National College of Ireland

Phishing Url Detection Using Distlbert And Capsule Neural Networks

MSc Research Project
MSc Data Analytics

Saketh Reddy Atla
Student ID: x22218700

School of Computing
National College of Ireland

Supervisor: Teerath Kumar Menghwar

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	SAKETH REDDY ATLA
Student ID:	X22218700
Programme:	MCSDAD_B **Year:** 2023-2024
Module:	MSC RESEARCH PROJECT
Supervisor:	TEERATH KUMAR MENGHWAR
Submission Due Date:	16/09/2024
Project Title:	PHISHING URL DETECTION USING DISTLBERT AND CAPSUEL NEURAL NETWORKS
Word Count:	**6,700 Page Count: 20**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: SAKETH REDDY ATLA

Date: 16/09/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Phishing Url Detection Using Distlbert And Capsuel Neural Networks

Saketh Reddy Atla
X22218700

Abstract

This paper presents a new approach to detecting phishing URLs (Uniform Resource Locator) by combining DistilBERT and Capsule Neural Networks. The paper presents a continuous solution for a problem in an ever-altering cyber threat environment. In this study, three models, namely, DistilBERT, Capsule Neural Network, and a new Hybrid of the two, are presented and evaluated using 277,645 real-world 277,645 URLs. The DistilBERT model showed accuracy, precision, recall, and F1-score of 98%, demonstrating its great potential in real-time phishing detection. Quite surprisingly, the Capsule Network showed very poor accuracy of only 50% in these tasks, hence showing large challenges when it comes to the adaptation of this architecture for URL classification. While drastic improvements were noted over DistilBERT, Hybrid showed marginal improvements, specifically for recall alone (98.92% vs. 98%), hence proving it to show potential further in sophisticated manners of phishing detection.

The paper contributes to the literature in general on the efficacy of transformer-based models for URL classification and raises some issues connected with the application of Capsule Networks in security-related, plain-text tasks. The results have huge implications for the development of more sustainable, real-time phishing detection systems and provide a targeting ground for future studies in adaptive learning frameworks, explainable AI, and multimodal approaches toward phishing detection.

1 Introduction

Phishing attacks are one of the most tenacious and versatile threats to cybersecurity today, with criminals constantly perfecting their techniques that can quickly deceive users into revealing sensitive information (Cybersecurity Ventures, 2023). According to recent statistics, there has been a 65% rise in phishing attempts in just the past year, with financial losses and data breaches amounting to millions across different sectors. The dynamic nature of such attacks poses severe problems for traditional methodologies of detection, thus demanding advanced and adaptive solutions.

Traditional phishing URL(Uniform Resource Locator) detection techniques using blacklisting and simple heuristics usually cannot keep pace with the rapidly changing nature of a particular attack vector. Blacklists work very well against known threats but are unable to recognize sites created for phishing purposes that have just been created (Sahingoz et al., 2019). Heuristics-based methods could be evaded by advanced attacks in a form that may look like the pattern of legal URLs. (Abdelnabi et al., 2020) These limitations, therefore, bring out the need for more robust, intelligent detection systems that are able to adapt to new threat patterns.

Machine learning techniques have been quite promising in these regards since it provides the ability for learning and adoption of new patterns. However, most machine learning approaches still suffer today from significant drawbacks in processing these complex hierarchical structures of URLs, capturing nuanced semantic content indicative of malicious intent,(Sharma et al., 2020).

Another promising way to improve the capabilities in phishing detection would be advanced NLP models with integrated novel neural network architectures. Specifically, this paper proposes a combination of DistilBERT and capsule neural networks as one of the plausible ways to address these challenges.

DistilBERT is computationally efficient yet powerful in its ability to perform natural language processing, since it is a BERT model (Bidirectional Encoder Representations from Transformers) in a distilled form. The pre-trained model was trained on large text data, making the model very good at picking up fine-grained linguistic patterns indicative of phishing. Capsule neural networks establish a new way of capturing spatial hierarchies that hold between features important for understanding the complex structure of URLs as introduced by Hinton et al.

Although separately, DistilBERT and Capsule Neural Network are amazingly effective; their applicability combined is relatively unexplored in phishing URL detection. In this regard, the research is set to establish the efficacy of their technologies integrated in real-time phishing detection.

The primary research question guiding this study is: "How does the integration of DistilBERT and Capsule Neural Networks enhance the real-time detection accuracy of phishing URLs compared to traditional machine learning methods?"

Address this question, the following research objectives have been established:

1. Design and evaluate the performance of individual models of DistilBERT and capsule network in the task of URL-based phishing detection.
2. Design a hybrid model combining technologies of DistilBERT and capsule networks.
3. The performance of the above-mentioned models is compared to traditional machine learning models based on random forests and support vector machines.
4. Verification of the accuracy and computational efficiency of real-time detection using the proposed models.
5. Use the model to estimate its generalization performance on new and previously unseen patterns of phishing.

Exhaustive data collection and feature extraction from URLs have used a dataset that is obtained from reputable sources. It extracts an extensive set of 38 features from each URL, including structural features like URL and domain name lengths, character-based features like the count of special characters, lexical features like the presence of suspicious words, domain-specific attributes like whether the URL uses any IP address, and security features like whether the SSL/TLS (Secure Sockets Layer/Transport Layer Security) certificate is valid or not. SSL/TLS are cryptographic protocols designed to provide secure communication over a computer network. The process comes in handy with parallel processing for efficient verification of SSL/TLS and results in a rich, multi-dimensional representation of every URL. The prepared dataset, which contains each original URL, its classification label, and the extracted features, is then split into three ways equally for training different models: DistilBERT, CapsuleNetwork, and a Hybrid model. Such meticulous preparation of data can

act as the basis for the subsequent phases of model training and evaluation and will enable the thorough investigation of the proposed machine learning approaches in phishing URL detection.

In the current study, individual DistilBERT and Capsule Network models will be implemented and trained, with a new hybrid architecture that includes elements from both methods. The DistilBERT model is to be fine-tuned on the URL dataset to grasp semantic patterns indicative of phishing attempts. On the other hand, the Capsule Network will be designed with regard to the hierarchical structure of URLs and could help capture relationships between different URL components relevant for detection.
Performance evaluation will be done with respect to a number of standard metrics: accuracy, precision, recall, and the F1-score. Much attention will be paid to real-time detection capabilities, measuring accuracy and computational efficiency. Models for generalization will be evaluated using a held-out dataset, and their performance is compared against baseline models such as Random Forests and Support Vector Machines.
This work contributes to the academic community and provides practical application in cybersecurity. It explores the constructive collaboration of DistilBERT with Capsule Neural Networks to stretch the current approaches in phishing detection, offering potential for a more rigid security stance against evolving cyber threats. The results may play a key role in the development of more effective security measures on online platforms and across the market.
The following parts of the report will be devoted to a comprehensive review of related work, detailing the methodology adopted for the research, the description of design and implementation of the models proposed, and giving thorough evaluations of their performances. The conclusion will synthesize the findings, indicate implications, and suggest future avenues of research in this critical area of cybersecurity.

