National College of Ireland

Implementing a Hybrid System for Accurately Detecting Phishing URLs with Machine Learning and Deep Learning Techniques

MSc Data Analytics
Research Project

Preetham Madeti
22142258

School of Computing
National College of Ireland

Supervisor: Abdul Qayum

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Preetham Madeti		
Student ID:	22142258		
Programme:	MSc Data Analytics	**Year:**	2023/2024
Module:	Research Project		
Supervisor:	Mr. Abdul Qayum		
Submission Due Date:	31/01/2024		
Project Title:	Implementing a Hybrid System for Accurately Detecting Phishing URLs with Machine Learning and Deep Learning Techniques.		
Word Count:	7076	**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: Preetham Madeti

Date: 29/01/2024

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):	

Implementing a Hybrid System for Accurately Detecting Phishing URLs with Machine Learning and Deep Learning Techniques

Preetham Madeti
22142258

Abstract

Phishing attacks, especially through deceptive URLs, present a critical challenge in cybersecurity. This study embarks on addressing the increasing complexity of phishing attacks through the lens of advanced computational strategies. At its core, it examines the efficacy of combining machine learning (ML) and deep learning (DL) techniques to enhance the detection of phishing URLs. The primary aim is to surpass the capabilities of traditional detection methods, which are often outpaced by the evolving sophistication of phishing tactics. To this end, the research develops and assesses an array of hybrid models. These models integrate various algorithms, including AdaBoost, Random Forest, Gaussian Naïve Bayes, Decision Trees, and Multi-layer Perceptron (MLP), each chosen for their distinct strengths in data analysis and pattern recognition. A notable outcome of this research is the superior performance of the Random Forest-MLP hybrid model. It exhibits an impressive accuracy of 88%, striking an optimal balance between swift training (768.395 seconds) and quick response in testing phases (0.575 seconds), marking it as a robust solution for real-time threat detection. Other models like AdaBoost-MLP and Gaussian Naïve Bayes-MLP also show commendable accuracy, around 86%, albeit with variations in training and testing durations.

The implications of these findings are substantial for the field of cybersecurity. They highlight the versatility and heightened effectiveness of hybrid approaches in countering phishing URLs. This research not only contributes valuable insights to academic discourse but also paves the way for practical advancements in cybersecurity measures, advocating for innovative strategies in the ongoing battle against digital phishing threats.

1 Introduction

In the digital era, the exponential growth of online platforms and services has been paralleled by an increase in cyber threats, with phishing attacks being one of the most pernicious. Phishing attacks, particularly those using deceptive URLs, are a major concern for internet users and organizations worldwide. These attacks aim to fraudulently acquire sensitive information by masquerading as trustworthy entities, and their increasing sophistication poses a significant challenge to cybersecurity. The traditional methods for detecting phishing URLs, such as blacklisting and heuristic-based approaches, are becoming increasingly ineffective. As cybercriminals continuously evolve their tactics, these conventional methods struggle to keep up, often failing to detect new or sophisticated phishing attacks. This has led to substantial financial and data losses (Abdul Samad et al., 2023) underscoring the urgent need for more effective detection systems. Recognizing the limitations of existing methodologies, this research project proposes a novel approach: the implementation of a hybrid system that combines machine learning (ML) and deep learning (DL) techniques to detect phishing URLs accurately and efficiently.

This project is driven by the research question: "Which is the most effective combination of algorithms that can detect phishing URLs with the highest precision and with the least training and testing time?" The motivation behind this research is twofold. First, it aims to address the growing

sophistication of phishing attacks by developing a system that can adapt to and recognize new patterns more effectively than existing methods. Second, it seeks to contribute to the broader field of cybersecurity by exploring the potential of hybrid AI systems in threat detection, an area that is currently underexplored in academic literature.

In our current digital age, the remarkable growth of internet-based platforms and services has been shadowed by a corresponding rise in cyber threats, with phishing attacks standing out as a particularly menacing challenge. These attacks typically use misleading URLs to deceive internet users and organizations around the globe, attempting to illicitly gather confidential information by posing as legitimate entities. The increasing complexity of these schemes has significantly amplified the difficulties faced in cybersecurity efforts (Jalil et al., 2023). Conventional strategies for identifying phishing URLs, such as compiling blacklists or applying heuristic methods, are finding it harder to cope with these advanced tactics. The resulting increase in cyber threats has led to notable financial and informational losses, underscoring the critical need for more sophisticated systems for detecting such threats (Abdul Samad et al., 2023).

In response to this escalating issue, our research introduces a groundbreaking method: the fusion of machine learning (ML) and deep learning (DL) to create a composite system adept at both effectively and accurately identifying phishing URLs. This innovative method aims to combine the self-learning attribute of deep learning for feature detection with the established pattern recognition capabilities of machine learning. This could offer a significant advantage over traditional detection methods in identifying advanced phishing schemes (Tajaddodianfar et al., 2020).

Research Objectives:

- To implement machine learning algorithms such as AdaBoost, random forest, Gaussian Naïve Bayes, and Decision Tree.
- To implement deep learning algorithms such as MLP classifier with various optimizers.
- To design and implement a Hybrid Model that integrates multiple machine learning and deep learning methodologies.
- To evaluate the performance of both machine learning and deep learning algorithms in terms of accuracy and precision.
- Compared the Training and Testing Time to identify the best combination of algorithms to detect phishing attacks.
- To evaluate the effectiveness of the hybrid system against current phishing threats.

Research Question:

- which is the effective combination of algorithms that detect phishing URLs with highest precision and with least training and testing time?
- What are the performance differences between the proposed hybrid system and existing detection methods?

The structure of this research paper is methodically organized for comprehensive understanding and clarity. Section 2 offers an in-depth review of the current state of phishing URL detection, particularly focusing on the roles of machine learning and deep learning in enhancing cybersecurity measures. Section 3 delves into a critical analysis of related works, highlighting the strengths and limitations of existing methodologies and setting the stage for the need for this research. Section 4 details the methodology adopted, explaining the integration of machine learning and deep learning techniques in our proposed hybrid system. In Section 5, the architecture and process flow of the hybrid model are meticulously outlined, showcasing the innovative approaches and strategies employed in this study. This structured approach ensures a thorough understanding of the research's context, methodology, and the innovative aspects of the proposed hybrid system for phishing URL detection.

2 Related Work

2.1 Recent research using machine learning and feature selection algorithms.

(Jalil et al., 2023) proposed a machine learning system for accurately predicting phishing URLs. Using features like URL entropy and brand name matching, their model, applied on various datasets including Kaggle, showed Random Forest achieving the highest accuracy rates, peaking at 96.25%. However, the study didn't explore the system's adaptability to new phishing strategies.

(Abdul Samad et al., 2023) focused on improving machine learning models for phishing detection. Their approach, involving data balancing and hyperparameter optimization, utilized datasets from UCI and Mendeley. Results showed significant accuracy improvements, especially with Random Forest and XGB models. Fig 1 shows the results of the datasets with accuracy of every Model. However, the impact of these tuning elements on real-world applicability remains unexplored.

