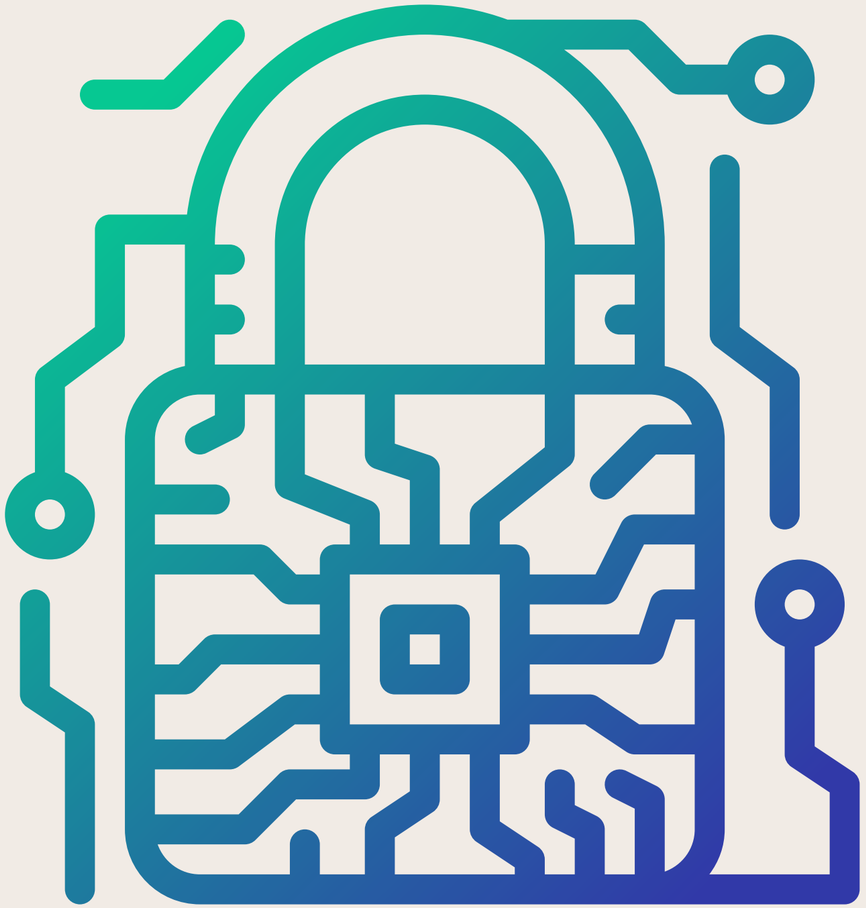
TRANSFORMER-BASED NLP MODEL FOR MALICIOUS URL DETECTION

PRESENTED BY:
NORA ALJOMUH
MOZOON ALKHALIS
RITAJ ALHAMLI

SUPERVISED BY: DR. MUSTAFA YOULDASH



INTRODUCTION

- Phishing and malicious URLs pose major cybersecurity threats.
- Attackers increasingly hide malicious intent inside complex or obfuscated URLs.
- Traditional ML models rely heavily on handcrafted lexical features.
- We apply transformer-based NLP to analyze URLs as text sequences.

PROBLEM STATEMENT

- How can we detect malicious URLs accurately, efficiently, and without manual feature engineering?

URLs contain:

- *Homoglyphs*
- *Hidden subdomains*
- *Random strings*
- *Embedded text*

Goal: Build a model that understands URL structure contextually, like language.

LITERATURE REVIEW

1. Tokenization Approaches

- URLTran (Maneriker et al.) explored tokenization strategies for URLs.
- DomURLs-BERT introduced special markers: [DOMAIN], [PATH], etc.
- Proper URL segmentation improves contextual representation.

2.Transformer Models

- Transformer models (BERT, DistilBERT, RoBERTa) excel at capturing contextual meaning. URLTran showed robustness
- SecureNet compared transformer models for phishing detection.

3.Embedded URL Detection

- PhishTransformer extracts all URLs embedded in webpage content.
- Detects phishing even when the main URL appears benign.
- Inspired hybrid systems that analyze multiple URL signals.

