNING:

- RE
- STRING
- NLTK
- TRANSFORMERS (HUGGING FACE)
- TORCH / TENSORFLOW

3.USER INTERFACE:

- STREAMLIT
- OS
- PICKLE (FOR SAVING TOKENIZERS/MODELS)

1. DATA HANDLING & PROCESSING

PANDAS:

A POWERFUL LIBRARY FOR DATA ANALYSIS AND MANIPULATION USING DATAFRAMES, SUPPORTING OPERATIONS LIKE FILTERING, GROUPING, MERGING, AND CLEANING STRUCTURED DATA.

NUMPY:

A FUNDAMENTAL LIBRARY FOR NUMERICAL COMPUTING IN PYTHON, SUPPORTING FAST OPERATIONS ON ARRAYS, MATRICES, AND ADVANCED MATHEMATICAL FUNCTIONS.

_2. TEXT CLEANING & MACHINE LEARNING

TRANSFORMERS (HUGGING FACE):

A CORE LIBRARY PROVIDING THOUSANDS OF PRE-TRAINED MODELS (LIKE BERT, DISTILBERT) FOR NLP TASKS. WE USE IT TO LOAD MODELS AND TOKENIZERS, AND FOR FINE-TUNING AND INFERENCE.

PYTORCH / TENSORFLOW:

DEEP LEARNING FRAMEWORKS THAT PROVIDE THE COMPUTATIONAL BACKBONE FOR TRAINING AND RUNNING THE LARGE NEURAL NETWORK MODELS.

RE (REGULAR EXPRESSIONS):

A BUILT-IN PYTHON MODULE FOR MATCHING PATTERNS IN STRINGS; USED FOR INITIAL TEXT CLEANING (E.G., REMOVING HTML TAGS).

NLTK (NATURAL LANGUAGE TOOLKIT):

USED FOR FUNDAMENTAL NLP TASKS LIKE STOPWORD REMOVAL, WHICH CAN BE APPLIED DURING THE INITIAL DATA PREPROCESSING STAGE BEFORE TOKENIZATION.

.

3. USER INTERFACE

1. STREAMLIT

A PYTHON FRAMEWORK USED TO BUILD INTERACTIVE AND USER-FRIENDLY WEB APPS EASILY.
USED TO CREATE THE PHISHING DETECTION INTERFACE WHERE USERS CAN INPUT EMAILS AND SEE PREDICTIONS.

2. OS

A BUILT-IN PYTHON MODULE FOR INTERACTING WITH THE OPERATING SYSTEM.
USED TO NAVIGATE FOLDERS AND FILES DYNAMICALLY — FOR EXAMPLE, TO LOAD THE LATEST MODEL OR VECTORIZER FILE.

3. PICKLE

A BUILT-IN PYTHON MODULE TO SAVE AND LOAD PYTHON OBJECTS (LIKE TRAINED MODELS).
USED HERE TO LOAD TRAINED MACHINE LEARNING MODELS AND VECTORIZERS (PICKLE.LOAD()).

SYSTEM ARCHITECTURE

MODELS USED

1. DISTILBERT

WHY USED?

- A DISTILLED VERSION OF BERT THAT IS SMALLER, FASTER, AND CHEAPER TO RUN, WHILE RETAINING 95% OF BERT'S PERFORMANCE.
- IDEAL FOR A REAL-TIME WEB APPLICATION WHERE SPEED IS CRUCIAL.
- PERFECT AS OUR DEFAULT MODEL.

2. BERT (BASE)

WHY USED?

- BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS. UNDERSTANDS CONTEXT FROM BOTH LEFT AND RIGHT OF A WORD.
- PROVIDES HIGHLY ACCURATE CONTEXTUAL UNDERSTANDING OF EMAIL TEXT, CATCHING SUBTLE PHISHING CUES.
- USED FOR COMPARISON AND AS A HIGH-ACCURACY OPTION.

3. ROBERTA (BASE)

WHY USED?

- A ROBUSTLY OPTIMIZED BERT PRETRAINING APPROACH. AN IMPROVED VERSION OF BERT WITH OPTIMIZED TRAINING PROCEDURES.
- OFTEN ACHIEVES STATE-OF-THE-ART RESULTS ON MANY NLP BENCHMARKS. USED FOR MAXIMUM ACCURACY.

4. MINILM

WHY USED?