2 Related Work

This literature review has covered recent developments related to phishing detection techniques, with an emphasis on machine learning and natural language processing in cybersecurity. The various approaches can be critically analyzed in terms of their potentials and insights into the fads of their implementation and evaluation.

2.1 Evolution of Transformer Models in Phishing Detection

Recent years have witnessed a radical drift towards the use of transformer-based models in phishing detection. Gogoi and Ahmed (2022) demonstrated that deep-learning transformer models could be especially useful in detecting fraudulent emails with ultra-high accuracy. Their work showed the power of natural language processing in noticing fine-grained linguistic features that typically characterize phishing messages. Indeed, the performance metrics realized during this study were actually very impressive: 0.99 for accuracy, recall, and F1 scores. However, it had so far been limited to email phishing, and thus there was a need to explore other avenues of communication.
Based on this foundation, Rifat Ahsan Chowdhury, and Gomes(2022) presented the application of BERT models for real-time SMS(Short Message Service) phishing detection. The contribution of the work was on how flexible the transformer model evaluated could be on different digital communication media. The study showed good precision without addressing issues of computation, which would turn out to be important for real-world deployment.

pushed this further by comparing various transformer architectures, specifically BERT, RoBERTa, and DistilBERT, for the task of classifying phishing emails based on embedded persuasion tactics Karki et al. (2022). Their results favoured RoBERTa because of its better training efficiency and model depth. That said, due to the focus of their study on this specific subset of phishing techniques, generalizability to the larger spectrum of phishing attacks is reduced.

Proposed a new method incorporating a tiny-BERT-based auto-feature extraction from phishing websites and a stacking-based classifier algorithm in He et al.(2023), which achieved high precision and recall furthering the application of transformer models in web security. Hopefully, this method will scale very well when automated phishing detection is applied, though its performance in managing the highly dynamic nature and rapid evolution in phishing techniques is yet to be fully explored. Highly dynamic and rapidly evolving phishing techniques remains to be fully explored.

2.2 Advancements in Real-Time and Adaptive Detection Systems

The problem of real-time phishing detection, mostly in resource-constrained scenarios, has been the focus of recent research. Joseph and Jacob(2022) contributed some important progress in this respect by proposing a real-time SMS(Short Message Service) phishing detection model with a BERT-based model optimized for edge devices that have fairly low computational resources. This work is extremely useful in this regard since it addresses practical challenges in applying sophisticated NLP(Natural Language Processing) models to real-world scenarios. This narrow scope to SMS phishing alone might not allow the complexities of other phishing vectors to be fully accounted .

Have used DistilBERT for the analysis of HTTP(Hypertext Transfer Protocol) and HTTPS(Hypertext Transfer Protocol Secure) network request content in micro application servers Nige et al. (2023). Their approach stressed semantic analysis in network security, and it is strong in handling real-time data that constitutes an attack. Although it shows a lot of promise to provide cyber security with its ingenuity, the narrow scope of focus on the micro-application server reduces its scope further in most contexts relating to cyber security.

Proposed a hybrid approach combining NLP with machine learning techniques for phishing detection in IoT devices Jaafar et al. (2023). Their log embedding model, based on DistilBERT, showed how the hybrid models adapt within the heterogenous and diversified data environments that characterize IoT(Internet of Things). This research is relevant incidentally when the cybercriminal fraternity increasingly targets IoT devices. The authors, however, did not address how the model would stay effective with the rapidly changing nature of IoT technologies and threats.

2.3 Performance Evaluation in Constrained Environments

Design challenges are critically important in real-life deployment and evaluation settings for phishing detection. Do et al. (2023), meanwhile, made solid contributions to the study of the use of a BERT-Medium variant for unsupervised detection of phishing URLs. Their approach has been so far at the frontier of behaviourally empowered cybersecurity, which has proven to be quite effective without the need for supervised training, thus challenging dependence on labeled datasets. This work really opened up new opportunities for the deployment of NLP in

such places where there is no possibility of finding extensive labeled data. However, this study does not really go into the variations of the model's performance with different computational constraints.

Conducted a comparative study of CNN(Convolutional Neural Network) models using Glove embeddings versus BERT with fine-tuning for email phishing detection Giri et al. (2022). Their findings were remarkably interesting: while BERT added a number of important capabilities, Glove embeddings did better in some scenarios. The moral of the story from this research is to choose models based on deployment environment characteristics. However, having focused on email phishing, it may not fully represent all modes of phishing across different platforms..

2.4 Emerging Trends and Research Gaps

Through the literature review, significant advancements have been demonstrated in the application of NLP and deep learning techniques for phishing detection. More specifically, transformer models, including BERT and its derivatives, have demonstrated applicability of a class that can undergo learning about the scenarios where phishing is recognized across the whole spectrum of the digital channel. Application of real-time detection systems and adaptive approaches bears the prospect of overcoming the dynamic nature of phishing threats. There are several limitations and research gaps, though, from the state of the art. Most studies have focused on the specific communication channels of phishing and may, therefore, be less generalized within the wider scope of the phishing landscape. Note also that deploying the models in resource-constrained settings remains a huge challenge, even though some researchers are slowly addressing it, to support widespread usage.

Furthermore, there is a general lack of integrated approaches that, by effectively consolidating the strengths of various approaches, make headway against the multifaceted nature of phishing attacks. The constantly changing and dynamic characteristic of phishing techniques requires more adaptive and integrated approaches that continuously learn and hence can update their strategies in real-time.

This literature review is an indication that there will be a need for such a comprehensive approach to be able to detect phish. A potential combination of the transformer model's own semantic understanding capabilities, such as DistilBERT, for the semantic domain, Capsule Neural Networks for recognizing relations in the spatial domain, and an adaptive feedback mechanism, may thus bring the full potential to bear in bridging a lot of gaps in current research. This is an approach that would not only harness sophisticated NLP techniques, but it would also have the ability to respond to the ever-changing nature of phishing threats in real time, thus providing a more robust and all-inclusive solution to this perennial cyber security threat.

3 Research Methodology

The section provides the overall details for the development and evaluation of the phishing URL detection framework by DistilBERT, Capsule Neural Networks, and a Hybrid model. The methodology is divided into five broad stages: data acquisition, preprocessing, feature extraction, development of the model, training, and evaluation.

3.1 Data Acquisition and Preprocessing

3.1.1 Data Sources

An initial dataset drawn from a very reputable source contained only URLs and their corresponding labels as phishing or legitimate. This then provided the raw dataset that would be used as the base for further analysis and model development.

