IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Outsmarting Cyber Threats: AI-Powered Deep Learning and NLP Frameworks for Proactive Malicious URL Detection

Innocent Paul Ojo[1*]; Ashna Tomy [2]

[2]UNIVERSITY OF HERTFORDSHIRE
School of Physics, Engineering and Computer Science, Hatfield, United Kingdom

[3]UNIVERSITY OF HERTFORDSHIRE
School of Physics, Engineering and Computer Science, Hatfield, United Kingdom

Abstract

This study investigates the integration of advanced deep learning and natural language processing (NLP) frameworks for the proactive detection of malicious URLs, encapsulated in the theme "Outsmarting Cyber Threats: AI-Powered Deep Learning and NLP Frameworks for Proactive Malicious URL Detection." By employing a character-level embedding strategy combined with robust regularization techniques, the research enhances both the accuracy and generalization of the models within a cybersecurity context. Three models are rigorously evaluated: Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Multi-Layer Perceptron (MLP).The LSTM model, achieving an accuracy of 78.1%, demonstrated a moderate ability to capture sequential patterns in URL structures. In contrast, the BiLSTM model, with an improved accuracy of 95.3%, effectively harnessed bidirectional context to detect nuanced threats such as phishing and malware. Remarkably, the MLP model achieved an accuracy of 99.2%, showcasing its superior efficiency in processing non-sequential data while maintaining high performance.

These results underscore the transformative potential of combining deep learning and NLP techniques to develop agile, real-time threat detection systems capable of handling vast data volumes and adapting to evolving cyber threats. The study also addresses practical challenges such as computational intensity, dataset quality, and class imbalance, and it offers recommendations for future research to explore more advanced architectures, diversify datasets, and streamline deployment strategies.

Keywords: *Malicious URL detection, BiLSTM, LSTM, MLP, cybersecurity, deep learning, phishing detection, machine learning models, feature extraction, sequential data processing, real-time threat detection, computational efficiency, model scalability, cyber threat analysis, URL classification.*

1. Background

Digitalisation offers significant conveniences but also introduces critical cybersecurity challenges (AlSalem, Almaiah, and Lutfi, 2023). As digital transactions and internet-based services expand, so does the scope for cyber threats, with criminals continuously adapting to exploit system vulnerabilities (Bederna and Rajnai, 2022; McKinsey & Company, 2022).

Malicious URLs are a major threat, acting as gateways for phishing, malware, and other harmful activities. Traditional detection methods, such as blacklists and heuristic-based techniques, struggle to keep pace with the rapid evolution of these URLs, creating notable security gaps (Sun et al., 2020; Ghaleb et al., 2022).

Deep learning and natural language processing (NLP) have emerged as powerful tools to address these issues. Deep learning leverages neural networks to model complex data patterns, achieving success in fields like image recognition, language understanding, and cybersecurity (Dong et al., 2023b). Meanwhile, NLP enables the analysis of textual content in URLs and related communications to detect malicious intent (Jia and Liang, 2023). Combining these approaches allows systems to learn and distinguish between benign and malicious URLs, even when cybercriminals design them to bypass traditional defenses (Santosh Kumar Birthriya and Ankit Kumar Jain, 2021; Afzal et al., 2021).

NLP further enhances detection by analyzing URL strings, web content, and metadata (Liang et al., 2021), offering a more robust solution (Ziems and Wu, 2021). The rise of social media has compounded these challenges, as these platforms are exploited to spread malicious URLs, phishing attacks, and malware, posing serious risks to individuals and organizations (Özkent, 2022; Herath, Khanna, and Ahmed, 2022).

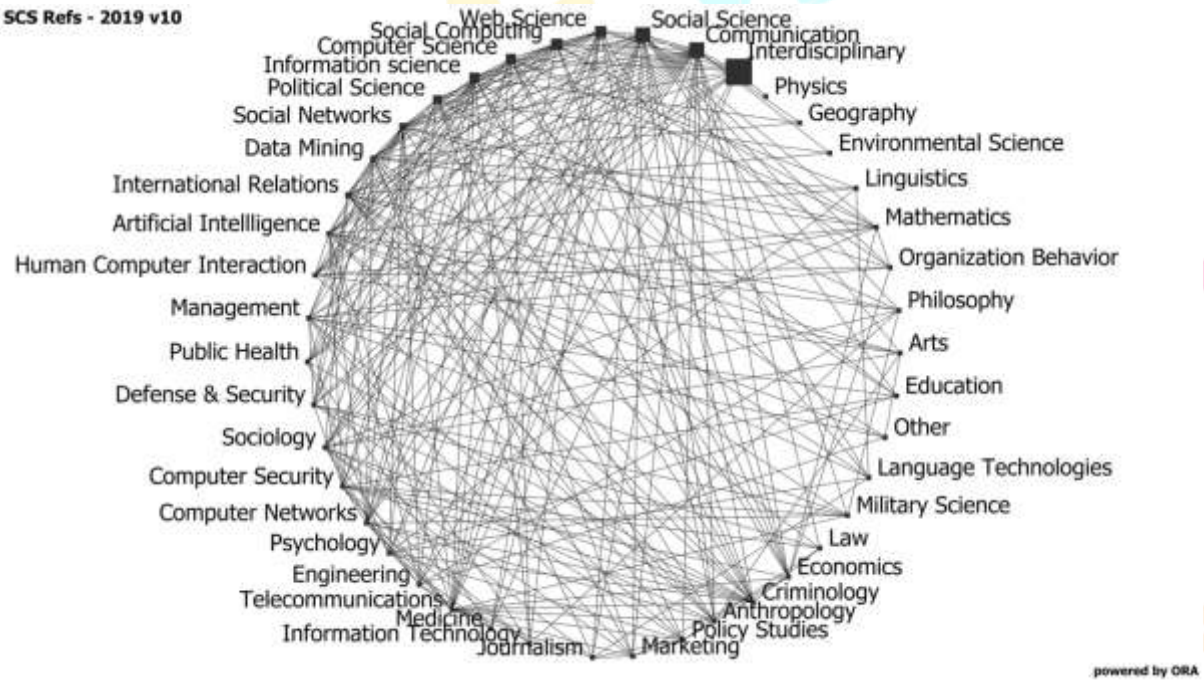

Figure 1: Social Cybersecurity: an emerging science (Carley, 2020)

Figure 1 shows how social cybersecurity integrates multiple disciplines, with nodes representing specific fields and lines indicating research overlaps.Deep learning and NLP techniques have proven effective in analyzing large volumes of social media data for malicious URL detection, enabling proactive threat responses (Garg, Gupta, and Srivastava, 2024; Sarker, 2021). Recent studies also demonstrate the success of models like RNNs, BiLSTMs, and MLPs in detecting complex patterns within URL structures (Garg, Gupta, and Srivastava, 2024; Zhao, Du, and Zhang, 2022). Moreover, integrating explainable AI into these frameworks enhances model transparency and trust, allowing cybersecurity professionals to better understand and validate detection processes (Othmane Niyaoui and Oussama Mohamed Reda, 2024; Hassija et al., 2023; Charmet et al., 2022).

Malicious URLs are critical cybersecurity threats. They facilitate phishing by disguising links, distribute malware that can lead to data breaches, and trigger drive-by downloads without user awareness. Cybercriminals exploit these URLs for command and control operations and quickly modify them to evade detection. Traditional approaches, relying on static blacklists or heuristic rules, are reactive and often unable to detect sophisticated tactics such as URL shortening, typosquatting, and obfuscation (Rao and Pais, 2017; Ghaleb et al., 2022b).

Advances in artificial intelligence, particularly deep learning and NLP, have transformed various domains including cybersecurity (Zhou et al., 2020; Xu et al., 2021). Modern NLP models like BERT and GPT-3 excel at understanding and generating human language (Jamin Rahman Jim et al., 2024), while deep learning models such as BiLSTM, RNNs, and MLPs effectively identify complex data patterns. These technologies enhance the detection of malicious URLs by analyzing both structural and textual features, thereby reducing errors and enabling proactive threat detection (Lin et al., 2022).Deep learning models are proficient at uncovering hidden patterns in vast datasets, making them well-suited for detecting subtle anomalies in URL structures (Burbela, 2023; R et al., 2020). Meanwhile, NLP techniques process and interpret textual data, crucial for identifying phishing and other text-based attacks. This combination improves detection accuracy, reduces false positives and negatives, and adapts to sophisticated evasion tactics, ultimately strengthening cybersecurity defenses (Mittal et al., 2022; Aldakheel et al., 2023; Saeed et al., 2023).

1.2 Problem Statement

Traditional methods for detecting malicious URLs, such as blacklists and heuristic approaches, are increasingly ineffective due to the rapid generation and mutation of these threats, resulting in high false positive and false negative rates (Orozco-Fonseca, Marín, and Lara, 2024; Chaudhari, Thakur, and Rajan, 2024). Malicious URLs mislead users into accessing harmful sites, leading to severe consequences like financial loss, data theft, and compromised system integrity (Su and Su, 2023b; Aysar Weshahi et al., 2024b). This project addresses the need for an advanced detection system that combines deep learning and NLP techniques to accurately identify and mitigate the risks posed by malicious URLs on social media.

1.4 Objectives of the Study

This study aims to enhance the detection of malicious URLs by integrating deep learning and NLP, with the following objectives:

1. Develop a system that combines deep learning models with NLP techniques to detect malicious URLs on social media.
2. Compare the effectiveness of various deep learning algorithms—including BiLSTM, RNNs, and MLP—in identifying malicious URLs.
3. Utilize NLP methods to analyze URL text and associated content to improve contextual detection accuracy.
4. Create a monitoring system capable of processing large volumes of social media data and providing timely threat alerts.

1.5 Research Questions

The study seeks to answer the following:

1. Which deep learning algorithms (BiLSTM, RNNs, MLP) are most effective at identifying malicious URLs?
2. How do these deep learning models compare in terms of accuracy and efficiency?
3. How can NLP techniques such as tokenization, word embeddings, and sentiment analysis be leveraged to interpret URL content and detect malicious intent?

