Arab American University
Faculty of Graduate Studies
Department of Natural, Engineering, and Technology Sciences
Master Program in Cybersecurity

Building and Evaluating Phishing Detection Systems with Machine Learning and Deep Learning

Tahany Sadeq Kmail

202317103

Supervision Committee:

Dr. Mohammad Hamarsheh

Dr. Maher Abufarha

Dr. Fadi Draidi

This Thesis Was Submitted in Partial Fulfilment of the Requirements for the Master Degree in Cybersecurity.

Palestine, Feb/2026

Arab American University
Faculty of Graduate Studies
Department of Natural, Engineering, and Technology Sciences
Master Program in Cybersecurity

Thesis Approval

Building and Evaluating Phishing Detection Systems with Machine Learning and Deep Learning

Tahany Sadeq Kmail

202317103

This thesis was defended successfully on 14/2/2026 and approved by:

Thesis Committee Members:

Name	Title	Signature
1. Dr. Mohammad Hamarsheh	Main Supervisor	
2. Dr. Maher Abufarha	Members of Supervision Committee	
3. Dr. Fadi Draidi	Members of Supervision Committee	

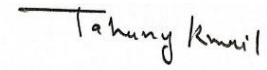
Palestine, Feb/2026

Declaration

I declare that, except where explicit reference is made to the contribution of others, this thesis is substantially my own work and has not been submitted for any other degree at the Arab American University or any other institutions.

Student Name: Tahany Sadeq Mahmoud Kmail

Student ID: 202317103

Signature: 

Date of Submitting the Final Vision of the Thesis: 1.3.2026

Dedication

I dedicate this thesis to the memory of my loving parents, whose love, kindness, and guidance have shaped who I am today and continue to inspire me even in their eternal rest. To my brothers and sisters, to my family, whose support has consistently been something I could rely on. Finally, my close friends and colleagues, I am indebted to you for the companionship and support that have sustained me and helped me throughout my career in academia. This achievement carries a part of each one of you, and I thank you all from the bottom of my heart.

Tahany Sadeq Mahmoud Kmail

Acknowledgment

My heartfelt thanks and appreciation go to Dr Mohammad Hamarsheh, Dr Maher Abufarha, and Dr. Fadi Draidi for their supervision, hard work, and precious time during my thesis writing. Their words of encouragement and constructive criticism have done a great deal to frame this paper, removing obstacles along the way in completing it. I am very much thankful for their commitment and suggestions.

Building and Evaluating Phishing Detection Systems with Machine Learning and Deep Learning.

Prepared By: Tahany Sadeq Kmail

Supervision Committee:

Dr. Mohammad Hamarsheh

Dr. Maher Abufarh

Dr Fadi Draidi

Abstract

Phishing attacks are still a significant cybersecurity enable attackers to compromise users by taking advantage of social engineering, together with the deceptive form of URLs and content obfuscation that bypass many rules-based defenses. Since phishing techniques are not fixed, and new attack patterns may emerge, we need to respond with adaptive detection strategies against the plethora of context-dependent threats. In this paper, we assess and compare the performance of various Machine Learning (ML)/Deep Learning (DL) methodologies in phishing detection, cross-dataset evaluation from global to local data. The experiment was implemented in Palestine using real internet browsing traces of local institutional domains and published global phishing datasets during the data collection and experimental evaluation phase. In this context, an experimental quantitative approach was pursued by implementing and assessing several detection methods, such as Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory networks (BiLSTM), and transformer-based ones like Distilled Bidirectional Encoder Representations from Transformers (DistilBERT) or advanced ensemble classifiers. The dataset was made up of phishing and legitimate URLs, with a sample size in research scale accounting for 11,430 global samples and over 6,000 locally collected instances. For feature collection, we used URL-based features and HTML-based features as well as domain-based features with some preprocessing and feature engineering processes to improve the quality of data. For a fair comparison, model evaluation was also conducted using common classification metrics. detection performance clearly. The discriminative score of clustering methods was low, CNN models served as a good

base method, and BiLSTM and DistilBERT were able to improve the results by modelling sequential and contextual patterns. Ensembles accounted for the best stability and regularity, with an ensemble attaining .0966 accuracy on the local Palestinian dataset. An important contribution of this work is a new and evaluated localized phishing dataset that can be used for realistic evaluations in regionally bounded context. The proposed detection pipeline achieved stable performance on both local and global datasets, proving its generalization ability. Thus, the study suggests using ensemble - based detection mechanisms, considering localized datasets while training, and focusing on adaptability and interpretability of models to improve phishing detection systems in real-life settings.

Keywords: Phishing Detection, Machine Learning, Deep Learning, Ensemble Models, and Datasets.

Table of Contents

List of Tables

List of Figures

List of Definitions of Abbreviations

URL	Uniform Resource Locator.
SSL	Secure Sockets Layer.
CNNs	Convolutional Neural Networks.
RNNs	Recurrent Neural Networks.
ML	Machine Learning.
DL	Deep Learning.
BEC	Business Email Compromise.
WPPDD	Web Page Phishing Detection Dataset.
AOL	America Online.
SVD	Singular Value Decomposition.
NMF	Non-negative Matrix Factorization.
TF-IDF	Term Frequency – Inverse Document Frequency.
KNN	K-Nearest Neighbors.
LSTM	Long Short-Term Memory.
GCN	Graph Convolutional Network.

NLP	Natural Language Processing.
GRU	Gated Recurrent Unit.
ROC	Receiver Operating Characteristic.
RCNN	Recurrent Convolutional Neural Network.
DBN	Deep Belief Networks.
BERT	Bidirectional Encoder Representations from Transformers.
LLMs	Large Language Models.
RoBERTa.	Robustly Optimized BERT Approach.
DNS	Domain Name System.
DBSCAN	Density-based spatial clustering of applications with noise.
SMOTE	Synthetic Minority Over-sampling Technique.
XGBoost	Extreme Gradient Boosting.
LightGBM	Light Gradient Boosting Machine.
DistilBERT	Distilled Bidirectional Encoder Representations from Transformers.
IPSDM	Improved Phishing and Spam Detection Model.
K-means	K-Means Clustering.
HTTPS	Hyper Text Transfer Protocol Secure.

ANOVA	Analysis of Variance.
AUC	Area Under the Curve.
WCSS	Within-Cluster Sum of Squares.

Chapter One: Introduction to the Study

1.1 Introduction

Phishing is a type of cybercrime that has rapidly spread and permeated the world; it also impacts both individual users and organizations in terms of finance and security. These malicious activities make extensive use of social engineering to dupe victims into revealing personal information, including usernames, passwords, and financial information, through fake websites and deceptive e-mails that appear as if they originated from legitimate organizations.

Phishing attacks have become more sophisticated with time, and attackers have therefore evolved to tactics by taking advantage of vulnerabilities in digital security ecosystems. One natural result is that the traditional security mechanisms fail to detect phishing attacks effectively since they are static-based and rule-based; therefore, it is difficult to solve the dynamic and rapidly changing behaviors of the current phishing threats *(Tang & Mahmoud, 2021)***.** As an indicative example of how phishing attacks are usually perpetrated, this sort of cybercrime can be broadly categorized according to a few principal stages involved, as shown in Figure 1.1.

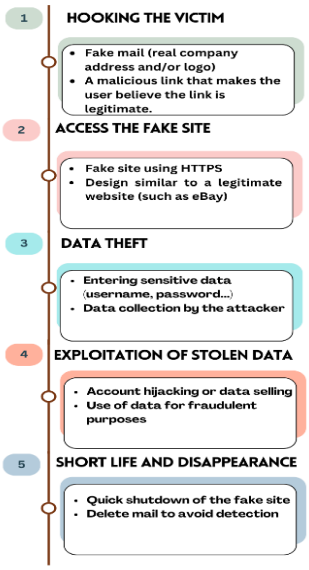


Figure 1.1: Phishing Email Life Cycle.

Phishing detection has used static, rule-based systems in the past, where blacklists comprising known bad URLs or emails are applied. The systems are, however, not effective in countering the changing nature of phishing techniques. There are more sophisticated phishing attacks, which include social engineering, an approach in which attackers rely on psychological manipulation or the exploitation of trust and social relationships to persuade individuals to reveal confidential information or perform certain actions, such as clicking on malicious links or entering credentials on fake websites. These attacks also include spear phishing, which is an attack that targets specific individuals or organizations via carefully crafted emails or communications, often after gathering detailed information about the victim to increase the credibility of the attack and persuade the target to respond to the message. These attacks also include domain name spoofing.

Domain spoofing is an attack where an attacker claims a domain that looks almost the same as a real website (a few letter or extension differences), and users are fooled into thinking they are visiting the real site. Eventually, attackers can also exploit bogus SSL certificates or those issued by untrusted parties to simulate a secure connection on phony sites and hence successfully deceive users into inputting their sensitive information. Therefore, there is a serious need to develop more dynamic and adaptive phishing systems in order to cope with this and the ever-growing cyber threat *(Musa et al., 2023)*.

1.2 Phishing in Palestine

Although there is scant literature and statistics on phishing attacks in Palestine, global and regional supply-side trends suggest at least a steady increase in such threats. Previous research has also shown that the frequency of phishing attacks goes up markedly in "politically unstable" areas, which, like it or not, is a label that could be applied to several countries in the Middle East. It is not unreasonable to infer, therefore, that as the Palestinian information arena is only lightly covered by quantitative surveys, phishing similar to that perpetrated against individuals and organizations in those areas is also aimed at a considerable scale at Palestinians.

Recent regional reports indicate that the Middle East has become a major hotspot for phishing attacks. These reports counted approximately 447 cyber incidents in the region in just one year, many of which were “politically motivated” and targeted government and private institutions. The Anti-Phishing Working Group report also showed that the region is witnessing a steady increase in the volume and sophistication of phishing attacks, making it one of the most vulnerable regions globally, as shown in Figure 1.2. These indicators highlight the fragility of the digital environment in the Middle East and illustrate the magnitude of the risks facing countries close to Palestine, given the absence of in-depth local studies that reveal the reality of cyber threats*(Anti-Phishing Working Group, 2024; CloudSEK, 2024)*.

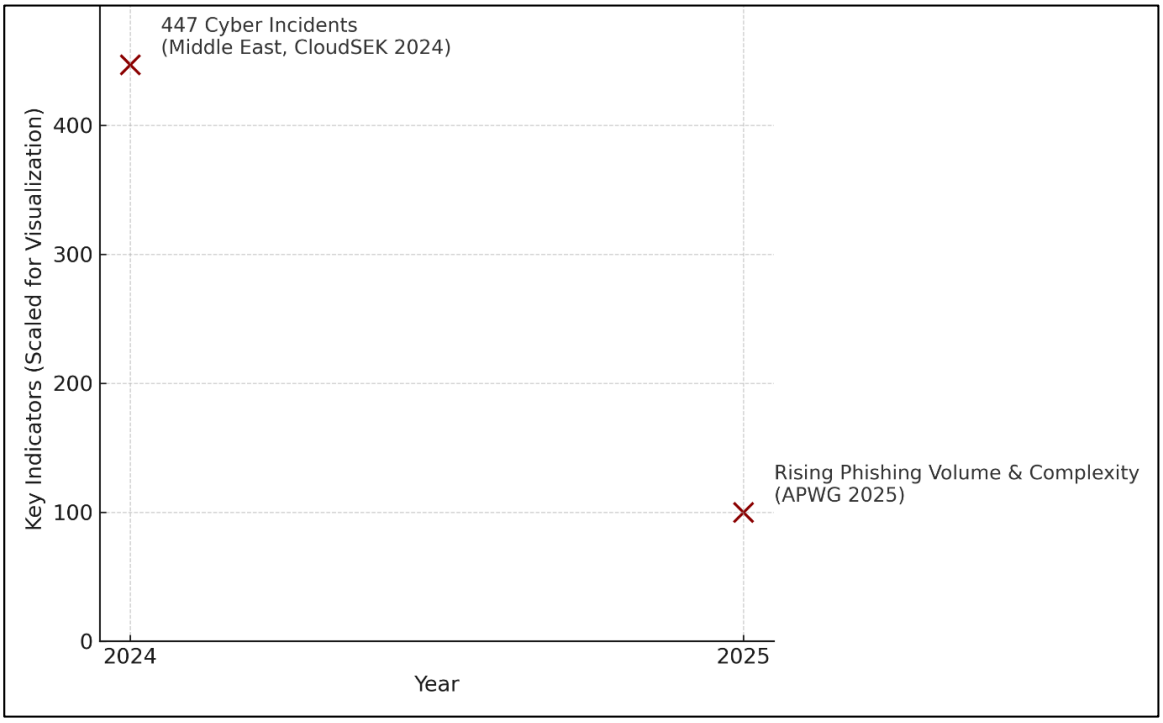


Figure 1.2: The phishing trends in the Middle East (2024-2025).

1.3 Phishing Detection

Traditionally, phishing detection is performed on the basis of heuristics-based solutions and blacklists, which can no longer be effectively applied to sophisticated phishing attacks *(Tang & Mahmoud, 2021)*. These attacks are not as blunt as they used to be because they are now more

focused and may involve attacks such as spear phishing, where recipients of the message are chosen, and the message is tailored to them. Moreover, the fact that phishing activities are currently disguised by SSL certificates and, in practice, by domain spoofing, as well as by manipulation of URL structure, has made it even more challenging to identify phishing websites based on the traditional measures *(Al-Sabbagh et al., 2024).*

The AI technology based on machine learning (ML) and deep learning (DL) provides a potential answer. In contrast to the traditional rule-based systems, ML and DL models can learn and identify complex patterns given large data sets, and even identify phishing techniques that have never been seen before. To keep up with the novel phishing tactics, these models can keep updating according to the sheer amount of data, such as URLs, email contents, characteristics and features of websites, and user activity *(Catal et al., 2022).*

Convolutional Neural Networks (CNNs) have been effectively used to detect phishing sites based on the format of URLs and web content structure. On the same note, Recurrent Neural Networks (RNNs), especially the Long Short-Term Memory (LSTM) networks, are good at identifying phishing emails by perceiving the order of words and phrases used in the text. Ensemble learning is also being researched as a promising technique for phishing detection, which integrates the advantages of several classifiers to advance generalization and robustness. Ensemble methods combine different learning models to improve the performance and stability of the detection, especially for complex and variant phishing patterns without using synthetically generated data. Models based on transformers, such as BERT, can be very useful for handling complicated text pattern analysis online.

Such systems have the capability of processing not only structured data, such as the URLs, but also unstructured data, such as the content of emails or the characteristics of websites, thereby expanding their capacity to recognise advanced phishing exploits *(Gilpin et al., 2019)*.

1.4 Significance of the Study

The scientific and practical contributions of this work are twofold, with afocus on the domain of cybersecurity, in particular phishing, which has emerged as one of the most widespread,

costly, and fast-moving cyber threats globally. With increasingly sophisticated phishing methods, legacy rule-based detection approaches might start performing less efficiently, and that raises the demand for more exact, adaptable, and data-driven detection mechanisms.

On the scientific side, this research helps to advance the state of knowledge by conducting a comprehensive comparison and evaluation of ML vs. DL with users' data. It contributes to the existing literature by investigating the performance of advanced learning methods, like deep neural models and ensemble classifiers, in URL-based phishing detection. The work provides an understanding of the practicalities, challenges, and limitations that techniques have to deal with sophisticated and evolving phishing patterns, as well as the way to generalize solutions on global benchmark datasets locally generated. By filling a gap in localized phishing studies, and especially within the Palestinian context, this study continues previous research that has overwhelmingly been based on global datasets and creates a basis for future regional focuses on cybersecurity.

Practically, this research has significant implications for the design and improvement of practical phishing detection systems that can be deployed in practice. The framework can be incorporated with organizational security facilities like e-mail gateways or internal monitoring that act as a precursor for threat mitigation. The model automatically examines incoming URLs and determines whether they are legitimate or phishing to minimize unauthorized access, credential compromise, and data leakage. In addition, using local Palestinian data is beneficial for enabling the model to detect context-specific phishing patterns so that it can increase true positive alerts and consequently improve system reliability.

The relevance of this study is also supported by recent international and local cybersecurity studies. Phishing attacks rose notably in 2024 and politically motivated and hacktivist-led cyber incidents were commonplace in the Middle East. *(Anti-Phishing Working Group, 2024), (Positive Technologies, 2024)* have published reports detailing how phishing attacks are becoming more distributed, complex, and costlier to public and private institutions, as illustrated in Figure 1.3. Regardless of this increasing threat, it is assumed that the issue of detecting phishing within the Palestinian context is under-addressed. Therefore, the result of this study can be used to enhance current phishing detection systems in Palestine with more advanced features and capabilities to

reduce cyber threats that cause minimal damage to people. Companies and government entities in Palestine can enhance their cybersecurity capacity.

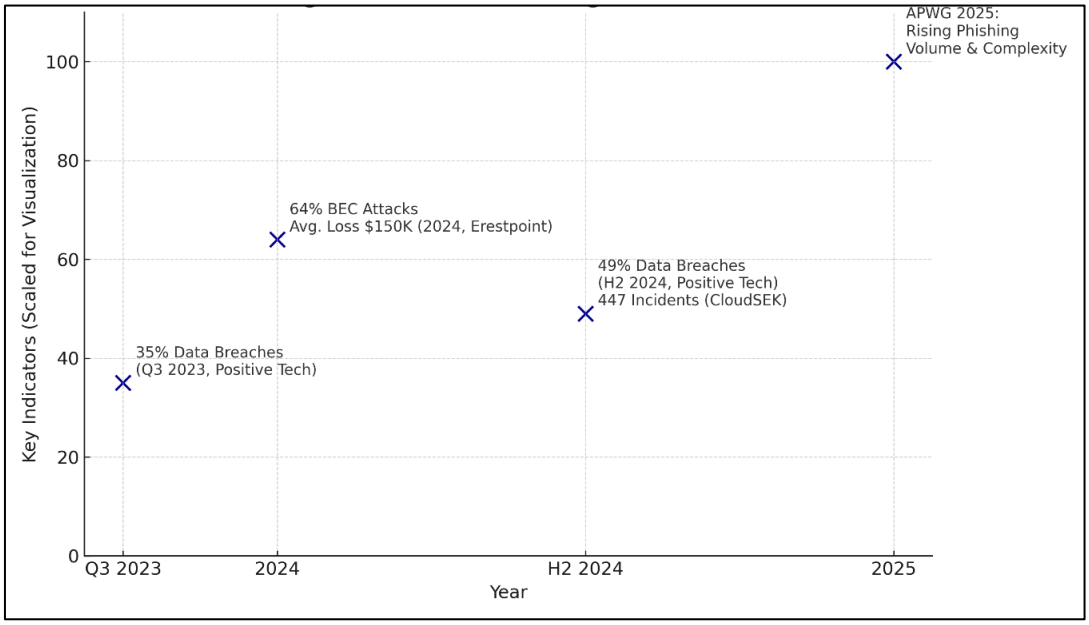


Figure 1.3: The Evolution of Cyber Attacks, Especially Phishing, Globally and Regionally During the Period 2023-2025.

1.5 Problem Statement

With the rapid maturation of the digital service, phishing attacks have increasingly become one of the most severe cybersecurity threats to both individuals and business entities. Complicated and varied onslaughts are harnessing security loopholes in ever more creative ways. These could range from simple fakes to disguise the site as a legitimate service, obfuscate the domain name, or attempt to elude being caught. The constantly changing nature of social engineering attack vectors makes adaptive detection that much more critical in light of the extremely dynamic threat landscape that cyberattacks present.

While these latest deep learning approaches achieved much better results and the extraction of high-level indicators to identify suspicious websites, their generalization is limited to the multifaceted feature space of the suspicious domains. Specifically, they have difficulty in

distinguishing between domain names with an explicit intention to be concealed and legitimate (phishing) domain names being registered by taking advantage of well-established organizations.

In addition, their effectiveness is undermined by the high dimensionality of URL and text features, which leads to an overfitting risk with scarce training data. These problems are even more emphasized by the massive computational resources needed to train and fine-tune such models properly. Furthermore, the general reliance on publicly available labeled datasets constrains these models from being able to account for locally collected data or new phishing techniques.

1.6 Research Objectives

The main objective of this research is to improve phishing detection in Palestine by achieving the following objectives:

1. Evaluating and comparing the performance of machine learning and deep learning techniques, and examining how effectively each approach identifies phishing attacks based on the characteristics of the datasets used.

2. Identifying the phishing detection technique that achieves the lowest false positive rate on global datasets while maintaining reliable performance based on standard evaluation metrics.

3. Fine-tuning and calibrating the algorithms and attempting to find the best model that can contribute to raising the accuracy of training and testing to the highest possible level.

4. Improving Palestinian phishing detection by collecting local datasets and applying the best detection techniques previously tested on global datasets, which will inform further cybersecurity measures.

1.7 Research Questions

1. How do ML and DL models differ in their ability to detect phishing attacks across different dataset characteristics?
2. Which phishing detection model gives the lowest false positive rate on the global dataset while still performing well on the main evaluation measures?
3. How do hyperparameter tuning and model calibration affect phishing detection performance, and which tuned model reaches the highest accuracy?
4. What level of improvement in accuracy and robustness is achieved when the best global phishing detection model is applied to the local Palestinian dataset?

1.8 Structure of the Thesis

The remainder of the thesis is structured as follows:

1. Introduction: This is where your topic, problem statement, objectives, and significance of the study should be stated.
2. Related Work: Discusses the state of the related work in phishing, ML, and DL in computer security, also reviews the literature's shortcomings, and finally outlines contributions or progress made.
3. Method: The study design, data collection, model development, and validation are presented.
4. Results: Provides the results of the study (e.g., model performance indices, comparison).
5. Discussion & Recommendations: Discusses the findings, identifies the significance of the study with reference to objectives, and concludes with recommendations that can be carried out on the basis of the above observations, including limitations.
6. References: It includes all the references cited in the thesis.
7. Appendices: Any supplementary data, code, or analyses.