Table 16. Dataset-1 performance (accuracy %) comparison.

Classification Algorithm	Untuned	Balanced Dataset	Hyper-parameter Tuned	Feature Selection
Logistic Regression (LogR)	92.537	92.261	92.374	92.33
Support Vector Machine (SVM)	94.744	94.949	96.898	97.04
Bernoulli Naïve Bayes (BNB)	90.285	90.596	90.531	91.27
K-Neighbors Classifier (KNN)	94.708	94.924	96.832	97.04
Decision Tree (DT)	96.182	96.694	96.475	96.690
Random Forest (RF)	97.223	97.425	97.466	97.440
Gradient Boosting (GB)	94.699	94.713	97.076	97.47
Extreme Gradient Boosting (XGB)	97.286	97.028	97.182	97.260

Table 17. Dataset-2 performance (accuracy %) comparison.

Classification Algorithm	Untuned	Hyper-Parameter Tuned	Feature Selection
Logistic Regression (LogR)	93.01	93.82	93.88
Support Vector Machine (SVM)	94.48	96.66	96.79
Gaussian Naïve Bayes (GNB)	83.73	84.4	86.99
K-Neighbors Classifier (KNN)	93.63	94.73	95.41
Decision Tree (DT)	95.57	94.52	96.17
Random Forest (RF)	97.62	97.74	97.79
Gradient Boosting (GB)	97.25	97.98	98.27
Extreme Gradient Boosting (XGB)	98.15	98.26	98.21

Fig 1: Results of Abdul et al. (2023)

The research (Karim et al., 2023) employed a hybrid LSD model combining LR, SVC, and DT for phishing URL detection. The model, which used canopy feature selection and Grid Search Hyperparameter Optimization, demonstrated superior outcomes compared to other models. Fig 2 shows the performance on a Specific Task. The study's limitation lies in its lack of testing the model in dynamically changing phishing environments.

Models	Accuracy	Precision	Recall	Specificity	F1-score
Linear Regression	58.83	100	26.37	100	41.74
Decision Tree	95.41	95.8	96	94.66	95.91
Random Forest	96.77	96.73	97.51	95.83	97.12
Naive Bayes	88.39	94.92	83.71	94.32	88.96
Support Vector Machine	71.8	96.34	49.81	97.606	65.67
Gradient Boosting Machine	70.34	99.65	47.24	99.79	64.1
Hybrid (LR+SVC+DT) soft	95.23	95.15	96.38	93.77	95.77
Hybrid (LR+SVC+DT) hard	94.09	93.31	96.33	91.25	94.79
Proposed approach	98.12	97.31	96.33	96.55	95.89

Fig 2: Results of Karim et al. (2023)

In (Isarhan et al., 2023), introduced a method for detecting phishing URLs using algorithms like j48 and Naïve Bayes. Their approach, though effective in identifying risks, lacked in-depth analysis of the algorithms' performance in differentiating sophisticated phishing tactics.

In this study, Bu et al, (2023) developed the DPN network for phishing detection, focusing on disentangled URL prototypes. Their method showed high accuracy, especially in scenarios with limited data. However, the model's effectiveness in real-time phishing detection scenarios was not addressed.

(Kumar et al., 2023) developed an innovative machine learning technique to identify phishing URLs in TLS 1.2 and 1.3 encrypted traffic, eliminating the need for decryption. Their method, based on analyzing transport layer characteristics, effectively differentiated between authentic and phishing URLs. The study produced high accuracy rates, with the Light GBM model achieving a notable 95.40% accuracy. However, the research did not thoroughly address the model's performance across varied network conditions, a crucial aspect for real-world application.

(Ripa et al., 2021) conducted a comprehensive study on phishing attacks, focusing on their prevalence and various forms such as URL, email, and website attacks. They noted an increased targeting of users on social media and online gaming platforms. Their research utilized machine learning, including a Twitter spear phishing bot, to identify phishing threats. They tested various classifiers, finding XGBoost most effective for URLs, Naïve Bayes for emails, and Random Forest for websites, with accuracies of 94.44%, 95.15%, and 96.80% respectively. However, the study did not address real-world implementation challenges of these machine learning solutions.

(Chawla 2022) focused on detecting phishing websites by analyzing specific characteristics common to such sites. The study employed a range of models, including RF, DT, LR, KNN, ANN, and a Max Vote Classifier combining RF, ANN, and KNN. It achieved notable accuracy, particularly with the Max Vote Classifier (97.73%). The research suggests practical applications, like a web application where users can verify website links. However, the study's exploration of false positives in varied web environments was limited.

(Prasad & Chandra 2023) introduced PhiUSIIL, a method leveraging a Similarity Index and Incremental Learning for phishing URL detection. This system effectively identifies sophisticated phishing techniques like homographs and Punycode. PhiUSIIL's dataset, containing over 235,000 URLs, significantly improved detection accuracy in both pre-training and incremental training scenarios, achieving up to 99.79% accuracy. Despite its high effectiveness, the study did not fully address the model's adaptability to continuously evolving phishing strategies, a key aspect for maintaining long-term efficacy in cybersecurity.

(Mossano et al., 2023) conducted a study involving 200 participants to assess the effectiveness of URL formatting techniques in identifying phishing URLs. They tested "Who-Area Highlighting" and "Who-Area Only" against a control group with standard URLs. The study found only a minor difference in detection rates, ranging from 71% to 76%. While informative, the research suggests that the impact of URL formatting on phishing identification might be limited in real-world scenarios. Additionally, the study uncovered other factors, such as gender and attention span, influencing users' ability to recognize phishing URLs, warranting further investigation.

In (Sameen et al., 2020) developed PhishHaven, a novel ensemble machine learning system designed to detect both AI-generated and human-crafted phishing URLs. The system stands out for its use of lexical analysis and URL HTML Encoding for real-time categorization, addressing the challenge of detecting small URLs with a unique URL Hit method. PhishHaven's multi-threading and impartial voting process enhances its accuracy and real-time detection capabilities. In tests with a benchmark dataset of 100,000 URLs, PhishHaven achieved an impressive 98% accuracy rate, showcasing its effectiveness in identifying phishing URLs.

(Tajaddodianfar et al., 2020) introduced Texception, an innovative deep learning structure for analyzing phishing URLs. Texception uniquely uses both character-level and word-level data from URLs, relying on multiple parallel convolutional layers for adaptability. This approach allows Texception to accurately identify phishing URLs without manual feature engineering, demonstrating superior generalization capabilities with unfamiliar URLs. The study highlighted Texception's effectiveness, showing a significant increase in True Positive Rate (TPR) and a low False Positive Rate (FPR), but did not extensively explore its adaptability to new and emerging phishing methods.

