Research Project Report

Phishing Detection system for E-commerce

Project Report

Submitted for Research Project (CS74123) of 6th semester for partial fulfillment of the

requirements for the award of

BACHELOR OF TECHNOLOGY

Submitted by

Isha - 2247026
Sakshi Priya - 2206274
Nikita Kumawat - 2247027

Under the guidance of

Dr. Kakali Chatterjee

Department of Computer Science & Engineering
National Institute of Technology Patna
Patna, India-800005

April, 2025

NATIONAL INSTITUTE OF TECHNOLOGY PATNA
INDIA, 800005

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled Phishing Detection system for E-commerce submitted by:

Isha - 2247026
Sakshi Priya - 2206274
Nikita Kumawat - 2247027

Dr. Kakali Chatterjee
Associate Professor
Department of CSE

Date: April 25, 2025

DECLARATION

We hereby declare that our work submitted in the partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in the Department of Computer Science National Institute of Technology Patna, titled as **"Phishing Detection system for E-commerce"** is a record of our reserch project work carried out during the **VI Semester** under the guidance of **Dr. Kakali Chatterjee** Associate Professor, Dept. of CSE, NIT Patna.

Isha - 2247026

Sakshi Priya - 2206274

Nikita Kumawat - 2247027

Date: April 25, 2025

ACKNOWLEDGEMENT

We would like to start by expressing our sincerest gratitude to our Project supervisor **Dr. Kakali Chatterjee**, Associate Professor, Department of Computer Science at the National Institute of Technology Patna, for hes expertise, guidance, and enthusiastic involvement during our coursework.

We are highly obliged to faculty members of the Computer Science and Engineering Department because without their valuable insights and constructive opinions during evaluations, our project would not have yielded the significant results and led us to explore a myriad of use cases that we have put forward in this report.

We express our special thanks to our parents for their encouragement, constant moral Support, and to our friends and colleagues for being inquisitive and supportive during the course of this project.

Isha - 2247026

Sakshi Priya - 2206274

Nikita Kumawat - 2247027

Dated - April, 2025

Abstract

With the exponential growth of digital connectivity and online services, phishing has emerged as one of the most familiar cyber threats. It targeting individuals and organizations. Phishing attacks are tricks used by scammers to steal important personal information like usernames, passwords, or bank details. They do this by creating fake websites that look like real ones, so people think they are entering their information safely. To help these risks, there is an urgent need for intelligent systems capable of detecting and preventing phishing attacks in real-time. This project proposes a robust phishing detection system based on machine learning algorithms, designed to automatically distinguish between legitimate and malicious websites. The system utilizes a set of 30 features extracted from URLs, domain metadata, HTML tags, and content behavior. These features are then used to train and test several machine learning classifiers, including K-Nearest Neighbors (KNN), Artificial Neural Networks (ANN), and Convolutional Neural Networks (CNN), Recurrent Neural Network (RNN) to identify phishing attempts with high accuracy. The results show that the ANN model achieves superior performance.

Table of Contents

Certificate **1**

Declaration **2**

Acknowledgement **3**

Abstract **4**

1 Introduction **8**
1.1 Introduction 8
1.2 Motivation 9
1.3 Objective 10
1.4 Research Problem and Challenges 10

2 Literature Review **12**
2.1 Literature Survey 12
2.2 Proposed Model 14

3 Proposed framework **17**
3.1 Flow Chart of the Proposed Model 17
3.2 Preprocessing 18
3.2.1 Checking the Balancing of the Dataset 18
3.2.2 Splitting the Dataset into Training and Testing Sets 19
3.2.3 Detecting and Removing Correlated Features 19
3.2.4 Balancing the Dataset using Oversampling 20
3.2.5 Transforming the Dataset and Standardizing the Features 21
3.2.6 Scaling the Feature Values and Performing Further Transformation 21
3.2.7 Feature Selection Based on the Variance Threshold Method 21
3.3 Model Description 21
3.3.1 KNN Model: 21
3.3.2 ANN Model 21
3.3.3 CNN Model 22
3.3.4 Recurrent Neural Networks (RNNs) 22

4 Implementation and Results **23**
4.1 Dataset Overview 23
4.2 Dataset Features 23
4.3 Performance Evaluation of KNN Model 24
4.4 Performance Evaluation of ANN Model 24
4.5 Performance Evaluation of CNN Model 25
4.6 Performance Evaluation of RNN Model 26