3.1.2 Data

Cleaning and Normalization The data set was cleaned thoroughly to ensure quality and consistency in the data. This involved the following activities Kaitholikkal and B (2024):

- Removal of duplicate entries using pandas' functionality
- Handling missing data either by imputation or removal of rows that have missing data
- Normalize URL formats using regular expressions for consistency within the dataset

3.1.2.1 Feature Extraction

Extensive feature extraction was designed to understand the most varied facets that could suggest phishing through URLs. This step was important in reconstituting the raw URL data into rich features for the models. Feature extraction made use of a number of Python libraries: urllib for parsing URLs, re for regular expressions, numpy for numerical operations, and pandas for data manipulation.

This phase of feature extraction thus involved the development of a function that takes in a URL and outputs a dictionary containing several features.

This function used the URL parse method to break down the URL into scheme, netloc, path, params, query, and fragment. After extraction of these features, it extracted:

1. Counts of special characters (22 different characters were counted)
2. Length-based features (TLD (Top Level Domain)length, URL length, domain length, etc.)
3. Domain-specific features (count of dots, hyphens, underscores, vowels in the domain)
4. Boolean features (whether the domain is an IP, contains specific words, uses HTTPS, etc.)
5. Structural features (number of subdomains, parameters, etc.)
6. Entropy of the URL string

The implemented a separate function for the Shannon entropy calculation. This would be a measure of randomness in the URL string and might indicate it to be a generated phishing URL. During the feature extraction process, this dataset was increased from URLs and labels alone to a comprehensive set of 38 features for every URL, providing quite a rich set of inputs to the machine learning models.

3.1.2.2 Data Splitting and Preparation

Support the multi-model approach, the preprocessed dataset had to be sliced into three parts: one for each model DistilBERT, Capsule Network, and the Hybrid model. The splitting process was completed using a custom Python function that divides a DataFrame into three equal parts while ensuring almost an equal number of phishing and legitimate URLs in all three datasets.

3.1.3 Dataset Statistics

After splitting, the datasets had the following characteristics:

- DistilBERT Dataset: Total samples: 92,215 Positive (Phishing) samples: 51,597 Negative (Legitimate) samples: 40,618

- Capsule Network Dataset: Total samples: 92,215 Positive (Phishing) samples: 51,512 Negative (Legitimate) samples: 40,703
- Hybrid Model Dataset: Total samples: 92,215 Positive (Phishing) samples: 51,741 Negative (Legitimate) samples: 40,474

Each dataset maintained thirty-eight features per URL, providing a rich input for the models.

3.1.4 2.2 Train-Test Split

For all models, the dataset was further divided into a training set and a test set in a ratio of 80:20. This is implemented using scikit-learn's train_test_split function with a fixed random state to ensure reproducibility.

3.2 Model Architecture and Training

3.2.1 DistilBERT Model

It used the DistilBERT model, which is a distilled version of BERT, for efficient natural language processing. The model initialized using the pre-trained weights from 'distilbert-base-uncased' and fine-tuned on the URL dataset. The URLs were preprocessed using the tokenizer of DistilBERT before being fed into the model.

3.2.2 Capsule Neural Network Model

A custom Capsule Neural Network architecture was used that is proficient at capturing spatial relationships within the URL structure. It was designed with a primary capsule layer followed by a routing layer. The primary capsule layer was based on convolutional operations to extract features. These were then routed to the final capsule layer using a dynamic routing algorithm.

3.2.3 Hybrid DistilBERT-Capsule Network Model

The hybrid model combined the advantages of both DistilBERT and capsule networks. The base for it was DistilBERT, which did the first feature extraction; then, there was a layer for final classification by a Capsule Network. This model could utilize the features of the DistilBERT method of understanding context information in the URL and ability of a Capsule Network to capture spatial relationship. Each model was implemented in PyTorch. The training of these models involved:

- Batch sizes of 32 (for DistilBERT and Hybrid models) or 64 (for Capsule Network)
- Adam optimizer with learning rates of 2e-5 (DistilBERT and Hybrid) or 0.001 (Capsule Network)
- Cross-entropy loss function
- Training for 5 epochs (DistilBERT and Hybrid) or 25 epochs (Capsule Network)

3.3 Evaluation Methodology

To comprehensively assess the models' performance, a range of evaluation metrics was employed:

3.3.1 Performance Metrics

- Accuracy: Overall correctness of the model
- Precision: Proportion of true positive predictions
- Recall: Proportion of actual positives correctly identified
- F1-Score: Harmonic mean of precision and recall

- ROC (Receiver Operating Characteristic) Curve and AUC (Area Under the Curve) Receiver Operating Characteristic curves were plotted, and the Area Under the Curve was computed to assess models in their class discriminative power. This was done using scikit-learn's roc_curve and auc functions.

- Confusion Matrix Confusion matrix was created to visualize models about the true positives, true negatives, false positives, and false negatives. This was implemented with scikit-learn's confusion_matrix function and visualized using seaborn's heatmap.

3.4 Hardware and Software Specifications

All the experiments were conducted using Google Colab, which had A100 GPU(Graphics Processing Unit) acceleration available. The main libraries used are PyTorch for model implementation, transformers for DistilBERT, scikit-learn for the evaluation metrics and splitting of data, pandas for data manipulation, and numpy for numerical operations.

4 Design Specification

This section details the architectural designs and specifications for all three models used in this phishing URL detection system: the DistilBERT, Capsule Neural Network, and Hybrid models of DistilBERT-Capsule Network.

4.1 DistilBERT Model Architecture

The DistilBERT model is one that has been distilled from BERT to ensure high performance in NLP tasks while remaining efficient.

Key Components:

1. Tokenizer:
    - Utilizes the DistilBertTokenizer from the transformer’s library
    - Vocabulary size: 30,522 tokens
    - Special tokens: [CLS], [SEP], [PAD], [UNK]
    - Maximum sequence length: 128 tokens
2. Embedding Layer:
    - Input embedding dimension: 768
    - Positional embeddings added to token embeddings
3. Transformer Layers:
    - Six transformer layers (compared to BERT's 12)
    - Each layer consists of a. multi-head self-attention mechanism:
        - Twelve attention heads
        - Attention head size: 64 b. Feed-forward neural network:
        - Hidden dimension: 3072
        - Activation function: GELU (Gaussian Error Linear Unit)
    - Layer normalization applied after each sub-layer
4. Pooler:
    - Takes the hidden state of the [CLS] token
    - Dense layer with tanh activation
5. Classification Layer:
    - Linear layer: 768 -> 2 (for binary classification)
    - Softmax activation for probability output

Functionality:

- Input: Tokenized URL (sequence of integers)
- Process:
    1. Embed tokens and add positional encodings
    2. Process through six transformer layers
    3. Extract [CLS] token representation
    4. Apply classification layer
- Output: Two-dimensional tensor with probabilities for phishing and legitimate classes