In (Maneriker et al., 2021) extensively explored transformer models for detecting phishing URLs. Their study included evaluating conventional masked language models and domain-specific pre-training tasks and comparing these to fine-tuned BERT and RoBERTa models. This led to the development of URLTran, a transformer-based system that significantly improved phishing URL identification, achieving a high True Positive Rate (TPR) and a low False Positive Rate (FPR). Despite its effectiveness, the study did not thoroughly address URLTran's performance in real-time phishing scenarios, an important aspect for practical applications.

In (Subasi & Kremic 2020) developed an intelligent framework for detecting phishing websites, utilizing various machine learning algorithms. Their approach, which included several categorization strategies, used metrics like classification accuracy, F-measure, and AUC for assessment. Adaboost with SVM emerged as the most effective technique, achieving an accuracy of 97.61%. However, the study did not explore the framework's ability to handle advanced phishing techniques. Figure 3 illustrates the results of Subasi et al.'s research.

	Accuracy		
	Single	MultiBoosting	Adaboost
SVM	96.42	97.28	97.61
k-NN	97.18	97.21	97.23
ANN	96.91	96.94	96.91
Random Forest	97.26	97.28	97.30
CART	95.79	96.98	97.15
C4.5	95.88	97.10	97.24
Rotation Forest	96.79	97.30	97.37
REP Tree	95.33	96.18	96.95
Random Tree	96.37	97.10	96.99

Figure 3 Results of Subasi et al, (2020)

2.2 Research using deep learning.

In their (Nagy et al., 2023) innovatively applied multiprocessing and multithreading strategies in Python to enhance the training of machine learning models. Focusing on a dataset comprising 54,000 training and 12,000 testing records, the study conducted five different experiments. These included sequential execution and four parallel execution methods, notably Python's parallel backend threading. The research utilized models such as Random Forest (RF), Naïve Bayes (NB), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM), and observed significant performance improvements, with NB model showing the highest accuracy at 96.01%. Figure 4 illustrates the performance metrics of four different machine learning algorithms. The outcomes indicated substantial advancements in computational efficiency, the study's exploration did not extend to the practical application of these models in real-world phishing detection scenarios, leaving a gap in understanding their effectiveness outside controlled environments.

	RF	NB	CNN	LSTM
Accuracy	95.14%	96.01%	95.13%	95.14%
Precision	87.28%	95.65%	87.24%	87.28%
Recall	100%	92.25%	100%	100%
F1-score	93.21%	93.92%	93.19%	93.21%

Figure 4 Results of Nagy, et al (2023)

Jishnu et al., in their 2023 research, presented a groundbreaking approach for phishing URL detection, integrating RoBERTa, a transformer-based model, for feature extraction, and LSTM for classification. The process involved RoBERTa retrieving semantic and contextual information from URLs and encoding them into contextualized embeddings. This enabled the model to understand the intricate meanings and contexts associated with the URLs. LSTM was then used to classify these URLs, considering their sequential relationships. The system was rigorously tested on a vast dataset containing 300,000 URLs. The method resulted in a high success rate, differentiating authentic from phishing URLs with an impressive accuracy of 97.14%. Despite its high accuracy, the study did not delve into the model's adaptability to varied and evolving phishing scenarios, which is crucial for understanding its long-term applicability and resilience against sophisticated phishing attacks.

(Huang et al., 2019) study introduced 'PhishingNet,' a deep learning-based approach for rapid and efficient phishing URL detection. The method uniquely employed a CNN module to extract spatial feature representations of URLs at the character level. Additionally, an attention based hierarchical RNN module was used to recover temporal feature representations at the word level. These extracted features were then integrated using a three-layer CNN to develop a sophisticated phishing URL classifier. Extensive testing was conducted on a verified dataset collected from the Internet, demonstrating that the automatically derived feature representations significantly improved the model's generalization ability on new and emerging URLs. While PhishingNet showed promising results in terms of accuracy and generalization, the study did not address the system’s adaptability to evolving phishing tactics, a critical aspect for real-world application and long-term effectiveness of phishing detection systems.

In (Yang et al., 2019) proposed an innovative deep learning algorithm for phishing detection, based on a multidimensional feature analysis approach. The initial phase of their methodology involved extracting character sequence features from URLs, which were then swiftly categorized using deep learning techniques. This novel process eliminated the need for external assistance or pre-existing

phishing knowledge. In the next phase, the researchers combined various features, including URL statistics, website code, webpage text characteristics, and the initial deep learning classification results, into a multidimensional feature set. This comprehensive approach significantly accelerated the detection process, enabling efficient threshold establishment. The model was tested on a dataset containing millions of authentic and phishing URLs, achieving a remarkable accuracy of 98.99% and a low false positive rate of 0.59%. Despite these impressive results, the study did not thoroughly investigate the method's efficiency and adaptability in diverse real-world phishing scenarios, which is essential to understand its practical applicability and performance consistency.

(Rasymas & Dovydaitis 2020) study, undertook an in-depth comparison of different characteristics of phishing URLs, aiming to identify the most effective detection method. The research analyzed three distinct features: lexical features, character-level embeddings, and word-level embeddings. They proposed a modern design for deep neural networks, which integrated multiple Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) layers into a unified structure. This combination allowed the model to achieve an accuracy rate of 94.4%. Fig 5 explains an array of models that were used to compare the performance. While the study demonstrated significant progress in identifying phishing URLs using deep neural networks, it did not explore the models' effectiveness against newly emerging and sophisticated phishing techniques. This gap highlights a need for further research to evaluate the resilience and adaptability of these models in the face of evolving phishing strategies.

Table 3: Comparison of model performance

Models	accuracy	precision	recall	f1	roc_auc
Lexical model	0.861354	0.935280	0.726785	0.817956	0.844533
Char model	0.941215	0.972225	0.888211	0.928321	0.934590
Word model	0.918192	0.956640	0.847528	0.898785	0.909359
Lexical, Char and Word model	0.942801	0.972089	0.892152	0.930407	0.936470
Char and Word model	0.944001	0.971699	0.895414	0.931998	0.937927
Expose model	0.938098	0.960799	0.891954	0.925097	0.932330

Figure 5 Results of Rasymas, et al, (2020)

This combination allowed the model to achieve an accuracy rate of 94.4%. While the study demonstrated significant progress in identifying phishing URLs using deep neural networks, it did not explore the models' effectiveness against newly emerging and sophisticated phishing techniques. This gap highlights a need for further research to evaluate the resilience and adaptability of these models in the face of evolving phishing strategies.

(Subasi & Kremic 2020) presented a unique deep learning-based model for phishing URL detection, utilizing a character-level CNN approach. This model was innovative in its ability to collect and analyze sequential patterns of URL strings without requiring access to the content of the targeted website or third-party services. The researchers compared this model against various classical machine learning and deep learning models, employing different feature sets like hand-crafted, character embedding, and character-level TF-IDF. The model demonstrated superior performance, achieving an accuracy of 95.02% on the dataset and even higher accuracy on benchmark datasets. The results, detailed in Fig 6, affirm its potential for real-world application, considering the evolving nature of phishing attacks. Despite its impressive accuracy, the study did not address how the model

might perform against emerging phishing techniques, which are continuously evolving in complexity and sophistication.