Research Gap

Many models rely on:

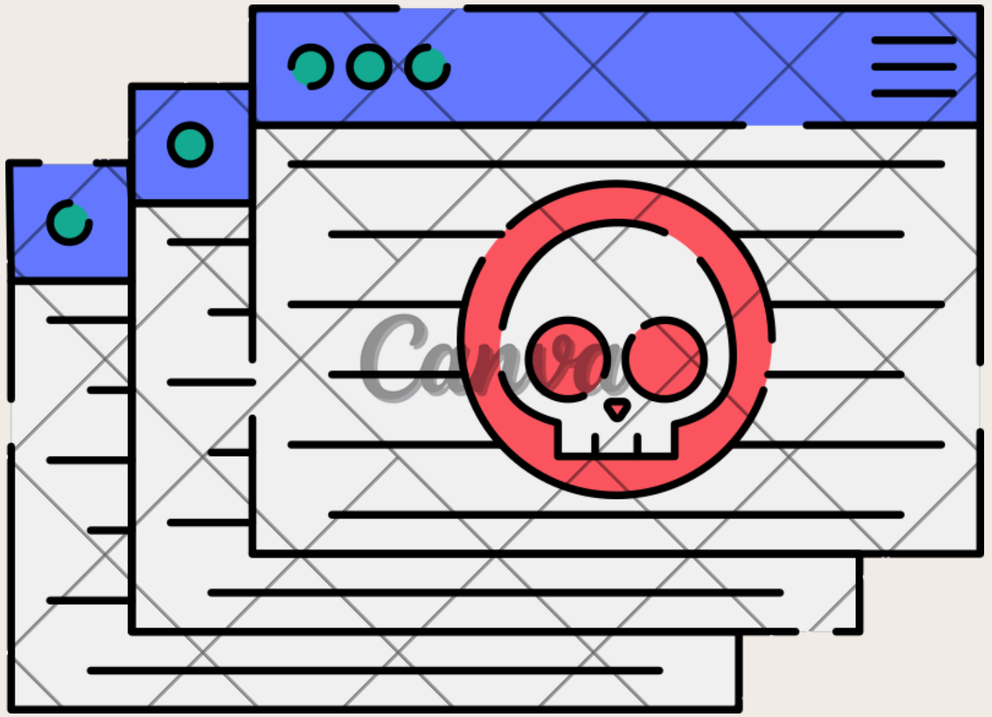
- Handcrafted features
- URL-only analysis
- Heavy transformer architectures
- ***Need a lightweight, fast, and context-aware NLP model.***

PROJECT OBJECTIVES

1. Develop a transformer-based system for URL classification.
2. Use DistilBERT for efficiency and speed.
3. Achieve high accuracy across four classes:

- Benign
- Phishing
- Malware
- Defacement

4. Produce a real-time prediction tool.



DATASET OVERVIEW

- **Dataset:** Malicious URLs Dataset (Kaggle)
- **4 classes:** Benign, Phishing, Malware, Defacement
- Includes varied URL patterns and obfuscation styles.

code snippet of loading the dataset

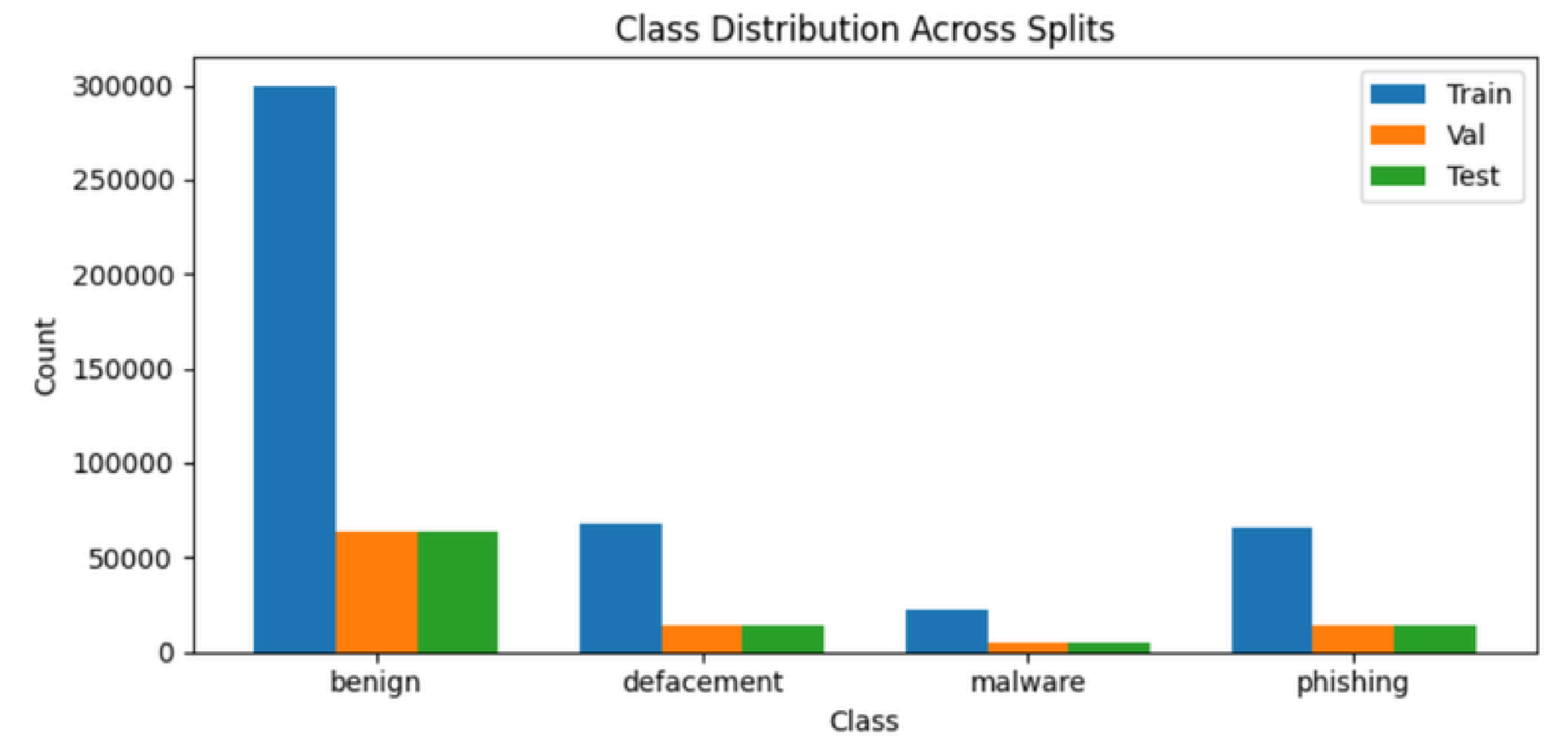
```
# Loading the dataset
import pandas as pd

df = pd.read_csv("malicious_phish.csv")
table = df['type'].value_counts().reset_index()
table.columns = ['Class', 'Count']
table
```