To make this evolution more understandable this evolution, Figure 1.4 shows a graphical overview of the structure of the thesis:

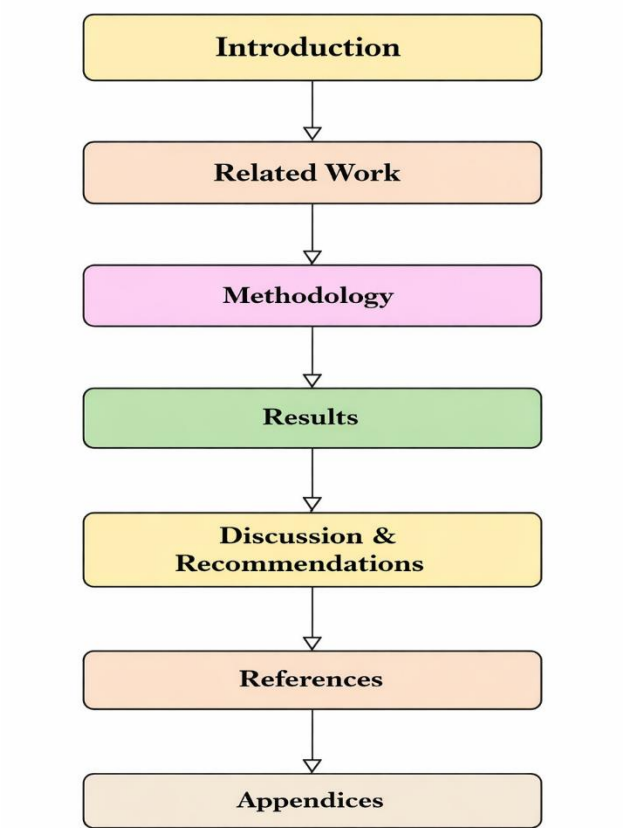


Figure 1.4: Flowchart Illustrating the Structure of the Thesis.

1.9 Limitations of the Study

The acquisition of the local dataset was problematic in two ways from the outset because, on one hand, it was not permitted to access the security logs due to strict data protection regulations in a reliable and systematic way, as already stated directly within an official government department. As a result, several security and administrative processes were imposed on the research, such as obtaining multiple institutional approvals, removing identifying elements from server logs, and limiting access to raw logs. Some of these restrictions prolonged the time required to collect data and resulted in a small dataset, but they allowed compliance with ethical norms and institutional privacy policies.

Second, because of these same security limitations, the produced data were not stored for long durations and shared for external validation. This limitation prevented people from running such large-scale repeated experiments or cross-institutional comparisons with the same dataset. However, this restriction is in fact a realistic scenario as phishing defense systems are usually operated under similar circumstances.

Third, we collected our data within time windows, so we may not observe all long-term or seasonal phishing patterns. Despite attempts to secure equal representation and diversity of behavior within samples, some temporal variation may be underrepresented in what was actually collected.

Another obstacle is that the comparison between models was not entirely fair in some cases, not because of differences in the data itself, but because of differing data segmentation mechanisms sometimes imposed to suit the structure and training requirements of certain algorithms. Therefore, the results should be viewed more as indicators of overall performance than as a rigid numerical comparison between models.

1.10 Definition of Key Terms

- Phishing: Social Engineering An attack that is designed to trick a user into giving away sensitive information, such as login details or credit card information, by pretending to be a trusted source like official communication between employees.
- Machine Learning (ML): A branch of AI that allows machines to learn from data patterns and make predictions or decisions without being explicitly programmed.
- Deep Learning : an advanced topic in machine learning, utilizing multi-layer neural networks to learn predefined (or automatically learned) complex representations from large-scale data.
- Ensemble Learning : The learning paradigm using multiple classifiers, and their joint optimization to improve detection accuracy, stability and generalization.
- False Positive Rate (FPR) : The proportion of non-phishing cases that are wrongly classified as a phishing attack by a detection model.
- URL Features : The anatomy of a URL for differentiating phishing sites from legitimate sites.

- Local Data : local data that belong to the area-specific profile of phishing characteristics in its geographic field.

1.11 Conclusion

Phishing is one of the most insidious and detrimental cybersecurity risks, and it is continually developing, as destructive players are learning to make reactive changes to take advantage of exploited cracks in digital assets. Such attacks usually trick the user into divulging their valuable data like usernames, passwords, and monetary information through false websites, emails, or other social engineering scams. Phishing methods have evolved over the years to the extent that they are posing a challenge to traditional security measures. This intricacy shows the importance of having sophisticated detection mechanisms that are capable of responding to new phishing methods.

In the past, the only phishing-related tools available were static, rule-based tools, examples of which include blacklists of known malicious URLs and emails. Nevertheless, they cannot be effective in identifying the ever more advanced phishing practices, e.g., spear phishing attacks, social engineering, and SSL certificate tampering *(Tang & Mahmoud, 2021)* As a counter to this, ML and DL models have become an effective solution for detecting phishing, and the advantage of adapting to new strategies to large amounts of data *(Catal et al., 2022)*. In particular, complex phishing activity detection has been successfully facilitated by models such as CNNs, RNNs, and Ensemble models Although institutions and ministries use tools such as Sophos and Microsoft Defender, constant improvement of detection is still a necessity. This study will help meet this requirement as it will help to use global and local data, to create a sophisticated phishing detection system that will be based on ML and DL algorithms. This study will be conducted phishing-related data available at the MOE, such as phishing incident reports, email and attack logs and patterns.

Chapter Two: Literature Review

2.1 Introduction

Phishing has become one of the most common and dynamically evolving threats to cybersecurity, posing serious threats to internet users and enterprises. Online scammers never stand still, and their latest trick is using more complex dodges to steal the personal information of UK citizens. As these attacks become increasingly sophisticated, legacy detection solutions that utilize rules-based approaches or static blacklists are struggling to keep pace with new, dynamic, and advanced phishing strategies. This has led to an urgent demand for new phishing detection techniques.

Machine learning and DL approaches offer strong potential to improve the accuracy of phishing detection. These technologies allow systems to learn and evolve by identifying patterns in data and, therefore, offer a dynamic and agile solution to counter phishing attacks. The latest trends in ML tools, such as CNN, RNN, transformers, and Ensemble-based models, are also widely used for enhancing phishing detection techniques. These strategies have various advantages and disadvantages; one of their key capabilities is to explore URL, email text, website body, etc., features all correlated to phishing attacks. These sophisticated methods are likely to have a significant place in the defense against expanding cyber threats as phishing monitoring systems are developed.

2.2 Phishing Definition: Background and History of Phishing

Phishing is the practice of tricking users into giving away sensitive, personal details, such as usernames, passwords, or credit card numbers, which can be obtained by masquerading as a reliable entity in an electronic communication *(Priya et al., 2024).* The first documented use of phishing occurred in the mid-1990s when cybercriminals exploited AOL Instant Messenger to dupe account holders into releasing sensitive account information. Since then, phishing has continued to advance, becoming more sophisticated and widespread.

At first, most phishing attacks occurred via email that pretended to be from trusted organizations, and were, in many cases, loaded with spoofed logins or links to malware. These

early phishing attacks were easier to detect because they often included sloppy language, the wrong logos, and suspicious links. However, as tricked users grew more familiar with these attacks, so too did the cybercriminals, who began shaping the tactics that underpin phishing attempts in order to make them harder to spot.

In the early 2000s, phishing attacks became increasingly focused on both banking and e-commerce websites and moved towards bespoke phishing attacks like spear phishing when criminals started creating tailor-made messages to target particular individuals or organizations. Meanwhile, explosive growth in social media and the explosion in mobile devices provided additional vehicles for phishing, they noted, making detection even more complex.

The attackers started off as basic phishers but over time have grown more sophisticated and have begun using domain and social engineering phishing and stealing certificates (SSL) to trick users into thinking they are interacting with trusted entities. These methods have made phishing more difficult to detect using the traditional approaches and resulted in the creation of enhanced, AI-based solutions. Phishing is still a significant security concern today, and phishing attacks nowadays also often include fake websites, malicious links, and fake login forms that would be less straightforward to detect.

As phishing attacks become increasingly sophisticated, the requirement for sophisticated detection systems, such as machine learning and deep learning, is more important than ever. These technologies have been embedded in new, more effective systems, enabling real-time detection of phishing even with new or novel phishing attacks.

2.3 Exploring Effective Approaches in Phishing Detection: A Study of Modern Techniques

Phishing remains a significant and increasing cyber risk for both people and businesses. As scammers invent increasingly elaborate ways to fool people, we need both better and more bulletproof ways to track down and halt phishing attacks. Conventional phishing detection techniques, such as rule-based mechanisms or static blacklists, are becoming increasingly barriers to unknown, evolving, or new phishing attacks. In order to tackle this challenge, ML and DL

based methods promise to be useful to make phishing detection accurate by automatically extracting patterns from the data and attempting to tackle new types of cyber-attacks.

At the beginning of phishing detection, classical ML methods mainly used manual feature engineering. For instance, *(Yasin & Abuhasan, 2016)* used stemming, WordNet with five classifiers extending Random Forest, which achieves up to .0991 of accuracy on a labelled phishing set, which proves that an enrichment of text representation adds value for phishing detection. Also, *Halgas et al. (2020)* investigated Singular Value Decomposition(SVD), NMF, and its application to the Term Frequency Inverse Document Frequency(TF-IDF) vector representation of phishing and benign emails. They showed that dimensionality reduction can be used to enhance phishing detection accuracy by using classifiers such as Decision Tree, Logistic Regression, K-Nearest Neighbors(KNN), etc.

introduced a hybrid method where the focus was on the behavioral aspects, i.e., the IP characteristics of the sender or credentials used for frequent email sending *(Hamid et al., 2013)*, that are harder for phishers to imitate. They reached .094 accuracy with this approach when applied to a dataset containing 6923 e-mails from the Nazario and Spam Assassin repositories. *(Zamir et al., 2020)* studied a dataset of 11,055 Phishing sites and compared different feature selection methods, including Gain Ratio, Information Gain, and Recursive Feature Elimination *(Tiwari, 2020).* The authors observed that ensemble learning, such as Random Forests, had the best results in comparison to other classifiers, with an accuracy of .0973, and when the ensemble stacking is performed, the accuracy improved to .0974.

With increasingly advanced phishing techniques being developed, it has become imperative to utilize deep learning models that are able to learn hierarchies of features in an automatic manner. As an instance, *(Alhogail & Alsabih, 2021)* suggested the use of a model by incorporating Graph Convolutional Networks (GCN) with Natural Language Processing(NLP) for detecting phishing emails, which was able to achieve high accuracy (.0982) with a very low false positive rate (0.015). *(Brindha et al., 2023)* introduced a more intelligent method, which consisted of the cuckoo search algorithm using recurrent units, and arrived at a significant success of .09972 in

order to begin the spam emails from the dataset Enron records, which, when joined with strategies of optimizations with deep recurrent networks, can yield even better results.

Phish Responder *(Dewis & Viana, 2022)* also introduced "Phish Responder", a hybrid system that relies on LSTM models and NLP for signature and marking detection of phishing & spam emails. Using an LSTM model, they achieved a .099 ROC accuracy in text-based datasets settled that sequence modeling is powerful in identifying phishing. Similarly, introduced a fashion for an architecture to detect phishing via a Recurrent Convolutional Neural Network (RCNN) with Bidirectional LSTM layers and achieved a low false positive rate (0.043).

There are also more recent studies that look at other deep learning models, including reinforcement learning, RNN, or Deep Belief Networks (DBN) for spam and phishing detection. These trends indicate the increasing interest in using the cutting-edge neural networks in the field of cybersecurity defense.

In this part, we'll review those 5 techniques and see how they can be applied to phishing. We adopted CNNs as they are capable of learning hierarchical information about the URL and web content. In this paper, we refer to unsupervised types of methods, such as a cluster method, for seeking phishing campaigns which do not need the labeled data. LSTM networks, and RNNs more in general, were shown to be efficient models for processing text examples of threats such as phishing emails.

This section gives an overview of those methods by discussing the advantages, techniques, and the application of the methods to detecting phishing. By understanding the fundamental strengths and weaknesses of these models, we are able to provide a more accurate assessment of how they might help to prevent the continuously changing phishing attacks. Taken together, these various means of detection could allow more effective and resilient phishing detection systems to catch both new and old threats.

2.3.1 Convolutional Neural Networks CNNs for Phishing Detection

Phishing detection has been one of the open research problems as the number of phishing websites and attacks increases. Phishing: Cybercriminals use this to deceive individuals into sharing private information by creating bogus websites. The papers reviewed indicate to us how CNNs can be leveraged to extract phishing websites, and here we are specifically interested in exploiting DL to categorize both URLs and website contents. The aim of this approach is to make it harder for phishing detection systems to cope with the reality that phishing efforts are getting bigger and more complex.

On the contrary, extended this concept slightly, and they employed a larger dataset of 6,157 real websites vs 4,898 fake websites. The dataset included features such as server form handlers (SFH), domain registration lengths, and SSL final states that improved the model's generalizability. The CNN approach outperformed classical models (Naive Bayes, Decision Trees, and Support Vector Machines (SVM)), reaching 0.982 accuracy and an F1 of 0.976. This approach demonstrated how capable deep learning is of finding phishing sites that have not previously been encountered. This also indicated that CNNs are effective in real-time phishing site discovery.

The researchers selected the datasets for these studies very carefully so that they featured real websites along with phishing websites. This ensured there was a good sample size for each type. The data sets contained more than just URLs. They also had knowledge about how phishing sites were constructed and operated, such as web traffic, DNS records, and SSL certificate use. CNN could learn simple and complex patterns that are indications of phishing by training the models using a variety of features gathered from such websites.

has proposed a CNN-based model for the detection of phishing websites based on URL features *(Yerima & Alzaylaee, 2020)*. These features were converted to feature vectors, which in turn were described as photos. The dataset consisted of 1,353 URLs and was composed of three types of URLs: phishing, legitimate, and suspicious. Every URL was measured with 10 distinctive clues: the length of a URL, its age, and unknown characters, for example. Transforming the

features of URLs into images made it simpler to apply CNNs to the classification, since each feature vector was represented as a gray area inside the image. The model was able to predict right .0865 of the time. RELUs were introduced to address issues like the vanishing gradient problem, which helped the deep learning model to perform much better*(Kulkarni, 2023)*. This work demonstrated that CNNs can actually be an effective tool in locating phishing websites, but it also demonstrated how difficult it is to discover new methods that don't behave in the manner one expects.

The studies that were analyzed employed CNNs for phishing website feature extraction and classification. We took other features of the URLs (length, domain age, etc.) and turned them into feature vectors. We then converted the feature vectors into images for the CNN to process. We evaluated the models with common metrics such as accuracy, precision, recall, and F1-score. Both works conducted training-testing split, but applied 10-fold cross-validation to ensure the robustness of them. CNNs allowed the data to learn complex patterns itself, and we did not have to handcraft features.

These studies' findings indicate that there is much to be hopeful for. Kulkarni's CNN model reached .0865 accuracy, while Yerima and Alzaylaee's model achieved .0982 accuracy *(Yerima & Alzaylaee,2020)*. These findings hint at the potential usefulness of CNNs in identifying phishing sites, especially the ones being seen for the first time. In order to defend against the fast-moving world of cyberattacks, it is necessary to be able to discover new phishing sites in real time. The fact that these models succeed at all is a demonstration of the power of DL in solving tough cybersecurity problems, in particular, detecting phishing emails, which traditional techniques, for the most part, fail to do.

Nonetheless, these studies had some limitations. One of the main issues is the difficulty of obtaining labeled datasets for training. Moreover, CNNs are very effective, but they tend to easily overfit, especially when they are trained on low-diversity datasets. We didn't mention any regularization approaches like dropout and batch normalization, and their lack may be the reason for overfitting in some cases. And, not to mention, URL and web content features are great, but they're not going to be able to catch a phishing site that leverages domains or forged SSL. CNNs,

especially when applied to large datasets or deep architectures, also consume substantial amounts of computational resources, making them impractical for smaller companies. Last but not least, the models could fail to recognize phishing sites that are not taking the usual shapes in the space of the training domain name.

2.3.2 Using Unsupervised Learning to Detect New Phishing Attempts

Phishing, in which fake messages are used to trick victims into revealing sensitive information, is an enduring and sophisticated cyber threat. For labelling phishing emails, the paper considered unsupervised ML techniques such as clustering algorithms. As phishing methods became more intricate, these methods use clustering algorithms, sophisticated feature extraction techniques, and large datasets to combat them. They primarily work in the context of clustering-based methods for phishing campaign detection.

Email-based, this study explored the applicability of K-means clustering in a decade-long phishing email collection. This study was to investigate the progress of phishing maneuvers and the distribution of spam/ phishing e-mails in terms of year, month, and day of the week. The data used in the experiments contained around nine thousand phishing emails. All emails had specific information such as sender, subject, message body, attachment, etc. The shape of clusters of phishing campaigns (tax info, bank account, and email ID (email validation)) became more apparent after clustering. Results of the study on phishing campaign timing will help to improve the accurate detection of new and shifting phishing threats *(Zawoad et al., 2013)*. Another study employs clustering techniques like Mean Shift and DBSCAN for the purpose of classifying phishing emails into campaigns. This is especially effective because phishers often use multiple email formats in order to circumvent spam detection, meaning that it can be infeasible for security researchers to identify all phishing campaigns. The subject line, URLs, sender attribute, and email body information were some of the features used that were extracted from the 60,000 phishing corpus for this study. Based on the attributes like email source domains and URLs, the research found that Mean Shift performed better than DBSCAN. It was also a huge help to IT, grouping like phishing emails together. The effectiveness of the algorithm as a tool for automatic incident response management was illustrated as it could form homogeneous clusters. Rather than

reviewing each system individually, teams could focus on larger clusters of emails *(Althobaiti et al., 2023)*.

URL-based, this study more advanced phishing detection system to further improve the semantic understanding of phishing URLs via leveraging large language models (LLMs), like BERT. Extracting URLs for embedding is a trivial task in this framework. Finally, clustering is enhanced based on these embedding. Clusters are created with higher coherence, and phishing campaigns are detected with higher accuracy in the LLM-enhanced solution. It is more intuitive to grasp the clustering process with the inclusion of LLMs due to the system containing visualizations such as scatter plots and explanations given with the help of keywords. As per *(Petukhova et al.,2025)*, this method makes the phishing detection systems clearer by helping to trace padding and padding content, and can even identify the majority of the novel phishing content, especially the ones made with generative AI.

Although the dataset of each study is not particularly shared with other studies, all these datasets with a large amount of phishing emails have their own characteristics. Here are a few common features taken from emails: time features, topic features, body features, sender/recipient features, URL features, and topic features such as length and words. The clustering approach is exploited for the recognition of real and phishing emails by extracting well-defined features, depicting clear patterns in the data. The application of unsupervised learning with these feature sets allows the identification of phishing attempts, in particular, when the general supervised learning frameworks are restricted due to the very small labeled datasets.

Clustering approaches improved phish detection systems according to researchers by grouping emails into “campaigns” which in turn made it easier to respond to incidents. By combining LLMs with sophisticated clustering algorithms, such as Mean Shift, the resulting system is capable of adapting to evolving phishing schemes, particularly those being driven by AI, and scaling in direct proportion to the volumes of email it must digest. Nevertheless, problems of high computational cost in dealing with high-dimensional data and feature reliance still exist. Nevertheless, there exists considerable room for enhancing phishing detection using unsupervised

machine learning methods. One that would offer IT teams a more efficient and scalable way to combat phishing threats.

These studies also have limitations, including the potential for incorrect clustering because of differences in phishing emails and difficulties in dealing with high-dimensional data in an efficient way. Although LLM has been incorporated to achieve better clustering results, the computing time of LLM is still suffering. Furthermore, the use of labeled datasets to evaluate the clustering results is impractical in terms of scalability and the inability to adapt to new phishing methods. Our future work will concentrate on further developing the effectiveness of clustering algorithms themselves so that they are more robust to such attacks and the algorithms scale well enough to more precisely deal with increasing phishing threats.

2.3.3 Using Recurrent Neural Networks (RNNs) to Detect Phishing

Phishing is still one of the most common and impactful threats to cybersecurity. And these programs put countless people and businesses at risk, as they trick consumers into divulging personally-identifiable information. There are several methods to identify phishing attempts, like blacklists and URL based Analysis; however, they are not sufficiently effective against the new phishing tactics. To overcome this, machine learning-based approaches, especially deep learning-based algorithms, are getting more attention for detecting phishing sites.

URL-based, for instance, the performance of CNNs, when applied to the detection of phishing websites, is illustrated by the work of *(Yerima & Alzaylaee, 2020)*. Their algorithm was trained using a dataset of 6,157 legitimate websites and 4,898 phishing sites. It obtained an accuracy of .0982 and an F1-score of 0.976, which performed better than other machine learning classifiers. This approach is so powerful because it can recognize features such as URLs and web material. This means it can detect new phishing sites that it has never visited before. In another study, a work attempts to show how CNNs can be leveraged to classify phishing URLs. This method transforms URL feature vectors to photos, and they use CNNs to cluster groups. A total of 1,353 URLs were included in the dataset for this analysis. They were divided into three types:

legitimate, suspicious, and web spoofing. The classification accuracy was .0865. What also characterizes our model is that it can digest raw URL features without much pre-processing.