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)	Training Time (s)	Test Time (s)
D2	98.58	98.55	98.62	98.56	98.58	5281.81	32.70
D3	95.46	96.63	94.02	95.30	95.43	6063.63	39.82
D4	95.22	95.01	95.30	95.16	95.22	6027.27	37.48

Figure 6 Results of Aljofey et al, (2020)

3 Research Methodology

Introduction:

This chapter on research methodology provides an in-depth examination of the various approaches utilized in our study. It presents a spectrum of strategies that range from abstract and theoretical concepts to more concrete and practical methods. This section also delves into the selection of methodologies, encompassing data preprocessing, label encoding, and the application of Machine Learning and Deep Learning algorithms. Also, the use of Hybrid Algorithms, which combine elements of both Machine Learning and Deep Learning, is thoroughly explored, and discussed.

Dataset:

The URLs in the Kaggle, Phishing Site URLs dataset have been classified as either phishing or not phishing. 5,49,346 URLs total in the sample, of which 2,74,673 are phishing URLs and 2,74,673 are not. Machine learning algorithms can be trained on the dataset to recognize phishing URLs (Kumar, et al., 2023).

The dataset is balanced, so machine learning algorithms' findings won't be skewed in favor of URLs that are phishing or not. This dataset is perfect for training machine learning algorithms because it is big, balanced, and recent. The dataset download URL is displayed in the section below.

https://www.kaggle.com/datasets/taruntiwarihp/phishing-site-urls`

Machine Leaning Models:

Machine learning algorithms are ways for finding and understanding patterns in data. The method consists of a wide range of algorithms that can discover, discriminate, and evaluate patterns in data by learning from information previously collected known as the training set.

1.AdaBoost

AdaBoost is a machine learning algorithm that is often used to improve the performance of decision trees. Its method is merging many inaccurate classifiers to generate a powerful classifier (Chawla, 2022). Each classifier in AdaBoost is trained on data, with specific attention given to changing the weights of mistaken examples to tackle difficult cases more successfully in later rounds. This method of iteration is repeated until either a certain number of classifiers are built or no more improvements can be made. The final model incorporates all the classifier assumptions, resulting in a more exact forecast. The AdaBoost algorithm is working, with a succession of weak classifiers like (basic decision trees) being trained. Each classifier focuses on examples that were incorrectly categorized by the preceding one, as evidenced by bigger representations of such occurrences. The last panel depicts the union of several weak classifiers to form a strong classifier.

2.Random Forest

Random Forest is a combination of learning techniques that can be used for both classification and regression. During training, it generates a huge number of decision trees and outputs

the class that is the mean prediction (for regression) or mode of the classes (for classification) of the individual trees (Alsarhan, et al., 2023). Random forests correct decision trees' tendency to overfit to their training set by integrating randomization into the tree construction process, hence boosting the model's resilience. A visual illustration of the Random Forest method, demonstrating the construction of many decision trees. Each tree is trained on a different collection of data and features. Collecting the predictions from all trees (majority voting for classification, average for regression) yields the final output.

3. Gaussian Naive Bayes

Gaussian Naive Bayes is a variant of the Naive Bayes algorithm that assumes that the continuous values linked to each category follow a Gaussian distribution. This approach works well when dealing with features that are believed to have a distribution (Ripa, et al., 2021). The algorithm itself is straightforward yet powerful, for classification purposes. It particularly shines in tasks, like text categorization and spam filtering. The Gaussian Naive Bayes model is depicted in this illustration showcasing a dataset that is divided into classes. Each features values contribute to a distribution. The image illustrates how this method calculates probabilities, for example by utilizing these distributions enabling classification.

4. Decision Tree

Decision trees are a method that is used in domains such as machine learning, image processing, and finding patterns. They are made up of nodes and branches that connect to one another. Nodes represent attributes associated with categories, and each category defines a value that helps in node identification. Decision trees are widely used because of their analysis methods and consistency across data sources. They are commonly used in machine learning models for training or classification. These trees solve issues by making successive decisions depending on outcome variables that have been predicted. Fig.7. shows the structure of Model Learning Algorithms in action. It starts at the top of the data and works its way down, leaf by leaf, using discrete values. Each leaf node represents a predicted outcome obtained from the learning stage of the algorithm. The method of classification starts at the root node and works its way up the tree branches to find the output class depending on the features provided. Using training cases, the ID3 method is used to build this decision tree (Jalil, et al., 2023). When joined with in the event then rules, it improves decision making. We choose the category feature that delivers the most information gained from a collection of test variables at each node. The decrease of uncertainty by separating data into value attributes results in an increase in information.

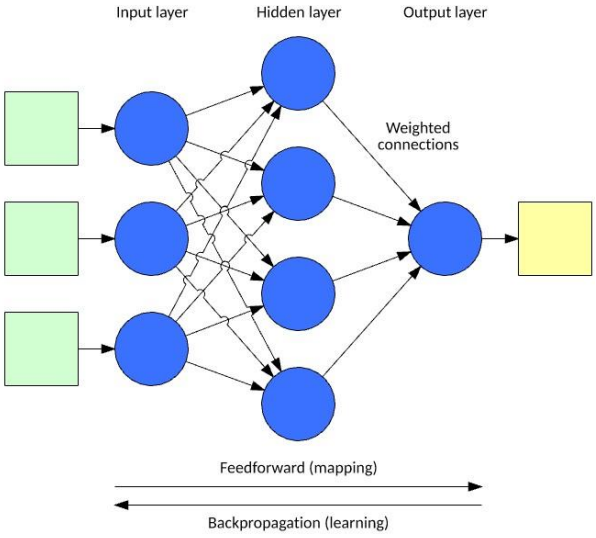

Fig 7: Structure of Model Learning Models

Deep Learning Models:

MLP

The Multi-Layer Perceptron (MLP) Classifier, a fundamental neural network used in machine learning, pattern recognition, and image processing, consists of several layers of interconnected neurons. Each neuron in these layers transforms incoming data through activation functions, enabling the MLP to capture complex, nonlinear relationships in the data. The process begins with the input layer, moves through multiple hidden layers for data processing, and concludes at the output layer, which uses functions like SoftMax for classification. The MLP's strength lies in its ability to model intricate patterns, thanks to its deep architecture and the use of backpropagation and gradient descent algorithms for training. This training is often enhanced with optimizers like SGD or Adam and requires careful tuning of hyperparameters and regularization techniques to prevent overfitting, ensuring the model's effective generalization to new data. The MLP Classifier's adaptability and capacity to handle high-dimensional datasets (Maneriker, et al., 2021) make it a valuable tool in diverse machine learning applications. Fig.8. explains how the deep learning models work effectively.