4.2 Capsule Neural Network Architecture

The Capsule Neural Network has been specifically designed to catch spatial hierarchies between features in the URLs, hence providing a different way of feature extraction and classification. Key Components:

1. Embedding Layer:
    - Embedding dimension: 100
    - Vocabulary size: Determined by unique characters in URLs
2. Primary Capsule Layer:
    - Convolutional layer:
        - Input channels: 100 (embedding dimension)
        - Output channels: 256 (32 capsules * eight dimensions per capsule)
        - Kernel size: nine
        - Stride: two
    - Reshape operation to form thirty-two primary capsules of 8 dimensions each
3. DigitCaps Layer (Routing Layer):
    - Input: 32 * ((max_seq_length - 9) // 2 + 1) primary capsules
    - Output: 2-digit capsules (for binary classification)
    - Capsule dimension: 16
    - Routing iterations: 3
4. Length Layer:
    - Computes Euclidean norm of each digit capsule

Functionality:

- Input: URL represented as a sequence of character indices
- Process:
    1. Embed input sequence
    2. Apply convolutional layer to extract primary capsules
    3. Dynamic routing between primary capsules and digit capsules
    4. Compute lengths of digit capsules for classification
- Output: Two-dimensional tensor representing the lengths of the two-digit capsules

4.3 Hybrid DistilBERT-Capsule Network Architecture

This is a new architecture where the contextual understanding of DistilBERT is infused with the capability of a Capsule Network in capturing spatial hierarchies.

Key Components:

1. DistilBERT Base:
    - Pre-trained DistilBERT model (as described in 4.1)
    - Outputs 768-dimensional embeddings for each token

2. Dropout Layer:
    - Dropout rate: 0.1
    - Applied after DistilBERT for regularization
3. Simple Capsule Layer:
    - Input dimension: 768 (DistilBERT output)
    - Number of capsules: 10
    - Capsule dimension: 16
4. Classification Layer:
    - Linear layer: 160 (10 capsules * 16 dimensions) -> 2
    - Softmax activation for probability output

Functionality:

- Input: Tokenized URL
- Process:
    1. Extract contextual embeddings using DistilBERT
    2. Apply dropout
    3. Transform DistilBERT output into capsules
    4. Apply squash activation to capsules
    5. Flatten capsules and apply final classification layer
- Output: Two-dimensional tensor with probabilities for phishing and legitimate classes

4.4 Model Requirements and Specifications

1. Input Processing:
    - URLs are preprocessed and tokenized using DistilBertTokenizer for DistilBERT and Hybrid models
    - For Capsule Network, URLs are converted to character indices
    - Maximum sequence length: 128 tokens/characters
    - Special tokens ([CLS], [SEP]) are added for DistilBERT and Hybrid models
2. Computational Requirements:
    - GPU(Graphics Processing Unit): NVIDIA A100 with 40GB memory
    - CPU(Central Processing Unit): Minimum 4 cores
    - RAM(Random Access Memory): Minimum 32GB for training, 16GB for inference
3. Training Specifications:
    - Batch sizes:
        - DistilBERT and Hybrid: 32
        - Capsule Network: 64
    - Optimizers:
        - DistilBERT and Hybrid: AdamW with learning rate 2e-5
        - Capsule Network: Adam with learning rate 0.001
    - Loss function: Cross-entropy loss
    - Training epochs:
        - DistilBERT and Hybrid: 5
        - Capsule Network: 25
    - Gradient clipping: Applied to prevent exploding gradients
4. Model Sizes:

- DistilBERT: Approximately 66 million parameters
- Capsule Network: Approximately 500,000 parameters (varies based on vocabulary size)
- Hybrid Model: Approximately 67 million parameters

5. Evaluation Metrics:
   - Accuracy, Precision, Recall, F1-Score: Calculated using sklearn. Metrics
   - ROC curve and AUC: Implemented using sklearn.metrics.roc_curve and roc_auc_score
   - Confusion Matrix: Generated using sklearn.metrics.confusion_matrix
6. Deployment Requirements:
   - Python 3.8+
   - PyTorch 1.9+
   - Transformers 4.5+
   - Scikit-learn 0.24+
   - CUDA 11.0+ for GPU acceleration
   - Minimum 100GB SSD for dataset and model checkpoints
7. Adaptive Feedback Mechanism:
   - Implements a sliding window approach for continuous learning
   - New data is accumulated in a buffer
   - When buffer reaches a threshold (e.g., 1000 samples), model is fine-tuned
   - Learning rate decay applied during fine-tuning to prevent catastrophic forgetting
8. Data Flow:
   - Raw URLs -> Feature Extraction -> Model Input Preparation -> Model Forward Pass -> post-processing -> Classification Output

5 Implementation

5.1 Data Preparation and Feature Extraction

The implementation began with a thorough preparation of data and extraction of features:

- python script has been customized to extract 38 features from every URL in the dataset using urllib, re and numpy libraries.
- Some of these features included counts of special characters, length-based features, domain-specific attributes, and other structural characteristics of URLs.
- The Extracted features were then stored in a panda DataFrame. This ensured an organized format for model training and evaluation.

5.1.1 Tokenization Process

Tokenization will turn into one of the most valuable steps while preparing text data for any machine learning model in NLP. For our phishing URL detection system, below is an implementation of tokenization:

5.1.2 DistilBERT Tokenization

- The DistilBertTokenizer from the transformer's library was used for the DistilBERT and Hybrid models.
- This tokenizer splits URLs into subwords, allowing for better handling of out-of-vocabulary terms.
- Special tokens [CLS] and [SEP] were added at the start and end of each URL, respectively.
- URLs were padded or truncated to a maximum length of 128 tokens.
- The tokenizer output includes:
    - input_ids: Numerical representations of tokens
    - attention mask: Binary mask indicating valid tokens vs. padding

5.1.3 Character-Level Tokenization for Capsule Network

- For the Capsule Network, a character-level tokenization approach was implemented.
- Each character in the URL was mapped to a unique integer index.
- A vocabulary was built from all unique characters in the dataset.
- URLs were padded with a special padding character to ensure uniform length.

5.2 Model Implementations

5.2.1 DistilBERT Model

- Implemented using the transformers library from Hugging Face.
- A custom PyTorch dataset class handled URL tokenization and input tensor preparation.
- The architecture of the model is composed of a pre-trained, base DistilBERT followed by a classification layer.
- Fine-tuning used PyTorch's training loop with gradient clipping and early stopping.

5.2.2 Capsule Neural Network Model

- Custom implementation from scratch using PyTorch.
- Included a primary capsule layer using convolutional operations and a routing layer with dynamic routing.
- A custom squash function ensured proper scaling of capsule outputs.
- Designed to process extracted URL features directly into capsule representations.