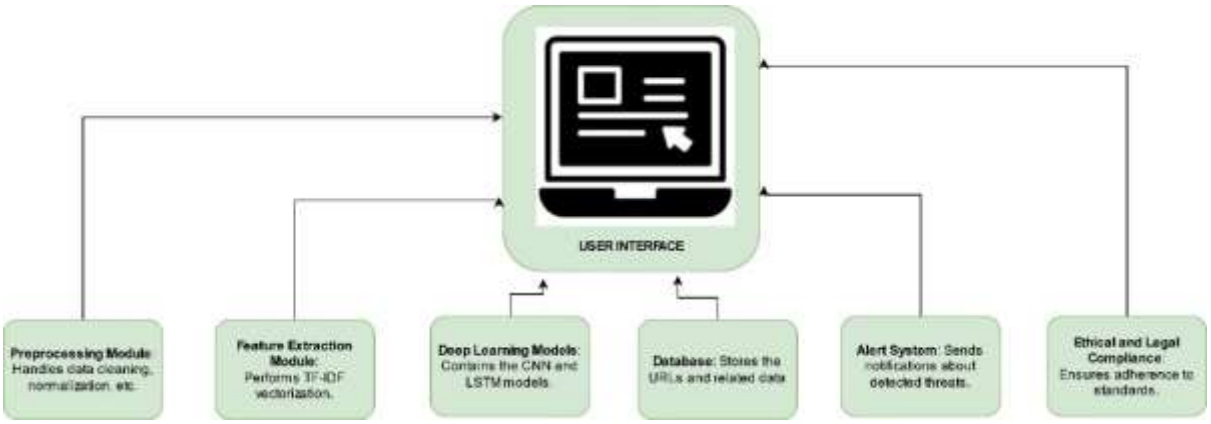

Figure 3: Architectural Diagram

Figure 3 outlines the proposed system architecture integrating deep learning and NLP for proactive malicious URL detection.

2. Literature Review

This chapter reviews existing research on detecting malicious URLs, emphasizing the integration of deep learning and natural language processing (NLP) techniques. Traditional methods—such as blacklists, signature-based, and heuristic approaches—have been widely used but often fall short against evolving threats. Their limitations underscore the need for more adaptive, intelligent solutions.

Deep learning has emerged as a powerful tool in cybersecurity, with models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) excelling in pattern recognition and anomaly detection (Kasongo, 2022; Ananya Redhu et al., 2024). These models can process vast amounts of data and identify subtle patterns that static methods might miss. NLP techniques further enhance these capabilities by analyzing the textual content of URLs. Methods such as tokenization and word embeddings allow for a deeper understanding of URL intent, providing a more comprehensive threat detection approach (Tyagi and Bhushan, 2023; Khan et al., 2023b).

2.1 Overview of Cybersecurity Threats

Cybersecurity threats have grown in complexity, posing significant risks to individuals, organizations, and governments. This section focuses on the various threats facilitated by malicious URLs.Phishing attacks deceive users into divulging sensitive information by embedding malicious URLs in emails, social media, or fake websites. These URLs redirect victims to harmful sites designed to steal personal data, leading to unauthorized access, financial loss, and identity theft (Alkhalil et al., 2021; Kosinski, 2024).Malware—including viruses, worms, trojans, and ransomware—is often distributed via malicious URLs that lead to compromised sites where harmful software is silently downloaded. These infections can cause severe system damage, data loss, and unauthorized control over devices (Ghosh and Soumen Kanrar, 2023; Ghanem, Rosso and Rangel, 2018; Chaithanya and Brahmananda, 2021).Drive-by download attacks occur when users inadvertently download malicious software simply by visiting a compromised website. Exploiting vulnerabilities in browsers and plugins, these attacks install malware without the user's knowledge, compromising system integrity and exposing sensitive data (Kaspersky, 2019; Ghosh and Soumen Kanrar, 2023b).Malicious URLs can connect infected devices to Command and Control (C&C) servers, allowing attackers to coordinate operations, launch large-scale Distributed Denial of Service (DDoS) attacks, and maintain ongoing access for data theft and system manipulation (Gardiner, Cova and Nagaraja, 2014; Ogu et al., 2019; Lohachab and Karambir, 2018).

SEO poisoning manipulates search engine rankings to boost malicious URLs in search results, increasing the likelihood that users will click on harmful links. This undermines the credibility of search engines and

exposes users to various cyber threats (The MITRE Corporation, 2023; BlackBerry, 2024).Social engineering attacks exploit human psychology, using malicious URLs embedded in seemingly trustworthy messages to trick users into revealing sensitive information or executing compromising actions. The deceptive nature of these URLs makes them particularly challenging to detect (Othmane Niyaoui and Oussama Mohamed Reda, 2024b; Ejaz, Mian and Manzoor, 2023).

Traditional detection techniques primarily rely on heuristic-based methods and blacklists. While straightforward to implement, these methods struggle to adapt to the rapid evolution of cyber threats.Heuristic-based approaches analyze features and patterns within URLs—such as structure, domain characteristics, and webpage content—using predefined rules. However, these methods require constant updates and can produce high false positive or negative rates when attackers modify URL patterns (Kumi, Lim and Lee, 2021; Silva, Feitosa and Garcia, 2020).Blacklist methods maintain and reference lists of known malicious URLs to block access. Although common in browsers and security systems, blacklists are reactive and can quickly become outdated as attackers use tactics like URL shortening and obfuscation to bypass them (Bell and Komisarczuk, 2020; Souppaya and Scarfone, 2013).

Traditional methods for detecting malicious URLs face significant challenges. Attackers use evasion techniques such as polymorphism—altering URL structures or payloads—and encrypted communications to bypass these systems, rendering heuristic and blacklist-based approaches less effective (Mamun et al., 2016c; Alraizza and Algarni, 2023). Due to their static, rule-based nature, these methods struggle to adapt to rapidly emerging attack vectors and lack the contextual awareness needed to distinguish benign from malicious URLs accurately (Beaman et al., 2021; Khraisat et al., 2019). Additionally, as web traffic increases, maintaining up-to-date blacklists and heuristics becomes complex, leading to scalability issues and reduced detection accuracy (Abad, Gholamy and Mohammad Reza Aslani, 2023).

2.2 Advances in Deep Learning for Cybersecurity

Deep learning has revolutionized cybersecurity by offering advanced methods for threat detection. These methods leverage various neural network architectures to identify complex patterns in large datasets, proving especially effective in detecting malicious URLs.Recurrent Neural Networks (RNNs) are well-suited for sequential data, making them effective for analyzing URL structures where the order of characters or words is crucial. Variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) overcome issues like vanishing gradients, allowing for the capture of long-term dependencies (Al-Selwi et al., 2024; Liu and Zhang, 2021). Transformers, which utilize attention mechanisms, process entire input sequences simultaneously and are particularly effective in understanding the contextual relationships within URL components (Krishna Teja Chitty-Venkata et al., 2023). Multi-Layer Perceptrons (MLPs) and Bidirectional LSTMs (BiLSTMs) also enhance detection capabilities by recognizing patterns across URL features and capturing contextual information from both directions (Hnamte and Hussain, 2023; Cheah and Fellows, 2023; Zhu et al., 2024).

Deep learning models excel in distinguishing harmful URLs from benign ones. RNNs, GRUs, BiLSTMs, and transformers analyze the sequential structure of URLs—examining subdomains, paths, and query strings—to detect anomalies and potential threats (Al-Selwi et al., 2024b; Liu and Zhang, 2021b; Hnamte and Hussain, 2023b; Cheah and Fellows, 2023b; Zhu et al., 2024b). In phishing detection, these models process website content and URL structures to identify fraudulent patterns, such as typosquatting and domain spoofing (Alkhalil et al., 2021b; Kosinski, 2024). For malware detection, deep learning aids in analyzing executables, network traffic, and behavioral patterns to detect links associated with malware downloads or command-and-control activities (Ghosh and Soumen Kanrar, 2023). Additionally, Intrusion Detection Systems (IDS) benefit from deep learning's enhanced anomaly detection capabilities, which help identify deviations in network traffic and system logs indicative of attacks (Gardiner, Cova and Nagaraja, 2014; Ogu et al., 2019).

2.3 NLP in Cybersecurity

Natural Language Processing (NLP) enables systems to analyze and interpret textual data, proving invaluable in cybersecurity for evaluating URL content and associated online communications.

Key NLP techniques include:

- **Tokenization:** Breaking text into words or characters. For example, tokenizing "http://example.com/login" yields components like ["http", "example", "com", "login"] (Murel, 2024).
- **Text Normalization:** Standardizing text via lowercasing, stemming, and lemmatization to ensure consistency (Murel, 2024b).
- **Part-of-Speech (POS) Tagging:** Assigning grammatical roles to words, which aids in contextual understanding (Murel, 2024c; nltk.org, 2023).
- **Named Entity Recognition (NER):** Identifying entities such as names, organizations, or locations in text (IBM TechXChange, 2024).
- **Word Embeddings:** Representing words as dense vectors to capture semantic relationships. Techniques include Word2Vec, GloVe, and FastText (Barnard, 2024).
- **Sequence Models:** Utilizing RNNs, LSTMs, GRUs, and Transformers to capture dependencies in text, with Transformers using self-attention for parallel processing.

NLP techniques are crucial for analyzing textual components of URLs and associated content. In phishing detection, NLP analyzes webpage and email text for indicators like urgent language or misleading hyperlinks, helping to identify fraudulent attempts (Benavides-Astudillo et al., 2023). For URL analysis, tokenizing URLs into meaningful parts—such as protocol, domain, and path—reveals suspicious patterns, while character-level analysis detects anomalies and domain mimicking (Su and Su, 2023d). Additionally, NLP aids in web content analysis by detecting keywords and sentiment that signal potential threats. In log analysis, NLP models normal behavioral patterns and flag deviations that could indicate intrusions. Finally, NLP contributes to threat intelligence by extracting and analyzing data from social media and dark web sources to identify emerging threats and attack trends (Sufi, 2024; Arazzi et al., 2023b).

2.4 Integration of Deep Learning and NLP

The integration of deep learning and natural language processing (NLP) has led to advanced cybersecurity solutions capable of addressing complex threats like malicious URLs. By combining powerful techniques from both fields, these systems enhance threat detection and mitigation, as illustrated by existing frameworks and the identification of gaps for further improvement (Kaur, Gabrijelčič and Klobučar, 2023).**PhishNet** is a notable example that detects phishing URLs by integrating deep learning and NLP. It uses Convolutional Neural Networks (CNNs) to analyze URL structures and Long Short-Term Memory (LSTM) networks to process webpage text for persuasive language and sensitive information requests, effectively combining structural and content analysis (Najwa Altwaijry et al., 2024; Ozcan et al., 2021).