The results reveal that deep learning, especially CNNs, is very powerful in identifying phishing emails. This obviates the requirement for handcrafted features, which may take time implement and are error-prone. CNNs and LSTMs enable models to learn new phishing tricks by consuming raw data. And they are therefore also expandable for real-time inspection.

URL and HTML-based, these approaches employ datasets compared to URLs, which have a different domain name, length, special characters, and HTML information, such as meta tags, and text. For instance, have made use of LLMs to enhance embedding for the classification of phishing. Their approach enhances the detection of phishing campaigns that undertake new phishing tactics and are directed towards specific companies and do so without the need for labeled data. *(Ariyadasa et al., 2020)* proposed LSTM and the CNN are two categorization models that use website content and URL features to identify the phishing attacks.

The datasets used in these analyses differ with respect to scale and features utilized for extraction. (*Yerima & Alzaylaee, 2020)* 10,000 websites; constructs its dataset from 1,353 URLs. These datasets are vital to the training of deep learning models, as their quality and diversity can affect the model's tendency to generalize and learn new phishing techniques.

However, these architectures are successful; still, they are not free from limitations. The main challenge is still to have adequate datasets with large enough labels to train deep learning models well. The models, furthermore, may find it difficult to spot very sophisticated phishing tricks, like those that rely on domain obfuscation or phishing websites that look too much like legitimate ones. The computation cost associated with training models based on deep learning techniques is also nontrivial, especially in resource-constrained environments.

2.3.4 Transformers-Based Models for Phishing Detection

Phishing attacks continue to be one of the most serious threats to online security; scammers can get hold of your money and identity with just a single click. The problem is that

because the tactics of attackers are always changing, current detection systems have a tough time keeping up with new phishing maneuvers. With recent advances in machine learning, and especially the transformer architectures models like Distilled Bidirectional Encoder Representations from Transformers(DistilBERT) and Robustly Optimized BERT Approach(RoBERTa), we have interesting tracks to be able to predict with high precision phishing emails and websites earlier than before. These attributes can effectively detect phishing attacks by taking URL characteristics and webpage content into account.

URL-based *(Asiri et al., 2024)* proposed Phish Transformer, which leverages a new methodology to detect phishing attacks by analyzing the text of a page as well as the URLs it contains. This model leverages a combination of CNN and transformer encoders to process webpage content and inclined URLs. Phish Transformer reduces the complexity of this model and improves the detection of phishing websites by considering the URLs that are contained in the webpage, such as links and Iframes. The model was tested on a dataset of 10,000 URLs, where it yielded an F1-score, precision, and recall of .099. This indicates that it is capable of capturing the newly appearing techniques to carry out the phishing attacks, for example, Browser-in- the-Browser (BiTB), and Clickjacking.

Email-based, this study also works to enhance the DistilBERT and RoBERTa models that classify phishing, spam, and ham emails. The data set used in this study was highly imbalanced *(Jamal et al., 2024).* It contained both phishing emails, spam, as well as valid emails (ham). Oversampling methods were used by the researchers to equilibrate the data set. Model performance of the fine-tuned model was superior to the performance of the baseline model, both when balanced and unbalanced. The Improved Phishing and Spam Detection Model (IPSDM) was more effective in classifying which emails were phishing and spam with high precision and recall.

In general these studies, the quality of the data collection and preparation is essential to the performance of the model. Usually*(Uddin & Sarker, 2025)* , datasets are classified as phishing and genuine emails or webpages, and features are derived from URLs, contents of emails, or layout of webpages. The Phish Transformer dataset *(Asiri et al., 2024)* was dominated by URLs in

webpages, including links and Iframes. Preprocessing included capturing the html and JavaScript information and converted it to a form that can be used in deep learning models. (*Uddin et al., 2024)* tokenized and resampled the email dataset to balance the two classes before performing analysis on it. In this way, the model would work well on other forms of phishing. Similarly, *(Jamal et al., 2024)* resorted to oversampling techniques to reduce the effects of unbalanced sets, increasing then the robustness of the model. These experiments demonstrate that transformer-based model can detect phishing emails. .099 for Phish Transformer, and *(Uddin et al., 2024)* fine-tuned DistilBERT was effective across all balanced and unbalanced datasets. *(Jamal et al., 2024)* IPSDM model also did considerably better than the baselines, showing impressive results that transformers based models can have in phishing& spam and ham versus spam email detection. These findings indicate that transformer-based models were able to address the difficult problem of phishing detection with practical applicability and high interpretability.

2.3. 5 Recent Advances in Ensemble for Phishing Detection

For enhancing phishing detection rate in email and web page scenarios, ML have evolved as the recent favorite choice. Ensemble models may use multiple classifiers to take advantage of the complementary learning characteristics for improving detection performance.

Email-based, in this study presented a hybrid ensemble model that combines logistic regression, random forest, and decision tree classifiers along with NLP based textual features for phishing email identification. Their method obtained an accuracy of around .0972, highlighting the benefit of integrating classical machine learners in ensemble based frameworks*(Ahmed & Hammad, 2026)*.

URL-based, this study proposed an ensemble stacking framework from a meta-learning perspective with the combination of a variety of base learners (XGBoost, CatBoost, LightGBM, random forest, gradient boosting, extra trees and SVM, and AdaBoost) to predict phishing websites. Their stacking ensemble model could achieve an accuracy of up to 1 on balanced data sets and performed better than the baseline models for imbalanced phishing URLs data with sharp detection*(Alamri et al., 2025)*.

In this work, a model stacking ensemble learning framework for phishing email identification is proposed, relying on header, content, and URL-derived feature combination. Their approach achieved the accuracy of around .0995 using ensemble learning with a high F1-score and shows that ensemble learning is well-suited for detecting different sets of phishing characteristics across feature spaces*(Connolly & Atlam, 2025)*.

Collectively, these works demonstrate the increasing popularity of ensemble and stacking-based learning mechanisms in the literature of phishing detection and their effectiveness in improving detection performance for different phishing scenarios.

Phishing is still really hard to detect in the security world because the attacks keep getting cleverer. Using CNNs for detecting phishing web pages is also explored by transforming URL-based features into images, which allows the encoding of website properties such as URL length, domain age, existence or absence of suspicious characters *(Kulkarni, 2023)*. While CNN-based approaches have demonstrated promising predictive ability, most of them cannot be directly used for predicting advanced phishing tactics like domain obfuscation and fake SSL certificates.

Unsupervised learning techniques such as clustering methods have also been studied in the light of phishing campaign detection. With the use of big data analytics, phishing emails can be clustered into campaigns to such an extent that IT team is not concerned with drilling down via certain level of high-level cluster analysis rather analyzing individual incidences for separation*(Petukhova et al., 2025)*. However, scaling the clustering methods to high dimensions and also evolving the techniques according to phishing attacks are challenging. Modern lines of research, as is demonstrated by*(Althobaiti et al., 2023)*, focus on-the-fly clustering schemes and large language models (LLMs) to improve the coherency of clustering results and the accuracy of detection.

For phishing URL detection, the structure and text characteristics of URLs have been modeled with RNN. Although RNN-based models are effective in real-time detection scenarios, they often need large amounts of annotated training data that can be expensive and time-consuming to obtain. Additionally, RNNs will have difficulties detecting phishing attacks that are significantly

different from those observed during training*(Rangapur et al., 2022)*. In response to these limitations, reinforcement learning and transfer learning are proposed as promising directions for enhancing the adaptability of RNN-based phishing detectors*(Ariyadasa et al., 2020)*.

There have also been some remarkable works on phishing detection utilizing the transformer-based approaches, like DistilBERT and RoBERTa. These models make use of massive amounts of data and context-aware representations to detect phishing content in emails and web pages. Nevertheless, the high cost of training transformer- based architectures and its reliance on big annotated corpora still present major obstacles. As noted, more investigations are needed for making transformer models compact and scalable for small dataset-specific problems and real-time applications.

The ensemble learning techniques have been widely studied in the context of phishing detection, in view that they can combine several classifiers to improve decision performance. Unlike the one-model-fits-all method, ensemble-learning algorithms like Random Forest, Gradient Boosting, and XGBoost work on different learning properties to provide more robustness in models and a better generalization toward dynamic phishing situations. Nonetheless, recent evidences show that the performance of ensemble predictions highly depends on the quality of feature construction and labeled data size. For instance, have shown that hybrid ensemble models with classical machine learning classifiers can obtain accuracy levels competitive with ours; however, performance significantly depends on the choice of the features to be used and may lack overall applicability in real-world scenarios.

Advanced ensemble designs using stacking have also been offered to enhance the detection performance by adding a meta-learning layer, which combine soft outputs from multiple base classifiers *(Alamri et al., 2025)* demonstrate that carefully-tailored ensemble stacking models can obtain very high detection accuracy, especially on balanced data, and outperform the baseline model on an imbalanced phishing URL dataset. However, these stacked-security features-based models are usually more complex and require an expensive hyperparameter tuning process, resulting in longer training time even with parallel training. Similarly, a stacked ensemble learning

technique for phishing email detection applied to the data achieves good results *(Connolly & Atlam, 2025)*, but hybrid feature extraction of multiple email items at run-time increases the computation overhead and might hinder deployment in resource-constrained devices. More generally, though costing much higher than the single classifier, ensemble and stacking models do improve performance to a certain extent. Literature has viewed the dual factor of detection accuracy system complexity as more and more clearly inversely related for hamburger classes in recent years, without question to lead itself toward efficient practical use.

In total various machine learning and deep learning techniques are presented to address the phishing attacks. Semi-supervised and unsupervised learning have shown a great deal of promise in decreasing dependence on large labeled datasets, given a small amount of annotated data. Furthermore, ensemble learning has demonstrated to improve detection performance by capitalizing on feature engineering, model diversity, and knowledge reuse rather than creating extra synthetic data.

For more advanced phishing attacks, hybrid architectures comprising CNNs, RNNs, and transformer-based models associated with complex feature extraction mechanisms, e.g., web traffic analysis and SSL certificate inspection, can encapsulate both structural- and context-related properties of malicious websites. Computationally, optimization can be performed, such as pruning and quantization to keep model complexity low for real-time deployment. To enhance existing systems designed to identify phishing, we exploit recent approaches for the detection of phishing on a locally gathered dataset containing local phishing and legitimate websites with the aim of refining the model for potential use in other settings and improving accuracy levels as well as real-time applicability. Through combining these techniques, we aim to enhance the detection effectiveness against phishing and provide reasonable computational overhead suitable for real-world implementation.

Last but not least, performance of phishing detection mechanisms is bounded by the quality and availability of diverse real-world datasets. This has been emphasized by *(Asiri et al., 2024)*. For training accurate models, it is important to have comprehensive datasets. Despite its high

computational cost, the transformer-based methods such as DistilBERT and RoBERTa are showing promising potential in mitigating the emergence of new patterns of phishing by enabling scalable real-time detection.

Although many advances have been made in phishing detection with the help of machine learning, deep learning, and ensemble-based techniques, there are some critical limitations in the current literature. State-of-the-art models currently focus on maximizing detection accuracy based on publicly available or artificially balanced datasets and seldom consider challenges in putting such models into use empirically, as real-world scenarios present data scarcity, class imbalance, linguistic variety, and overcoming evolution. Specifically, transformers and sophisticated ensembles are highly accurate, but to train these models large amount of labeled data and computation resources are required, which does not apply to real time or resource-constrained cases.

Besides, there are few studies addressing the generalization capacity of phishing detection models over heterogeneous datasets, specifically crossing from a global benchmark dataset to locally gathered or region-based data. The influence of cultural, linguistic, and contextual differences on the performance of the model has not been fully studied. What's more, semi-supervised, unsupervised, and transfer learning methods have been presented to alleviate the reliance on massive annotated corpora; their performance on practical phishing detection scenarios is not yet fully investigated.

Therefore, there is a definite necessity for a phishing detection system that can effectively trade off between high detection rate and computational efficiency, as well as being robust against new attack patterns and maintaining good generalization ability with different datasets. Bridging these gaps involves developing models that are evaluated on both locally and globally collected datasets to test for their generalizability, scalability, and real-world utility.

Table 2.1 summarizes a comparative overview of ML and DL based phishing detection mechanisms; datasets used, approaches taken, performance evaluation metrics employed, and limitations identified in the reviewed works are compared across these studies.

Table 2.1: Comparison of Phishing Detection Techniques Using Machine Learning and Deep Learning Models.

Paper	Techniques Used	Limitations	Evaluation Metrics	Dataset Used
(Kulkarni, 2023)	CNN	Needs labeled data, overfitting	Accuracy (.0865)	1,353 URLs (Phishing, Legitimate, Suspicious)
(Ariyadasa et al., 2020)	LSTM for URL analysis, CNN for HTML feature extraction	Needs large labeled datasets, struggles with sophisticated phishing techniques	Accuracy (.09834)	URLs and HTML data
(Asiri et al., 2024)	CNN + Transformer Encoders (Phish Transformer)	Requires large, high-quality datasets, and challenges with computational power for small businesses	F1-score (.099), Precision, Recall	10,000 URLs, including links and Iframes in webpages
(Rangapur et al., 2022)	RNN	Requires labeled data, may struggle with new phishing tactics	Efficiency in real-time detection of phishing URLs	Fraudulent URLs with a domain structure
(Alamri et al., 2025)	Optimized ensemble stacking models	High computational complexity, increased training time, sensitivity to model configuration, and hyperparameter tuning	Accuracy (up to 1.00), Precision, Recall, F1-score	Phishing and legitimate website datasets (balanced and imbalanced URL datasets)
(Connolly & Atlam, 2025)	Stacked ensemble learning using hybrid feature selection	Increased training complexity, dependency on feature engineering	Accuracy (.0995), F1-score (0.99)	Phishing and legitimate email datasets (header, body, and URL features)
(Adrita et al., 2025)	Random Forest, SVM, XGBoost	Limited to traditional ML models; relies on handcrafted URL and web features; may struggle with evolving phishing patterns	Accuracy (RF ≈ .097, SVM ≈ .096, XGBoost ≈ .095)	11,430 phishing and legitimate URLs with 87 URL and web-based features
(Al-Ahmadi et al., 2022)	PDGAN (LSTM as generator, CNN as discriminator)	Relies on URL data only, may miss content-based phishing threats	Accuracy (.09758), Precision (.09802)	Phish Tank, Dom Cop
(Rangapur et al., 2022)	LSTM, CNN	High computational cost, hard to deploy on low-resource devices	Scalability, the ability to detect new phishing tactics	Phishing datasets
(Halgas et al., 2020)	RNN, NLP	High computational cost, challenges in scalability	Ability to detect subtle phishing patterns in text	Phishing email dataset with text patterns
(Uddin et al., 2024)	DistilBERT Fine-tuning	Depends on labeled data, the computational cost of transformer models	Phishing Email Detection Accuracy, Explainability with LIME and Transformer Interpretation	Phishing and legitimate emails dataset
(Jamal et al., 2024)	DistilBERT, RoBERTa Fine-tuning (IPSDM)	Challenges with big annotated corpora, expensive training requirements	F1-score, Precision, and Recall on imbalanced datasets	Phishing, spam, and ham emails
(Ahmed & Hammad, 2025)	Hybrid ensemble learning (L R, RF, Decision Tree – NLP-based)	Depends on labeled data, performance is affected by feature engineering, and moderate computational cost	Accuracy (.0972), Precision, Recall, F1-score	Phishing and legitimate email datasets with NLP-based textual features

2.4 Conclusion

Phishing is still one of the most prevalent and harmful threats in cybersecurity, with attack methods becoming more sophisticated. Conventional detection approaches have been increasingly failing to deal with such advanced techniques, and thus there has been a growing interest in ML and DL based solutions. Models such as CNN, LSTMs, and transformer-based models, including DistilBERT and RoBERTa, have been proven to achieve superior performances in phishing attacks detection of emails, websites, and campaigns with increased accuracy and efficiency than most of all the existing solutions. Such methods are effective, however have a number of inherent limitations which one must be cautious about. Most of the deep learning techniques need a large amount of labeled data for training which is hard to maintain in the poisonous phishing domain because fraud attack patterns are changing rapidly. Moreover, performance overhead of advanced architectures, especially transformer-based models, and high computational costs also make it difficult to deploy them to resource-strained small organizations and edge devices. Ensemble learning is a practical and effective solution to overcome these difficulties. Ensemble algorithms enhance both robustness and generalization by exploiting complementary decision patterns among different classifiers, and they achieve a better performance than any of the component models. In this paper, ensemble models performed solid detection pertained to a small amount of similar corpus without the complicated computation overhead in deep transformer models, which are more applicable to practical phishing detection situations.

Even if deep learning models such as CNN and LSTM work well to capture structure information and sequential dependency features of phishing context, they are based on a single detection mechanism, which may fail to adjust themselves with new threats as they emerge. The multimodal detection with ensemble learning and deep learning combines these two approaches, resulting in a robust methodology that also overcomes the limitations of single models.

Thus, in future work, to explore scalable and effective spamming detection systems using ensemble learning with advanced deep learning models for local as well as global deployment. Such methods could be incorporated in the context of cybersecurity frameworks to provide better overall protection against known and emerging phishing

threats, thereby improving network security and protecting end users from constantly evolving attackers.

Chapter Three: Methodology

3.1 Introduction

In this chapter, we introduce the methodology employed for the implementation, testing, and validation of the proposed phishing-detection system. The approach was designed to provide a structured and transparent procedure of comparing learning approaches in different data conditions. The results were leveraged on two data sets: a global benchmark one and another that was collected locally, so as to capture web usage behavior from the Palestinian perspective. A combination of both data sets allows the model performance to be evaluated in internationally standardized as well as local "real life" conditions. The same pre-processing pipeline was performed on the two datasets in order to standardize feature representation and avoid structural and statistical differences that might introduce bias model. Upon data preparation, we trained and assessed several ML/DL models on the international dataset to determine the most accurate and optimal solution. This comparison was between neural architectures, transformer-based language models, and ensembles.

The experimental results showed that the ensemble model obtained the best performance and most stable trend in other evaluation criteria. Hence, this model was chosen for the final stage of the suggested method and used then on the Palestinian dataset to check its capability to counter more efficiently against local phishing patterns beyond the global benchmark. Our systematic approach design provides a solid ground to build our phishing-detection system upon, one that is robust and transferable to real-life cybersecurity needs.

3.2 Framework for Phishing Detection

We adopted a structured methodology in this work to integrate the global and local phishing data for constructing an effective detector. The workflow began with initial preprocessing, including data cleaning and label encoding, and continued to a set of feature-engineering routines whose purpose was to enrich the dataset through polynomial interactions and statistical summaries. For the deep learning experiments on the global dataset, we performed additional pre-processing steps: tokenizing of character sequences, padding to equal length, and reshaping data to be suitable for CNNs, LSTMs, and

transformer-based architecture data formats. The samples were split using stratified sampling. Then, a few models were trained, including DL architectures such as CNN, LSTM, and BERT- based ones, and a classical ML algorithm which I combined using an ensemble technique. Hyperparameter learning was employed to further train the ensemble models, so that the final training could be implemented with the optimal settings. The performance of all the models was evaluated using various metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. Transfer and local learning were further repeatedly processed with the best global model in Figure 3.1 to examine if a global model can generalize under different life scenario situations.

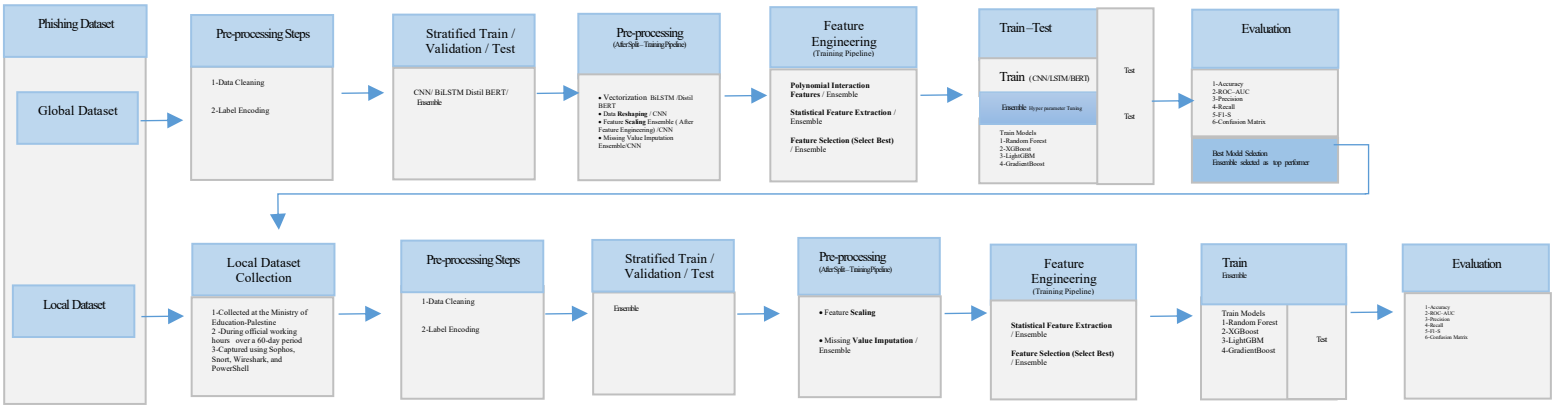


Figure 3.1 : The proposed Methodology.

3.3 Data Collection

The data used in this study was collected from two sources, and the procedures are presented in this section. The first source is a worldwide available dataset, containing URLs labeled as benign or malicious. A second source is from a database which was collected locally at the Ministry of Education in Palestine. The area statistical data were generated in two months working days by integrating security monitoring tools, with analysis system. This in this real-world dataset that we capture from our local sandbox represents an in-the-wild phishing situation.