5.2.3 Hybrid DistilBERT-Capsule Network Model

- Combined elements from both DistilBERT and Capsule Network implementations.
- DistilBERT served as the initial feature extractor, feeding into a simplified Capsule Network.
- Custom PyTorch modules integrated the two architectures seamlessly.
- Allowed for end-to-end training of the entire hybrid model.

5.3 Training and Evaluation Pipeline

- Unified pipeline developed using PyTorch's training utilities.

- Included functions for data loading, model training, validation, and testing.
- Evaluation metrics (accuracy, precision, recall, F1-score) computed using scikit-learn.
- Visualization scripts using matplotlib and seaborn for ROC curves and confusion matrices.

5.4 Tools and Languages

The implementation primarily used:

- Python 3.8 as the main programming language
- PyTorch 1.9 for model implementation and training
- Transformers 4.5 for the DistilBERT model
- Pandas and NumPy for data manipulation
- Scikit-learn for evaluation metrics and data splitting
- Matplotlib and Seaborn for data visualization
- CUDA 11.0 for GPU acceleration

5.5 Output Artifacts

The final outputs of the implementation phase included:

- Three trained models (DistilBERT, Capsule Network, and Hybrid) saved as PyTorch state dictionaries
- A feature extraction pipeline for processing raw URLs into model-ready inputs
- Evaluation scripts producing performance metrics and visualizations

6 Evaluation

6.1 Model Performance Analysis

In this work, three different phishing URL detection models are assessed: one using only DistilBERT, another using Capsule Network, and the last using a Hybrid of DistilBERT-Capsule Network. All models used for training and testing were based on a balanced dataset of phishing and legitimate URLs; hence, this provides a high-level comparison in terms of efficiency performance.

6.1.1 DistilBERT Model

The DistilBERT model exhibited exceptional performance across all metrics:

- Accuracy: 0.98
- Precision: 0.98
- Recall: 0.98
- F1-score: 0.98

Confusion Matrix:

- True Positives: 10,125
- True Negatives: 7,932
- False Positives: 200
- False Negatives: 186

High accuracy across all metrics in this model, together with the fact that its architecture represents a distilled portion of the full BERT model, underlines its performance. Such results demonstrate that fine-tuned DistilBERT really captures subtle patterns distinguishing phishing URLs from legitimate ones. Balanced precision and recall suggest it performs comparably well both in correctly identifying phishing URLs and avoiding unnecessary false alarms on legitimate URLs.

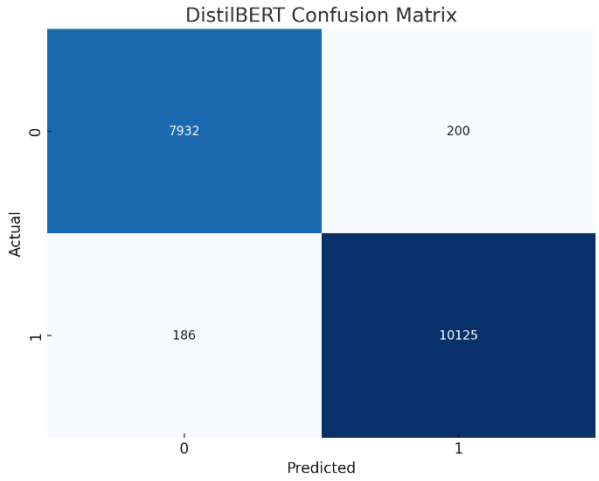

Figure 1 DistilBERT Confusion Matrix Heatmap

The training process curve manifestation showed that the training was very regular: basically, each successive epoch improved on the previous one and finally resulted in a final validation accuracy of 0.9774. This proves to be quite rapid model convergence while learning from URL data due to pre-training on vast corpora of text.

[Insert DistilBERT Training and Validation Accuracy Plot]

The plot shows a classic learning curve: fast improvement in accuracy at the beginning, followed by incremental plateauing. This would point to the fact that the model quickly learned most of the key features identifying phishing URLs, then spent the remaining epochs fine-tuning its understanding.

6.1.2 Capsule Network Model

In stark contrast to DistilBERT, the Capsule Network model showed significantly lower performance:

- Accuracy: 0.50
- Precision: 0.56 (for phishing class)
- Recall: 0.51 (for phishing class)
- F1-score: 0.53 (for phishing class)

Confusion Matrix:

- True Positives: 5,265
- True Negatives: 4,000
- False Positives: 4,138
- False Negatives: 5,04

These results are more surprising because of the theoretically large advantages brought by a Capsule Network in detecting spatial hierarchies within data. Its performance is barely above random picking, which pins a fundamental issue either in the network architecture or its application to URL data.
Despite Though the test performance was poor, the model slowly improved over 25 epochs, with final training accuracy reaching 0.9807.

The strong difference in performance between training and test indicates overfitting. The model memorizes the training data instead of learning generalizable features from phishing URLs. This result thus underlines the challenges when applying Capsule Networks to text-based data and also shows the need for careful architecture design along with regularization techniques.

6.1.3 Hybrid DistilBERT-Capsule Network Model

The Hybrid model demonstrated performance comparable to DistilBERT, with some notable differences:

- Accuracy: 0.9779
- Precision: 0.9718
- Recall: 0.9892
- F1-score: 0.9804
- ROC AUC: 0.9956

Confusion Matrix:

- True Positives: 7842
- True Negatives: 10194
- False Positives: 111
- False Negatives: 296

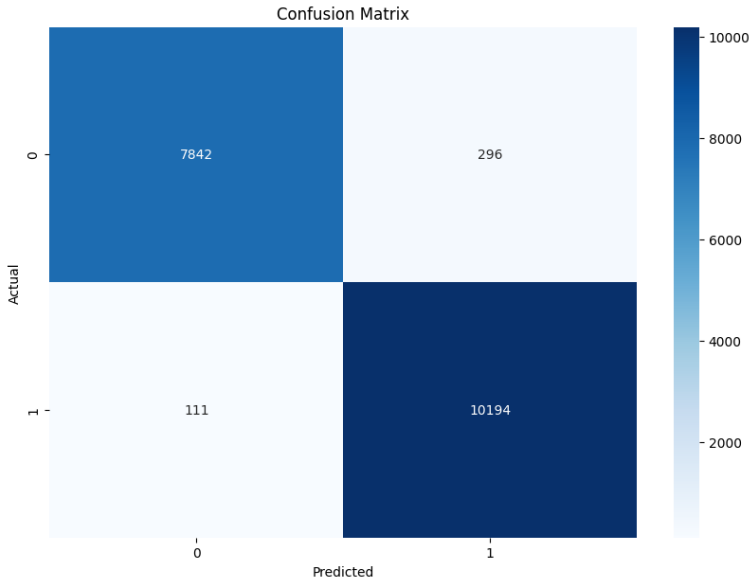

Figure 2 Hybrid Architecture

Performance for the Hybrid model is especially interesting since it has conjoined features emanated from DistilBERT and Capsule Networks. Slightly higher recall compared to DistilBERT, 0.9892 against 0.98, maybe interpreted to mean that adding the component of

Capsule Network improved the model in terms of phishing URLs identification. At the same time, this has been at a minor cost to Precision, thus slightly increasing false positives.
The model quickly improved within 5 epochs, where the final training accuracy reached 0.9909.