Another system, **URLNet**, examines both character-level and word-level features. CNNs extract features from URL embeddings, while NLP techniques analyze word embeddings to capture semantic meaning, thus improving detection of obfuscation tactics (Le et al., 2018a).

Frameworks employing transformer models, such as **BERT**, leverage attention mechanisms to analyze entire URL sequences simultaneously. This approach captures complex dependencies and contextual relationships, enhancing the detection accuracy of malicious URLs (Yu et al., 2024b).

Despite advancements, current models face challenges in adapting to evolving threats. Continuous learning systems could enable real-time model updates to handle new malicious URL patterns (Shams Forruque

Ahmed et al., 2023b). Data quality and diversity also impact performance; expanding datasets and using data augmentation can improve model robustness (Aldoseri, Khalifa and Hamouda, 2023).

Feature engineering remains an area for improvement, where combining automated extraction with expert-driven insights could capture subtle indicators of malicious behavior more effectively (Gibert et al., 2022). Additionally, enhancing model explainability is crucial. Techniques that clarify decision-making processes can build trust and improve practical application in cybersecurity contexts (Balasubramaniam et al., 2022).

2.7 Justification of the Approach

This section outlines why deep learning and NLP techniques were chosen over traditional and conventional machine learning approaches, emphasizing their superior ability to handle complex data and detect sophisticated threats like malicious URLs.Deep learning techniques, including Recurrent Neural Networks (RNNs) and Multi-Layer Perceptrons (MLP), excel at analyzing intricate data patterns. Models such as Long Short-Term Memory (LSTM) networks capture temporal dependencies crucial for tracking evolving URL patterns, as demonstrated by systems like PhishNet and URLNet (Sarker, 2021d; DiPietro and Hager, 2020). These models offer scalability by processing large datasets efficiently with modern computing resources (GPUs and TPUs) and adaptability through transfer and continuous learning (Das et al., 2023).

NLP techniques are essential for analyzing the textual data associated with URLs. Word embeddings and transformer models capture the semantic and contextual nuances of language, which are vital for detecting sophisticated phishing attempts (Arazzi et al., 2023b). Tokenization breaks URLs into meaningful components, while character-level analysis detects subtle obfuscations. Integrating NLP with deep learning allows for comprehensive analysis of both URL structure and content, resulting in improved threat detection and timely response (Kaur, Gabrijelčič and Klobučar, 2023b).

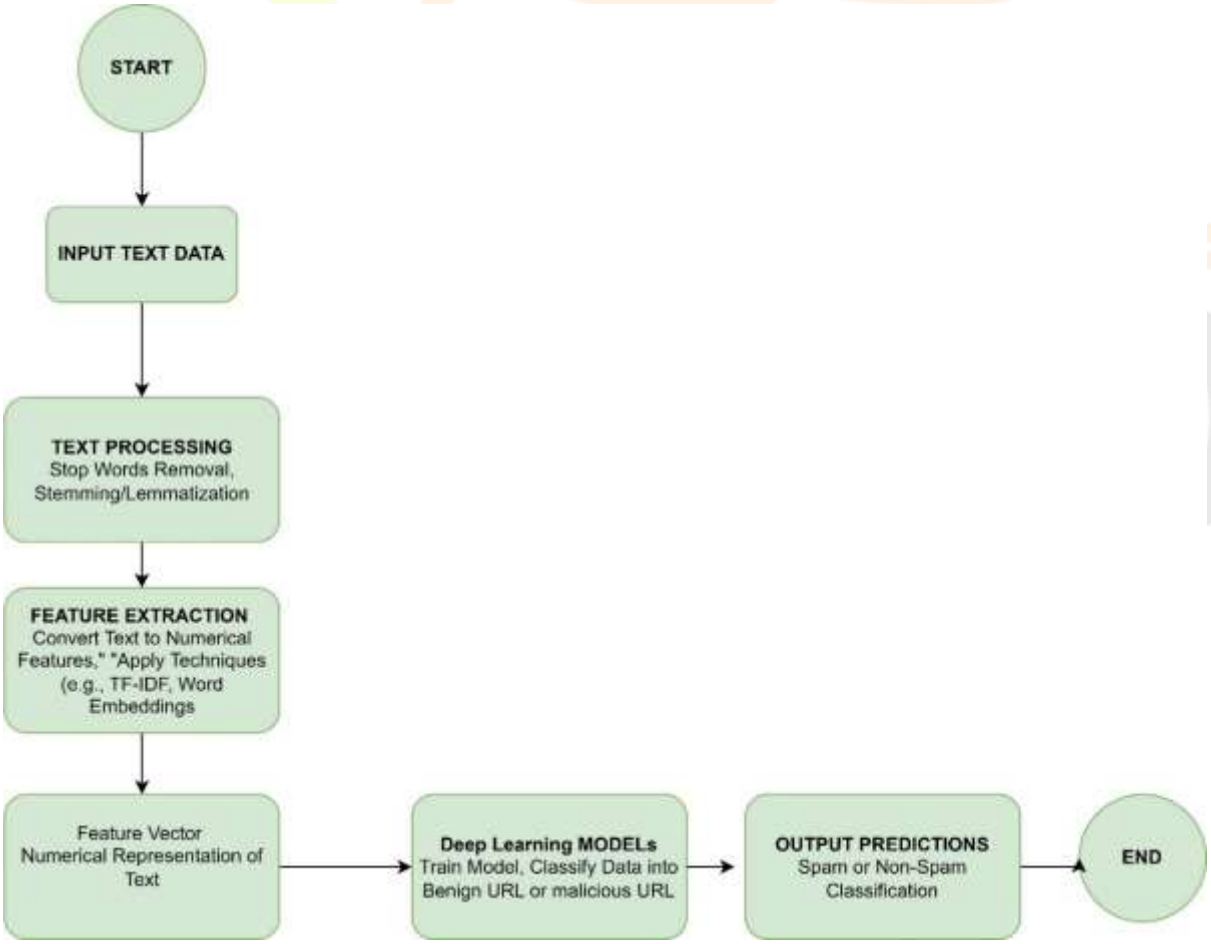

Figure 3: NLP Flow Chart

Traditional detection methods, such as heuristic-based and blacklist-based approaches, rely on static rules and struggle to adapt to new threats. They often suffer from high false positive and negative rates and require constant updates, leading to increased maintenance costs and delays (Gupta and Jain, 2020). In contrast, deep learning and NLP methods offer dynamic, adaptive learning that automatically extracts high-dimensional features from diverse data. This leads to improved performance in detecting malicious URLs and a significant advantage over conventional approaches (Manakitsa et al., 2024; Taye, 2023).

3. Methodology

3.1 Data Collection and Description

This chapter details the methodology used to develop and implement a deep learning and NLP-driven approach for detecting malicious URLs. It outlines each step—from data collection to system integration and evaluation—ensuring the research process is transparent and reproducible.The research employs an experimental and applied approach, combining deep learning with natural language processing (NLP) to build an effective cybersecurity tool. The development is structured into distinct phases, as summarized in the tables below.

Table 1: Data Collection and Preprocessing

Objective	Methods
Assemble a diverse dataset of URLs and prepare it for analysis.	Data is gathered from various sources, cleaned, normalized, and formatted for model training.

Table 2: Model Development

Objective	Methods
Create and train models using deep learning and NLP techniques to identify malicious URLs.	Selection of models such as MLP and RNNs, training with preprocessed data, and fine-tuning parameters to enhance performance.

Table 3: System Integration

Objective	Methods
Integrate the trained models into a real-time monitoring system.	Develop a pipeline for real-time URL analysis, combining deep learning and NLP components, and ensure smooth integration within the cybersecurity setup.

Table 4: Evaluation and Validation

Objective	Methods
Evaluate the models' effectiveness and efficiency.	Use metrics like accuracy, precision, recall, and F1 score to test the models against a separate dataset and compare with traditional methods.

Table 5: Implementation and Testing

Objective	Methods
Deploy the system in a real-world setting and test its performance.	Implement the system, monitor its performance, gather feedback, and adjust as necessary.

Table 6: Research Phases

Phase	Details
Exploratory Phase	- *Literature Review*: Review existing research on malicious URL detection, deep learning models, and NLP techniques. - *Problem Definition*: Define the problem, identify gaps, and establish research questions and hypotheses.
Data Collection Phase	- *Data Sources*: Obtain datasets from platforms such as Kaggle and PhishTank. - *Data Aggregation*: Combine data from multiple sources to form a comprehensive dataset of benign and malicious URLs.
Data Preprocessing Phase	- *Data Cleaning*: Address duplicates, errors, and missing values. - *Feature Engineering*: Extract relevant features from URLs (e.g., domain, path, query parameters). - *Data Transformation*: Normalize and tokenize URL components for modeling.
Model Development Phase	- *Model Selection*: Evaluate various deep learning architectures and NLP techniques. - *Model Training*: Train selected models using preprocessed data, applying cross-validation to optimize performance. - *Hyperparameter Tuning*: Adjust parameters to improve accuracy and prevent overfitting.
System Integration Phase	- *Pipeline Development*: Develop a pipeline that integrates deep learning and NLP for real-time URL analysis. - *Real-Time Monitoring*: Implement mechanisms for ongoing threat detection.
Evaluation and Validation Phase	- *Performance Metrics*: Define metrics such as accuracy, precision, recall, and F1 score to assess model performance. - *Comparative Analysis*: Compare the models with traditional heuristic and blacklist methods. - *Testing*: Validate models using separate test datasets and real-world scenarios.
Implementation and Testing Phase	- *System Deployment*: Deploy the system in a real-world environment. - *Performance Monitoring*: Continuously monitor performance, collect feedback, and make improvements as needed.

3.2 Ethical Practices

The research prioritizes data privacy and security. Although no formal ethical approval was required, proper permissions and attributions for data usage are secured. Data is anonymized where possible, ensuring responsible and effective integration of advanced technologies in detecting malicious URLs.