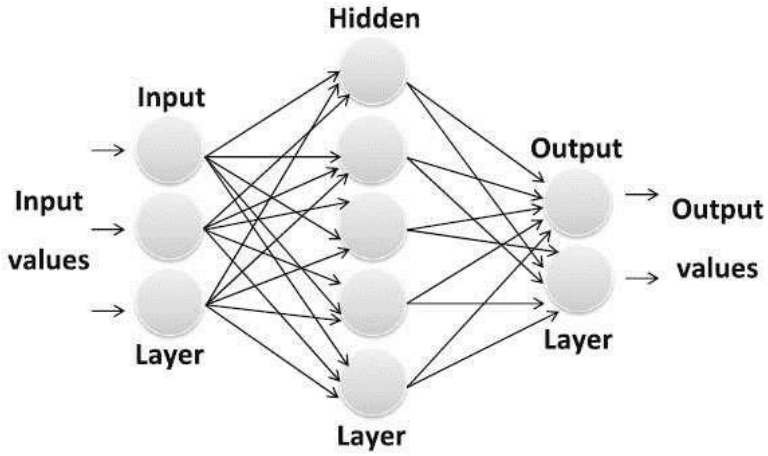

Fig 8: Structure of Deep Learning Model

4 Design Specification

Our phishing detection system is architected on a sophisticated framework that synergizes machine learning (ML) and deep learning (DL) techniques to analyze and detect Phishing URLs. The process begins with an exploratory data analysis, followed by rigorous data cleaning and preprocessing that includes text vectorization and feature extraction. We then strategically split the data for training and testing purposes. The core of our system comprises various ML algorithms such as Random Forest and AdaBoost, complemented by an MLP deep learning model for intricate pattern recognition. These models are individually tuned and subsequently fused into a hybrid model that leverages the predictive power of both ML and DL. This hybrid approach utilizes a voting mechanism to improve accuracy and robustness in phishing URL detection. The implementation, conducted in Python with libraries like Scikit-learn and TensorFlow, ensures a modular and scalable design, facilitating robust evaluations based on precision and recall, and is tailored for adaptability to evolving cyber threats. The Complete process is visually explained in Fig .9.

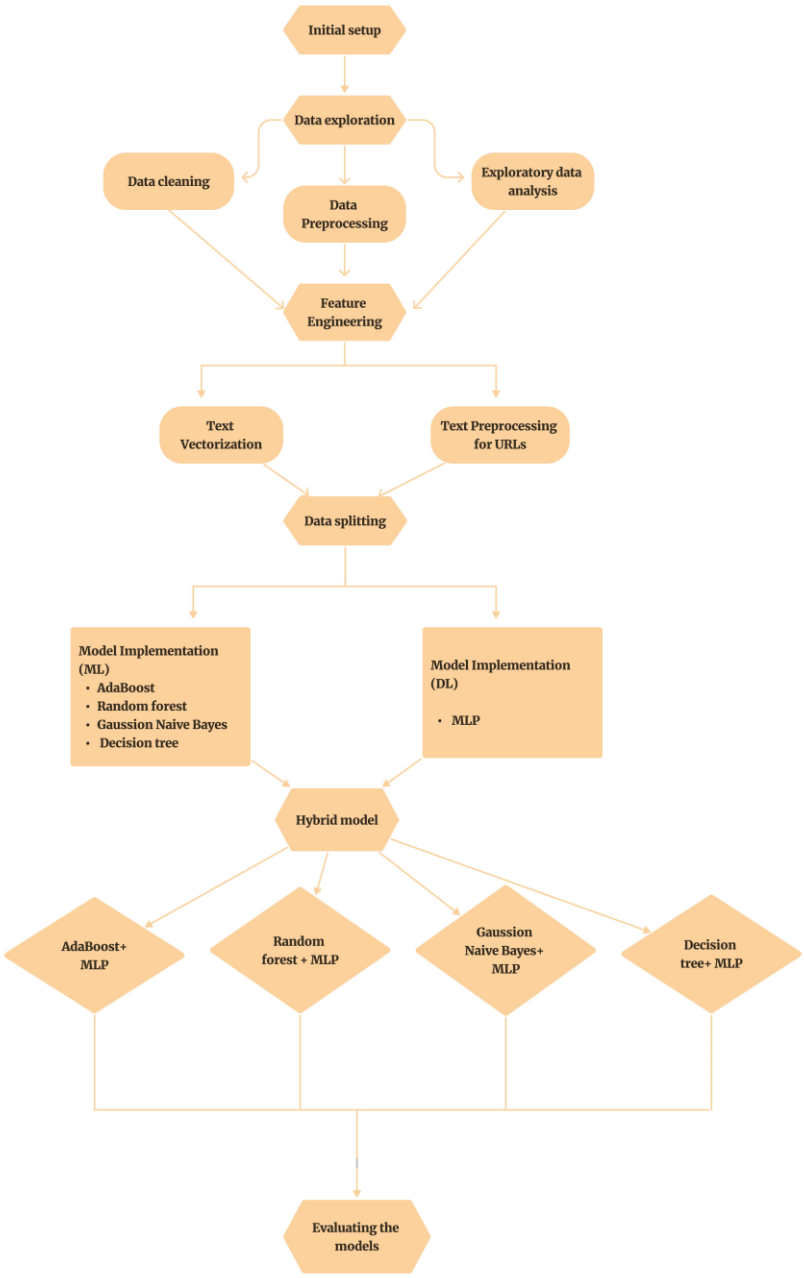

Fig 9: Flow chart for Full Implementation

5 Implementation

The experimental section of this project report focuses on the validation and performance assessment of the developed model. Utilizing TensorFlow and Keras libraries within the Jupyter Notebook environment, an open-source platform was employed for the deep learning implementations. A comprehensive description of the methodology, including the implementation steps and configurations used, is provided in the subsequent section of the report.

5.1 Data Preprocessing:

Importing the Data:

For implementing machine learning algorithms and Deep learning algorithms the first step is to import the dataset. Dataset is imported by utilizing the powerful data manipulation library, pandas, the dataset is loaded into the Python environment from a CSV file named phishing_site_urls.csv. Preliminary inspection of the dataset is performed using methods such as .head(), .unique(), and .value counts () to understand the distribution of data and identify the proportion of benign to phishing URLs.

Data Preprocessing:

Data preprocessing involves transforming data into a format that can be easily understood by machines. One of the techniques used in data preprocessing is the removal of values and null entries which will be discussed further.

Removing Duplicates:

When working with datasets it is common to encounter values. These duplicates can negatively impact the performance of models. To address this issue, it is necessary to eliminate these duplicates. In this study we have successfully removed duplicates using the duplicated function as illustrated in Figure 10.

```
urlphish_hyrd.duplicated()

0         False
1         False
2         False
3         False
4         False
          ...
549341     True
549342     True
549343     True
549344     True
549345     True
Length: 549346, dtype: bool
```

Fig 10: Removing Duplicates

Removing Null Values:

In large datasets, null or missing values can adversely affect machine learning models. This research employed a procedure to eliminate null values, ensuring data integrity. Removal of these values reduces noise, enhances model accuracy, and minimizes the risk of biased outcomes, crucial for reliable data analysis. Fig .11 shows the removal of Null Values.

```
urlphish_hyrd.isna().any()

URL      False
Label    False
dtype: bool
```

Fig 11: Removing Null Values

Thus, after removing the duplicates and Null Values in the dataset, there are 507196 counts of data. Thus 42150 duplicates values in the dataset as shown in Table 1.