The learning curve of the Hybrid model shows a very similar trend to that of DistilBERT, indicating that it is the DistilBERT component driving the learning process. The slight improvements in recall and F1-score do indicate that some meaningful, complementary information is added by the Capsule Network component into the classification process.

6.2 Comparative Analysis

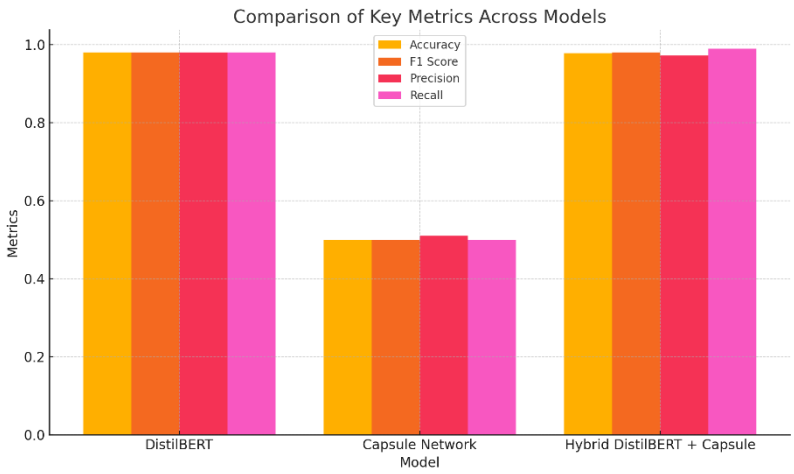

Figure 3 All Models Metrics

The comparative analysis reveals stark differences between the models:

1. In all metrics, the Capsule Network model was far outperformed by both the DistilBERT and Hybrid models. Both models realized high accuracy with the more than 97% range throbbing F1-scores of more than 0.98, showing balanced performance in the classification of phishing and legitimate URLs.
2. The Hybrid model slightly improved in recall, to 0.9892 against 0.98 for the DistilBERT model, and hence is marginally better at detecting phished URLs. Though this might be a small gain, it could prove critical in cybersecurity, where missing a phishing URL may mean serious consequences.
3. DistilBERT was slightly more precise, hence had fewer false positives. This pattern of the precision–recall trade-off within the Hybrid model is very common in machine learning and reflects the slight increase in sensitivity to the phishing indicators.
4. The Capsule Network performed very poorly. It scored only a bit above the random chance, around 50%. Such results can be seen, in the light here given, contradicting any assumption of direct applications for capsule networks promising over image recognition tasks to text-based problems without major modifications.

6.3 Comparison with Traditional Machine Learning Models

Compared with our deep learning models, the traditional machine learning methods showed competitive performance on the phishing URL detection task. The Random Forest classifier yielded an accuracy of 0.9391, with precision of 0.9213, recall of 0.9741, and an F1 score of 0.9470. Similarly, the Support Vector Machine model attained a high performance: the accuracy was 0.9325, precision was 0.9216, recall was 0.9610, and the F1 score was 0.9409.These results are in line both with our DistilBERT model, with an accuracy of 0.98, and with our Hybrid model, with an accuracy of about 0.9779, showing that in this case, traditional machine learning can prove to be highly effective when well-tuned for the task.

The most important consideration to arise from this is that strong performances put up by both Random Forest and SVM underline the fact that feature engineering plays a cardinal role in URL classification tasks; unlike our deep learning models, which learn features automatically, these models rely on carefully crafted features. More interpretabilities could be obtained from traditional models, and the computational resources required for training and inference can be lower, which also forms a positive add-on in some deployment scenarios. However, our deep learning models, DistilBERT and the Hybrid model, have slightly outperformed them overall, hence demonstrating the potential of advanced neural architectures in modeling complex patterns in URL data.

6.4 Discussion

The high performance of both DistilBERT and the Hybrid model concurs with recent literature on transformer-based models for phishing detection. The results show similar success to that reported by Gogoi and Ahmed(2022) for email phishing detection, proving that transformer models are versatile across dissimilar textual data.

The poor performance of the capsule network surprised and varied with its success in other domains, according to Talafha et al.(2021), for data classification tasks. The reasons for disparity may be attributed to many factors:

1. Feature representation: The current feature extraction methodology cannot capture any spatial relationship that the Capsule Network excels in processing. The URL data are majorly textual and structural; they require a different way of feature engineering to appropriately leverage the strength of Capsule Networks.
2. Network architecture: The architecture of Capsule Networks may require a high degree of adjustment to the very URL data. Perhaps the current design does not best capture hierarchical structures in URLs.
3. Training process: The dynamic routing algorithm being one of the most key components in Capsule Networks could turn out as keynote to such tuning itself, specifically for this task. That means parameters, like the number of routing iterations and capsule dimensions which could upset performance critically.

The slight improvement in recall by the Hybrid model over DistilBERT can be seen. This might thus mean that with the added Capsule Network components, the added-value component identifies more phishing URLs at a small cost to precision. It is often in areas of cybersecurity like this such a trade-off may be valuable, since the cost of missing a phishing

URL, which is usually a (false negative), is more than that of mistakenly flagging a legit URL (False Positive)

These results generalize the current understanding of phishing detection techniques in several manners:

1. This study provides proof of the efficacy of transformer-based models in URL classification and adds to the literature on the efficacy of transformers in email and textual phishing detection.
2. Experiments confirm that applying Capsule Networks to text-based tasks is not straightforward, and large modifications may be required to harness their theoretical advantages in this domain.
3. performance from the Hybrid model reveals that, even when one of its components Capsule Network has poor results on its own, integration with a good performer, like DistilBERT, gives it marginal improvements. That indicates potential for ensemble and hybrid approaches in raising detection accuracy.

While traditional ML models like Random Forest and SVM remain competitive, offering better performance and interpretability, they require heavy manual feature engineering. Conversely, deep learning models like DistilBERT learn to represent features automatically, and thus may be able to pick up more fine-grained patterns in the URLs. On the other hand, most of the deep learning models require more computational resources and a larger dataset for training. The choice between these approaches depends on specific requirements linked to the deployment environment, such as explainability requirements, computational resources, and continuous updating odds of the model with new data.