3.3 Data Collection

The dataset for this project was sourced exclusively from Kaggle, specifically the *Malicious Phish Dataset*. It comprises 651,191 URLs categorized as follows:

- **Benign:** 428,103
- **Defacement:** 96,457
- **Phishing:** 94,111
- **Malware:** 32,520

This dataset, compiled from reputable sources such as PhishTank, OpenPhish, and URLHaus, provides a diverse and comprehensive foundation for training and evaluating models.The dataset was preprocessed through the following steps:

- **Data Cleaning:** Removing duplicate entries and handling missing values by filling gaps or discarding incomplete records. Malformed URLs were corrected to maintain data integrity.
- **Normalization:** Standardizing URLs (e.g., converting to lowercase, removing unnecessary slashes) and breaking them down into key components (domain, path, query parameters).
- **Feature Extraction:** Analyzing domain names for characteristics like length and suspicious keywords, and examining URL paths and query parameters for patterns indicating malicious activity.
- **Transformation:** Encoding categorical features using one-hot encoding and converting textual content into numerical vectors via TF-IDF and word embeddings. Scaling techniques ensured uniform contribution of features during training.
- **Data Splitting:** Dividing the dataset into training (80%), validation, and testing sets to fine-tune model parameters and prevent overfitting.

3.4 Algorithm Selection

Choosing appropriate algorithms is crucial for effective detection of malicious URLs. The process involved evaluating various deep learning models and NLP techniques based on their ability to recognize patterns indicative of potential threats. Preference for models capable of handling sequential data, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks.Evaluation based on accuracy, precision, recall, and F1 score to ensure balanced identification of malicious and benign URLs.

The final selection included:

- **RNNs (LSTMs):** For managing sequential dependencies within URLs.
- **Multi-Layer Perceptrons (MLPs):** For pattern recognition across various URL features.
- **Bidirectional LSTMs (BiLSTMs):** For capturing contextual information in both directions.

NLP techniques were selected based on their ability to extract and analyze textual content from URLs. Key considerations included:Both character-level and word-level tokenization to break down URLs into meaningful components.Use of Word2Vec and FastText to capture semantic relationships. Adoption of transformer models like BERT to provide contextual understanding of URL components.

The chosen NLP techniques comprised:

- **Tokenization:** Character-level and word-level tokenization.
- **Word Embeddings:** Implementing Word2Vec and FastText.
- **Sequence Encoding:** Using BERT for deep contextual analysis.

This strategic selection, combining advanced deep learning models with robust NLP techniques, ensures the development of a reliable and scalable system for detecting malicious URLs with high accuracy and efficiency.

3.5 Model Training

Training deep learning models for malicious URL detection requires a structured approach, ensuring effective learning while minimizing overfitting.The dataset is divided into training (70%), validation (15%), and test (15%) sets to balance learning and evaluation. Data augmentation enhances variability by modifying URL components. Feature extraction follows, capturing key attributes such as domain length, special characters, and path depth. NLP techniques—including Word2Vec, FastText, and BERT—are then applied to tokenize and vectorize URL elements for model input.

The deep learning architectures (MLP, RNN, LSTM) are configured with appropriate layers, activation functions, and hidden units. Weight initialization methods like Xavier and He initialization ensure stable learning. Mini-batch gradient descent updates model weights iteratively, with batch sizes determined by computational constraints. Each iteration involves forward propagation for prediction, loss computation via Binary Cross-Entropy, and backpropagation to optimize weights. Optimizers such as Adam, RMSprop, and SGD adjust learning rates dynamically. Regularization techniques (dropout, L2 regularization, batch normalization) mitigate overfitting.

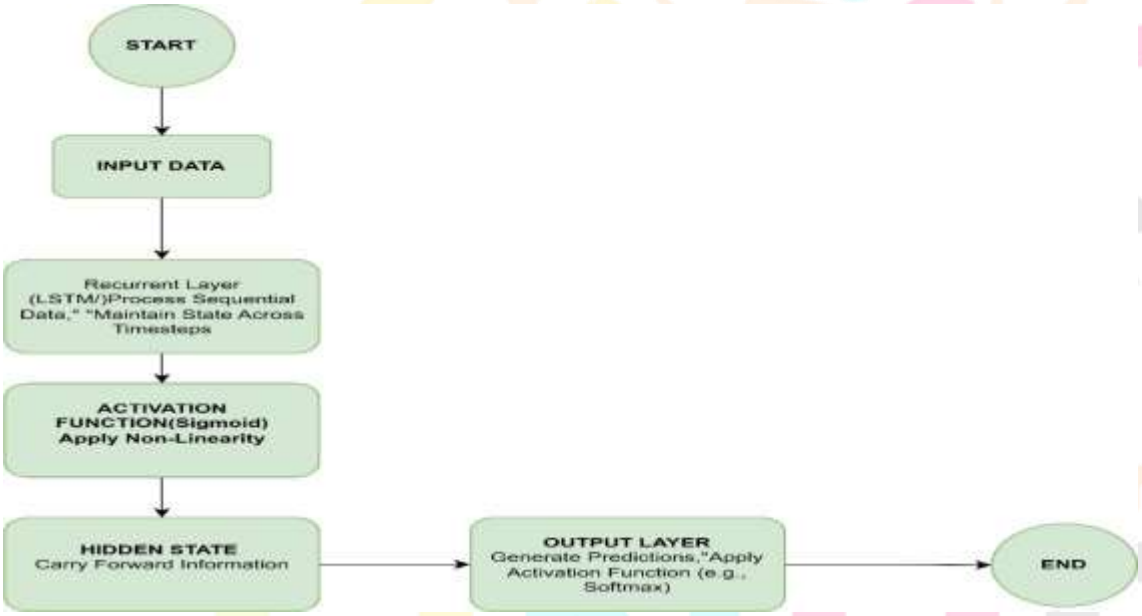

Figure 4:RNN Flow Chart

After each epoch, performance is evaluated using accuracy, precision, recall, and F1-score. Hyperparameters—including learning rate, batch size, and epoch count—are optimized using grid search and random search. Early stopping prevents overfitting by halting training when validation performance plateaus.The trained model is assessed on the test set using key evaluation metrics. A comparative analysis of different architectures and NLP techniques identifies the most effective configuration.

Model performance depends on careful parameter tuning:

- **Learning Rate:** Starts between 0.001 and 0.01, adjusted dynamically.
- **Batch Size:** Ranges from 32 to 256 based on hardware capacity.
- **Epochs:** Set between 50 and 100, with early stopping applied.
- **Optimizers:** Adam (default), with RMSprop and SGD as alternatives.

- **Regularization:** Dropout (0.2–0.5), L2 regularization (0.0001–0.01).
- **Word Embeddings:** Dimension range of 50–300.
- **Tokenized Sequence Length:** Adjusted between 50–200 tokens based on dataset characteristics.

3.6 Model Testing and Evaluation

Assessing model performance ensures robustness in detecting malicious URLs.Test data undergoes the same preprocessing (tokenization, vectorization, normalization) as during training. The trained models generate probability scores, which are thresholded for classification. Performance is analyzed via a confusion matrix, breaking down true positives, false positives, true negatives, and false negative.A high accuracy suggests effective classification, but the trade-off between precision and recall determines practical usability. A high precision ensures minimal false positives, while high recall ensures broad threat detection. The F1-score balances both aspects, guiding model refinements.

3.7 System Integration

Deploying deep learning and NLP models into a real-world malicious URL detection system involves strategic implementation. Ensemble modeling (averaging, stacking, voting) enhances performance by leveraging multiple architectures (RNN, LSTM). NLP processing integrates tokenization, word embeddings (Word2Vec, FastText), and sequence encoding (BERT) within the feature extraction pipeline. Tools and Frameworks are as follow**:**

- **Deep Learning:** TensorFlow, Keras, PyTorch.
- **NLP Processing:** NLTK, SpaCy, Hugging Face Transformers.
- **Data Manipulation:** Pandas, NumPy, Scikit-learn.
- **Visualization:** Dash (real-time monitoring), Matplotlib, Seaborn.

Traditional blacklist-based systems struggle against evolving cyber threats. Deep learning and NLP enhance detection by recognizing complex patterns. Studies (Birthriya & Jain, 2021; Yamsani et al., 2024; Tung et al., 2022) validate the effectiveness of these approaches. The chosen methodology integrates MLP, BiLSTM, and RNNs for pattern recognition, combined with tokenization and word embeddings for content analysis, ensuring robust threat detection. To addressing limitations the following were considered :

- **Data Quality:** Datasets sourced from Kaggle and PhishTank ensure diverse URL samples.
- **Model Variability:** Ensemble methods (model averaging, stacking) mitigate inconsistencies.
- **Overfitting:** Hyperparameter tuning, cross-validation, and early stopping optimize model generalization.

4. Results and Discussion

This chapter presents the development and evaluation of a deep learning and NLP-based system for malicious URL detection. It details the model training process, performance assessment, and results. Key evaluation metrics include accuracy, precision, recall, and F1 score to measure the system's effectiveness.The analysis begins with training multiple machine learning models, outlining their configurations and parameter settings. Performance comparisons highlight the classification capabilities of the deep learning models against traditional detection techniques. The results are then analyzed to assess how well the models detect malicious URLs and how these findings align with the research objectives.

4.1 Data Analysis

The dataset, **malicious_phish.csv**, contains **651,191 entries** with two columns: url and type. The type column categorizes URLs as **benign, phishing, defacement,** or **malware**. While relatively balanced, the dataset has a higher proportion of benign URLs.

Table 7 summarizes the dataset structure:

Attribute	**Details**
Total Rows	651,191
Unique URLs	641,119
Categories	4 (benign, phishing, defacement, malware)
Most Frequent URL	http://style.org.hc360.com/css/detail/mysite/s...
Frequency of Most Frequent URL	180
Top Category	Benign
Frequency of Top Category	428,103

A statistical overview of the dataset is presented in **Table 8**:

Attribute	**Description**
url	URLs categorized into different types (benign, phishing, defacement, malware).
type	Classification label for each URL.

Table 9 presents summary statistics:

Statistic	**URL Count**	**Type Count**
Total	651,191	651,191
Unique	641,119	4
Most Frequent	http://style.org.hc360.com/css/detail/mysite/s...	Benign
Frequency (Top)	180	428,103

The dataset's size and diversity provide a solid foundation for training and evaluating models for malicious URL detection.