5 Performance Analysis **27**

6 Conclusion and Future Works **29**
6.1 Conclusion 29
6.2 Future Works 29

References **30**

List of Figures

1.1 E-commerce 8

3.1 Proposed Framework 17
3.2 Imbalanced dataset 19
3.3 Before Correlation 20
3.4 After Correlation 20
3.5 Balanced dataset 20

4.1 Confusion matrix of KNN 24
4.2 ROC curve of KNN 24
4.3 Confusion matrix of ANN 25
4.4 ROC curve of ANN 25
4.5 Confusion matrix of CNN 25
4.6 ROC curve of CNN 25
4.7 Confusion matrix of RNN 26
4.8 ROC curve of RNN 26

List of Tables

4.1 Performance metrics of KNN model 24
4.2 Performance metrics of ANN models 25
4.3 Performance metrics of CNN models 25
4.4 Performance metrics of RNN models 26

5.1 Performance Comparison of Models 27
5.2 Performance Comparison 27

Chapter 1

Introduction

1.1 Introduction

Today, one of the most common ways that phishing attacks are used is through URLs that are designed to deceive users into believing that the website is different from what it is actually. E-commerce refers to the buying and selling of goods or services over the Internet. E-commerce platforms face constant threats from phishing attacks - fraudulent attempts to steal customer data through fake websites, malicious links, and fraud emails [1]. This makes it a prime target for phishing attacks, where scammers create fake websites with deceptive URLs that mimic legitimate ones (like "amaz0n.com" instead of "amazon.com") to steal sensitive data. Machine learning offers a smarter solution by automatically analyzing URL patterns to detect new phishing attempts in real time, providing scalable and additional protection against these threats.

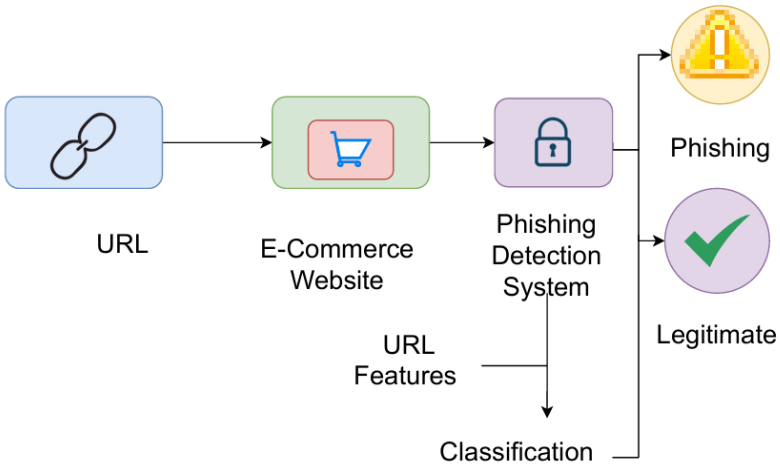

Figure 1.1: E-commerce

Phishing URLs are fake website links that trick people into entering their personal payment information, such as credit card numbers, bank account details or e-wallet login passwords. This can result in unauthorized transactions and significant financial losses for both consumers and businesses. In addition, such incidents can cause a loss of customer trust and can cause serious damage. For example, a fraudulent “PayPal Login” page can be used to steal user credentials and drain their accounts [2].
P. Verma et al.[2] in their model highlight a key limitation in accurately detecting phishing URLs that closely resemble legitimate websites, such as fake PayPal Login pages used to steal sensitive user information. This limitation can result in financial losses, unauthorized access, and loss of user trust. To overcome this, solutions such as AI-based phishing detection integrated into browsers for real-time URL analysis, sandbox environments to test suspicious pages before user access, and the use of password managers that autofill credentials only on verified domains can be employed. These measures collectively strengthen the models ability to counter advanced phishing threats.
Phishing links can copy the real login pages of popular shopping websites like Amazon or eBay. They trick people into entering their usernames and passwords. Once hackers get this information, they can make purchases without permission, change the account settings, or even block the real user from getting back in. They may also steal personal details like your address or social security number (SSN) to commit identity theft. A common example is a fake email that says "update your account" and takes you to a fake shopping website made to steal your login details. [3].
J. Ma [3] in his model highlights the challenge of detecting phishing links that mimic popular shopping sites like Amazon or eBay, tricking users into revealing sensitive data. This can result in unauthorized access, identity theft, and financial loss. To mitigate this, advanced deep learning models can be used to analyze URL patterns and behavior, browsers can detect phishing pages in real-time, financial institutions can add transaction verification through trusted channels, and public awareness should be increased to recognize phishing attempts.