The comprehension of both types of data sources make each learning from the complemental basis, receiving large scale signals of the global phenomenon and on local phishing behavior appreciation to estimate their generalizability.

3.3.1 Global dataset

The worldwide dataset adopted in this work is known as Web Page Phishing Detection Dataset (Version 3) and can be found at the Mendeleyev Data repository. It has been created, made available and published in Hannousse and Yahiouche (2021) with the following DOI: https://doi.org/10.17632/c2gw7fy2j4.3 *(Hannousse & Yahiouche, 2021).*

It is composed of 11,430 URLs which are all split equally into its positive and negative samples, 5,715 respectively as shown in Figure 3.2 to provide an equal representation of both classes.

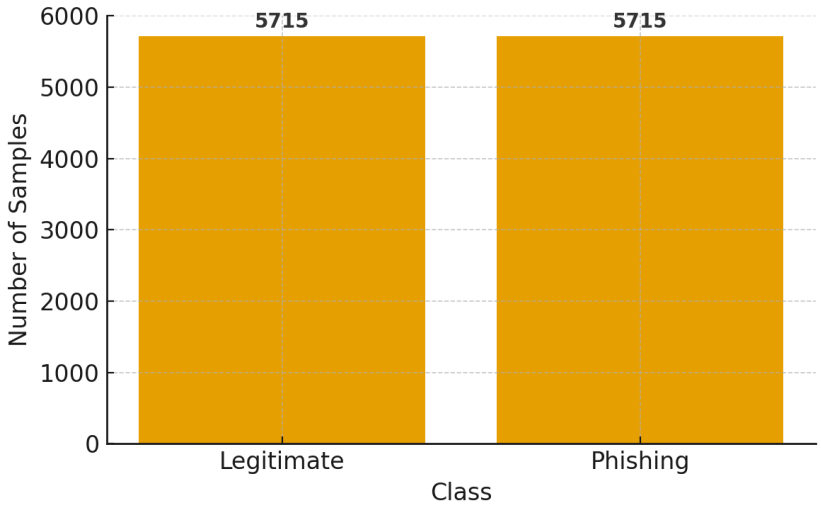


Figure 3.2**:** Phishing data sample distribution chart (global dataset)**.**

Each of the URLs is represented through a feature vector composed by 87 carefully engineered characteristics that reflect various dimensions of web property behavior and structure. Such characteristics can be divided into several classes, such as:

- URL-derived features (56): including URL length, domain structure, IP address in the center of a host name segment with varying suffixes and lengths,

number of parameters, patterns that correspond to special characters, and presence of HTTPS.

- HTML and content-level attributes (24): such as the number of scripts, objects, and internal/external links per page.
- details, expiration dates, WHOIS details and website ranking.

This dataset is of particular value due to its registry, feature richness, and correlation with current phishing methodologies. By taking into account lexical, structural, content-based, and behavioral features of phishing webpages, this system achieves realistic recognition of modern phishing techniques. It is therefore is no only a solid basis to develop machine-learning and deep-learning models but also gives researchers the opportunity to test such models in a controlled and realistic experimental environment. In order to better understand the world data, Figure 3.3 shows a comparison of URL length distributions between the worldwide legitimate and phishing servers. From the plot we can observe that legitimate URLs have smaller concentrated area of lower values while phishing URLS clearly have a bigger and longer stretched structure. This longer pattern fits to a common practice of the phishing attackers who add extra parameters or nested paths to lengthen URL in order to mask up their malicious contents .

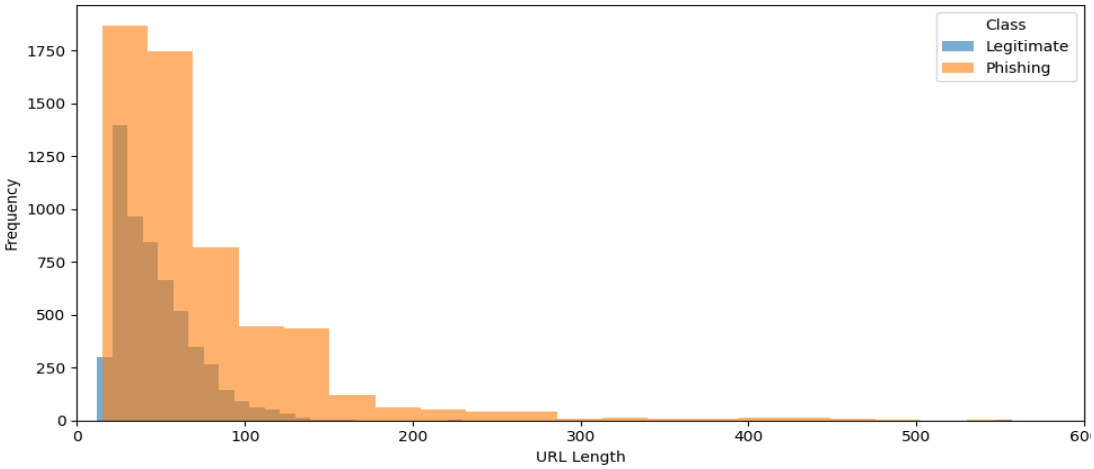


Figure 3.3: URL Length Distribution for Legitimate and Phishing Websites.

Overall, this distinction reinforces URL length as a meaningful and discriminative feature that supports the effectiveness of the feature-engineering process described earlier.

In the local dataset, not all classifications were initially fully available, so the classification process was supplemented with expert review in the field, along with reliance on trusted sources and known blocklists. This helped improve the accuracy of the classifications and ensure the reliability of the data used in training and evaluation.

3.3.2 Local dataset

One of the main contribution in this study is a local dataset that was obtained from Palestinian Ministry of Education's cyber security environment. . Every data instance in our dataset corresponds to a URL bedsides' 87 engineered features for each sample. These features were developed according to common sets used in global phishing dataset for detection and state-of-the-art techniques, and regarded as a baseline method to have consistency with other existing research. and is identified as phishing or not, without keeping webpage content, user comments or personal information.

Data collection was implemented using various security solutions that have been deployed across the ministry infrastructure, such as Sophos (for Web and email threat detection), Snort for intrusion monitoring or Wireshark tool to perform packet-level analysis; while a tailored PowerShell script injected on regular basis, relevant alerts and log data into the same structure. These two tools worked in conjunction to produce a constant stream of phishing related events that mirror real operations, which are shown in Figure 3.4.These systems were curated by a cybersecurity expert using alerts produced by these security systems. These alerts coupled with human verification enabled determination of the phishing and legitimate nature of each URL. All information was obtained and managed under confidentiality agreements, and utilized strictly for research. Personal identifiers (such as usernames, IPs or email content) were removed to ensure compliance with privacy and ethical research .

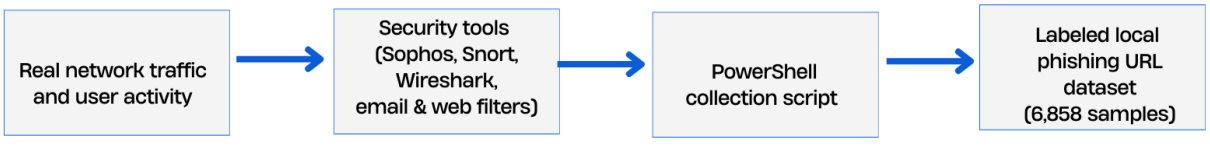


Figure 3.4: The process of collecting Local Phishing URL Workflow.

The dataset contains a total of 6,858 examples, being roughly balanced between the two classes general (3448) and phishing (3410). This natural balance, shown Figure 3. 5 supplies a descent learning condition for model training and data balances the bias of either class.

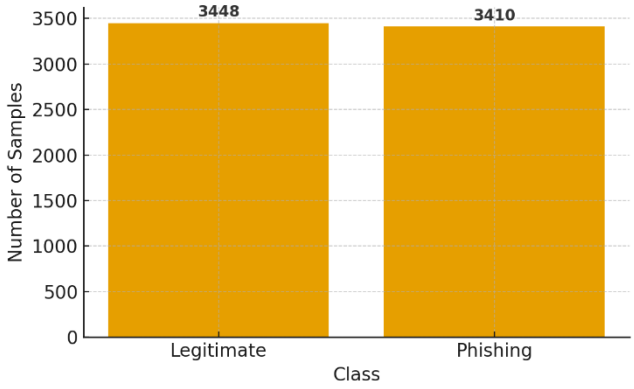


Figure 3.5 **:**Phishing data sample distribution chart**.**(local dataset).

Figure 3.6 shows correlation of top ten features, it confirms several intuitive structural and behavioral properties. Especially, the ratio digits URL feature is strongly linked to the link IP-based. Although some highly correlated features were observed, there was no feature removal at this stage; redundancy was handled in the downstream steps of ANOVA feature selection methods, and tree ensemble models proved to be robust towards correlated inputs.

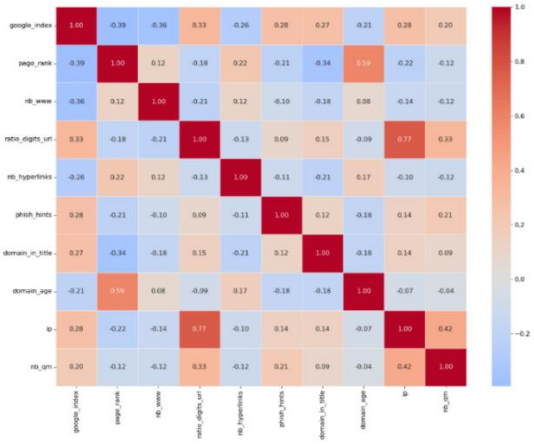


Figure 3.6: Phishing Data(Correlation Matrix For Top 10 Most Important Features)**.**

It enables the evaluation of model performance under local operating conditions, offering insights that cannot be captured solely through global, publicly shared datasets.

3.4 Data Preprocessing

Before discussing how we preprocessed, it should be noted that the input data are structured slightly differently for the various types of models used in this work. Sequence-based neural models such as BiLSTM and DistilBERT take URL inputs in the format of text sequences, which means each input URL is converted into a sequence of textual tokens. However, the traditional machine-learning model ensemble classifier and CNN each represent a URL as an 87-dimensional feature vector that is manually engineered to measure the structural and lexical features of the URL. We emphasize that this study does not use any webpage content or HTML features; all input is taken either from URL strings, in their raw text form, or as engineered numerical representations.

Data pre-processing is a basic step in constructing a phishing detection model since the raw URL data are not directly used for training before it becomes structured and learnable. We pre-processed the data in a way entirely consistent with what is actually required of our models.

The preprocessing routine of the phishing URLs aimed at extracting successful valuable numerical representations which help the models discriminate legitimate and malicious patterns. As phishing threats are widely reported to be due to the fact that minor URL features manipulation that phishers use can allow for more advance sophisticated URLs attacks, this is why (in the preprocessing step)it's been necessary to isolate informative characteristics and make sure these attributes are well-suited for Machine Learning & Deep learning models. These transformations made sure the data going into the models was in standard form and ready for feature engineering and model building.

Preprocessing is normally a basic lifetime step of constructing a working anti-phishing solution such as classification model, given that raw URL data can be hardly processed directly without being properly transformed to some structured format suitable to learning. Preprocessing was carried out to exactly match the demands of used models in

this work, so that each operation would be part of the training pipeline and no data leaking happened.

Data cleaning was performed as the first preprocessing step before splitting to data. Samples with missing or invalid URL strings or target labels were eliminated for the purpose of data integrity and consistency. This preprocessing was performed consistently on all the supervised models (CNN, BiLSTM, BERT and ensemble classifiers,).

The target variable was also transformed to a numeric value before data splitting through label encoding. The valid URLs were encoded as 0 and the phishing URLs as 1. Such transformation was needed to enable ensemble classifiers in generating decision boundaries, and for deep-learning models to estimate loss functions during training. This provided us with uniform target representation across all models, but still we had to convert the datasets from 3-class format to ‘1‘and ‘0’.

Stratified sampling was employed to split the cleaned and label encoded dataset as between the test set, validation set, and train. Split ratios were not exactly the same in order to guarantee the original class distribution between legitimate and phishing URLs, and make the final test a totally unseen set. All remaining pre-processing steps that need fit parameters were then performed on this split only. The data was divided in the first phase to ensure that information leakage was prevented between the training, verification and testing groups.

Missing value imputation was only performed on ensemble machine-learning models after data fractionation. For numerical features, missing values were replaced with the mean calculated from the training set only and this statistic was extended to the validation and test sets. This was necessary as ensemble-based methods including Random Forest, Gradient Boosting, XGBoost and LightGBM do not support missing numeric data.

Feature scaling was performed after data split to the CNN models for numerical stability during training .It is important to clarify here that the scaling process in the ensemble took place after the feature engineering, which we will discuss in the next section, Scaling factors were estimated using the training set only, and reused on validation and

test sets. LSTM and BERT were not scaled since these architectures operate on vectorized textual representations of text without first converting the data into raw numerical features.

For models that treat links as strings, URLs were converted to numerical representations after data segmentation. In the LSTM model, the links were converted to numerical sequences using a Keras tool trained solely on the training data to prevent information leakage from the validation or test data into the learning phase.

Subsequently, padding and truncation were applied to adjust the length of all sequences to 150 elements, ensuring a consistent input size and allowing the BiLSTM model to handle links of varying lengths uniformly during training.

In the DistilBERT-based model, the links were encoded using a pre-trained encoder after data segmentation. This resulted in input IDs and attention masks based on word subdivision. Because the encoder was pre-trained, it did not need to be retrained, and the encoding was consistently applied to the training, validation, and test data.

In the deep learning models of this study, URLs were treated as raw text without manual feature extraction. Therefore, there is no derivation of a specific number of features, nor is there any correlation analysis between features. The URLs are directly converted into a numerical representation using markup tools, allowing the model to automatically learn patterns from the textual data without the need for traditional feature engineering.

3.5 Feature Engineering

Polynomic feature interactions were incorporated in the training pipeline to cover additional numerical representations for ensemble's models. In addition, non-linear relations were captured between core phishing indicators (URL length, domain features and hyperlink-based features) by creating second-degree interaction terms. These interaction terms allowed tree-based ensemble classifiers such as Random Forestand LightGBM to model non-linear effects between the features, which would not be apparent if looking at each feature in isolation. This is especially true when applying the proposed framework to a global dataset. However, when applying the model to a local dataset, this approach was intentionally excluded because local data are more homogeneous, and adding

polynomial expansion does not add any real-value information and may lead to increased complexity and a higher risk of overlearning.

The original numerical features are used for statistical feature extraction in order to get compact and informative representations. Descriptive statistics, mean, standard deviation, minimum, and maximum, rounded off to the closest copybook, were used to describe characteristics of behaviors commonly seen on phishing URLs. These statistical features yielded robust handcrafted signals for ensemble classifiers to differentiate between benign and phishing URLs.

Feature selection is used as a post process to eliminate redundancy and select informative features for ensemble classification. Feature rankings according to relevance scores for the target were established using the Select-Best method, and then only top ranked features were passed to the final ensemble models. This latter action enhanced model efficiency by removing noise and reducing the risk of overfitting.

The processing line in feature-based ensemble models involves several systematic stages, beginning with feature engineering to extract properties derived from the relationships. Polynomial features are also created to enhance the representation of nonlinear relationships between variables. Feature selection is then applied to reduce redundancy and improve model efficiency, followed by feature scaling to adjust the range of values and ensure the stability of the ensemble models' performance during training.

3.6 Model Training

The trained models were evaluated according to a rigorous experimental protocol that guarantees fair comparison and data non-leakage, once the pre-processing job for feature construction has been performed. All learning strategies were trained via the stratified data splits, with model-specific settings used based on the provided code.

The CNN model was trained to capture spatial interactions between the numerical URL features. Numerical features were standardized using the Standard Scaler that was applied only on the training set after splitting. Reshaping: The reshaping method was

applied to the scaled feature vectors and used a custom strategy which tries to find near-square factor dimensions. Dimensions whose difference was greater than zero were zero-padded or truncated. The output tensors were organized as four-dimensional inputs which are ready for the convolutional layers. The CNN comprised a stacking of convolution and pooling layers, embedding output was fed to fully connected layers and sigmoid output unit for binary decision making. The model was trained with binary cross-entropy loss and the Adam optimizer. A stop condition was decided based on validation performance to limit overfitting and improve convergence.

The BiLSTM model was constructed along the patterns of the URL strings. In other words, URLs were considered as character-based text sequences (not numerical features). Finally, we also vectorized the URLs post data split but in this step we fit a Keras-based vectorization tool on the training set only. Padded or trimmed sequences were to a maximum length of 150 tokens in order to maintain consistent input dimensionality. The model had an embedding layer, followed by stacked bidirectional LSTM (BLSTM) layers and fully connected layers with sigmoid activation as the output. Training was performed by using the Adam optimizer with a binary cross-entropy loss. Overfitting was avoided by checking validation performance during training. No feature engineering, feature scaling or resampling is performed since the model learns representative from sequential raw inputs.

Instead of building a recurrent neural network-based model to learn context-aware representations for URL strings, Transform-based model was employed by DistilBERT to capture the contextualized representation of URL sequences. Following data splitting, we encoded the URLs using the pretrained DistilBERT encoder, which produces sub word token sequences and corresponding attention masks. Input sequences were padded or shortened to a constant length for doable tensor shapes. The sequence classification model DistilBERT is fine-tuned with the Adam optimizer and a suitable binary prediction loss function. As the encoder and embedding were pre-trained we didn't need to fit anything specific for this dataset during encoding. Validation-loss and -accuracy were tracked during fine-tuning to avoid overfitting and obtain convergence.

Ensemble learning was used as the main supervised method with numerical attributes from the GTLD set, which form numeric features. Several tree-based classifiers, such as Random Forest, Gradient Boosting, XGBoost, LightGBM, and were trained independently on stratified folds. After the split, missing value imputation and feature scaling were performed only in the training pipeline based on statistics calculated from the training set. Subsequently, an input to ensemble models would be obtained from feature engineering outputs. In particular, hyper-parameter tuning for the ensemble classifiers was conducted utilizing a random search method with cross-validation only on the training dataset. No resampling methods like SMOTE was performed. Model performance was assessed on a completely unseen test set based on the accuracy, precision, recall, F1-score, and ROC-AUC. The best ensemble model was chosen as the global classifier and tested on the Palestinian local .

3.7Hyperparameter Tuning Strategy

In this step, I performed hyperparameter optimization for the ensemble-based machine learning models to further enhance their effectiveness, using the same approach used in setting up the experimental code. We tuned the ensemble classifiers which are hyperparameter-sensitive: Random Forest, Gradient Boosting, XGBoost, and LightGBM. This specific focusing enabled a boost in performance, yet without unnecessary computations. Randomized Search CV was used for hyper parameter tuning with stratified 5-fold cross validation such that phishing and legitimate samples were equally divided between folds. In contrast to exhaustive grid search, the randomized strategy permitted a more effective exploration of the hyper parameter space, by sampling a fixed number of configurations for each model.

In the case of Random Forest, random search was used to look for different number of trees, maximum depth of a tree, features to consider when looking for the best split and node splitting constraint. For Gradient Boosting, the grid search on number of estimators, learning rate, tree depth, subsampling and minimum split and leaf size. For XGBoost and LightGBM, all selected configurations included when present random sampling over the boosting iterations, learning rate, tree depth and related controls and over feature/instance

subsampling parameters. The prediction accuracy using cross-validation was calculated for each sampled parameter configuration, and we automatically identified the best combination of hyperparameters in this optimization process.

Once having determined the best configuration for each ensemble model, their corresponding classifier was re-trained from scratch on the entire training partition based on the chosen hyper parameter setting. This permitted that the ultimate models were trained totally with the available data of training and learning from improvements observed in cross-validation. Finally, all pre-processing steps in the pipeline were fitted on training set only and no access to test set was allowed during tuning. Our investigations show average performance enhancement among the tuned ensemble models, indicating robustness of the final ensemble pipeline to hyper parameter tuning on global phishing dataset and emphasizing need for parameter tuning in tree-based methods. This tuning strategy was applied comprehensively to the global dataset and comprehensively to the local dataset.

3.8 Model Evaluation

Once the training stage was finished, the models were then tested on a hold-out test set. The scoring was based on a combination of supervised classification metrics that measure the system performance to successfully detect phishing URLs with as few false alarms on benign URLs as possible. To evaluate it in a more effective manner, we calculated numerous metrics skill as the accuracy, precision, recall, F1-score, specificity and AUC-ROC each offering different point of view about how good the model works. Confusion matrices were also produced in order to visualize the distribution of predictions over true positives, true negatives, false positives and false negatives that helps to gain more insight on the patterns of errors. Furthermore, ROC and Precision-Recall curves were observed to analyze its performance at varying levels of classification thresholds and assessing sensitiveness with respect to subtle aspects of phishing. These two evaluation techniques combined allow us to have both a thorough and fair idea of how well each model differentiate phishing from legitimate URLs, and ensure that CNN, BiLSTM,

DistilBERT and ensemble classifiers are compared fairly under the same testing environment.

- Accuracy: Is an overall measure of how effective a classification model is, then reflecting the high rate of correct predictions. It reflects the percentage of URLs classified as phishing or not phishing among all samples, including true positive and true negative prediction as shown in Equation (1). Accuracy only gives a rough overview on model performance but does not always reflect the overall detecting-quality in phishing cases, compared to the case of an imbalanced dataset.

$$\text{Accuracy} = \frac{\text{True Positives+True Negatives}}{\text{Total Samples}} \quad (1)$$

Despite its usefulness for a potential metric, accuracy is not always reliable especially in the case of dealing with an imbalanced dataset like phishing detection where the non-phishing websites are generally greater than that of phishing ones *(Chicco, 2017)*.