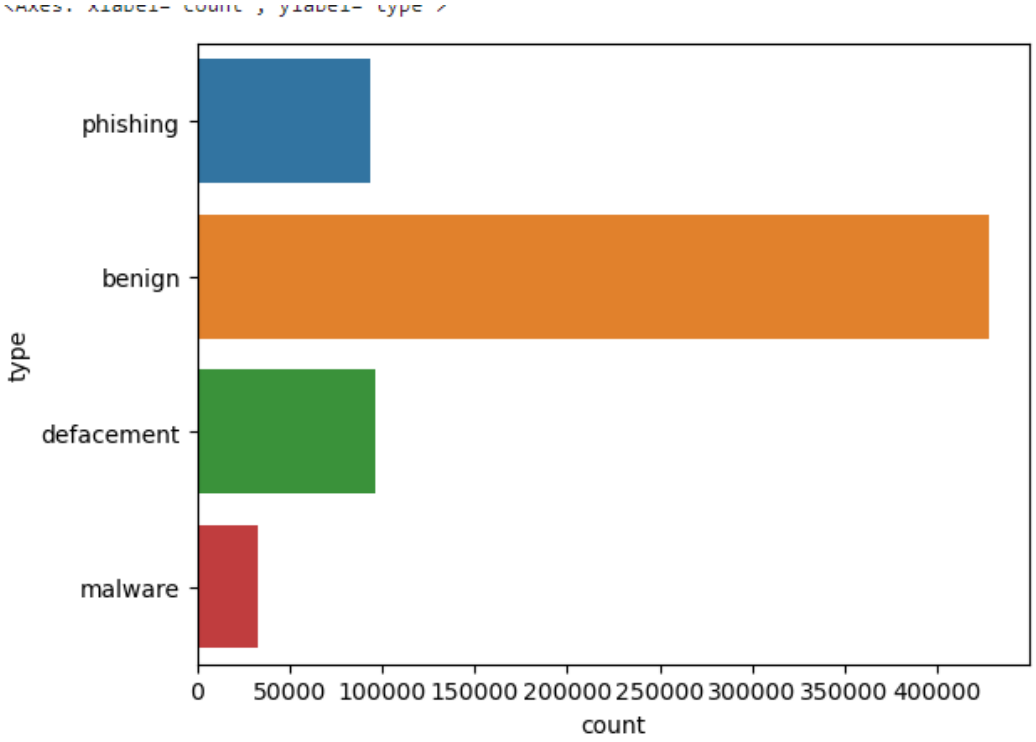

Figure 6 illustrates the distribution of unique URL types in the dataset.

4.2 Model Performance

The LSTM and BiLSTM models were trained and tested, showing performance variations across different configurations. The model was designed for binary classification using **Long Short-Term Memory (LSTM)** networks, which are well-suited for processing sequential data such as URLs.

Embedding Layer: The model starts with an Embedding layer, which converts words into dense vectors of fixed size (output_dim=100), based on the number of unique words (input_dim=len(tokenizer.word_index)+1). Spatial Dropout: The next layer, SpatialDropout1D, helps prevent overfitting by randomly setting a fraction of input units to zero during training. LSTM Layer: An LSTM layer with 100 units follows, designed to capture temporal dependencies in the input sequence. Both standard dropout and recurrent dropout are applied to reduce overfitting. Dense Layer: Finally, a Dense layer with a sigmoid activation function is added, which outputs a single value between 0 and 1, representing the probability of belonging to one of two classes. Compilation:

The model is compiled using the Adam optimizer and binary cross-entropy loss, which are standard for binary classification tasks. It also tracks accuracy during training. Label Encoding:

The code uses LabelEncoder from Scikit-learn to convert categorical labels in y_train to integer values, ensuring they can be processed by the model. Training:

The model is trained on padded sequences (X_train_pad) and the encoded labels for 5 epochs, using a batch size of 32. A validation split of 20% is applied to monitor the model's performance on unseen data during training.

For the LSTM model, the training accuracy steadily increased from 15.70% in the first epoch to 78.60% by the fifth epoch. Similarly, the validation accuracy rose from 61.97% to 74.72%. However, the loss values indicated a negative trend, with the training loss dropping significantly from 0.60 to -1021.06, and the validation loss reaching -1074.09 by the final epoch. The rapid decline in loss suggests a potential issue with the model's training process, such as an improper learning rate or gradient explosion.

Table 1 Training and Validation Epoch

Epoch	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
1	0.1570	0.6197	0.6045	-16.2649
2	0.6891	0.7069	-71.9256	-253.0174
3	0.7548	0.7354	-357.1562	-518.8596
4	0.7808	0.7397	-695.4479	-786.6046
5	0.7860	0.7472	-1021.0640	-1074.0868

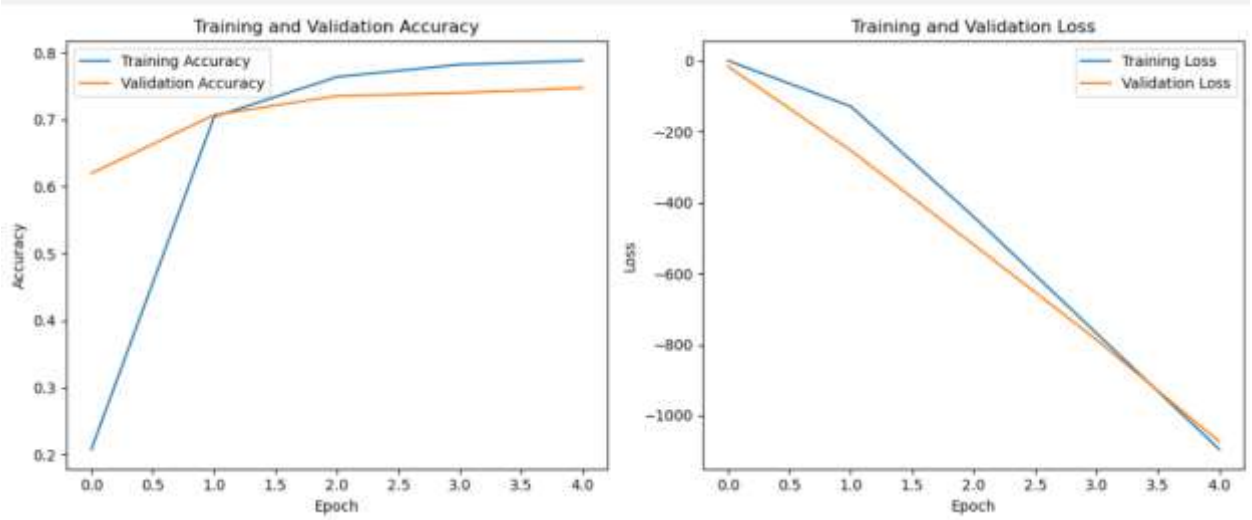

Figure 1:Training and Validation plots

The BiLSTM model exhibited stronger performance, with training accuracy starting at 86.85% and rising to 95.66% by the fifth epoch. Validation accuracy also improved significantly, beginning at 91.47% and reaching 95.27% by the end. Unlike the LSTM, the BiLSTM model showed more stable and positive loss values, indicating that the training process was more controlled and effective.

Table 2Epoch For BiLSTM

Epoch	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
1	0.8685	0.9147	0.3694	0.2306
2	0.9242	0.9396	0.2064	0.1668
3	0.9425	0.9493	0.1597	0.1423
4	0.9517	0.9535	0.1340	0.1299
5	0.9566	0.9527	0.1201	0.1313

The classification report for the BiLSTM model shows high precision, recall, and F1-scores across all classes, with an overall accuracy of 95.33%. The confusion matrix further confirms the model's effectiveness, with minimal misclassifications across the different types of URLs.

For comparison, the Multi-Layer Perceptron (MLP) model performed with an accuracy of 98.2% on the test set, with near-perfect precision, recall, and F1-scores for both benign and malware classes. This model demonstrated slightly better performance in classification tasks, particularly for the balanced dataset used.

Table 3 Summary Model Results

Model	Test Accuracy	Precision	Recall	F1-Score
LSTM	0.7472	0.95	0.93	0.94
BiLSTM	0.9533	0.95	0.95	0.95
MLP	0.982	0.99	0.98	0.98

In summary, the BiLSTM model with character vectorization showed strong performance, particularly in scenarios requiring sequential data processing. The MLP model, while simpler, outperformed the LSTM in terms of accuracy and efficiency, making it a viable option for the specific dataset used. The LSTM model,

despite improvements in accuracy, encountered issues with loss values, indicating a need for further tuning and optimization.

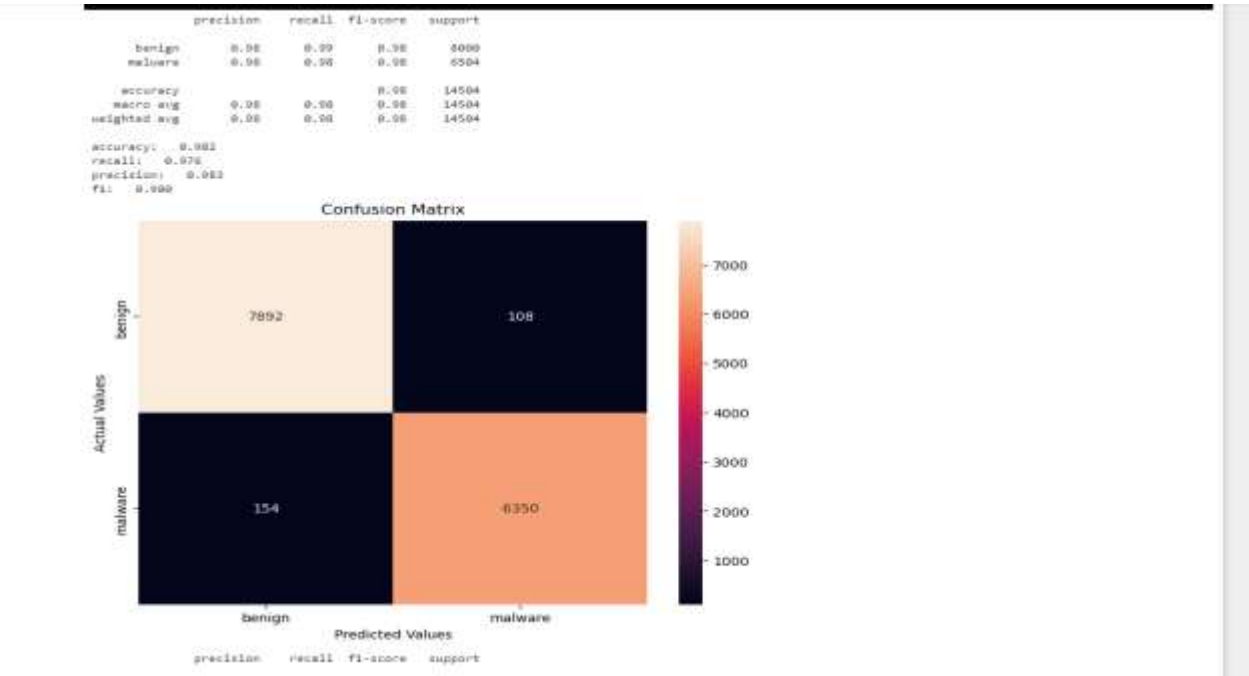

Figure 2: Confusion Matrix

Figure 3: Confusion Matrix

4.4 Discussion of Findings

The study demonstrates the effectiveness of LSTM, BiLSTM, and MLP models in detecting malicious URLs, highlighting their distinct strengths. BiLSTM outperformed LSTM in capturing contextual relationships within URLs, leading to superior classification accuracy. The MLP model, despite its simpler architecture, delivered high accuracy, challenging the assumption that complex models always yield better results. Its efficiency in handling character-level embeddings suggests that non-sequential models can be just as effective, particularly when computational efficiency is a priority.