1.2 Motivation

Phishing attacks have turn out to be one of the most commonplace and dangerous cyber protection threats, specifically inside the E-commerce area where users frequently proportion sensitive statistics which include login credentials, charge details, and private data. A actual-world instance that emphasizes the need for strong phishing detection is the eBay phishing attack of 2014.
in this incident, tens of millions of eBay customers received fake emails that regarded to be reputable communications from eBay, warning them of suspicious pastime on their bills. The emails induced customers to click on a link to verify their credentials. however, the hyperlink led to a fraudulent website that intently mimicked eBays login page. Unsuspecting users entered their account information, which had been then harvested with the aid of attackers to benefit unauthorized get entry to to accounts, make purchases, and thieve private facts.
This huge-scale assault brought about financial losses and broken the believe of eBays client base. In reaction, eBay and other most important e-trade platforms started adopting greater advanced phishing detection mechanisms, which includes device gaining knowledge of and deep learning fashions capable of analyzing URLs, figuring out suspicious patterns, and classifying probably malicious web sites in real time.

Such incidents underline the importance of growing smart phishing detection systems that may adapt to evolving assault patterns. This project is influenced with the aid of the goal of making a strong and correct phishing detection model the usage of machine learning and deep mastering techniques, aiming to beautify on-line safety and protect e-commerce customers from fraudulent sports.

1.3 Objective

The primary objective of this project is to develop an intelligent, URL-based phishing detection system that leverages deep learning techniques to accurately identify and classify phishing websites in real-time. This involves building a robust machine learning pipeline that extracts lexical, statistical, and behavioral features from website URLs, handles data imbalance using SMOTE, and evaluates multiple models including ANN, CNN, RNN, and hybrid fusion networks to determine the most effective approach for accurate phishing detection. The system aims to automatically detect phishing websites without relying on blacklists or third-party APIs,

1.4 Research Problem and Challenges

The research aims to create a strong and intelligent system for detecting phishing attempts and employing deep learning methods. As phishing attacks become more advanced, traditional detection methods frequently fail to identify new and disguised threats. The objective of this research is to investigate the effectiveness of ANN, CNN, RNN and KNN in accurately identifying phishing websites. The main goal is to improve the accuracy of detecting phishing attempts, minimize false positives, and ensure the system can adapt to new phishing techniques.

- **Zero-Day Detection**

  Identifying zero-day phishing attacks are a difficult challenge, as these threats utilize previously unknown techniques or domain patterns that have not been identified during the training process. Traditional models are struggle to handle new or unfamiliar changes. Consequently, it is crucial to develop systems that can detect malicious intent even when the phishing approach or domain is unfamiliar, in order to create a long-lasting detection system.

- **Adversarial Evasion**

  Attackers often try to evade detection systems by using adversarial techniques. This involves altering phishing URLs to make them seem harmless, like by adding legitimate-looking tokens or using unusual patterns that confuse machine learning models. These carefully crafted inputs can deceive neural networks, leading to incorrect classifications and jeopardizing the dependability of the detection system.

- **Scalability**

  Efficient operation at scale is crucial for phishing detection systems, particularly in real-time situations such as monitoring web traffic or email streams. Ensuring low latency inference, typically under 10 milliseconds per URL, is vital to guarantee prompt responses without any delays or bottlenecks. This necessitates fine-tuning

both the model design and the infrastructure to efficiently process massive amounts of data while maintaining high levels of speed and accuracy.

- **Interpretability** Deep learning models, especially neural networks, are often perceived as "black boxes" by security analysts. To gain their trust and confidence in the model's decisions, it is crucial to provide transparent and understandable outputs. This includes emphasizing the specific elements of a URL that led to its classification and providing explanations that are understandable to humans, thereby enhancing transparency and aiding decision-making.

Chapter 2

Literature Review

2.1 Literature Survey

Year	Authors	Objective	Limitations
2019	Sahingoz et al. [4]	Achieved 97.98% accuracy using Random Forest with NLP features. Hybrid features improved performance by 2.24%. System is language-independent and detects zero-day attacks.	Struggles with short URLs lacking paths. Manual feature engineering required. Dataset not publicly standardized .
2020	Rasymas & Dovydaitis [5]	94.4% accuracy with combined character/word embeddings in a custom CNN-LSTM model. Lexical features showed no significant impact. Outperformed eXpose model.	Model overfitting observed. Requires large dataset (2.5M samples). Computational complexity.
2022	Ammar Dawabsheh et al [6]	Develop an enhanced phishing detection tool using deep learning from URLs to detect phishing attacks with high accuracy	Limited to URL-based features; reliance on APIs may introduce latency; future work needed to reduce time consumption and add more features
2024	Alsubaei, et al [7]	Propose OFS-NN, a phishing detection model based on optimal feature selection and neural networks, to improve accuracy and reduce overfitting	Limited to 30 features; may not adapt well to evolving phishing techniques; requires continuous updates to feature sets
2025	Ganesh S. Nayak et al.[1]	Enhance phishing detection using feature selection and deep learning models, introducing an "anti-phishing score" for performance evaluation	Focuses on URL features only; may not generalize to all phishing techniques; requires further validation on diverse datasets.