- Precision is how trustworthy the model while predicting phishing, I.e. if the website truly to be considered as phish. It indicates the ratio of phishing info to non-phishing info within all URL is the classifier classified as phishing, in other words a measure indicating frequency correctness. A high precision means the model has less false alarms because there are less legitimate sites being misclassified to the phishing sites, as shown in Equation (2).

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives+False Positives}} \quad (2)$$

For a high precision, this means the model makes a correct prediction when it says that phish and this is also very important in phishing detection because one does not want to block a clean website for phish *(Buda et al., 2018)*.

- Recall: Is the ability of the model to detect phishing websites when they are indeed present. It quantifies the proportion of genuine phishing URLs that are detected by the model; hence, it evaluates how well the missed attacks are reduced

using our model as shown in Equation (3). This is especially critical for phishing detection since the failure to detect a hostile webpage may result in severe risks.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives+False Negatives}} \quad (3)$$

The high recall means that the model can detect most phishing sites, which leads to the fact that it is hard for us to admit a phishing site and miss a phishing attack (false negatives). In tasks such as maintaining the one about phishing detection, emphasis should be given to recall since one prefers to decrease chances of an identified/predicted not fraud than increasing more false predictions *(Buda et al., 2018)*.

- The F1-score allows to give a fair evaluation of the classifier by combining precision and recall into one single measure. It demonstrates the compromise between the desirability of accurate detection of phishing websites and that of avoiding false alarms, which is an important information tool for analyzing the unbalanced dataset like phishing detection. Because it takes into account both the false positive and the false negative, F1-score provides a more descriptive measure than accuracy, as shown in Equation (4).

$$\text{F1} - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+ Recall}} \quad (4)$$

When detecting phishing sites, the F1-score is particularly appropriate because it discards having in practice available site-catching ability (the recall) of the model under consideration and not the precision with which it should catch its own measure*(Chicco, 2017)*. The F1-score can be understood, which is the higher its value, the better the model that gives us the balance between breakpoints and false positives.

- The ROC curve portrays a classification model's capability to discern between phishing and normal websites as discriminatory thresholds are altered. It is a graph of the true positive rate against the false positive rate, showing how the model performs at different levels of sensitivity. The AUC statistic, which takes a value between 0 and 1, is the size of the area under

this curve and provides a single quantitative measure of how discrimination depends on risk. The higher the AUC value is, the stronger the classification performance will be because it represents a better separation between phishing sites and legitimate ones than a random guess. Therefore, AUC-ROC is a popular robust evaluation metric in phishing detection as it provides invariance to the threshold when comparing models on consistent testing conditions.

- Confusion Matrix: The confusion matrix of true positives, false negatives (type 2 error), false positive (type 1 error), and true negative are used to analyze the performance of the classification model. It is also useful for understanding how the model classifies. The confusion matrix helps visualize:

1. True Positives (TP): Correctly classified phishing websites.
2. False Positives (FP): Legitimate websites incorrectly classified as phishing.
3. True Negatives (TN): Legitimate websites correctly classified as legitimate.
4. False Negatives (FN): Phishing websites are incorrectly classified as legitimate.

The Confusion matrix is also particularly important for identifying potential areas of improvement, such as reducing false positives or negatives *(Chicco, 2017)*.

3.9 Conclusion

This chapter introduced a fully implementation-oriented process to construct and evaluate a phishing-detection system based on deep learning, transformer architectures, and ensemble learning. The pipeline went from a structured pre-processing section of data cleaning, scaling, vectorization and reshaping into specialized training pipelines constructed to the exact needs of each model as specified in the code. The two algorithms each brought distinctive analysis perspectives:

- The engineered numerical features were phased as spatial patterns by the CNN model, which, in result, allowed learning local inter-relationships on URL attributes through convolutional filters.
- The BiLSTM was able to capture sequential and positiont positional dependencies between vectorized URL, allowing it to model how character sequences and their orders would contribute to phishing behaviors.
- DistilBERT used self-attention mechanisms to model the contextual dynamics across disjoint segments of each URL, capturing long-range dependencies that originate beyond mere sequential patterns.
- The ensemble models could capture interaction of order more than 1 between the engineered features with tree-based learners, which allowed learning over a larger space and fitted multiple decision boundaries to get fine, stable phishing detection performance.

The assessment of the model was conducted split using a variety of metrics comprising accuracy, precision, recall, F1-score, specificity, AUC-ROC, and confusion matrix. These two procedures combined to provide a fair evaluation of the models for identifying phishing attacks and prudent classification legitimacy websites and assisted in maintaining good reliability.

As a whole, this chapter establishes an open, technically sound, and truly reproducible experimentation framework for conducting an in-depth phishing detection study with strong practical relevance.

Chapter Four :Experiments and Results

4.1 Introduction

In this chapter, we have provided a thorough review of various phishing detection methods, which incorporate DL and ensemble techniques and neural models such as Convolutional Neural Networks (CNNs) for spatial features extraction and Bidirectional LSTMs (BiLSTMs) for sequential patterns learning. We also further examine transformer-based models, i.e., DistilBERT, as well as ensemble-based classifiers. We evaluate performance with a comprehensive range of performance metrics -accuracy, precision, recall, F1-score and AUC- combined with confusion matrices, ROC curves, Precision-Recall analyses and feature importance ranking. Combined, these comparisons serve as a strong baseline between the strengths and weaknesses of these methods, and it relevancy towards real world application in phishing detection.

In addition to the algorithmic assessments, this chapter describes aspects of experimental design and methodology that guided the implementation. Computational data processing steps with both machine learning and deep learning to enhance the classification. The entire classification process was implemented using Python libraries within the Jupyter Notebook environment. This approach allowed to directly control as much of the middleware work stages as possible like data processing, building model and learning parameters adjustment and performance evaluation. Specialized libraries like scikit-learn, TensorFlow/Keras were utilized to perform all steps of training and testing in a more flexible manner making the results reproducible with accurate analysis

The classifier type was selected based on the nature of the classification problem being addressed at each stage.

This study compares four classification methods, including our proposed approach, using more than just accuracy as a metric. We examined the results from different angles such as false positives, false negatives, sensitivity, specificity, and the balance between precision and recall to better understand each model's strengths and weaknesses. The work also fills an important gap in phishing detection research. Although previous studies used clustering, deep learning, or transformer-based models, very few have compared these

techniques side-by-side under a consistent experimental setup. Most past research focused mainly on accuracy and overlooked factors like training efficiency and how easy it is to interpret the model's decisions. By looking at robustness, stability, and practical usefulness not just the highest accuracy this study provides a more realistic view of which methods are truly effective for real-world cybersecurity applications.

4.2 Deep Learning and Machine Learning Techniques for Global Dataset Results

Global detection of phishing on web holdings demands for models that can capture structural, sequential and semantic patterns in URLs. These DL techniques from convolutional neural networks (CNN), bi-directional long short-term memory networks (BiLSTM), to transformer-based architectures such as DistilBERT incrementally enhance performances through spatial, temporal, and contextual modeling. Ensemble learning exploits these advantages, achieving state of the art performance via a family of classifiers and revealing the interconnectedness between algorithmic innovation and the natural difficulty of phishing data.

4.2.1 CNN-based Phishing Detection

In this section, we show experimental results using the Convolutional Neural Network (CNN) model for the phishing dataset. The model was tested with a series of performance metrics to evaluate its capability to differentiate between benign and phishing web sites. Quite pleasing to note is that in Table 4.1, the classification performance of CNN model reached an overall accuracy of **0**.8241, which is considered reasonable due to the structural features of inputs employed. In addition to accuracy, the model has precision of 0.7874, recall of 0.8880 and F1-score is 0.8347 which indicates this model has a balanced behavior for correctly recognizing phishing websites as well as keeping false alarm rate of legitimate URLs at an acceptable level.

These results support the observations of *(Kulkarni, 2023)*, show that simplified CNN models can obtain modest yet competitive performance of phishing detection on various datasets and feature representations, which supports the validity of our experimental results.

To get a better sense of these results, we show an in-depth visual inspection of the performance obtained by the CNN model in Figure 4.1. The confusion matrix of the CNN model against the test set can be seen in Figure 4.1(A). On the x-axis are the predicted labels, and on they y axis are all possible values; that is legitimate vs phishing URL. As illustrated in the figure, the model accurately identified 869 benign URLs and 1,015 phishing URLs, which indicates that our model has a good detection ability on malicious websites. At the same time, 274 real URLs were incorrectly classified as phishing (false positives), and 128 phishing URLs were misclassified as legitimate (false negatives). The second low false negative rate demonstrates that the model can reduce missed Phishing attacks, a crucial aspect in any phishing detection system.

The Receiver Operating Characteristic (ROC) curve is generated in Fig. 4.1(B), where the False Positive Rate (FPR) and True Positive Rate (TPR) are plotted as x-axis and y-axis, respectively. The ROC curve has an AUC of 0.9052 (i.e., very strong discrimination between benign and malicious URLs at various classification thresholds). The curve's steep upward training toward upper left-hand corner indicates that has a high true positive rate at low level of false positives.

The Precision–Recall curve is given in Figure 4.1(C) X-axis as Recall and Y-axis as Precision. The model attained a PR-AUC of 0.9034, suggesting an excellent balance between precision and recall in class imbalance scenario period. As shown in the figure, their precision is at a high level for recall values from 0.0 to .090, which demonstrates that the model can capture most of phishing URLs without much reliability loss.

The training history of the CNN model is shown in Figure 4.1(D), as a function of training epochs on the x-axis, and classification accuracy on y-axis. Both the training and validation accuracy curves are similar, rising steadily together, maintaining a tight separation from each other and ending up around .081-.083. Such a lack of visible gap between the 2 curves means steady convergence behavior and hints that our model is not learning to fit on the noise. This finding also proves the efficiency of introduced regularization methods and training method.

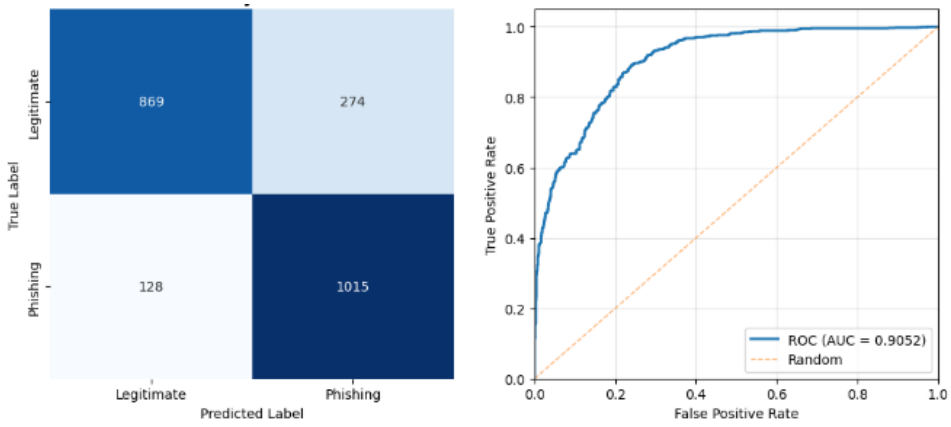
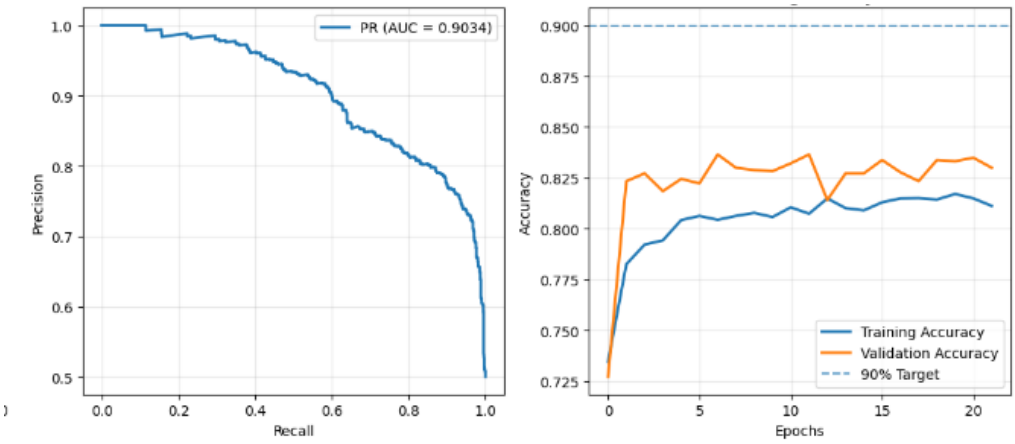


Figure 4.1:(A). Confusion Matrix (CNN). (B). ROC Curve (CNN).



(c).Precision–Recall Curve (CNN). (D). Training History (CNN).

The model's performance on the test set (2,286 samples) is summarized by the classification report above. The macro average precision, recall, and F1-score were 0.83 (precision), 0.82 (recall) and 0.82 (F1-score), the weighted versions of which were also 0.83 (precision), 0.82(recall) and 0.82 (F1-score). The nearly identical values of the macro

and weighted averages indicate that performance is relatively stable across classes, without any strong impact due to class imbalance. At the class level, for the legitimate class the model received an F1-score of 0.81 (precision 0.87, recall 0.76), and for phishing one this was equal to 0.83 (precision 0.79, recall 0.89). This demonstrates a good phishing URL detection ability (high phishing recall), maintaining a reasonable trade-off between phishing detection and overall precision/recall.

Table 4.1 Model classification report : precision, recall and F1-score for each category which help to gauge the balance between correct predictions and false positives is presented in below Table 4.1 Model classification report.

Table 4.1: classification report for CNN model.

	Precision	Recall	f1- score	support
Legitimate	0.87	0.76	0.81	1143
Phishing	0.79	0.89	0.83	1143
accuracy			0.82	2286
macro avg	0.83	0.82	0.82	2286
weighted avg	0.83	0.82	0.82	2286

4.2.2Phishing Detection Using LSTM Networks

For a complete assessment of the proposed Bidirectional Long Short-Term Memory (BiLSTM) model, various performance features are investigated regarding to learning behavior and classification power of unseen samples. The dynamics of learning behavior learned by model as it trains, and discrimination ability on the test set are shown in figure. 4.2.The training and validation accuracy versus epochs are shown in Figure 4.2(A). The training accuracy is initially about 0.84 and reaches close to 0.98, which indicates that our

model is learning sequential URL patterns well. Consequently, however the validation accuracy is quite constant around 0.91textendash 0.92 without noticeable decrease. The small gap between the two curves indicates relatively little overfitting. The training and validation loss curves are shown in Figure 4.2(B). The training loss plummets from around 0.38 to less than 0.10, which shows that the optimization and convergence processes are successful. Meanwhile, the validation loss seems to have stabilized between 0.21 and 0.25 with a small upturn in later epochs. This is a mild form of overfitting, common in deep recurrent network, but such divergence can be effectively controlled by early stopping to achieve convergence and training stability.

The confusion matrix achieved on the unseen test set is observed in figure. 4.2(C). The legitimate and phishing URLs were correctly classified by the BiLSTM model as 1,035 and 1,054 respectively, with 108 false positives while having just 89 false negatives. This well-distributed error curve indicates the model's capacity to achieve high performance in detecting phishing URLs, meanwhile with low false alarm rate for benign websites. The Receiver Operating Characteristic (ROC) curve of the BiLSTM model tested on test data can be found in Figure 4.2(D). The curve is well above the diagonal reference line with an AUC of around 0.976, suggesting good discriminative ability across various levels on the classification spectrum.

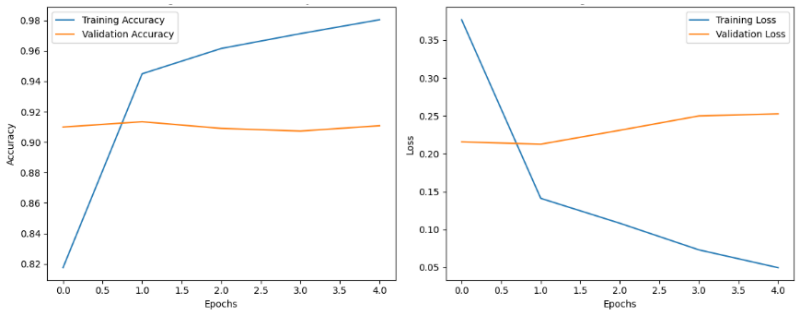
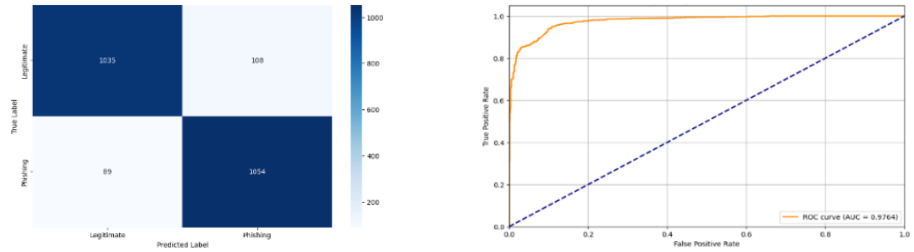


Figure 4.2 (A) Accuracy curves (B) Loss curve (BiLSTM) .



(C) Confusion matrix (BiLSTM) (D) ROC curve (BiLSTM).

The classification report additionally verifies the performance of the BiLSTM model. The performance of the model in the test set, which consisted of 2,286 samples, was an overall accuracy of 0.91. The legitimate and phishing classes had an F1-score of 0.91 with superior recall in the phishing class (0.92) making it sensitive for catching phishers. The proximity between macro-average and weighted average metrics refutes the notion that performance of the model are highly skewed and not affected much by classes distribution. specific numerical performance scores from the classification report are shown in Table 4.2.

Table 4.2: classification report for BiLSTM model.

	precision	recall	f1- score	support
Legitimate	0.92	0.91	0.91	1143
Phishing	0.91	0.92	0.91	1143
accuracy			0.91	2286
macro avg	0.91	0.91	0.91	2286
weighted avg	0.91	0.91	0.91	2286

4.2.3 Phishing Detection via Transformer Based Models (DistilBERT)

This section discusses the empirical results of the DistilBERT model used to detect whether a URL is phishing or not, on an out-of-sample test set for providing evidence that this model can indeed effectively classify legitimate and phishing URLs.

The results, shown in Figure 4.3, indicate that the DistilBERT model is capable of strong classification performance, performing at high accuracy and with consistent precision and recall between classes. The generation of confusion matrix reflects large

number of true classified samples, which means qualified phishing URLs detection and the controlled false alarm for legitimate website.

Additional examination based on the Receiver Operating Characteristic (ROC) analysis verifies the good discrimination ability of the model for a variety of classification threshold points, showing a high ROC–AUC value.

Figure 4.3(A) shows how the training process of the DistilBERT model progresses with respect to classification accuracy. The number of training epochs is drawn on x-axis and the classification accuracy are one the y- axis. Moreover, the training accuracy gradually increases from around 0.88 in first epoch to nearly 0.99 in last epoch, which reveals our model has sufficient ability to learn textual patterns through URL information. Validation accuracy seems to be more or less constant between 0.94 and 0.96 with little spread between the two curves. The model is good with mild overfitting, judging from the small discrepancy.

The loss values for training and validation sets vs. number of epochs are shown in Figure 4.3(B). It can be observed that the value of training loss decreases from about 0.27 in the 1st epoch to less than 0.03 in the last, indicating a continuous increase in optimization level of model. Simultaneously, the validation loss still stays firm between 0.15 and 0.17 with little raise only in later epochs . The lack of any spikes or sudden increases in validation loss indicates that the learning dynamics are stable and show there is indeed generalization behavior at play, rather than only memorizing the training data.

Visualization of classification outcomes for the unseen test set is presented in Fig. 4.3(C) The legitimate URLs were correctly identified, while 45 legitimate samples were misclassified as phishing. In like manner, 812 phishing URLs were well classified, while only 45 of the samples of legitimate URLs were wrongly classified. This fairly even spread of errors shows the model to be robust in its ability to find phishing sites as well as keeping a check false alarm rate for correct URL's. The Receiver Operating Characteristic (ROC) curve in Figure 4.3(D) shows the diagnostic capacity of DistilBERT model in terms of different decision thresholds. The AUC is getting to around 0.99, showing the model has good capability to differentiate phishing URLs from legit URL. The steep rise of curve in the upper-left corner means the high true positive rate and low false positive rate,

which also demonstrates that such model would be worked quite well in detection of phishing URL Request tasks.

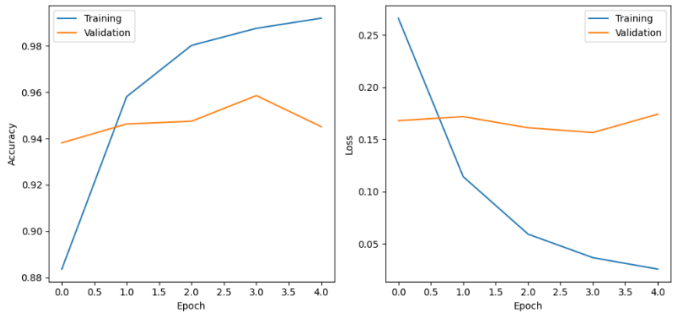
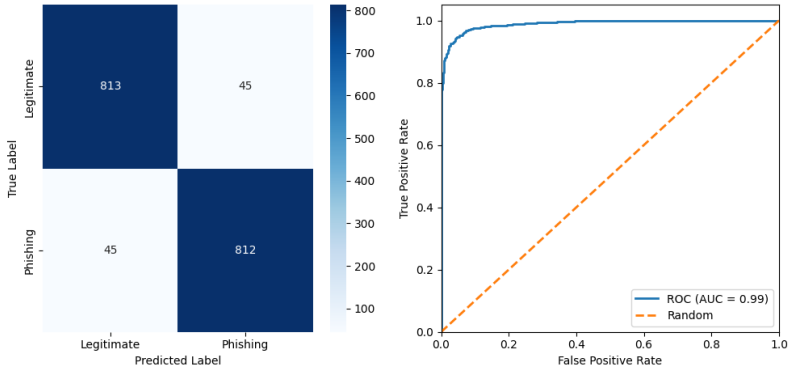


Figure 4.3:(A). Training accuracy (B). Training loss (DistilBERT).