Feature extraction played a crucial role in model performance, with analysis focusing on common malicious URL patterns, including repeated substrings, URL shortening services, and domain structures. A word cloud visualization provided an intuitive representation of prevalent elements in malicious URLs, aiding pattern recognition. The BiLSTM model's sequential processing capabilities aligned well with state-of-the-art

research, confirming its suitability for detecting sophisticated phishing and malware attacks. The MLP model's strong performance further validated the effectiveness of feedforward networks in large-scale cybersecurity applications, particularly when speed and resource efficiency are critical.

Despite these successes, certain challenges emerged. The BiLSTM model required significant computational resources, making it less viable for real-time applications where speed is paramount. While achieving high accuracy, it occasionally misclassified benign URLs due to sensitivity to unusual patterns. In contrast, the MLP model excelled in real-time monitoring, swiftly processing large datasets with minimal computational overhead. The system's modular design allows for adaptability, enabling dynamic switching between models based on specific deployment needs. These findings reinforce the potential of deep learning in cybersecurity while emphasizing the importance of balancing accuracy, efficiency, and scalability in practical applications

5. Conclusion and Future Work

This study highlights the effectiveness of deep learning models like BiLSTM and MLP in detecting malicious URLs, demonstrating their potential to enhance cybersecurity by automating threat detection. By leveraging character-level embeddings and advanced architectures, these models achieve high accuracy in identifying harmful web traffic, making them valuable tools for real-time cybersecurity applications. Their ability to adapt to new threats through continuous training further strengthens their role in protecting digital environments, particularly in industries with high-security requirements such as finance, healthcare, and government. However, challenges such as computational demands, dataset limitations, and class imbalance remain critical considerations for real-world deployment. Addressing these issues through improved data diversity, refined sampling methods, and integration with anomaly detection can enhance model robustness and generalization.

Future research should explore advanced architectures, including transformers and attention mechanisms, to improve detection capabilities and pattern recognition. Expanding datasets with more varied malicious URLs and implementing continuous retraining will help models remain effective against evolving threats. Additionally, practical deployment studies should examine seamless integration into existing cybersecurity frameworks, focusing on real-time adaptability, false positive management, and workflow efficiency. Ethical and legal considerations, including privacy regulations and responsible AI use, must also be addressed to ensure that these technologies align with cybersecurity best practices. By tackling these challenges, future work can build upon this study to develop more resilient and scalable AI-driven security solutions

References

Abad, S., Gholamy, H. and Mohammad Reza Aslani (2023). Classification of Malicious URLs Using Machine Learning. *Sensors*, 23(18), pp.7760–7760. doi:https://doi.org/10.3390/s23187760.

Afzal, S., Asim, M., Javed, A.R., Beg, M.O. and Baker, T. (2021). URLdeepDetect: A Deep Learning Approach for Detecting Malicious URLs Using Semantic Vector Models. *Journal of Network and Systems Management*, [online] 29(3). doi:https://doi.org/10.1007/s10922-021-09587-8.

Al-Selwi, S.M., Hassan, M.F., Abdulkadir, S.J., Muneer, A., Sumiea, E.H., Alqushaibi, A. and Ragab, M.G. (2024). RNN-LSTM: From applications to modeling techniques and beyond—Systematic review. *Journal of King Saud University - Computer and Information Sciences*, [online] 36(5), p.102068. doi:https://doi.org/10.1016/j.jksuci.2024.102068.

Aldakheel, E.A., Zakariah, M., Gashgari, G.A., Almarshad, F.A. and Alzahrani, A.I.A. (2023). A Deep Learning-Based Innovative Technique for Phishing Detection in Modern Security with Uniform Resource Locators. *Sensors*, 23(9), p.4403. doi:https://doi.org/10.3390/s23094403.

Aldoseri, A., Khalifa, K.N.A. - and Hamouda, A.M. (2023). Re-Thinking Data Strategy and Integration for Artificial Intelligence: Concepts, Opportunities, and Challenges. *Applied Sciences*, [online] 13(12), pp.7082–7082. doi:https://doi.org/10.3390/app13127082.

Alkhalil, Z., Hewage, C., Nawaf, L. and Khan, I. (2021). Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. *Frontiers in Computer Science*, 3(1). doi:https://doi.org/10.3389/fcomp.2021.563060.

Alraizza, A. and Algarni, A. (2023). Ransomware Detection Using Machine Learning: A Survey. *Big Data and Cognitive Computing*, [online] 7(3), p.143. doi:https://doi.org/10.3390/bdcc7030143.

AlSalem, T.S., Almaiah, M.A. and Lutfi, A. (2023). Cybersecurity Risk Analysis in the IoT: A Systematic Review. *Electronics*, [online] 12(18), p.3958. doi:https://doi.org/10.3390/electronics12183958.

Ananya Redhu, Choudhary, P., Srinivasan, K. and Tapan Kumar Das (2024a). Deep learning-powered malware detection in cyberspace: a contemporary review. *Frontiers in physics*, 12. doi:https://doi.org/10.3389/fphy.2024.1349463.

Arazzi, M., Arikkat, D., Nicolazzo, S., Nocera, A., Rehiman, R. and Conti, M. (2023). *NLP-Based Techniques for Cyber Threat Intelligence*. [online] *https://arxiv.org/pdf/2311.08807*. Available at: https://arxiv.org/pdf/2311.08807 [Accessed 6 Jul. 2024].

Aysar Weshahi, Feras Dwaik, Khouli, M., Ashqar, H.I., Amani Shatnawi and Mahmoud ElKhodr (2024b). IoT-Enhanced Malicious URL Detection Using Machine Learning. *Lecture notes on data engineering and communications technologies*, [online] 203(167), pp.470–482. doi:https://doi.org/10.1007/978-3-031-57931-8_45.

Balasubramaniam, N., Kauppinen, M., Hiekkanen, K. and Kujala, S. (2022). Transparency and Explainability of AI Systems: Ethical Guidelines in Practice. *Requirements Engineering: Foundation for Software Quality*, pp.3–18. doi:https://doi.org/10.1007/978-3-030-98464-9_1.

Barnard, J. (2024). *What are Word Embeddings? | IBM*. [online] www.ibm.com. Available at: https://www.ibm.com/topics/word-embeddings.

Beaman, C., Barkworth, A., Akande, T.D., Hakak, S. and Khan, M.K. (2021). Ransomware: Recent advances, analysis, challenges and future research directions. *Computers & Security*, [online] 111(1). doi:https://doi.org/10.1016/j.cose.2021.102490.

Bederna, Z. and Rajnai, Z. (2022). Analysis of the cybersecurity ecosystem in the European Union. *International Cybersecurity Law Review*, [online] 657(161). doi:https://doi.org/10.1365/s43439-022-00048-9.

Bell, S. and Komisarczuk, P. (2020). An Analysis of Phishing Blacklists: Google Safe Browsing, OpenPhish, and PhishTank. *Proceedings of the Australasian Computer Science Week Multiconference*. doi:https://doi.org/10.1145/3373017.3373020.

Benavides-Astudillo, E., Fuertes, W., Sanchez-Gordon, S., Nuñez-Agurto, D. and Rodríguez-Galán, G. (2023). A Phishing-Attack-Detection Model Using Natural Language Processing and Deep Learning. *Applied Sciences*, 13(9), p.5275. doi:https://doi.org/10.3390/app13095275.

BlackBerry (2024). *What Is SEO Poisoning?* [online] Blackberry.com. Available at: https://www.blackberry.com/us/en/solutions/endpoint-security/ransomware-protection/seo-poisoning [Accessed 20 Aug. 2024].

Burbela, K. (2023). *Model of detection of phishing URLs based on machine learning*. [online] Available at: https://www.diva-portal.org/smash/get/diva2:1773760/FULLTEXT02.

Carley, K.M. (2020). Social cybersecurity: an emerging science. *Computational and Mathematical Organization Theory*, [online] 26(4), pp.365–381. doi:https://doi.org/10.1007/s10588-020-09322-9.

Chaithanya, B.N. and Brahmananda, S.H. (2021). Detecting Ransomware Attacks Distribution Through Phishing URLs Using Machine Learning. *Lecture notes on data engineering and communications technologies*, 1290(23), pp.821–832. doi:https://doi.org/10.1007/978-981-16-3728-5_61.

Charmet, F., Tanuwidjaja, H.C., Ayoubi, S., Gimenez, P.-F., Han, Y., Jmila, H., Blanc, G., Takahashi, T. and Zhang, Z. (2022). Explainable artificial intelligence for cybersecurity: a literature survey. *Annals of Telecommunications*, 11(231). doi:https://doi.org/10.1007/s12243-022-00926-7.

Chaudhari, S., Thakur, A. and Rajan, A. (2024). An Efficient Malicious URL Detection Approach Using Machine Learning Techniques. *Lecture notes in electrical engineering*, [online] 789(112), pp.485–495. doi:https://doi.org/10.1007/978-981-99-7077-3_48.

Cheah, P. and Fellows, C. (2023). *Multi-Layer Perceptron Neural Network for Improving Detection Performance of Malicious Phishing URLs Without Affecting Other Attack Types Classification*. [online] Available at: https://arxiv.org/pdf/2203.00774.

Coyac-Torres, J.E., Sidorov, G., Aguirre-Anaya, E. and Hernández-Oregón, G. (2023). Cyberattack Detection in Social Network Messages Based on Convolutional Neural Networks and NLP Techniques. *Machine Learning and Knowledge Extraction*, [online] 5(3), pp.1132–1148. doi:https://doi.org/10.3390/make5030058.