Table 1: Table illustrating before and after Preprocessing.

	Count of data
Before data preprocessing	549346
After data preprocessing	507196

Data Visualization:

Data visualization serves as an integral component of this research, transforming complex datasets into comprehensible visual formats. Through this approach, insights that may remain hidden in raw data are rendered visible, aiding in pattern recognition and anomaly detection.

Data visualization is the graphical representation of information, and the data is visualized using variety of methods including charts, graphs, and maps. In this research, the data is visualized using the Pie chart. Fig.12 indicates the classes such as Good URL's and Bad URL's.

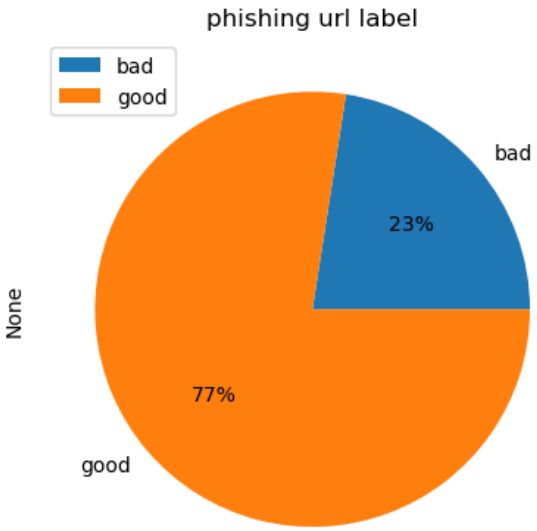

Fig 12: Pie chart indicating Good and Bad URL's

Further, the lengths of URLs were quantified and then represented as a histogram, illustrating the distribution and commonality of URL lengths within the dataset. Fig.13 histogram serves to highlight potential correlations between URL length and phishing likelihood.

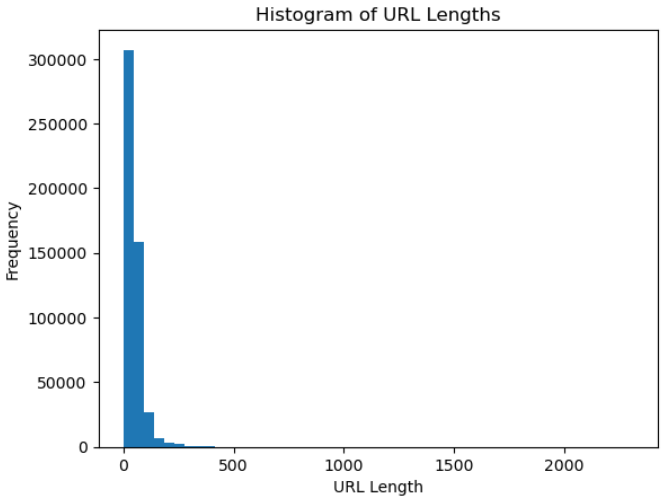

Fig 13: Histogram of URL lengths

Subsequently, a word cloud was generated, providing a visual summary of the most frequent terms within the dataset's URLs. Fig.14 visualization not only offers a snapshot of common terms but also assists in identifying keywords that might be indicative of phishing activity. Few words like "login," "verification," "secure," "update," "bank," "PayPal," are considered as Malicious, whereas some terms might include "index," "home," "search," "article," are more likely to be Non-Malicious.

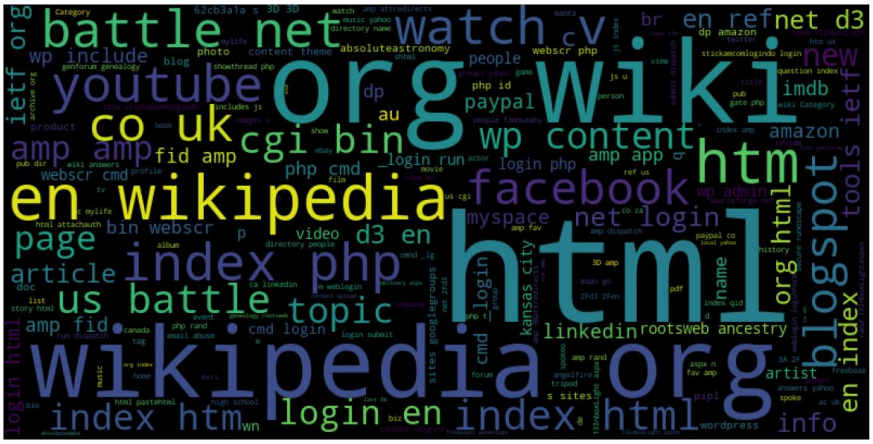

Fig 14: Word Cloud of Frequent terms within Dataset

Further analysis included evaluating the length of query strings within the URLs. A bespoke function was employed to compute these lengths, and the results were visualized using a histogram, differentiated by labels. Fig.15 represents the variance in query string lengths across different categories of URLs.

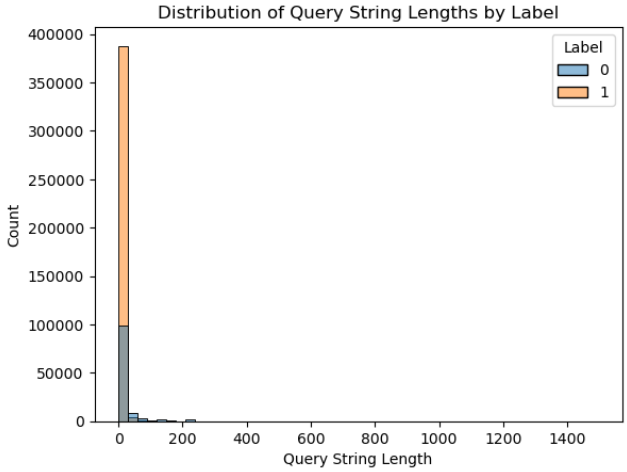

Fig 15: Distribution of Query String Length by Label

Lastly, the prevalence of HTTPS protocol usage in the URLs was scrutinized. Fig.16 depicts the count plot delineated the relationship between the use of HTTPS and the URLs' classification as either phishing or non-phishing. This visualization was critical in assessing the common belief that HTTPS might be less prevalent in phishing URLs. Each of these visualizations contributes a distinct perspective to the multifaceted domain of phishing detection, thus enhancing the analytical depth of the study.