(C). Confusion matrix (DistilBERT). (D). ROC curve (DistilBERT).

According to the classification report, the DistilBERT model had an accuracy of 0.95 on test set, which contains 1,715 samples. Precision, recall and F1-score of both macro and weighted averages all achieved 0.95, which demonstrated a highly consistent ability for distinguishing between legitimate class and phishing class, confirming the discrimination ability in instances involving multiple classes exhibited by the layered AUC.

At class level, the model yielded an F1 score of 0.95 for both benign and phishing URLs with same precision (0.95) and recall (0.95). This balanced result illustrates that the model is good not only in correctly identifying phishing URLs, but also maintaining a high precision for legitimate websites. In general, the results confirm that the DistilBERT model is robust enough and able to learn discriminative patterns directly from raw URL text in a reliable way. Specific numerical values extracted from the classification report is presented in Table 4.3.

Table 4.3: classification report for DistilBERT model.

	precision	recall	f1- score	support
Legitimate	0.95	0.95	0.95	858
Phishing	0.95	0.95	0.95	857
accuracy			0.95	1715
macro avg	0.95	0.95	0.95	1715
weighted avg	0.95	0.95	0.95	1715

4.2.4 Advanced Ensemble Learning Model for Phishing Detection

In this work, we presented a clean and verifiable pipeline to train phishing detection models using database through proper data preprocessing, tame feature engineering and sensible assembling algorithm fine-tune process resulting in the best prediction performance. We begin by loading the dataset and check systematically - We convert the target into binary representation (status), which is quite robust and we also fill in missing values for numerical feature set using mean interpolation. This dataset was partitioned for training, validation and testing (via stratified sampling) into 1,200 samples that were retained as validated test-set.

To increase the representational capacity by respecting data splitting, feature engineering was carried out in a combined scikit-learn Pipeline and fitted only on the training folds. This involved generating second-order-interaction-only polynomial features from a subset of the most informative input attributes and extracting row-wise statistical

descriptors (mean, standard deviation (std), min, max, and median) to describe the global behavior of each instance. Then, Select-Best based on ANOVA F-test was employed to select the most discriminative engineered features, after which each feature was Robust-Scaler normalized for numerical stability and resistance to outliers.

Contrary to up-sampling methods, there was no SMOTE or synthetic data creation used, and all the models have been trained based on naturally distributed data. Train and fine-tune with various ensemble-based classifiers (Random Forest, Gradient Boosting), also adding XGBoost or LightGBM with Randomized Search-CV with stratified cross-validation only on the training set was performed: A model was selected based solely on validation accuracy, and final evaluation was made only once on the held-out test set. The numerical results, presented in the next section, display that this disciplined approach provides an effective generalization as evidenced by high recall and strong discriminating power to away samples, which confirms the credibility of the proposed pipeline for phishing detection.

In general, CNN and DistilBERT exhibit mild overfitting, while BiLSTM peaked in a more stable way, with smaller train-validation gaps**.**

4.2.5 Comparison of tuning results for all ensemble models

In terms of validation, which is the main reference to select the model as shown in Figure4.4. The primary purpose of this figure is to assess and compare the generalization performance of different classifiers without taking their generalization performance on a testing set into account; the latter would require an additional analysis. With the combination of different evaluation angles such as judgment accuracy, type-specific judgment metrics, discrimination power, and incorrectness issue, this figure makes a model decision so that the model selection would not be sensitive in a particular evaluation perspective. Figure.4.4 (A) The comparison of the validation accuracy shows significant performance variety among the tested models. The validation accuracy of Random Forest Tuned was 0.9648, Gradient Boosting Tuned and XGBoost Tuned showed 0.9673 and 0.9692, respectively. LightGBM Tuned presented the highest validation accuracy, with

0.9746, and better predictive consistency, making this method a favorable option for further testing.

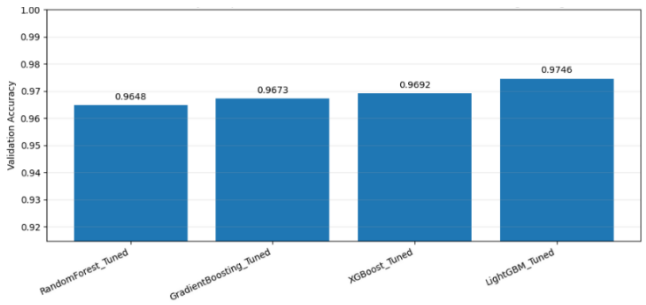


Figure 4.4 **:(A)**. Validation accuracy (Ensemble Models for Global Dataset).

In addition to accuracy, performance of the classification is analyzed based on precision, recall, F- 1 score**and AUC ,** Figure 4.4(B). All models exhibit excellent performance consistently ranging over 0.96 metrics, thus proving high reliability of predictions. It is worthy to mention that LightGBM-Tuned has a balanced trade-off between Precision and Recall with near-optimal AUC values, which is especially significant for phishing detection as missed attacks are costly.

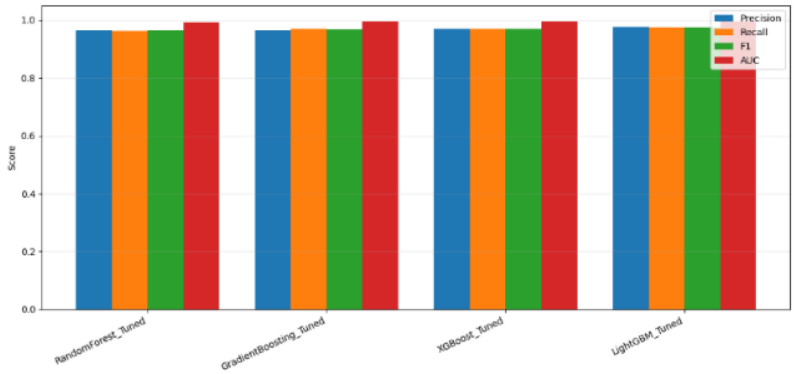


Figure 4.4**:**(B). Validation metrics (Ensemble Models for Global Dataset).

Tuned models' capacity to distinguish samples is **also well-illustrated** by ROC curves, as added in Figure 4.4(C). The curves are virtually indistinguishable and concentrated in the top-left region, with an AUC value of 0.993, which illustrates a good degree of separability between legitimate instances and phishing ones. Supplementing with the system retention of high precision values across wider recall regions on the Precision–

Recall curves Figure 4.4(D), it is further confirmed that our models remain robust under class-sensitive evaluation. Error rate analysis (Figure 4.4(E)) provides further insight into the stability and generalization of the model. The small distance between training, validation, and test error rates suggests that the models do not overfitting to the training data. This correlates to LightGBM-Tuned attaining the lowest validation/test error and also reflects its good generalization performance as well.

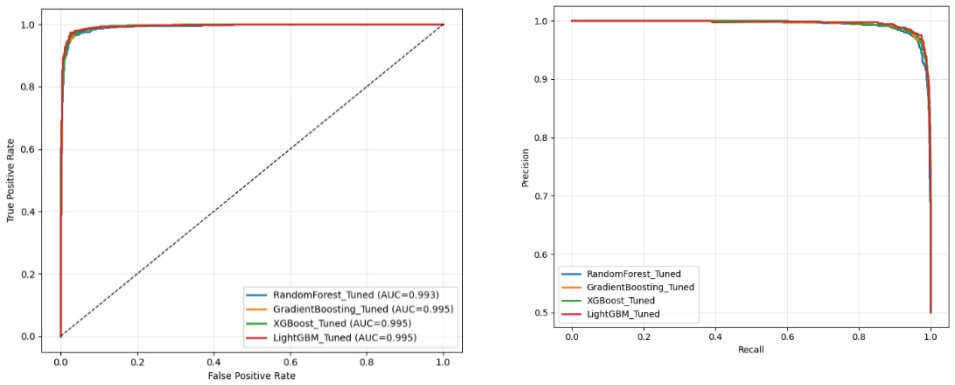
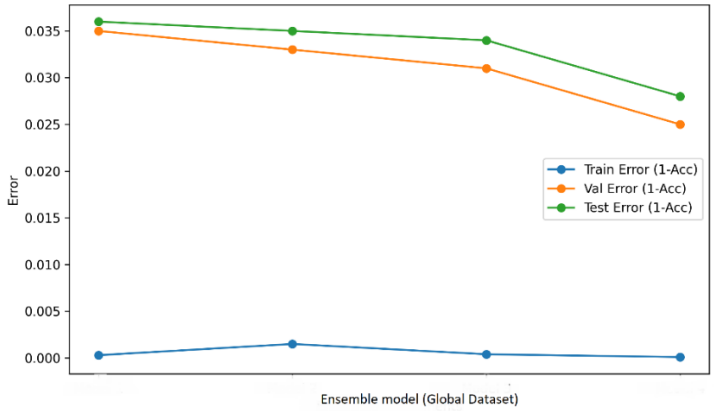


Figure 4.4**:**(C). Validation ROC curves (Ensemble Models for Global Dataset).

(D). Validation PR curves (Ensemble Models for Global Dataset).



(E). Error rates across models (Ensemble Models for Global Dataset).

Altogether, the general overview of Figure 4.5 determines unambiguously that LightGBM-Tuned is the best model according to validation-based evaluation and thus should be evaluated on independent data as a final practice.

Figure 4.6 is to highlight the ending evaluation of the model. The aim of this figure is to measure the absolute model performance in reality when all training and meta-feature selection choices are frozen without introducing any biases into an estimate of generalization ability. The learning curve in Figure 4.5(A) is the plot of training set size against model performance. Even if the training accuracy is kept close to 1.00, the cross-validation accuracy grows steadily from around 0.944 up to 0.971 with increasing numbers of training samples included. Indeed, such a convergence trend reflects a stable learning behavior and hints that the model actually succeeds at taking advantage of more external data without overfitting. The test summary confusion matrix is shown in Figure 4.5(B), which further breaks down how the test set has been classified. Test results show its performance is close to the real situation, with 581 legitimate and 585 phishing classified right out of 1,200 test samples, with only 19 false positives and 15 false negatives, therefore achieving a testing accuracy of more than 0.9717 or at least equal to it. The minimal number of misclassified phishing cases indicates the effectiveness of the proposed model for security-aware applications. Additional discrimination analysis is given by the ROC-curves in 4.5(C) comparing validation and test performance. The similarity of the curves, with AUCs of 0.993 (test), indicates a robust classification behavior on new data. The corresponding Precision–Recall curves in Figure 4.5(D) demonstrate a consistent precision maintenance under various recall levels, which indicates the stability of our model with regard to different working thresholds.

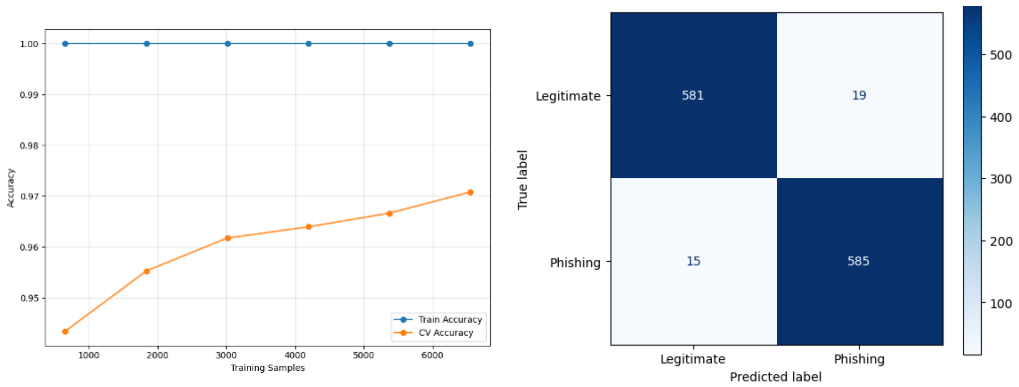
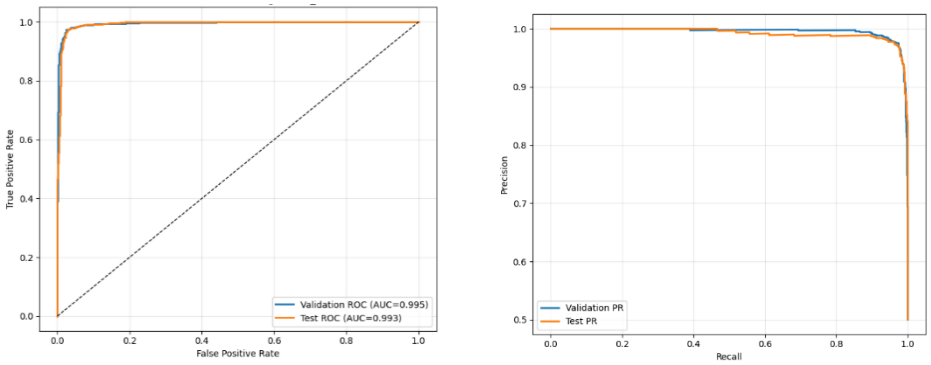


Figure 4.5:(A). Learning curve (Ensemble Models for Global Dataset).

(B). Test confusion matrix (Ensemble Models for Global Dataset).



(C). ROC curves (validation and test Ensemble Models for Global Dataset).

(D). PR curves validation and test (Ensemble Models for Global Dataset).

We can conclude that the results shown in Figures 4.4 and 4.5 reveal high generalization ability, strong recall capability, and discriminative robustness of the proposed phishing detection pipeline proves its effectiveness. Finally, the classification report in Table 4.4 revealed a very good overall success of the chosen model on test data. The proposed model has a precision of 0.9685, a recall of 0.9750, and an F1-score of 0.9718, which indicates that it achieves a well-compromised balance between detection accuracy and reliability. This is evidence that the model designed can distinguish effectively between phishing examples and show good generalization performance.

Table 4.4: Classification report of the model evaluated on the test set (Ensemble Models for Global Dataset).

	precision	recall	f1- score	support
Legitimate	0.97	0.96	0.97	600
Phishing	0.96	0.97	0.97	600

Accuracy			0.974	1200
macro avg	0.97	0.97	0.97	1200
weighted avg	0.97	0.97	0.97	1200

4.2.6 Phishing Detection Using Advanced Ensemble Learning Models for Local Dataset Results.

In the collected results, we performed yet another analysis with a locally obtained testing set to examine whether ensemble models tuned for globally were able to generalize to new data. The local data used the same primary learning pipeline and validation protocol, but did not include global-specific optimization steps such as hyperparameter search or polynomial feature expansion.

Such class balance in the local dataset was retained using stratified sampling, with no resampling or synthetic data generation introduced, thus preserving the source distribution's natural statistics. From the practical perspective, this controlled experiment setup allows for a realistic and computationally commercial vs. effective overview of how well models trained globally generalize to local operational conditions. Despite being so powerful and robust, can inherit complexity from multiple sub-models that it aggregates to tackle individual weaknesses of each sub-model. The overfitting risk in such models is reduced through diversity pooling across different components.

Figure 4.6 displays a validation comparison with the globally optimized ensemble models, performed on the local data set. This is, however, not a ranking of models but how well models behaved and certainly at the shakedown model performance. We then investigate whether or not the learning behavior learned on a global dataset is preserved locally.

Figure 4.6 (A) presents the validation accuracy of the tested ensemble classifiers, Random Forest, Gradient Boosting, XGBoost, and LightGBM, with their parameters fixed by the hypermeters obtained from the global optimization phase. On the X-axis we have

the classifiers tested, and on the Y-axis we see validation accuracy. It can be observed from the results of Table I that the performance of models is very similar among all classifiers, where the accuracy values are reported between 0.9638 and 0.9647. The best validation accuracy is obtained by Gradient Boosting and XGBoost (0.9647), closely followed by Random Forest and LightGBM (0.9638). This small difference in performance indicates that the local dataset has uniform structural properties among learners of the ensemble when performing after no more tuning.

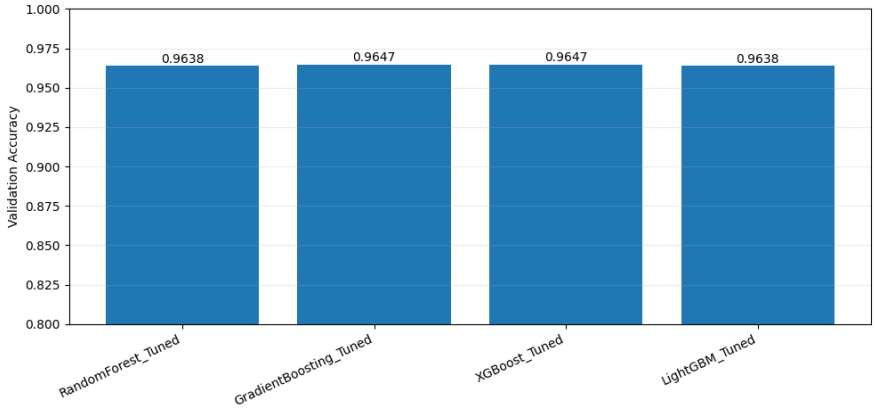
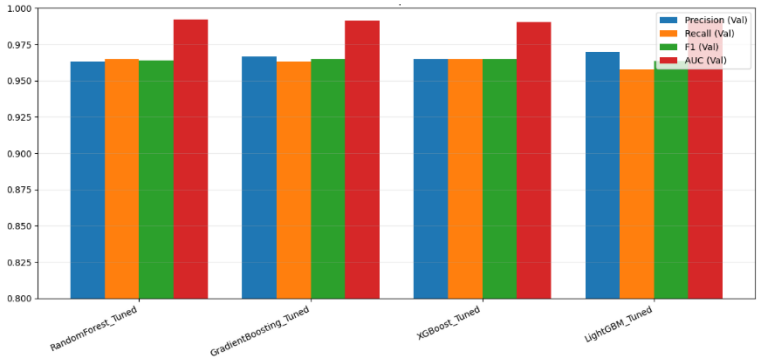


Figure 4.6**:** A**)**.Validation accuracy(Ensemble Models for local Dataset).

Figure 4.6(B) presents a comparison of validation metrics such as Precision, Recall, F1-score, and AUC. This visualization aims to inspect the quality of the classification beyond mere accuracy. The metrics of all models indicate relatively high values with AUC standing near 1.0, suggesting high discrimination power between phishing and benign objects on the local dataset.



(B). Validation metrics (Ensemble Models for local Dataset).

The cross-validation results indicate that the models' performance is similar on the **local dataset**, as we can only observe marginal accuracies and classification metric differences. This validates that the proposed pipeline is robust using locally acquired data.

Figure 4.7 shows the final model evaluation on a held-out test set of the **local dataset**. This phase offers an unbiased evaluation of the generalization performance of a model, once all training and selection choices have already been determined.

Error rates (1 accuracy) over training, validation, and test are depicted in Figure 4.7(A). Where the x-axis is the data splits, and the y-axis shows the error rate. The values found for them are 0.0000 (training), 0.0353 (validation), and 0.0333 (test), which results in a test accuracy of around 0.9667. The proximity of the validation and test errors indicates they are representative of stable generalization, while the zero training error demonstrates that the model has enough capacity to perfectly fit the training data. Figure 4.7 (B) Show the Error curve for the various data represent the error through different portion of data It can be observed from this feature that for smaller value of iteration, we obtained higher rate of error; but as we extended more iterations we got an improved result to reduce our future prediction errors over all along training Proving convergence in decreasing trends also which propagates several time in entire loop. The error grows from the training set to the validation, decreases a little on the test (i.e., it holds that the model does not exhibit catastrophic forgetting in unseen samples).

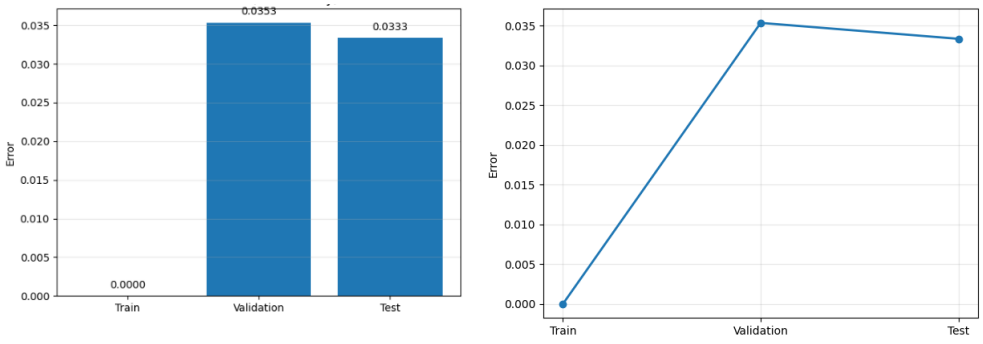


Figure 4.7**:**(A). Error rates(Ensemble Models for local Dataset).

(B). Error trend(Ensemble Models for local Dataset).