Dong, H., Dong, J., Yuan, S. and Guan, Z. (2023). Adversarial Attack and Défense on Natural Language Processing in Deep Learning: A Survey and Perspective. *Lecture Notes in Computer Science*, 11(4), pp.409–424. doi:https://doi.org/10.1007/978-3-031-20096-0_31.

Ejaz, A., Mian, A.N. and Manzoor, S. (2023). Life-long phishing attack detection using continual learning. *Scientific Reports*, [online] 13(1), pp.1–14. doi:https://doi.org/10.1038/s41598-023-37552-9.

Ekah, U.J. and Emeruwa, C. (2022). Penetration Depth Analysis of UMTS Networks Using Received Signal Code Power. *Journal of Engineering Research and Reports*, 25(12), pp.16–25. doi:https://doi.org/10.9734/jerr/2022/v23i7732.

Engage with IBM IBM TechXChange (2024). *What is named entity recognition? | IBM*. [online] www.ibm.com. Available at: https://www.ibm.com/topics/named-entity-recognition.

Gardiner, J., Cova, M. and Nagaraja, S. (2014). *Command & Control Understanding, Denying and Detecting*. [online] *In collaboration with Lastline, Inc*. Available at: https://arxiv.org/pdf/1408.1136.

Garg, R., Gupta, A. and Srivastava, A. (2024a). A Comprehensive Review on Transforming Security and Privacy with NLP. *Lecture notes in networks and systems*, 1290(1359), pp.147–159. doi:https://doi.org/10.1007/978-981-97-0641-9_10.

Gaspar, D., Silva, P. and Silva, C. (2024). Explainable AI for Intrusion Detection Systems: LIME and SHAP Applicability on Multi-Layer Perceptron. *IEEE Access*, [online] 12, pp.1–1. doi:https://doi.org/10.1109/access.2024.3368377.

Ghaleb, F.A., Alsaedi, M., Saeed, F., Ahmad, J. and Alasli, M. (2022). Cyber Threat Intelligence-Based Malicious URL Detection Model Using Ensemble Learning. *Sensors*, [online] 22(9), p.3373. doi:https://doi.org/10.3390/s22093373.

Ghanem, B., Rosso, P. and Rangel, F. (2018). An Emotional Analysis of False Information in Social Media and News Articles. *Association for Computing Machinery*, [online] 1(1). doi:https://doi.org/10.1145/1122445.1122456.

Ghosh, R. and Soumen Kanrar (2023a). Malware Analysis Based on Malicious Web URLs. *Lecture notes in networks and systems*, 738, pp.265–278. doi:https://doi.org/10.1007/978-981-99-4433-0_23.

Gibert, D., Planes, J., Mateu, C. and Le, Q. (2022). Fusing feature engineering and deep learning: A case study for malware classification. *Expert Systems with Applications*, [online] 207, p.117957. doi:https://doi.org/10.1016/j.eswa.2022.117957.

Gold, N., Udeh, A., Adaga, M., DaraOjimba, D. and Osato, N. (2024). ETHICAL CONSIDERATIONS IN DATA COLLECTION AND ANALYSIS: A REVIEW: INVESTIGATING ETHICAL PRACTICES AND CHALLENGES IN MODERN DATA COLLECTION AND ANALYSIS. *International journal of applied research in social sciences*, [online] 6(1), pp.1–22. doi:https://doi.org/10.51594/ijarss.v6i1.688.

Gopali, S., Namin, A., Abri, F. and Jones, K. (2024). The Performance of Sequential Deep Learning Models in Detecting Phishing Websites Using Contextual Features of URLs. *ACM Reference Format*, [online] 157. doi:https://doi.org/10.1145/3605098.3636164.

Gupta, B.B. and Jain, A.K. (2020). Phishing Attack Detection using a Search Engine and Heuristics-based Technique. *Journal of Information Technology Research*, 13(2), pp.94–109. doi:https://doi.org/10.4018/jitr.2020040106.

Hassija, V., Chamola, V., Mahapatra, A., Singal, A., Goel, D., Huang, K., Scardapane, S., Spinelli, I., Mahmud, M. and Hussain, A. (2023). Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence. *Cognitive Computation*, [online] 16(14). doi:https://doi.org/10.1007/s12559-023-10179-8.

Herath, T.B.G., Khanna, P. and Ahmed, M. (2022). Cybersecurity Practices for Social Media Users: A Systematic Literature Review. *Journal of Cybersecurity and Privacy*, [online] 2(1), pp.1–18. doi:https://doi.org/10.3390/jcp2010001.

Hnamte, V. and Hussain, J. (2023). DCNNBiLSTM: An Efficient Hybrid Deep Learning-Based Intrusion Detection System. *Telematics and Informatics Reports*, p.100053. doi:https://doi.org/10.1016/j.teler.2023.100053.

Holdsworth, J. (2024). *What Is Natural Language Processing?* [online] IBM. Available at: https://www.ibm.com/topics/natural-language-processing.

Islam, R., Islam, M., Afrin, M., Antara, A., Tabassum, N. and Amin, A. (2024). *PhishGuard: A Convolutional Neural Network-Based Model for Detecting Phishing URLs with Explainability Analysis*. [online] Available at: https://arxiv.org/pdf/2404.17960v1 [Accessed 9 Jul. 2024].

Jamin Rahman Jim, Apon, M., Partha Malakar, Md Mohsin Kabir, Nur, K. and M.F. Mridha (2024). Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review. *Natural Language Processing Journal*, 6(98), pp.100059–100059. doi:https://doi.org/10.1016/j.nlp.2024.100059.

Jia, J. and Liang, W. (2023). *A Review of Hybrid and Ensemble in Deep Learning for Natural Language Processing*. [online] arxiv.org. Available at: https://arxiv.org/html/2312.05589v1 [Accessed 2 Jul. 2024].

Jiang, J., Chen, J., Choo, K.-K.R., Liu, C., Liu, K., Yu, M. and Wang, Y. (2018). A Deep Learning Based Online Malicious URL and DNS Detection Scheme. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp.438–448. doi:https://doi.org/10.1007/978-3-319-78813-5_22.

kaggle.com (n.d.). *Malicious URLs dataset*. [online] www.kaggle.com. Available at: https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset.

Kasongo, S.M. (2022). A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. *Computer Communications*, 43(21). doi:https://doi.org/10.1016/j.comcom.2022.12.010.

Kaspersky (2019). *What Is a Drive by Download*. [online] Kaspersky.com. Available at: https://www.kaspersky.com/resource-center/definitions/drive-by-download.

Kaur, R., Gabrijelčič, D. and Klobučar, T. (2023). Artificial Intelligence for Cybersecurity: Literature Review and Future Research Directions. *Information Fusion*, [online] 97(101804), p.101804. doi:https://doi.org/10.1016/j.inffus.2023.101804.

Khan, W., Daud, A., Khan, K., Muhammad, S. and Haq, R. (2023a). Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and emerging trends. *Natural Language Processing Journal*, [online] 4(1000026), p.100026. doi:https://doi.org/10.1016/j.nlp.2023.100026.

Khraisat, A., Gondal, I., Vamplew, P. and Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, [online] 2(1), pp.1–22. doi:https://doi.org/10.1186/s42400-019-0038-7.

Khurana, D., Koli, A., Khatter, K. and Singh, S. (2022). Natural Language processing: State of the art, Current Trends and Challenges. *Multimedia Tools and Applications*, 82(3), pp.3713–3744. doi:https://doi.org/10.1007/s11042-022-13428-4.

Kosinski, M. (2024). *What is phishing? | IBM*. [online] www.ibm.com. Available at: https://www.ibm.com/topics/phishing.

Krishna Teja Chitty-Venkata, Mittal, S., Murali Emani, Vishwanath, V. and Somani, A.K. (2023). A survey of techniques for optimizing transformer inference. *Journal of Systems Architecture*, 144, pp.102990–102990. doi:https://doi.org/10.1016/j.sysarc.2023.102990.

Kumi, S., Lim, C. and Lee, S.-G. (2021). Malicious URL Detection Based on Associative Classification. *Entropy*, 23(2), p.182. doi:https://doi.org/10.3390/e23020182.

Le, H., Pham, Q., Sahoo, D. and Hoi, S. (2018a). *URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection*. [online] Available at: https://arxiv.org/pdf/1802.03162 [Accessed 20 Aug. 2024].

Liang, Y., Wang, Q., Xiong, K., Zheng, X., Yu, Z. and Zeng, D. (2021). Robust Detection of Malicious URLs with Self-Paced Wide & Deep Learning. *IEEE Transactions on Dependable and Secure Computing*, [online] 19(2), pp.717–730. doi:https://doi.org/10.1109/tdsc.2021.3121388.

Lin, T., Wang, Y., Liu, X. and Qiu, X. (2022). A survey of transformers. *AI Open*, 3(23). doi:https://doi.org/10.1016/j.aiopen.2022.10.001.

Liu, K. and Zhang, J. (2021). *A Dual-Layer Attention-Based LSTM Network for Fed-batch Fermentation Process Modelling*. [online] ScienceDirect. Available at: https://www.sciencedirect.com/science/article/abs/pii/B9780323885065500863.

Lohachab, A. and Karambir, B. (2018). Critical Analysis of DDoS—An Emerging Security Threat over IoT Networks. *Journal of Communications and Information Networks*, 3(3), pp.57–78. doi:https://doi.org/10.1007/s41650-018-0022-5.

Mamun, M.S.I., Rathore, M.A., Lashkari, A.H., Stakhanova, N. and Ghorbani, A.A. (2016a). Detecting Malicious URLs Using Lexical Analysis. *Network and System Security*, [online] 23(214), pp.467–482. doi:https://doi.org/10.1007/978-3-319-46298-1_30.

Manakitsa, N., Maraslidis, G.S., Moysis, L. and Fragulis, G.F. (2024). A Review of Machine Learning and Deep Learning for Object Detection, Semantic Segmentation, and Human Action Recognition in Machine and Robotic Vision. *Technologies*, [online] 12(2), p.15. doi:https://doi.org/10.3390/technologies12020015.

Marchal , S. (2014). *PhishStorm - phishing / legitimate URL dataset*. [online] Aalto University's research portal. Available at: https://research.aalto.fi/en/datasets/phishstorm-phishing-legitimate-url-dataset [Accessed 9 Jul. 2024].

McKinsey & Company (2022). *Cybersecurity trends: Looking over the horizon | McKinsey*. [online] www.mckinsey.com. Available at: https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/cybersecurity/cybersecurity-trends-looking-over-the-horizon [Accessed 18 Jun. 2024].