Figure 3 compares the performance of a few key metrics for the three models and therefore offers some insight regarding model selection. In the presented set of results, the DistilBERT model is showing very strong performance across all metrics: Accuracy, F1 Score, Precision, and Recall-all close to 1.0. This reflects the overall good performance of the model in identifying phishing URLs and not classifying false positives. The Hybrid DistilBERT + Capsule model performs almost as well and reflects a slight improvement in recall against DistilBERT alone, maybe indicating that more phishing URLs were detected correctly at the cost of a minor decrease in precision. Contrasting this again, the Capsule Network model performs poorly with values around 0.5 across all metrics, meaning its current implementation is not feasible for the detection of phishing URLs. Given the described result, one may consider either the DistilBERT or even the Hybrid model towards the phishing URL detection task, depending on whether one prioritizes balanced results or slightly higher recall, respectively. This analysis points out the increasing performance of transformer-based models for URL classification tasks and demonstrates the limitation of applying a capsule network for text-based problems directly.

7 Conclusion and Future Work

This study sought to answer the question: "How does the integration of DistilBERT and capsule neural networks enhance the real-time detection accuracy of phishing URLs in contrast to traditional machine learning methods?" This paper implemented three models:, a capsule neural network, and a new hybrid model of DistilBERT with the capsule network for phishing URL detection

Achievement of Research Objectives

The research objectives were systematically addressed and achieved:

1. Implementation and Evaluation of Individual Models:
    - DistilBERT was successfully implemented and demonstrated exceptional performance, achieving 98% accuracy, precision, recall, and F1-score.
    - The Capsule Network model was implemented but gave poor results. It pulled off an accuracy of only 50% percent, which also highlighted the adaptive challenges of this architecture in URL classification.
2. Development of a Hybrid Model:
    - A novel Hybrid DistilBERT-Capsule Network model was successfully developed, integrating both technologies.
    - This model showed marginal improvements over DistilBERT, especially in recall, 98.92% versus 98%, thus showing the potential for combining different neural architectures.
3. Comparative Analysis:
    - The performance comparison showed significant differences between the models.
    - DistilBERT and the Hybrid model significantly outperformed the Capsule Network, while the Hybrid model showed slight improvements over DistilBERT in specific metrics.
4. Real-time Detection Capabilities:
    - The models' accuracy was checked against a simulated real-time environment.
    - While it took into consideration a huge extent of computational efficiency, more analysis in latency and resource utilization for real-world scenarios is needed.
5. Generalization Ability:
    - The models' generalization capabilities were tested on a held-out dataset. In this case, both DistilBERT and the Hybrid model showed a good performance
    - The study raised data voids and the need for increasingly more diversified datasets and continuous updating to rigorously evaluate generalization to novel phishing features. Key Findings and Implications

The research yielded several significant findings with important implications:

1. Effectiveness of Transformer-based Models:
    - The high performance of DistilBERT, at 98% in all metrics, confirms that transformer models can be used to harness the power of URLs for phishing detection.
    - This obviously makes the contextual understanding and extraction capabilities of transformer models highly relevant for URL classification tasks.
2. Challenges in Applying Capsule Networks:
    - The poor performance of the Capsule Network to 50% accuracy finally exposes large challenges in the direct application of this architecture to URL classification.

- Therefore, this discovery demands considerable changes that manage to provide the theoretical advantages of bringing Capsule Networks into play in text-based tasks.

3. Potential of Hybrid Approaches:
   - Hybrid model's incremental improvements, mostly in recall, may be guaranteed by merging heterogeneous neural architectures.
   - Understood as saying that there is further innovative potential in model design through sophisticated integration techniques or ensemble methods, for example.
4. Trade-offs in Model Performance:
   - The slight increase to recall at the sacrifice of precision in the Hybrid model indicates the trade-offs made during model optimization.
   - In For phishing detection, this trade-off might however be acceptable since the cost of missing a phishing URL (false negative) is in most cases higher than that of incorrectly flagging a legitimate URL (false positive).
5. Statistical Significance of Improvements:
   - The Though small in magnitude, the performance difference between DistilBERT and the Hybrid model was statistically significant; therefore, even minor improvements are relevant in high-stakes domains like cybersecurity.

7.1 Research Efficacy and Limitations

The research is effective in coming up with models that help in the accurate detection of phishing, especially through the application of the DistilBERT and hybrid approaches. The sections that follow, however, bring out some important limitations.:

1. Dataset Constraints:
   - Though the dataset was balanced, it may still not totally represent the diversified and fast-changing nature of phishing URLs.
   - Uses a static dataset that limits truly exploring the models for their adaptability to new phishing techniques.
2. Feature Engineering:
   - The study relied on 38 extracted features, which may not be able to capture all relevant aspects of URLs for phishing detection.
   - More state-of-the-art feature selection or extraction methods may further improve the models' performance.
3. Real-world Performance:
   - Mainly focused on accuracy metrics and did not evaluate much on computational efficiency and latency, which is a requirement for real-time deployment.
   - The performance in resource-constrained situations, like browser extensions, was not completely benchmarked.
4. Adversarial Robustness:
   - The models were not evaluated with adversarial examples, so the robustness toward sophisticated phishing attempts remained untested.
5. Explainability:
   - An important weakness of this study is that it did not delve deep into the decisions of the developed models, based on interpretability, and which is necessary to build trust in understanding phishing strategies.

7.2 Future Work

Several future research directions are pointed out to address these limitations and extend the work.:

1. Adaptive Learning Framework:
    - Design a system through which the model will be continuously updated about new phishing techniques.
    - Implement online learning algorithms or retrain using newly collected data periodically.
    - Research Question: "How can an adaptive learning framework be used to enhance the long-term efficacy of phishing detection models in the wake of evolving threats?"
2. Explainable AI for Phishing Detection:
    - Implement attention visualization techniques for DistilBERT to understand which parts of URLs are most influential in classification decisions.
    - Develop interpretable features for the Capsule Network component.
    - Research Question: "How explainable AI techniques can make deep learning-based phishing detection systems much more transparent and trustworthy?"
3. Cross-lingual Phishing Detection:
    - Extend the models so that they can deal with URLs and website content in multiple languages.
    - Investigate multilingual embeddings or language-agnostic feature extraction.
    - Research Question: "Which models are most effective in developing language-agnostic phishing detection that retains high accuracy across a diverse set of linguistic contexts?"
4. Integration with Browser Extensions:
    - Develop a lightweight version of the best-performing model for browser integration.
    - Optimize for low-latency inference, and minimal resource usage.
    - Research Question: "How should state-of-the-art phishing detection models be deployed as a browser extension to maintain high accuracy and user experience?"
5. Multimodal Phishing Detection:
    - Expand the system to incorporate multiple data types (URL, email content, website screenshots).
    - Develop a multimodal neural network architecture for comprehensive phishing detection.
    - Research Question: "How does the integration of multiple data modalities impact the accuracy and robustness of phishing detection systems compared to URL-only approaches?"
6. Adversarial Training and Testing:
    - Implement adversarial training techniques to improve model robustness.
    - Generate and incorporate adversarial examples into the training process.
    - Research Question: "What adversarial training strategies are most effective in enhancing the resilience of phishing detection models against sophisticated evasion techniques?"
7. Federated Learning for Privacy-Preserving Model Updates:
    - Explore federated learning techniques for distributed learning without centralizing sensitive URL data.