The confusion matrix on the test is reported in Figure 4.7(C). On the horizontal axis are predicted labels and on the vertical, true labels. The model achieved the accuracy of 577 legitimate and 583 phishing samples, which were correctly classified, while20legitimatesampleswereclassified incorrectly as phishingand20phishingsamplesmisclassifiedaslegitimate. 1,160 out of the 1,200 test samples were classified correctly, which **equates to** a test accuracy of 0.9667. Crucially, the phishing class has very few false negatives, implying good coverage of malignant examples. ROC curve on the test set is shown in Figure 4.7(D). The FPR is represented by the x-axis; TPR, the y-axis. The model has an AUC of 0.9937, suggesting that it possesses excellent discrimination ability over a wide range of thresholds. We present the Precision–Recall curve from our testing data in Figure 4.7(E), which is especially informative here for phishing detection. This curve has an AP of 0.9932 and shows that at all recall values, even close to 1, the precision remains high, proving the robustness of the model with class-sensitive evaluation. The test results on local dataset indicate the high and consistent performance with a test accuracy of around .09667, balanced error characteristics, and good discriminative ability as shown by high AUC and AP values. A brief and quantified summary of the final model performance (on the local dataset) can be found. in a detailed classification report for the held-out test set.

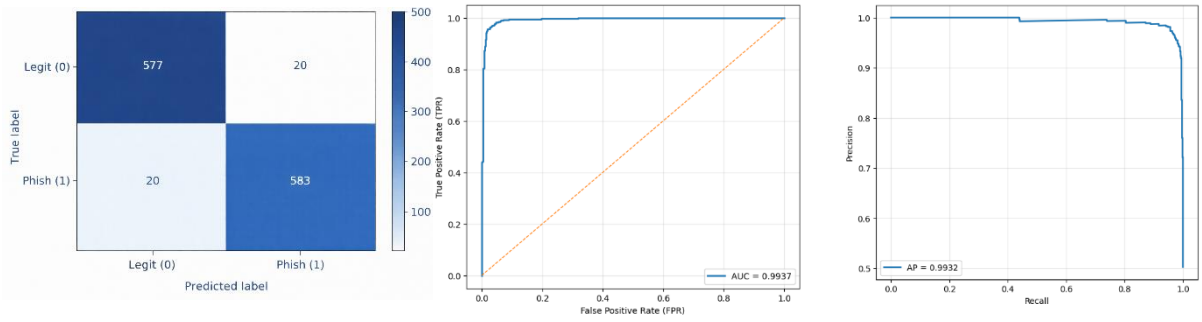


Figure 4.7(C). Confusion matrix(Ensemble Models for local Dataset).

(D) ROC curve(Ensemble Models for local Dataset).

(E). Precision–Recall curve(Ensemble Models for local Dataset).

This work provides a class-wise assessment of the model's predictive capabilities that complements the visual analyses, while giving out numerical performance measures. The classification results of the final selected model are shown in Table 4.5, class 0 being legitimate instances and class 1 representing phishing instances. This model has a good trade-off between precision and recall for both classes, which means that it can consistently discriminate the legitimate samples from phishing ones. For example, the phishing class achieves a precision of 0.9668 and recall of 0.9668, which demonstrates that our model has a high capability to correctly distinguish malicious instances at a low false alarm rate. A total accuracy of 0.9667 also confirms that the proposed method is robust and reliable against unseen local data.

Table 4.**5**: Classification report of the model evaluated on the test set (local Dataset).

	Precision	recall	f1- score	support
Legitimate	0.966	0.966	0.966	597
Phishing	0.966	0.966	0.966	603
accuracy			0.966	1200
macro avg	0.966	0.966	0.966	1200
weighted avg	0.966	0.966	0.966	1200

Each of the above result tables shows that not all compared models have been tuned using the same data splitting. Different model families have different requirements for training. Due to the nature of deep learning models, having explicit validation sets is necessary for early stopping and convergence testing; hence, we split the data into a 70/15/15 train/validation/test split in the case of DistilBERT, 70/10/20 for BiLSTM, and around 64/16/20 for CNN. The difference is that the stacking machine learning models were optimized with stratified k-fold cross-validation on the training set, and they were relatively less dependent on a huge independent validation subset. Stratified sampling was

used where applicable, with a fixed random seed in all of our experiments, and the test set was strictly held out and never reused more than once for final evaluation to avoid any optimistic bias .Further, the local dataset was processed in a different way than the global dataset because it represents a novel real-life Palestinian cybersecurity environment. Thus, the published results were obtained with different training and validation configurations but can be compared based on each model's learning needs and the purpose of the experiments. From a cybersecurity perspective, a high recall rate is important because false negative values mean phishing attacks go undetected. Furthermore, achieving a high recall rate and accuracy in an ensemble model indicates better protection in practice.

To make a transparent evaluation and comparison, Table 4.6 presents the input type and data split setup we utilized for each model, the dataset sizes, and these models' final performance.

Table 4. 6: Master Summary of Experimental Settings and Evaluation Results

Model	Input Type	Split Ratio	Train Size	Val Size	Test Size	Accuracy	Precision	Recall	F1
CNN (Global)	87 engineered URL features (numeric reshaped)	~64 / 16 / 20	7,315	1,829	2,286	0.82	0.83	0.82	0.82
BiLSTM (Global)	URL as text sequence	70 / 10 / 20	8,001	1,143	2,286	0.91	0.91	0.91	0.91
DistilBERT (Global)	URL as text sequence (transformer tokens)	70 / 15 / 15	8,001	1,715	1,715	0.95	0.95	0.95	0.95
Ensemble (Global)	87 engineered URL features (numeric)	CV on Train + fixed Test	10,230 (CV)	–	1,200	0.974	0.97	0.97	0.97
Ensemble (Local)	87 engineered URL features (numeric)	~64 / 16 / 20	4,526	1,132	1,200	0.966	0.966	0.966	0.966

Some models performed better than others, but it is not clear whether they can be considered "definitively best" since we evaluated each model on different input representations, data splits, and training protocols.

4.3 conclusion

In summary, the results of the experiments in this chapter have all shown that ensemble learning models deliver the most reliable and robust classification performance for detecting phishing. The performance of the MME was found to be stable and better than separate models in each measurement. When conducting the same ensemble-based procedure for the locally **constructed dataset**, its strong and consistent performance was sustained, demonstrating that it can cross-generalize beyond global benchmark data. The very high correlation between validation and test operations, as well as between precision and recall, demonstrates that the model can successfully discover discriminative phishing patterns without over-fitting. In general, the results of this chapter provide a solid empirical evidence that ensemble learning can be a manifest solution for phishing detection as systematically illustrated through this section.

Chapter Five: Discussion & Recommendations

5.1 introduction

This chapter compares four different phishing detection methods: CNN, BiLSTM, DistilBERT, and Ensemble Learning on the global and local datasets. The results demonstrate a substantial and consistent increase in performance by extending the complexity of the model. The deep-learning-based models exhibit an increasing level of discriminative power.

Additionally, the study reveals changes between global benchmark datasets and local samples, with a change in model response. The above findings suggest that the objective of this work is not just making a more accurate model but also constructing such a robust and general phishing detection system ready for deployment in practice.

5.2 Evaluation of Model Performance.

This section presents a comprehensive evaluation of the proposed models using multiple performance metrics. The observed differences in results can be attributed to the distinct learning mechanisms and representational capacities of the considered models. The Convolutional Neural Network (CNN) application is limited when dealing with structured and sequential data, such as URLs and content. Although CNNs are ideal for capturing spatial patterns, their fixed-size representation tends to limit performance in encoding the long-range dependency inherent in phishing-related data.

On the other end, BiLSTM involves two sequence model representing forward and backward sequential learning, making it able to better capture the contextual dependencies with textual input features. This leads to better performance for phishing detection, especially in recall, which is more important in malicious instance spotting. However, it still depends on word-level representations and is not as effective as the recent advanced language models.

In addition, we can take advantage of the fully pertained context-aware language representations by distillation. This enables the model to learn fine-grained semantics patterns, thus providing better performance than existing deep-learning networks. However, such performance improvement comes with a higher computational cost, which limits its application in resource-constrained scenarios.

In general, the Balanced Ensemble Learning shows the optimal and most robust performance across all methods tested. The ensemble model consolidates various base classifiers and diminishes individual model bias as well as variance, leading to better generalization and consistently robust performance not only on global but also local datasets. The lack of an obvious performance gap between training and testing stages also verifies that the proposed models are not overfitting, thus corroborating the rationality of our experimental results.

5.3 Comparative Analysis between techniques.

Classification performance of all the evaluated phishing detection models on unseen data. To report the overall classification performance of the considered models, we present a summary table (Table 5.1 for details), which entails several standard evaluation metrics (accuracy, precision, recall, F1-score, and AUC) computed on a hold-out test set. These values are the weighted average of performance of classes corresponding to each model, according to which a comprehensive and unbiased evaluation can be made.

The evaluation results show the evident performance ranking between different methods. CNN model has the poorest overall performance mainly because of its incompetence in capturing more intricate sequential and semantic features embedded within phishing data. On the other hand, with the BiLSTM model, all the metrics are improved considerably indicating that bidirectional sequence modeling indeed owns power in capturing contextual dependencies.

Finally, we observe that DistilBERT performs better than other deep learning-based models (CNN, BiLSTM) in the context of URLs, exploiting context-dependent language representations to learn sequential and semantic features. Yet, the ensemble models obtained the best global result in terms of all MI measures, demonstrating clear superiority in terms of classification balance and discrimination. This superiority can be interpreted by the ensemble's ability to integrate complementary abilities of many classifiers – Random Forest, Gradient Boosting, XGBoost, and LightGBM, where each model features different structural and statistical representations of the data. By averaging the predictions, ensemble models mitigate individual model biases and variances, thus resulting in robust generalization.

Additionally, in spite of this performance gain, there are differences between these ensemble methods and transformer-based models with respect to computation cost and service deployment. DistilBERT is potent in representation learning but expensive to learn and deploy as DistilBERT has high computational demands (e.g., GPU, memory, response time, etc.), therefore may be less scalable for real-time or resource-limited scenarios. While in production, ensemble models will often have quicker inference times, lower memory overhead, and a smaller footprint for running on average servers serving millions of client systems, thus they are better suited to the challenges of large-scale or real-time phishing detection. Thus, while achieving the best performance overall in this study, when to use ensemble versus transformer-based models should be determined not simply based on accuracy but also on latency, computational cost, and deployment requirements for a given operational context.

In essence, the performance trends reported in Table 5.1 support the model complexity succession and underpin the ensemble strategy as a superior yet reliable approach to phishing detection for our purpose.

Table 5.1: Performance Summary.

Model	Accuracy	Precision	Recall	F1-Score	AUC
CNN	0.82	0.83	0.82	0.82	0.90
BiLSTM	0.91	0.91	0.91	0.91	0.97
DistilBERT	0.95	0.95	0.95	0.95	0.99
Ensemble	0.97	0.97	0.97	0.97	0.993

5.4 Discussion with Related Work.

The comparison given situates the proposed work within existing literature regarding recent studies on phishing detection using deep learning, and ensemble approaches.

Instead of emphasizing incremental improvement in binary accuracy, this document concentrates on methodological design choices, learning paradigms, and the degree to which different approaches can generalize across varied data domains.

Many existing works showed a high level of detection accuracy in laboratory environment. For instance, *(Ariyadasa et al., 2020)*and *(Al-Ahmadi et al., 2022)*, respectively, obtained superior performance using hybrid CNN–LSTM architectures and adversarial data augmentation. These methods achieve good performance with respect to detection, but their evaluations are almost all performed on small modalities of data or synthetic URL corpora. Instead, our approach is end-to-end with fusion of structural, sequential, and contextual learning models and results in more stable behavior on varied and realistic datasets.

Transformer-based methods like the work by *(Asiri et al., 2024)* and the interpretable model of *(Uddin et al., 2024)*, which also enhanced phishing identification by considering advanced attack conditions as well as rendering the model more understandable. Although these algorithms work well, they may be computationally demanding and require large, high-quality datasets. Experiments conducted using the global benchmark and also local field data show that the proposed ensemble scheme not only performs as well as (and at times outperforms) existing approaches but is more adaptable and robust when put into real operation.

From an ensemble learning perspective, recent works by. Demonstrate the effectiveness of ensemble and stacking strategies for phishing detection. While these studies report state-of-the-art performance, such results are often achieved through intensive feature engineering and extensive hyperparameter tuning tailored to specific datasets. In contrast, the proposed framework applies feature engineering and hyperparameter tuning within a structured and reproducible pipeline, while promoting methodological diversity through the integration of multiple deep learning architectures in an ensemble-based strategy. This design reduces dataset-specific optimization and enhances robustness and generalization across heterogeneous datasets.

The study also emphasizes the potential of transformer-based models in extremely imbalanced settings. Although performing very well in cases of extreme data imbalance, their method needs large scale annotations and high-cost training. While the presented solution is not specifically tuned to extreme imbalance settings, it shows sustained and reliable performance over the balanced and realistic data distributions that are better suited for practical cybersecurity scenarios.

On the whole, the comparison indicates that our work, while not aiming at state-of-the-art performance on a single benchmark, is focused on promoting the development of a robust and generalizable phishing detection methodology. The proposed approach strikes a good balance between performance, robustness and adaptability by investigating several learning paradigms on the global and local data, thereby adding to current research and tackling critical issues relating to deploying a real-world phishing detection system

5.5 Unified Comparative and Local Replication Analysis

In this section, we provide a holistic comparison to position the proposed framework against existing solutions and investigate its behavior over global as well as local datasets. In Table 5.2, we do numerical comparison solely with studies on the same global benchmark, to make the comparison fair and avoid dataset bias. Experimental results show that the proposed framework achieves comparable and, in some cases, better performance when compared with traditional URL-based systems based on handcrafted features and classical ML models. Different from these approaches, our approach combines DL models and ensemble learning to exploit structural and sequential patterns in phishing URLs, which increases its representation ability beyond simple lexical features.

As can be seen in Table 5.2, our approach achieves an accuracy of 0.974 over the global benchmark, which slightly drops to 0.966 for the local Palestinian dataset. This decrease in performance can be explained by the greater noise and context dependency of real operational data. However, the structure remains intact and could be easily and practically employed on new data sources. Such observations are in line with the findings of previous research aimed at regional or domain-oriented phishing detection, where model performance decreases when moving to noisier and context-dependent data. A similar

tendency was found by*(Adrita et al., 2025)*, who noted that achieving comparable performance on more realistic or region-specific data was challenging, and reported an accuracy between 0.95 and 0.97 for global benchmark URLs, indicating the general performance degradation when models are applied to real-world applications is not only common but also expected.

Table 5.2: Unified Comparative and Replication Analysis

Case	Dataset Type	Method / Model	Learning Paradigm	Accuracy	Key Observation
(Adrita et al., 2025)	Global Benchmark (11,430 URLs)	RF, SVM, XGBoost	Handcrafted URL features + Classical ML	~0.95–0.97	Strong URL-only performance, limited representation learning
Our Study	Global Benchmark (11,430 URLs)	CNN, BiLSTM, DistilBERT, Ensemble	Deep Learning + Ensemble Learning	0.974	Stable and competitive performance using multiple learning paradigms
Our Study	Local Dataset (6,858 URLs)	Ensemble Model	Ensemble Learning	0.966	Slight drop due to local data specificity; confirms successful local replication using the same pipeline and consistent tuning procedure

After determining the optimal architecture and hyperparameters based on the global dataset, the same systematic design was adopted for the local dataset. The same model was retrained independently on the local data using the same architecture and the same range of parameters to ensure a fair comparison between the two experiments. However, polynomial interaction features were excluded in the local experiment due to the small size of the local dataset to minimize the risk of overfitting and maintain model stability. The same preprocessing steps were also applied within the pipeline, with the fit being performed only on the local training data and the transform being applied to the validation and test data. This prevented any data leakage and ensured a realistic assessment of the model's performance in the local context. In summary, the primary achievement of this work is a self-contained phishing detection framework built in an end-to-end and leakage-free training pipeline by integrating deep learning with ensemble learning. We further validate the framework on a global dataset which encompasses various learning paradigms, and subsequently replicated locally on real Palestinian data using the same pipeline structure, a consistent tuning procedure, by retraining the models with local data. This illustrates the robustness of the framework across datasets and feasibility.

5.6 Recommendations

The experimental results of our study indicates that deep learning based and ensemble models are more effective than traditional methods for phishing detection. Nevertheless, in order to turn that performance into practical cybersecurity solutions, we also need to handle some methodological and operational issues.

First, it is highly desirable to build bigger and more temporally variable, more representative datasets. Despite the balanced global and local dataset for our experiments, phishing attacks in real scenarios are dynamic, multilingual (generated automatically or adversarial). So, future collecting data should focus on regional and linguistically diverse sets of corpora such as the local phishing contents like Palestinian email phishing datasets. This diversity would improve the external validity of detection models, keeping them from overfitting when they confront new attack patterns.

Secondly, phishing detection systems should not be limited to static learning strategies. It is suggested to adopt adaptive learning methods, for instance: the online learning and reinforcement learning frameworks, because phishing tricks are always evolving. These techniques allow the models to change in response to new attack tactics in an incremental manner, while also being efficient computationally and responsive at runtime.

In addition, strong testing in an operational setting is critical before deployment. The assessment method applied in our work is a starting point to assess the model soundness and further validation on other organizational settings (e.g., email gateways, web browsers, financial services platforms) would be needed. This is key to determine aspects such as scalability, the latency bound above which a system cannot withstand adversarial manipulation, and its applicability for mission-critical cybersecurity applications.

Last but not least, an interpretable model should be considered as the primary need rather than a secondary goal. High detection accuracy is not enough for sensitive or regulated applications. On Transparency We recommend integrating transparency

methods, like SHAP and LIME, to increase transparency, aid analyst trust in Black Box systems, and comply with ethical regulations for an AI system.

In the future, based upon findings of the current study, a few potential future research avenues are suggested. One line of our approach is to construct a compact and effective multi-modality phishing detection framework that combines URL-based features with textual content information in the architecture so as to mimic real-world attack conditions. In this setting, sophisticated Natural Language Processing (NLP) can be used to extract linguistic, semantic, and stylistic properties from message content as well as structural and domain-level information about URLs *(Kmail et al., 2025)*.

The feature fusion approaches (feature-level and decision-level feature fusion) will be explored in our future work to investigate the effect of fusing different Independent features on model generalization and robustness. These techniques also enhance resilience to advanced phishing attacks that combine deceptive language with malicious link patterns.

Furthermore, possible future work also includes evaluating the proposed methodology across different communication channels (e.g., email, SMS, social media). This cross-platform evaluation would improve the transportability and deployment of phishing defense systems, as well as aid in unified defense against the persistently emerging digital threats.

5.7 Conclusion

Phishing is still one of the most intractable cyber threats, as it adapted itself over time and become more sophisticated. The empirical findings from the current article clearly show that learning-based methods pay attention to this issue across different data contexts.

The performance gap from simpler to more complex methods is evident in the comparative analysis. Unsupervised analysis was useful to understand the structure and distribution of the data; however can be considered as insufficient for obtaining a reliable detection accuracy, also showing that these methods are not appropriate as a standalone phishing detector. Deep learning models improved performance by orders of magnitude,

with accuracy >.080 in the 1st generation models increasing to <.090 when preserving sequential information well. These findings demonstrate a significant reduction in the misclassification errors and enhanced tradeoff between sensitivity and specificity.

Additional boost in performance was attained from models that could learn contextual representations, surpassing an accuracy of .095 with consistently high precision and recall values. It shows that the structural and semantic features of phishing patterns are crucial, especially in highly sophisticated and emerging threat scenarios.

The best performance was obtained by ensemble-based learning methods. The proposed ensemble reached an accuracy of about .097 on the global benchmark dataset and approximately 0.99 of AUC, misclassifying a smaller fraction of samples out of the entire test set. When tested on the local data, the ensemble proved to be persistent with its high performance in securing an accuracy of around .0966, a relatively low error rate.

In general, the reported results indicate that although single models of deep resource, arbitrary shape classifiers have strong detection power, ensembling is more robust to these types of attacks. Similar performance levels of global and local evaluation indicate that the transfer of sophisticated phishing defense solutions is applicable in regional context. This further validates that intelligent, data-driven detectors are capable of preventing phishing not only for the global population but also subpopulations and hence can be used in practice in security-sensitive applications.

References

Adrita, R. A. A., Islam, M. M., & Islam, M. S. (2025). Machine learning models for phishing website detection based on URL and web content features: A comparative analysis (EasyChair Preprint No. EIFPREPRINT15682). *EasyChair Preprints*. https://easychair.org/publications/preprint/GC3s

Ahmed, W., & Hammad, M. (2026). Hybrid machine learning approach for phishing email detection using NLP and ensemble models. In A. A. Abd El-Latif, M. A. ElAffendi, M. A. AlShara, & Y. Maleh (Eds.), *Cybersecurity, cybercrimes, and smart emerging technologies* (pp. 72–81). CRC Press. https://doi.org/10.1201/9781003614197-8

Al-Ahmadi, S., Alotaibi, A., & Alsaleh, O. (2022). PDGAN: Phishing detection with generative adversarial networks. *IEEE Access, 10*, 42459–42468. https://doi.org/10.1109/ACCESS.2022.3168235

Alamri, Z., Alhuzali, A., Alsulami, B., & Alghazzawi, D. (2025). Enhanced phishing website detection using optimized ensemble stacking models. *International Journal of Advanced Computer Science and Applications, 16*(8). https://doi.org/10.14569/IJACSA.2025.01608100

Alhogail, A., & Alsabih, A. (2021). Applying machine learning and natural language processing to detect phishing emails. *Computers & Security, 110*, 102414. https://doi.org/10.1016/j.cose.2021.102414

Almousa, M. M. (2022). *A comparative study of machine learning algorithms for the detection of social semantic cyberattacks* (Doctoral dissertation, North Carolina A&T State University). ProQuest Dissertations Publishing.