Mikołajczyk-Bareła, A. and Grochowski, M. (2023). *A survey on bias in machine learning research*. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.2308.11254.

Mittal, A., Engels, D., Kommanapalli, H., Sivaraman, R., Chowdhury, T. and Chowdhury, T. (2022). Phishing Detection Using Natural Language Processing and Machine Learning. *SMU Data Science Review*, [online] 6(2), p.14. Available at: https://scholar.smu.edu/cgi/viewcontent.cgi?article=1215&context=datasciencereview.

Mohammad, R., Saeed, F., Almazroi, A.A., Alsubaei, F.S. and Almazroi, A.A. (2024). Enhancing Intrusion Detection Systems Using a Deep Learning and Data Augmentation Approach. *Systems*, [online] 12(3), p.79. doi:https://doi.org/10.3390/systems12030079.

Murel, J. (2024). *IBM Developer*. [online] developer.ibm.com. Available at: https://developer.ibm.com/tutorials/awb-tokenizing-text-in-python/.

M. Vasek and T. Moore, "Empirical analysis of factors affecting malware URL detection," 2013 APWG eCrime Researchers Summit, San Francisco, CA, USA, 2013, pp. 1-8, doi: https://doi.org/10.1109/eCRS.2013.6805776. keywords: {Malware;IP networks;Indexes},

Nagendar Yamsani, K. Sarada, Mohammed Abbas Ahmed and K. Saikumar (2024). Estimate and prevention of malicious URL using logistic regression ML techniques. *AIP conference proceedings*, [online] 2919(1). doi:https://doi.org/10.1063/5.0190584.

Najwa Altwaijry, Isra Al-Turaiki, Alotaibi, R. and Alakeel, F. (2024). Advancing Phishing Email Detection: A Comparative Study of Deep Learning Models. *Sensors*, 24(7), pp.2077–2077. doi:https://doi.org/10.3390/s24072077.

nltk.org (2023). *NLTK :: nltk.tag.pos_tag*. [online] Nltk.org. Available at: https://www.nltk.org/api/nltk.tag.pos_tag [Accessed 20 Aug. 2024].

Ogu, E.C., Ojesanmi, O.A., Awodele, O. and Kuyoro, S. (2019). A Botnets Circumspection: The Current Threat Landscape, and What We Know So Far. *Information*, 10(11), p.337. doi:https://doi.org/10.3390/info10110337.

Opara, C., Chen, Y. and Wei, B. (2024). Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics. *Expert Systems with Applications*, [online] 236, p.121183. doi:https://doi.org/10.1016/j.eswa.2023.121183.

Orozco-Fonseca, D., Marín, G. and Lara, A. (2024). Taxonomy of Malicious URL Detection Techniques. *Lecture notes in networks and systems*, [online] 7(23), pp.73–81. doi:https://doi.org/10.1007/978-3-031-54235-0_7.

Othmane Niyaoui and Oussama Mohamed Reda (2024a). Malicious URL Detection Using Transformers' NLP Models and Machine Learning. *Lecture notes in networks and systems*, 13(9), pp.389–399. doi:https://doi.org/10.1007/978-3-031-54318-0_35.

Ozcan, A., Catal, C., Donmez, E. and Senturk, B. (2021). A hybrid DNN–LSTM model for detecting phishing URLs. *Neural Computing and Applications*, 132(109). doi:https://doi.org/10.1007/s00521-021-06401-z.

Özkent, Y. (2022). Social media usage to share information in communication journals: An analysis of social media activity and article citations. *PLoS ONE*, [online] 17(2). doi:https://doi.org/10.1371/journal.pone.0263725.

phishtank (n.d.). *PhishTank | Join the fight against phishing*. [online] www.https://phishtank.org. Available at: https://phishtank.org/ [Accessed 9 Jul. 2024].

R, vinayakumar, S, S., KP, S. and Alazab, M. (2020). Malicious URL Detection using Deep Learning. *TechRxiv* |, 64(231). doi:https://doi.org/10.36227/techrxiv.11492622.v1.

Rao, R.S. and Pais, A.R. (2017). An Enhanced Blacklist Method to Detect Phishing Websites. *Information Systems Security*, [online] 17(29), pp.323–333. doi:https://doi.org/10.1007/978-3-319-72598-7_20.

Saeed, S., Suayyid, S.A., Al-Ghamdi, M.S., Al-Muhaisen, H. and Almuhaideb, A.M. (2023). A Systematic Literature Review on Cyber Threat Intelligence for Organizational Cybersecurity Resilience. *Sensors*, [online] 23(16), p.7273. doi:https://doi.org/10.3390/s23167273.

Santosh Kumar Birthriya and Ankit Kumar Jain (2021). Analysis for Malicious URLs Using Machine Learning and Deep Learning Approaches. *Algorithms for intelligent systems*, 21(8), pp.797–807. doi:https://doi.org/10.1007/978-981-15-7533-4_63.

Sarker, I.H. (2021a). Deep Cybersecurity: A Comprehensive Overview from Neural Network and Deep Learning Perspective. *SN Computer Science*, 2(3). doi:https://doi.org/10.1007/s42979-021-00535-6.

Shams Forruque Ahmed, Bin, S., Hassan, M., Mahtabin Rodela Rozbu, Taoseef Ishtiak, Rafa, N., M. Mofijur, Ali and Gandomi, A.H. (2023a). Deep learning modelling techniques: current progress, applications, advantages, and challenges. *Artificial Intelligence Review*, [online] 56(112). doi:https://doi.org/10.1007/s10462-023-10466-8.

Silva, C.M.R. da, Feitosa, E.L. and Garcia, V.C. (2020). Heuristic-based strategy for Phishing prediction: A survey of URL-based approach. *Computers & Security*, [online] 88, p.101613. doi:https://doi.org/10.1016/j.cose.2019.101613.

Souppaya, M. and Scarfone, K. (2013). Guide to Malware Incident Prevention and Handling for Desktops and Laptops. *Guide to Malware Incident Prevention and Handling for Desktops and Laptops*, [online] 1. doi:https://doi.org/10.6028/nist.sp.800-83r1.

Su, M. and Su, K.-L. (2023a). BERT-Based Approaches to Identifying Malicious URLs. *Sensors*, [online] 23(20), pp.8499–8499. doi:https://doi.org/10.3390/s23208499.

Sufi, F. (2024). An innovative GPT-based open-source intelligence using historical cyber incident reports. *Natural language processing journal*, pp.100074–100074. doi:https://doi.org/10.1016/j.nlp.2024.100074.

Sun, B., Takahashi, T., Zhu, L. and Mori, T. (2020). Discovering Malicious URLs Using Machine Learning Techniques. *Intelligent systems reference library*, 15(8), pp.33–60. doi:https://doi.org/10.1007/978-3-030-38788-4_3.

Taye, M.M. (2023). Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions. *Computers*, [online] 12(5), pp.91–91. doi:https://doi.org/10.3390/computers12050091.

The MITRE Corporation. MITRE ATT&CK and ATT&CK (2023). *Stage Capabilities: SEO Poisoning, Sub-technique T1608.006 - Enterprise | MITRE ATT&CK®*. [online] attack.mitre.org. Available at: https://attack.mitre.org/techniques/T1608/006/.

Tung, S.P., Wong, K.Y., Kuzminykh, I., Bakhshi, T. and Ghita, B. (2022). Using a Machine Learning Model for Malicious URL Type Detection. *Lecture Notes in Computer Science*, 13158(4), pp.493–505. doi:https://doi.org/10.1007/978-3-030-97777-1_41

Tyagi, N. and Bhushan, B. (2023). Demystifying the Role of Natural Language Processing (NLP) in Smart City Applications: Background, Motivation, Recent Advances, and Future Research Directions. *Wireless Personal Communications*, 107(31). doi:https://doi.org/10.1007/s11277-023-10312-8

Viswan Vimbi, Noushath Shaffi and Mahmud, M. (2024). Interpreting artificial intelligence models: a systematic review on the application of LIME and SHAP in Alzheimer's disease detection. *Brain informatics*, [online] 11(1). doi:https://doi.org/10.1186/s40708-024-00222-1.

Wang, H., Singhal, A. and Liu, P. (2023). Tackling imbalanced data in cybersecurity with transfer learning: a case with ROP payload detection. *Cybersecurity*, [online] 6(1). doi:https://doi.org/10.1186/s42400-022-00135-8.

Xu, Y., Wang, Q., An, Z., Wang, F., Zhang, L., Wu, Y., Dong, F., Qiu, C.-W., Liu, X., Qiu, J., Hua, K., Su, W., Xu, H., Han, Y., Cao, X., Liu, E., Fu, C., Yin, Z., Liu, M. and Roepman, R. (2021). Artificial Intelligence: a Powerful Paradigm for Scientific Research. *The Innovation*, [online] 2(4). doi:https://doi.org/10.1016/j.xinn.2021.100179.

Young, T., Hazarika, D., Poria, S. and Cambria, E. (2017). *Recent Trends in Deep Learning Based Natural Language Processing*. [online] arXiv.org. Available at: https://arxiv.org/abs/1708.02709 [Accessed 2 Jul. 2024].

Yu, B., Tang, F., Daji Ergu, Zeng, R., Ma, B. and Liu, F. (2024a). Efficient Classification of Malicious URLs: M-BERT - A Modified BERT Variant for Enhanced Semantic Understanding. *IEEE Access*, 134(23), pp.1–1. doi:https://doi.org/10.1109/access.2024.3357095.

Zhao, Z., Alzubaidi, L., Zhang, J., Duan, Y. and Gu, Y. (2024). A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations. *Expert Systems with Applications*, [online] 242(122807), p.122807. doi:https://doi.org/10.1016/j.eswa.2023.122807.

Zhou, M., Duan, N., Liu, S. and Shum, H.-Y. (2020). Progress in Neural NLP: Modeling, Learning, and Reasoning. *Engineering*, 6(3), pp.275–290. doi:https://doi.org/10.1016/j.eng.2019.12.014.

Zhu, S., Huang, Y., Huang, L., Li, S. and He, P. (2024). URL Detection Based on Quantum Long Short-Term Memory Neural Network. *Scholars Journal of Engineering and Technology*, [online] 12(05), pp.151–155. doi:https://doi.org/10.36347/sjet.2024.v12i05.001.

Ziems, N. and Wu, S. (2021). *Security Vulnerability Detection Using Deep Learning Natural Language Processing*. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.2105.02388.