output:

Class	Count
Benign	428,103
Defacement	96,457
Phishing	94,111
Malware	32,52

VISUALIZATION



PREPROCESSING PIPELINE

1
Remove whitespace and invalid characters

2
Tokenize using DistilBERT tokenizer

3
Convert labels to integers

4
Train/validation/test split

5
No handcrafted features — fully automated NLP pipeline

MODEL ARCHITECTURE

Base model: DistilBERT

- 40% fewer parameters than BERT
- Faster inference (suitable for deployment)
- Classification head for 4 output classes

Trained with:

- AdamW optimizer
- Cross-entropy loss
- Learning rate scheduling

code snippet of loading the Base Model

```
from transformers import DistilBertForSequenceClassification

model = DistilBertForSequenceClassification.from_pretrained(
    "distilbert-base-uncased",
    num_labels=4
)
```

TRAINING SETUP

- **Batch size:** 16
- **Epochs:** 3
- **Learning rate:** 5e-5
- **Evaluation metric:** Accuracy + F1 Score
- **Tools:** PyTorch, HuggingFace Transformers

```
# Training arguments

from transformers import TrainingArguments

training_args = TrainingArguments(
    output_dir="results",
    eval_strategy="epoch",
    save_strategy="epoch",
    logging_steps=100,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    num_train_epochs=3,
    learning_rate=2e-5,
    weight_decay=0.01,
    report_to=[]          # disables wandb
)
```

```
# Trainer
from transformers import Trainer

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_ds,
    eval_dataset=val_ds
)
```

TRAINING SETUP

```
# Training

trainer.train()
```

[85470/85470 2:22:11, Epoch 3/3]

Epoch	Training Loss	Validation Loss
1	0.065300	0.050454
2	0.035800	0.051283
3	0.011100	0.059515

```
TrainOutput(global_step=85470, training_loss=0.044253682005225156, metrics={'train_runtime': 8531.4036, 'train_samples_per_second': 160.29,
'train_steps_per_second': 10.018, 'total_flos': 2.26444370333045 76e+16, 'train_loss': 0.044253682005225156, 'epoch': 3.0})
```

EVALUATION METRICS & RESULTS

- Accuracy
- Precision
- Recall
- F1-score

```
Accuracy: 0.9889

Classification Metrics Table:
      Class  Precision  Recall  F1-Score
   phishing     0.9641  0.9596    0.9619
     benign     0.9922  0.9944    0.9933
 defacement     0.9974  0.9991    0.9982
    malware     0.9908  0.9705    0.9805
  Macro Avg     0.9861  0.9809    0.9835
```

PREDICTION TOOL DEMO

- User inputs a URL → model predicts class
- Fast inference due to DistilBERT's lightweight architecture

```
Type a URL to classify it.
Type 'exit' to stop.

Enter URL: www.syfy.com/rewind/?p=50206
Prediction: phishing

Enter URL: account-verification-chase.com/login/secure
Prediction: benign

Enter URL: http://www.nptw103.com.tw/news.php?news_id=6
Prediction: defacement

Enter URL: soft-downloads247.com/windows/update/install.exe
Prediction: malware

Enter URL: exit
Exiting.
```

THANK YOU VERY MUCH!