Al-Sabbagh, A., Hamze, K., Khan, S., & Elkhodr, M. (2024). An enhanced K-means clustering algorithm for phishing attack detection. *Electronics, 13*(18), 3677. https://doi.org/10.3390/electronics13183677

Althobaiti, K., Vaniea, K., Wolters, M. K., & Alsufyani, N. (2023). Using clustering algorithms to automatically identify phishing campaigns. *IEEE Access, 11*, 96502–96513. https://doi.org/10.1109/ACCESS.2023.3310810

Anti-Phishing Working Group. (2024). *Phishing Activity Trends Report, 1st Quarter 2024*. https://docs.apwg.org/reports/apwg_trends_report_q1_2024.pdf

Ariyadasa, S., Fernando, S., & Fernando, S. (2020). Detecting phishing attacks using a combined model of LSTM and CNN. *International Journal of Advanced and Applied Sciences, 7*(7), 56–67.

Asiri, S., Xiao, X., & Li, Y. (2024). PhishTransformer: A transformer-based phishing detection dataset with HTML and JavaScript features [Dataset]. Dataset linked with: PhishTransformer: A novel approach to detect phishing attacks using URL collection and transformer. *Electronics, 13*(1). https://doi.org/10.3390/electronics13010030

Bahnsen, A. C., Aouada, D., & Ottersten, B. (2015). Cost-sensitive decision trees for fraud detection. *Expert Systems with Applications, 42*(13), 5371–5379. https://doi.org/10.1016/j.eswa.2015.04.002

Brindha, R., Nandagopal, S., Azath, H., Sathana, V., Joshi, G. P., & Kim, S. W. (2023). Intelligent deep learning-based cybersecurity phishing email detection and classification. *Computers, Materials & Continua, 74*(3), 5901–5914. https://doi.org/10.32604/cmc.2023.030784

Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks, 106*, 249–259. https://doi.org/10.1016/j.neunet.2018.07.011

Catal, C., Giray, G., Tekinerdogan, B., Kumar, S., & Shukla, S. (2022). Applications of deep learning for phishing detection: A systematic literature review. *Knowledge and Information Systems, 64*(6), 1457–1500. https://doi.org/10.1007/s10115-022-01672-x

Chicco, D. (2017). Ten quick tips for machine learning in computational biology. *BioData Mining, 10*(1), 35. https://doi.org/10.1186/s13040-017-0155-3

CloudSEK. (2024). *Threat landscape report for the Middle East 2024*. https://www.cloudsek.com/whitepapers-reports/threat-landscape-report-for-middle-east-2024

Connolly, A., & Atlam, H. F. (2025). An effective ensemble learning phishing detection system using hybrid feature selection. *Journal of Network and Computer Applications, 242*, 104251. https://doi.org/10.1016/j.jnca.2025.104251

Dewis, M., & Viana, T. (2022). Phish responder: A hybrid machine learning solution for phishing and spam email detection. *Applied System Innovation, 5*(4), 73. https://doi.org/10.3390/asi5040073

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2019). Explaining explanations: An overview of interpretability of machine learning. *arXiv*. https://doi.org/10.48550/arXiv.1806.00069

Halgas, L., Agrafiotis, I., & Nurse, J. R. C. (2020). Catching the phish: Detecting phishing attacks using recurrent neural networks (RNNs). In *Lecture Notes in Computer Science* (Vol. 11897, pp. 219–233). https://doi.org/10.1007/978-3-030-39303-8_17

Hamid, I. R. A., Abawajy, J., & Kim, T. H. (2013). Using feature selection and a classification scheme for automating phishing email detection. *Studies in Informatics and Control, 22*(1), 61–70. https://sic.ici.ro/documents/549/SIC_2013-1-Art7.pdf

Hannousse, A., & Yahiouche, S. (2021). *Web page phishing detection* (Version 3) [Dataset]. Mendeley Data. https://doi.org/10.17632/c2gw7fy2j4.3

Jamal, S., Wimmer, H., & Sarker, I. H. (2024). A more efficient transformer model to detect phishing, spam, and ham emails: A large language model perspective. *Security and Privacy, 7*(5), e402. https://doi.org/10.1002/spy2.402

Kulkarni, A. D. (2023). Convolution neural networks for phishing detection. *International Journal of Advanced Computer Science and Applications, 14*(4). https://doi.org/10.14569/IJACSA.2023.0140403

Musa, H., Gital, A. Y., Ali, U., & Kwami, A. M. (2023). A comprehensive review of phishing website detection using machine learning and deep learning techniques. *International Journal of Scientific Development and Research, 8*(2).

Petukhova, A., Matos-Carvalho, J. P., & Fachada, N. (2025). Text clustering with large language model embeddings. *International Journal of Cognitive Computing in Engineering, 6*, 100–108. https://doi.org/10.1016/j.ijcce.2024.11.004

Positive Technologies. (2024). *Cybersecurity threatscape: Q1 2024*. https://global.ptsecurity.com/en/research/analytics/cybersecurity-threatscape-2024-q1/

Priya, S., Gutema, D., & Singh, S. (2024). *A comprehensive survey of recent phishing attack detection techniques*. In *2024 5th International Conference on Innovative Trends in Information Technology (ICITIIT 2024)* (pp. 1–6). IEEE. https://doi.org/10.1109/ICITIIT61487.2024.10580446

Rangapur, A., Kanakam, T., & P, D. (2022). Phish-defence: Phishing detection using deep recurrent neural networks (arXiv:2110.13424). arXiv. https://doi.org/10.48550/arXiv.2110.13424

Smadi, S., Aslam, N., & Zhang, Y. (2018). Dynamic evolving neural network for online phishing email detection based on reinforcement learning. *Expert Systems with Applications, 106*, 87–100. https://doi.org/10.1016/j.eswa.2018.03.016

Tang, L., & Mahmoud, Q. H. (2021). A survey of machine learning-based solutions for phishing website detection. *Machine Learning and Knowledge Extraction, 3*(3), 672–694. https://doi.org/10.3390/make3030034

Tiwari, S. (2020). Web page phishing detection dataset [Dataset]. Kaggle. https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset

Uddin, M. A., & Sarker, I. H. (2024). An explainable transformer-based model for phishing email detection: A large language model approach (arXiv:2402.13871). arXiv. https://doi.org/10.48550/arXiv.2402.13871

Yasin, A., & Abuhasan, A. (2016). An intelligent classification model for phishing email detection. arXiv. https://doi.org/10.48550/arXiv.1608.02196

Yerima, S. Y., & Alzaylaee, M. K. (2020). High-accuracy phishing detection based on convolutional neural networks. In *2020 International Conference on Computer Applications & Information Security (ICCAIS)* (pp. 1–6). IEEE. https://doi.org/10.1109/ICCAIS48893.2020.9096869

Zawoad, S., Dutta, A. K., Sprague, A., Hasan, R., Britt, J., & Warner, G. (2013). Phish-Net: Investigating phish clusters using drop email addresses. In *2013 eCrime Researchers Summit* (pp. 1–13). IEEE. https://doi.org/10.1109/eCrime.2013.6805805

Appendices

APPENDIX A

An example snapshot of the global phishing data is depicted in Figure 1. Given privacy and confidentiality constrains, the local dataset is not displayed.

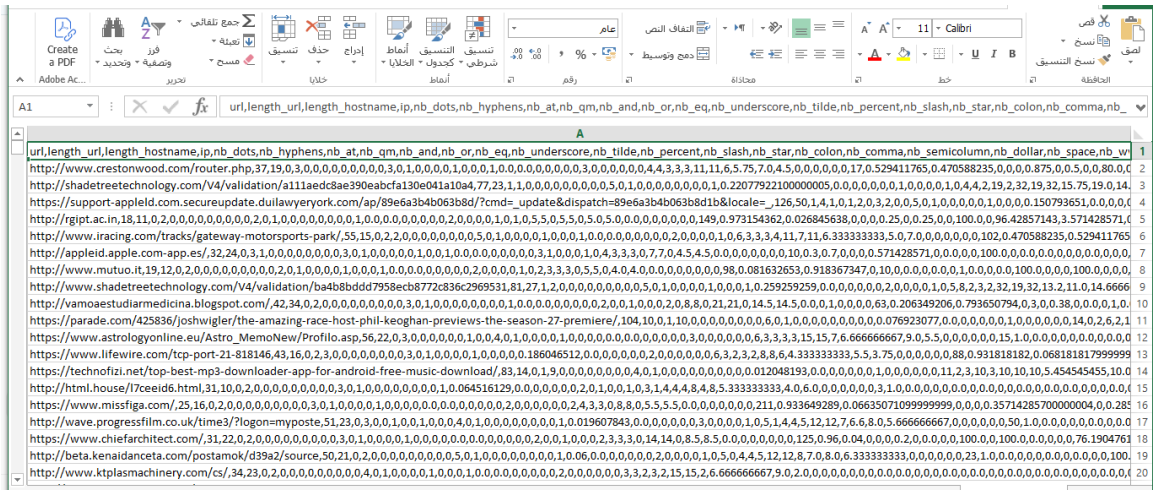


Figure A.1: Example Rows on the Structure of the Global Phishing Dataset.

APPENDIX B

Categorization of feature on the Global Phishing.

This appendix provides a brief classification of the characteristics embodied by the global phishing data set. According to their shape, the features can be classified into six types in terms of information content, resulting in a structured and minimal description of the dataset without involving specific features. This classification is proposed to aid in clear presentation and summary aspects of the bio-data employed for this investigation.

URL-Based Features

- These features characterize the lexical and structural attributes of the URL.
- URL length and depth
- Use of IP address
- Special characters, dots, and hyphens
- Subdomains and URL tokens

Domain-Based Features

- These are indications of domain registration and credibility.
- Domain age and registration length
- WHOIS and DNS records
- Domain registration country

Content-Based Features

These are elements of HTML layout or content.

- Forms and login fields
- JavaScript and Iframes
- Number of links and anchors

- Text-to-HTML and image-to-text ratios

Security Features

These characteristics stand for site safety and trust.

- SSL certificate validity
- Google Indexing and Browsing status
- Page rank and domain reputation

Behavioral Features

These characteristics model user interaction and director behavior.

- Redirection rate
- External links
- Pop-ups and user engagement indicators

Miscellaneous Features

These characteristics tend to capture other suspicious behaviors.

- Suspicious tokens and keywords
- Hidden fields
- Page size and hosting information
- Class Label

Each case is annotated for binary classification: **1: Phishing,0: Legitimate.**

APPENDIX C

Hyperparameter Tuning Strategy

This appendix presents the hyperparameter tuning approach implemented in this work for maximizing the potential performance of the machine learning and ensemble models proposed. It provides a tuning procedure, the leak-safe training workflow, cross-validation settings, and the search space for model optimization. The hyperparameter tuning was done on the training set only using Randomized Search CV with stratified cross validation for more stable model selection and to avoid data leakage.

A) Tuning Method

Method Name: Randomized Search CV with Stratified K-Fold Cross-Validation

Explanation:

We performed a hyperparameter tuning on the best model given by Randomized Search CV with Stratified Fold cross-validation (n = 5) to guarantee that we had an equal representation of phishing and legit URLs in each fold. All the preprocessing steps, including the tuning process was carried out on the training set to avoid data leakage and were implemented within the pipeline. Model choice relied on cross-validated accuracy, and the best model was automatically refitted for the entire training set with refit=True.

B) Leakage-Safe Training Pipeline

Pipeline Name: Train Pipeline (Fit On Training Data Only)

Explanation:

The configuration of the other methods had been well-tuned through cooperative pre-processing and one group classifier. All transformations and parameter fitting were

learned on training point only, and executed on validation and test. This had the effect of clearly separating the training and testing phases, with as little leakage as possible during the best hyperparameter search.

C) Cross-Validation Configuration

Configuration Name: Stratified Fold (5 Folds)

Explanation:

The tuning consisted of Stratified Fold with five-fold cross-validation, 'shuffle = True', and constant random seed (random-state = 42). Stratification preserved the original class distribution in those folds so that it is more consistent and less biased for estimating the model performance by tuning.

D) Search Control Parameters

Configuration Name: Randomized Search Settings

Explanation:

The random hyperparameter search is performed with the following parameters, specified directly in the code:

- scoring = "accuracy"
- n-jobs = -1 (parallel execution)
- random-state = 42 (reproducibility)
- verbose = 1 (progress monitoring)
- refit=True, (retrain the best model)
- Number of Search Iterations:
- Random Forest-Tuned: 20 iterations
- Gradient Boosting-Tuned: 20 iterations
- XGBoost-Tuned (if available): 25 iterations
- LightGBM-Tuned (if available): 25 iterations

Model tuning also used XGBoost and LightGBM when their implementations were present in the runtime environment.

E) Tuned Ensembles Model and Parameter Space

1) Random Forest-Tuned

Model Name: Random Forest Classifier (class-weight = "balanced")

Explanation:

Random Forest is applicable due to its sensitivity to tree depth, number of estimators, and feature sampling strategies, including purer algorithm can significantly affect generalization as well as overfitting behavior.

Tuned Hyperparameters:

- n-estimators: {300, 600, 900}
- max-depth: {None, 10, 15, 20}
- min-samples-split: {2, 5, 10}
- min-samples-leaf: {1, 2, 4}
- max-features: {"sqrt", "log2", None}
- bootstrap: {True, False}

2) Gradient Boosting-Tuned

Model Name: Gradient Boosting Classifier

Explanation:

For Gradient Boosting, we made an effort to trade model complexity off against learning stability, as performance is highly affected by learning rate, tree depth, and ensemble size.

Tuned Hyperparameters:

- n-estimators: {200, 400, 600}
- learning-rate: {0.03, 0.05, 0.08, 0.10}
- max-depth: {2, 3, 4}
- subsample: {0.7, 0.8, 0.9, 1.0}
- min-samples-split: {2, 5, 10}
- min-samples-leaf: {1, 2, 4}

3) (Optional – if library is available): XGBoost-Tuned

Model Name: XGB Classifier (eval-metric = "logloss")

Explanation:

We included XGBoost because of its well-acknowledged good performance on tabular data and high sensitivity to hyperparameter choice –especially with respect to tree depth, learning rate, and regularization terms.

Tuned Hyperparameters:

- n-estimators: {200, 300, 500, 800}
- max-depth: {3, 4, 5, 6, 8}
- learning-rate: {0.01, 0.03, 0.05, 0.08, 0.10}
- subsample: {0.7, 0.8, 0.9, 1.0}
- colsample-bytree: {0.7, 0.8, 0.9, 1.0}
- min-child-weight: {1, 3, 5, 7}
- reg-lambda: {0.5, 1.0, 2.0, 5.0}

4) LightGBM-Tuned (can be used if the library is present)

Model Name: LGBM Classifier (class-weight = "balanced")

Explanation:

We tuned the LightGBM models because of its leaf-wise growth approach, which has a strong dependence on controlling leaf count, depth, and sampling parameters.

Tuned Hyperparameters:

- n-estimators: {200, 300, 500, 800}
- max-depth: {−1, 4, 6, 8, 10}
- learning-rate = {0.01, 0.03, 0.05, 0.08, 0.10}
- num-leaves: {31, 63, 127, 255}
- subsample: {0.7, 0.8, 0.9, 1.0}
- colsample-bytree: {0.7, 0.8, 0.9, 1.0}
- min-child-samples: {10, 20, 30, 50}
- reg-lambda: {0.0, 0.5, 1.0, 2.0}

F) Tuning Output

Output Name: Best Estimator Selection

Explanation:

For each ensemble classifier, we again used Randomized Search CV to return the pipeline configuration with highest cross-validated training accuracy (best-estimator). Then these models were used for the validation-based selection and tested on the unseen test set.

APPENDIX D

Instructions for Data Collection in Local Dataset

This addendum contains the official facilitation material that provided guidance and legitimacy to local collection. The survey was made available to local institutions in order to provide an understanding of the research objectives, data needs, and obtain access to phishing-related local records. Including this document is a step toward full transparency in data collection, while honoring privacy and institutionally prohibited access.

Arab American University
Faculty of Graduate Studies

الجامعة العربية الأمريكية
كلية الدراسات العليا

2025/6/25

إلى من يهمه الأمر،

تسهيل مهمة بحثية

تحية طيبة وبعد،

تهديكم كلية الدراسات العليا في الجامعة العربية الأمريكية أطيب التحيات، وبالإشارة الى الموضوع أعلاه، تشهد كلية الدراسات العليا في الجامعة أن الطالبة **تهاني صادق محمود كميل** والتي تحمل الرقم الجامعي 202317103 هي طالبة ماجستير في برنامج امن المعلومات "الكتروني وتعمل على رسالة الماجستير الخاصة بها بعنوان:

"Building and Evaluating Phishing Detection Systems with Machine Learning and Deep Learning"

تحت اشراف الدكتور محمد حمارشة. نأمل من حضرتكم الإيعاز لمن يلزم لمساعدتها للحصول على المعلومات اللازمة للدراسة، علماً أن المعلومات ستستخدم لغاية البحث فقط وسيتم التعامل معها بغاية السرية، وقد أعطيت هذه الرسالة بناء على طلبها.

وتفضلوا بقبول فائق الاحترام

عميد كلية الدراسات العليا

د. نوار قطب

كلية الدراسات العليا
FACULTY OF GRADUATE STUDIES

Page 1 of 1

Jenin Tel: +970-4-2418888 Ext.:1471,1472 Fax: +970-4-2510810 P.O. Box:240
Ramallah Tel: +970-2-2941999 Fax: +970-2-2941979 Abu Qash - Near Alrehan
E-mail: FGS@aaup.edu ; PGS@aaup.edu Website: www.aaup.edu

Figure C.1: Instructions for Data Collection in Local Dataset.

بناء و تقييم أنظمة الكشف عن التصيد الاحتيالي باستخدام التعلم الالي و التعلم العميق.

تهاني صادق محمود كميل

أعضاء اللجنة :

د. محمد حمارشة

د. فادي دريدي

د. ماهر أبو فرحة

الملخص

لا تزال هجمات التصيد الاحتيالي تشكل خطرًا كبيرًا على الأمن السيبراني، إذ تُمكّن المهاجمين من اختراق حسابات المستخدمين عبر استغلال الهندسة الاجتماعية، بالإضافة إلى استخدام عناوين URLالخادعة وتقنيات إخفاء المحتوى التي تتجاوز العديد من أنظمة الحماية القائمة على القواعد. ونظرًا لأن أساليب التصيد الاحتيالي ليست ثابتة، وقد تظهر أنماط هجوم جديدة، فإننا بحاجة إلى الاستجابة باستراتيجيات كشف تكيفية لمواجهة الكم الهائل من التهديدات التي تعتمد على السياق. في هذه الورقة، نقوم بتقييم ومقارنة أداء منهجيات التعلم الآلي/التعلم العميق المختلفة فيما يتعلق بالكشف عن التصيد الاحتيالي، مع الأخذ في الاعتبار قدرتها على التعميم على مجموعات بيانات مرجعية عالمية، بالإضافة إلى مجموعة بيانات محلية مصممة خصيصًا تعكس الخصائص المجتمعية الإقليمية.تم تنفيذ التجربة في فلسطين باستخدام بيانات تصفح الإنترنت الحقيقية لمواقع المؤسسات المحلية، ومجموعات بيانات التصيد الاحتيالي العالمية المنشورة، وذلك خلال مرحلتي جمع البيانات والتقييم التجريبي. في هذا السياق، تم اتباع منهج تجريبي كمي من خلال تطبيق وتقييم عدة طرق للكشف، مثل طرق التجميع، والشبكات العصبية الالتفافية(CNN) ، وشبكات الذاكرة طويلة المدى

ثنائية الاتجاه(BiLSTM) ، والطرق القائمة على المحولات مثلDistilBERT ، أو المصنفات التجميعية المتقدمة.تتكون مجموعة البيانات من عناوين URL للتصيد الاحتيالي وعناوين URL شرعية، بحجم عينة مناسب للأبحاث يبلغ 11430 عينة عالمية وأكثر من 6000 عينة تم جمعها محليًا. لجمع الميزات، استخدمنا ميزات قائمة على عناوين URL وميزات قائمة علىHTML ، بالإضافة إلى ميزات قائمة على المجال، مع بعض عمليات المعالجة المسبقة وهندسة الميزات لتحسين جودة البيانات. ولإجراء مقارنة عادلة، تم أيضًا تقييم النموذج باستخدام مقاييس التصنيف الشائعة.من الواضح أن أداء الكشف يتحسن باستمرار. كانت درجة التمييز لأساليب التجميع منخفضة، بينما شكلت نماذج الشبكات العصبية التلافيفية (CNN) أساسًا جيدًا، واستطاعت نماذج BiLSTM و DistilBERTتحسين النتائج من خلال نمذجة الأنماط التسلسلية والسياقية. حققت النماذج المجمعة أفضل استقرار وانتظام، حيث بلغت دقة 0.966 على مجموعة البيانات الفلسطينية المحلية.من أهم إسهامات هذا العمل مجموعة بيانات جديدة ومُقَيَّمة للتصيد الاحتيالي، يمكن استخدامها لإجراء تقييمات واقعية في سياق إقليمي محدد. حقق مسار الكشف المقترح أداءً مستقرًا على مجموعات البيانات المحلية والعالمية، مما يثبت قدرته على التعميم. لذا، تقترح الدراسة استخدام آليات الكشف القائمة على النماذج المجمعة، مع مراعاة مجموعات البيانات المحلية أثناء التدريب، والتركيز على قابلية النماذج للتكيف والتفسير لتحسين أنظمة الكشف عن التصيد الاحتيالي في بيئات واقعية.

الكلمات المفتاحية: الكشف عن التصيد الاحتيالي، التعلم الآلي، التعلم العميق، النماذج المجمعة، مجموعات البيانات.