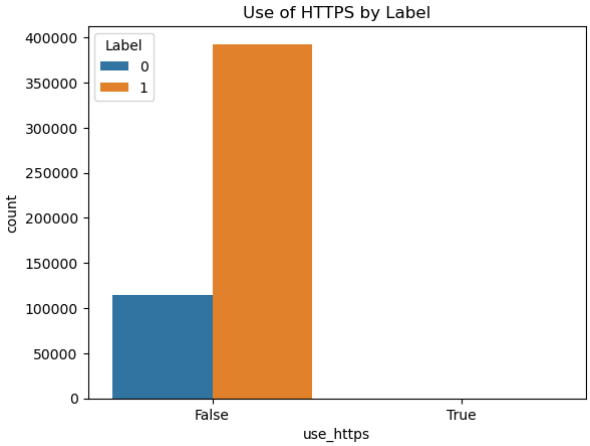

Fig 16: Use of HTTPS by Label

Finding Suitable Techniques:

The final stage of the proposed phishing detection system implementation involved a multifaceted approach to model development and evaluation. By leveraging the capabilities of Python and its libraries, a series of machine learning and deep learning models were developed, including AdaBoost, Random Forest, Gaussian Naive Bayes, Decision Trees, and Multi-layer Perceptron (MLP) classifiers. Each model was individually parameterized and optimized using GridSearchCV to identify the best-performing hyperparameters.

The data, initially raw URL entries, underwent preprocessing that included lemmatization and stop words removal using the Natural Language Toolkit (nltk) and regular expressions (regex), transforming it into a more analyzable format. The TF-IDF Vectorizer was employed to convert the text data into a numerical array suitable for machine learning processing.

A hybrid model approach was adopted by combining different algorithms through a Voting Classifier, which enhanced predictive performance and robustness. This ensemble technique integrated the individual strengths of both machine learning and deep learning models. Performance metrics such as classification reports and confusion matrices were generated to evaluate the models' effectiveness in distinguishing between phishing and non-phishing URLs. The outputs included not just the transformed dataset but also the models themselves, along with their evaluation scores and prediction times.

Throughout the implementation, warnings were managed to streamline the output, and Matplotlib was used to visually display the performance of the models in terms of accuracy, precision, and recall. These insights were pivotal in confirming the validity of the predictive models and their potential application in real-world cybersecurity scenarios.

Table 2: Comparison of Hybrid Models in Terms of Training and Testing Time, and Accuracy

Hybrid Models	Training Time	Testing time	Accuracy
AdaBoost-MLP	1339.642	0.937	0.86
Random Forest-MLP	768.395	0.575	0.88
Gaussian Naïve Bayes-MLP	704.858	0.485	0.86
Decision Tree-MLP	629.635	1.016	0.82
AdaBoost, Random Forest-MLP	633.133	24.263	0.88
Gaussian Naïve Bayes, Decision tree-MLP	8.417	0.968	0.86
AdaBoost, Random Forest, Gaussian Naïve Bayes, Decision Tree-MLP	715.035	1.909	0.86

The evaluation of hybrid machine learning and deep learning models for phishing URL detection, as summarized in Table 2, reveals varied performance trade-offs. The Random Forest and MLP hybrid model stand out with an impressive 88% accuracy, balanced by efficient training and testing times (768.395 seconds and 0.575 seconds, respectively), making it highly suitable for real-time applications. Confusion Matrix for Random Forest and MLP is shown in Fig.17. In contrast, the AdaBoost-MLP hybrid shows a comparable accuracy of 86% but requires a longer training time (1339.642 seconds), which might limit its use in rapid deployment scenarios, despite its swift testing time (0.937 seconds). Similarly, the Gaussian Naïve Bayes-MLP hybrid matches the AdaBoost-MLP in accuracy but with a reduced training time (704.858 seconds), indicating a more time-efficient training process. The Decision Tree-MLP hybrid is the quickest to train (629.635 seconds) but slightly lags in accuracy at 82%, suggesting a possible compromise between training speed and accuracy.

More complex models, like the AdaBoost, Random Forest-MLP hybrid, maintain high accuracy (88%) but have longer testing times (24.263 seconds), which could impact real-time usage. The Gaussian Naïve Bayes and Decision Tree with MLP hybrid balance a decent accuracy of 86% with a quick training time (8.417 seconds), offering a viable option for scenarios requiring swift model readiness. The most comprehensive model, integrating AdaBoost, Random Forest, Gaussian Naïve Bayes, and Decision Tree with MLP, achieves an accuracy of 86% with a moderate training time (715.035 seconds) but a higher testing time (1.909 seconds). These models collectively illustrate the diverse considerations of training and testing durations against accuracy, guiding the choice of the most suitable model based on the specific needs of phishing detection tasks.

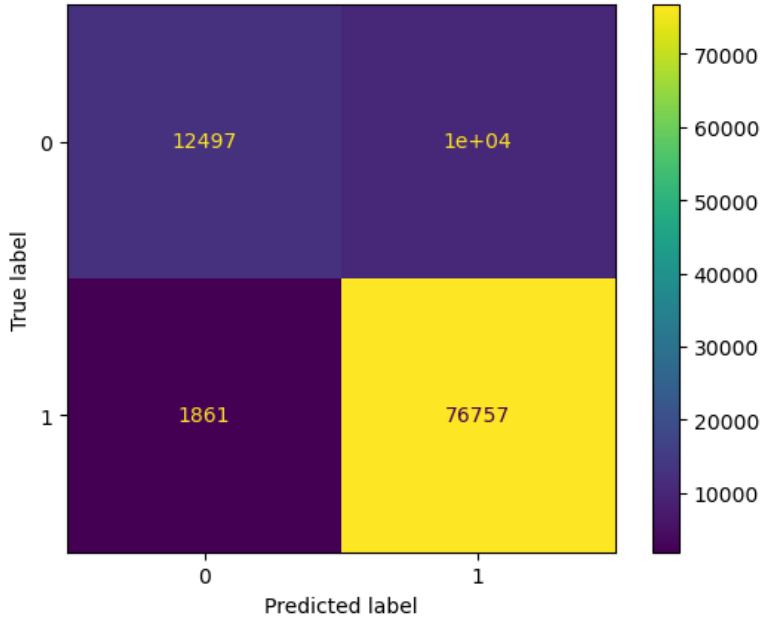

Fig 17: Confusion Matrix for Random Forest and MLP

6 Evaluation

6.1 Accuracy

In the first case study, we concentrate on the accuracy of our proposed hybrid machine learning and deep learning models for phishing URL detection. The accuracy is a pivotal performance metric that directly reflects the models' effectiveness. Our models were rigorously evaluated using a large dataset, and the accuracies were calculated using standard metrics such as precision, recall, and F1-score. The results were visualized using confusion matrices and ROC curves to provide a clear understanding of each model's performance. The highest accuracy was achieved by the Random Forest and MLP hybrid model, which suggests that ensemble methods combined with neural networks provide a robust approach to phishing detection. This has important implications for both academia and industry, as it demonstrates the potential for complex models to improve security measures against phishing attacks.