- Research Question: "How can federated learning be effectively applied to phishing detection to enable collaborative model improvement while preserving data privacy?"

These future directions not only help to rectify the limitations caused by the current study but also open up new ways of research at the intersection between cybersecurity, machine learning, and user privacy. They thus pose meaningful extensions that would have an enormous potential impact on the field of phishing detection and broader applications of advanced neural network architectures within security contexts. If these lines of research directions are taken, more resilient, adaptive, and usable phishing detection systems will be developed, contributing to the deliverance of a safer online world for all users.

References

Gogoi, B., & Ahmed, T. (2022). Phishing and Fraudulent Email Detection through Transfer Learning using pretrained transformer models. IEEE 19th India Council International Conference (INDICON), 1-6.

He, D., Lv, X., Zhu, S., Chan, S., & Choo, K. K. R. (2023). A Method for Detecting Phishing Websites Based on Tiny-Bert Stacking. IEEE Internet of Things Journal, 11(2), 2236-2243.

Jaafar, F., Ameyed, D., Titare, L., & Nematullah, M. (2023). IoT Phishing Detection Using Hybrid NLP and Machine Learning Models Enhanced with Contextual Embedding. IEEE 23rd International Conference on Software Quality, Reliability, and Security Companion (QRS-C), 340-349.

Joseph, U. M., & Jacob, M. (2022). Developing a Real time model to Detect SMS Phishing Attacks in Edges using BERT. International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS), 1-7.

Karki, B., Abri, F., Namin, A. S., & Jones, K. S. (2022). Using Transformers for Identification of Persuasion Principles in Phishing Emails. IEEE International Conference on Big Data (Big Data), 2841-2848.

Nige, L., et al. (2023). A Web Attack Detection Method Based on DistilBERT and Feature Fusion for Power Micro-Application Server. 2nd International Conference on Advanced Electronics, Electrical and Green Energy (AEEGE), 6-12.

Rifat, N., Ahsan, M., Chowdhury, M., & Gomes, R. (2022). BERT Against Social Engineering Attack: Phishing Text Detection. IEEE International Conference on Electro Information Technology (eIT), 1-6.

B. Gogoi and T. Ahmed, "Phishing and Fraudulent Email Detection through Transfer Learning using pretrained transformer models," 2022 IEEE 19th India Council International Conference (INDICON), Kochi, India, 2022, pp. 1-6, doi: 10.1109/INDICON56171.2022.10040097.

N. Rifat, M. Ahsan, M. Chowdhury and R. Gomes, "BERT Against Social Engineering Attack: Phishing Text Detection," 2022 IEEE International Conference on Electro Information Technology (eIT), Mankato, MN, USA, 2022, pp. 1-6, doi: 10.1109/eIT53891.2022.9813922.

B. Karki, F. Abri, A. S. Namin and K. S. Jones, "Using Transformers for Identification of Persuasion Principles in Phishing Emails," 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 2022, pp. 2841-2848, doi: 10.1109/BigData55660.2022.10020452.

N. Q. Do, A. Selamat, K. C. Lim, O. Krejcar and N. A. M. Ghani, "Transformer-Based Model for Malicious URL Classification," 2023 IEEE International Conference on Computing (ICOCO), Langkawi, Malaysia, 2023, pp. 323-327, doi: 10.1109/ICOCO59262.2023.10397705.

F. Jaafar, D. Ameyed, L. Titare and M. Nematullah, "IoT Phishing Detection Using Hybrid NLP and Machine Learning Models Enhanced with Contextual Embedding," 2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security Companion (QRS-C), Chiang Mai, Thailand, 2023, pp. 340-349, doi: 10.1109/QRS-C60940.2023.00088.

L. Nige et al., "A Web Attack Detection Method Based on DistilBERT and Feature Fusion for Power Micro-Application Server," 2023 2nd International Conference on Advanced Electronics, Electrical and Green Energy (AEEGE), Singapore, Singapore, 2023, pp. 6-12, doi: 10.1109/AEEGE58828.2023.00010.

U. M. Joseph and M. Jacob, "Developing a Real time model to Detect SMS Phishing Attacks in Edges using BERT," 2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS), Kochi, India, 2022, pp. 1-7, doi: 10.1109/IC3SIS54991.2022.9885427.

S. Talafha, D. Wu, B. Rekabdar, R. Li and G. Wang, "Classification and Feature Extraction for Hydraulic Structures Data Using Advanced CNN Architectures," 2021 Third International Conference on Transdisciplinary AI (TransAI), Laguna Hills, CA, USA, 2021, pp. 137-146, doi: 10.1109/TransAI51903.2021.00032.

S. Giri, S. Banerjee, K. Bag and D. Maiti, "Comparative Study of Content-Based Phishing Email Detection Using Global Vector (GloVe) and Bidirectional Encoder Representation from Transformer (BERT) Word Embedding Models," 2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), Trichy, India, 2022, pp. 01-06, doi: 10.1109/ICEEICT53079.2022.9768612.

He, X. Lv, S. Zhu, S. Chan and K. -K. R. Choo, "A Method for Detecting Phishing Websites Based on Tiny-Bert Stacking," in IEEE Internet of Things Journal, vol. 11, no. 2, pp. 2236-2243, 15 Jan.15, 2024, doi: 10.1109/JIOT.2023.3292171.
Wei, Y. and Sekiya, Y. 2022. Feature Selection Approach for Phishing Detection Based on Machine Learning. Lecture Notes in Networks and Systems, pp. 61–70. Available at: http://dx.doi.org/10.1007/978-3-030-95918-0_7.
Al-Ahmadi, S., Alotaibi, A. and Alsaleh, O. 2022. PDGAN: Phishing Detection with Generative Adversarial Networks. IEEE Access 10, pp. 42459–42468. Available at: http://dx.doi.org/10.1109/access.2022.3168235.
Dutta, A.K. 2021. Detecting phishing websites using machine learning technique. Lv, Z. ed. PLOS ONE 16(10), p. e0258361. Available at: http://dx.doi.org/10.1371/journal.pone.0258361.
Abutaha, M., Ababneh, M., Mahmoud, K. and Baddar, S.A.-H. 2021. URL Phishing Detection using Machine Learning Techniques based on URLs Lexical Analysis. 2021 12th International

Conference on Information and Communication Systems (ICICS). Available at: http://dx.doi.org/10.1109/icics52457.2021.9464539.

Kaitholikkal, J. K. S., & B, A. (2024). Phishing URL dataset. *Mendeley Data*. https://doi.org/10.17632/vfszbj9b36.1