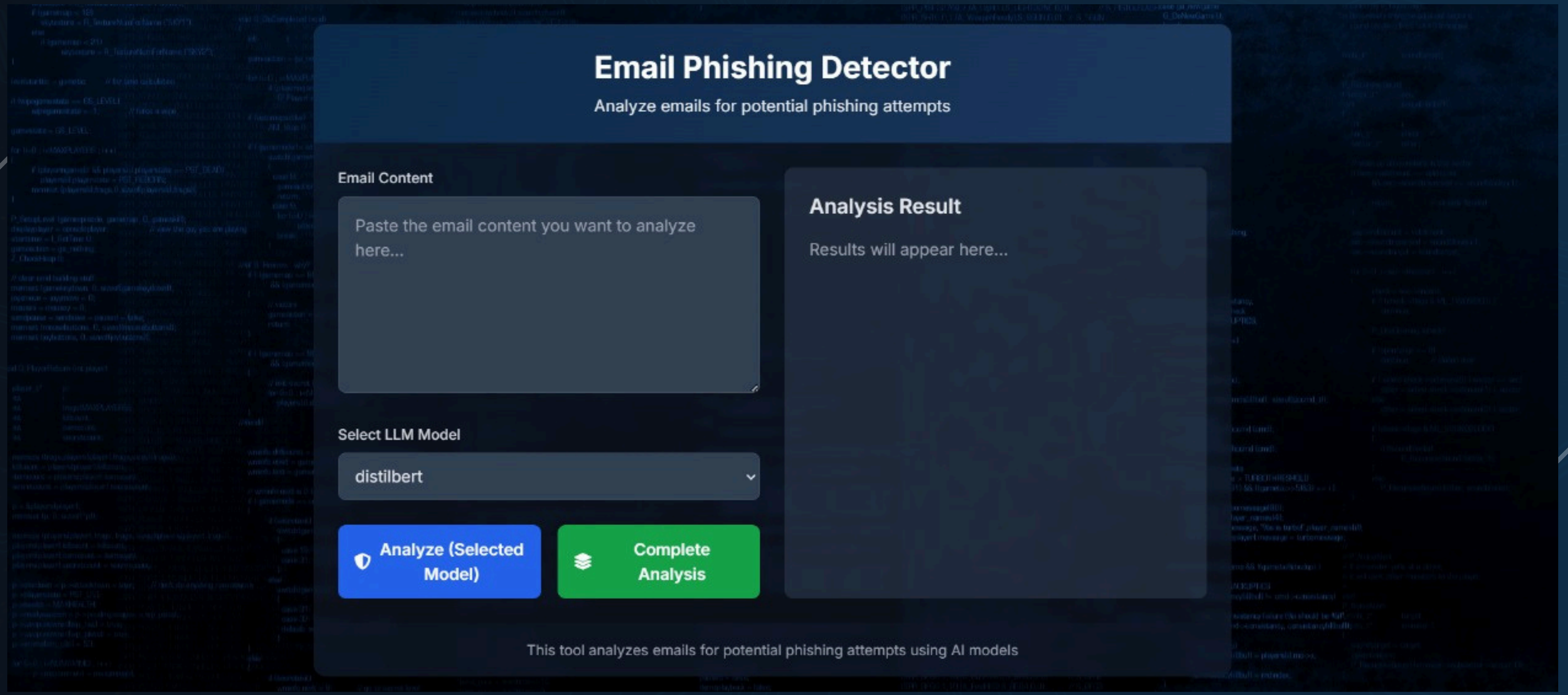
- A COMPRESSED MODEL THAT IS EXTREMELY SMALL AND FAST.
- DESIGNED FOR ENVIRONMENTS WITH LIMITED COMPUTATIONAL RESOURCES (E.G., MOBILE DEVICES OR EDGE COMPUTING).
- EXPLORED FOR ITS EFFICIENCY.

MODEL TRAINING & EVALUATION

HERE IS THE METRICS TABLE:

Model	Accuracy	Precision	Recall	F1 Score
distilbert	0.9756	0.9691	0.9823	0.9757
bert_base	0.968	0.9711	0.9653	0.9682
roberta_ba	0.9685	0.9768	0.9604	0.9685
minilm	0.9682	0.9698	0.9663	0.968

WEB APPLICATION INTERFACE



EXAMPLES OF PHISHING MAILS

✅ LEGITIMATE EMAIL EXAMPLE

SUBJECT: INVOICE FOR YOUR RECENT PURCHASE

BODY:

DEAR CUSTOMER,

THANK YOU FOR SHOPPING WITH US! PLEASE FIND THE ATTACHED INVOICE FOR YOUR RECENT ORDER #56789.

IF YOU HAVE ANY QUERIES, FEEL FREE TO CONTACT OUR SUPPORT TEAM AT SUPPORT@STORE.COM.

BEST REGARDS,

STORE TEAM

✅ PHISHING EMAIL EXAMPLE (HARD TO DETECT)

SUBJECT: URGENT: YOUR ACCOUNT WILL BE SUSPENDED

BODY:

DEAR USER,

WE DETECTED UNUSUAL ACTIVITY IN YOUR ACCOUNT. TO KEEP YOUR ACCOUNT ACTIVE, PLEASE VERIFY YOUR IDENTITY IMMEDIATELY BY CLICKING THE LINK BELOW:

VERIFY ACCOUNT

FAILURE TO ACT WITHIN 24 HOURS WILL RESULT IN ACCOUNT SUSPENSION.

THANK YOU FOR YOUR COOPERATION,

SECURITY TEAM

CONCLUSION & FUTURE SCOPE

CONCLUSION:

Our model successfully identifies phishing threats.

Helps prevent cyberattacks in real-time.

FUTURE SCOPE:

Deploy as a browser extension or API

Integrate with email clients like Gmail or Outlook

Add image or attachment-based detection

THANK YOU