6.2 Training Time

The second case study focuses on the training time of each model. Efficiency during the training phase is essential for a quick deployment of the detection system and reflects on the model's scalability. The Random Forest-MLP model showed a compromise between a reasonable training time and high accuracy, which can be appealing for practical applications where time and resources are limited. Time-series plots illustrate the training duration for each model, and statistical tests assess

the significance of the differences observed. The outcomes of this case study underscore the need for efficient training algorithms that do not sacrifice performance, guiding future research and practical applications in developing rapid response cybersecurity tools.

6.3 Testing Time

The third case study assesses the testing time, which is crucial for real-time detection systems where quick identification of phishing attempts can prevent data breaches. The testing time was evaluated to ensure that the model not only learns from the data but also responds swiftly to new, unseen URLs. Our findings, represented through histograms and scatter plots, revealed that certain models that performed well in accuracy had longer testing times, which could hinder their deployment in time-sensitive environments. This section discusses the balance between accuracy and prediction speed, drawing attention to the practical trade-offs that cybersecurity professionals may need to consider.

6.4 Discussion

This comprehensive discussion integrates the findings from the experiments, offering a critical appraisal of the results against the backdrop of existing literature. We address the strengths and limitations of our research design, acknowledging areas where the models could be improved, such as data preprocessing or hyperparameter optimization. Suggestions for future work might include employing more advanced neural network architectures or incorporating novel feature selection methods. Furthermore, we compare our results with previous studies, thereby contextualizing our contributions within the broader scope of cybersecurity research and practice. The discussion aims to provide actionable insights for both researchers and practitioners in the field of phishing detection.

7 Conclusion and Future Work

Our research effectively analysed the efficacy of hybrid machine learning and deep learning models in identifying phishing URLs. The standout performer, a Random Forest-MLP hybrid, demonstrated exceptional accuracy, validating our hypothesis that hybrid models can outpace traditional methods in phishing detection. These results not only underscore the potential of combining machine learning strategies but also signal a leap forward for cybersecurity defenses. Despite notable successes, challenges like potential overfitting and the need for computational efficiency were acknowledged, with the understanding that these factors could limit the application's scope. The dataset, while extensive, may not reflect the entire breadth of phishing threats. Looking ahead, integrating more sophisticated neural architectures, and utilizing transfer learning could further enhance model accuracy and generalization. Commercially, these models hold promise for development into real-time phishing detection systems, offering substantial market value.

Future research should test these models against a broader array of threats, potentially incorporating multilingual data to ensure robustness. A pragmatic next step would involve piloting these models in actual cybersecurity environments, complemented by user-friendly interfaces for monitoring and intervention. This approach would not only extend the practicality of our findings but also ensure that subsequent developments are rooted in real-world applicability and effectiveness.

References

Abdul Samad, S.R., Balasubaramanian, S., Al-Kaabi, A.S., Sharma, B., Chowdhury, S., Mehbodniya, A., Webber, J.L. and Bostani, A., 2023. Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection. Electronics, 12(7), p.1642.

Prasad, A. and Chandra, S., 2023. PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. Computers & Security, p.103545.

Karim, A., Shahroz, M., Mustofa, K., Belhaouari, S.B. and Joga, S.R.K., 2023. Phishing Detection System Through Hybrid Machine Learning Based on URL. IEEE Access, 11, pp.36805-36822.

Jalil, S., Usman, M. and Fong, A., 2023. Highly accurate phishing URL detection based on machine learning. Journal of Ambient Intelligence and Humanized Computing, 14(7), pp.9233-9251.

Mossano, M., Kulyk, O., Berens, B.M., Häußler, E.M. and Volkamer, M., 2023, October. Influence of URL Formatting on Users' Phishing URL Detection. In Proceedings of the 2023 European Symposium on Usable Security (pp. 318-333).

Nagy, N., Aljabri, M., Shaahid, A., Ahmed, A.A., Alnasser, F., Almakramy, L., Alhadab, M. and Alfaddagh, S., 2023. Phishing URLs Detection Using Sequential and Parallel ML Techniques: Comparative Analysis. Sensors, 23(7), p.3467.

Alsarhan, A., Igried, B., Bani Saleem, R.M., Alauthman, M. and Aljaidi, M., 2023, September. Enhancing Phishing URL Detection: A Comparative Study of Machine Learning Algorithms. In 2023 Asia Conference on Artificial Intelligence, Machine Learning and Robotics (pp. 1-7).

Bu, S.J. and Cho, S.B., 2023, August. Phishing URL Detection with Prototypical Neural Network Disentangled by Triplet Sampling. In Computational Intelligence in Security for Information Systems Conference (pp. 132-143). Cham: Springer Nature Switzerland.

Jishnu, K.S. and Arthi, B., 2023, August. Phishing URL detection by leveraging RoBERTa for feature extraction and LSTM for classification. In 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS) (pp. 972-977). IEEE.

Kumar, M., Kondaiah, C., Pais, A.R. and Rao, R.S., 2023. Machine learning models for phishing detection from TLS traffic. Cluster Computing, pp.1-15.

Sameen, M., Han, K. and Hwang, S.O., 2020. PhishHaven—An efficient real-time AI phishing URLs detection system. IEEE Access, 8, pp.83425-83443.

Tajaddodianfar, F., Stokes, J.W. and Gururajan, A., 2020, May. Texception: a character/word-level deep learning model for phishing URL detection. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2857-2861). IEEE.

Huang, Y., Yang, Q., Qin, J. and Wen, W., 2019, August. Phishing URL detection via CNN and attention based hierarchical RNN. In 2019 18th IEEE International Conference on Trust, Security and Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) (pp. 112-119). IEEE.

Maneriker, P., Stokes, J.W., Lazo, E.G., Carutasu, D., Tajaddodianfar, F. and Gururajan, A., 2021, November. URLTran: Improving phishing URL detection using transformers. In MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM) (pp. 197-204). IEEE.

Yang, P., Zhao, G. and Zeng, P., 2019. Phishing website detection based on multidimensional features driven by deep learning. IEEE access, 7, pp.15196-15209.

Rasymas, T. and Dovydaitis, L., 2020. Detection of Phishing URLs by Using Deep Learning Approach and Multiple Features Combinations. Baltic journal of modern computing, 8(3).

Subasi, A. and Kremic, E., 2020. Comparison of adaboost with multiboosting for phishing website detection. Procedia Computer Science, 168, pp.272-278.

Aljofey, A., Jiang, Q., Qu, Q., Huang, M. and Niyigena, J.P., 2020. An effective phishing detection model based on character level convolutional neural network from URL. Electronics, 9(9), p.1514.

Ripa, S.P., Islam, F. and Arifuzzaman, M., 2021, July. The emergence threat of phishing attack and the detection techniques using machine learning models. In 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI) (pp. 1-6). IEEE.

Chawla, A., 2022. Phishing website analysis and detection using Machine Learning. International Journal of Intelligent Systems and Applications in Engineering, 10(1), pp.10-16.