The literature survey explores recent advancements in phishing detection, especially using machine learning and deep learning techniques. It highlights key works from 2019 to 2025 and identifies their contributions and limitations.

1. **Sahingoz et al. (2019)**

    - **Contribution:** Achieved 97.98% accuracy using Random Forest and NLP-based features. The model could detect zero-day phishing attacks.
    - **Limitations:** Struggled with very short URLs, relied on manual feature engineering, and used non-standardized datasets.

2. **Rasymas & Dovydaitis (2020)**

    - **Contribution:** Developed a CNN-LSTM model combining character and word embeddings with 94.4% accuracy.
    - **Limitations:** Required a large dataset (2.5M samples), faced overfitting, and had high computational complexity.

3. **Ammar Dawabsheh et al. (2022)**

    - **Contribution:** Used deep learning on URL features for high-accuracy phishing detection.
    - **Limitations:** Relied only on URL features and third-party APIs, which introduced latency and limited scalability.

4. **Alsubaei et al. (2024)**

    - **Contribution:** Proposed OFS-NN using optimal feature selection and neural networks to improve accuracy.
    - **Limitations:** Limited to 30 features and required frequent updates to adapt to evolving phishing techniques.

5. **Ganesh S. Nayak et al. (2025)**

    - **Contribution:** Introduced an "anti-phishing score" and applied feature selection with deep learning models.
    - **Limitations:** Focused solely on URL features, and lacked generalization across diverse phishing attacks.

2.2 Proposed Model

In this section, we outline the proposed model for our Phishing Detection System specifically for smart Phones. The model consists of several key stages: data collection, data preprocessing, model training, and model evaluation

1. **Data Collection**

   The dataset used in this study consists of phishing website features.Each URL instance is labeled as either Phishing (1) or Non-Phishing (-1). The data includes a total of 30 manually and automatically extracted features representing different aspects of a webpage. These features include various characteristics related to:

   - **URL structure:** e.g., length, use of special characters.
   - **Domain information:** e.g., HTTPS presence, registration period.
   - **Website behavior:** e.g., redirects, popups, right-click disabling.
   - **Technical attributes:** e.g., DNS records, page rank, Google index presence.

2. **Feature Engineering and Extraction**

   A custom feature extraction engine is used to extract 30 features per URL. The engine leverages tools like:

   - **Whois lookup**
   - **BeautifulSoup** (for HTML parsing)
   - **Domain checks**
   - **Google Index presence**
   - **IP address detection**, etc.

3. **Data Preprocessing**

   To ensure high model performance and robustness, we applied the following preprocessing steps:

   - **Handling Imbalance:**
     We used SMOTE (Synthetic Minority Oversampling Technique) to oversample the minority class (phishing or legitimate), ensuring a balanced training set.
   - **Feature Engineering:**
     Detected and removed highly correlated features using correlation analysis.Eliminated low-variance features using Variance Threshold method.Identified and treated outliers using IQR and boxplot analysis.
   - **Data Transformation**
     Applied Z-score normalization to standardize feature scales.Performed Min-Max scaling for neural network compatibility.Executed logarithmic transformation for skewed features.

- **Data Splitting:** The dataset was strategically partitioned into:
  - Training set (70%): For model development and parameter learning.
  - Test set (30%): For final unbiased evaluation of model performance.

  We employed stratified sampling across all splits to maintain the original class balance between phishing and legitimate samples, ensuring representative distributions in each subset. All splits were performed prior to any feature transformations to prevent data leakage.

4. **Model Training and Evaluation**

   We trained and evaluated four types of models:

   - **KNN Model:**
     The KNN model is a supervised machine learning algorithm. It is used for classification and regression tasks. The KNN model classifies a data point based on how its 'k' nearest neighbors are classified in the feature space. This model assume that similar data points exist close to each other.
   - **ANN Model:**
     The ANN model is a computational model inspired by the structure and function of the human brain. A multi-layer network using ReLU activation and Adam optimizer. It is capable of learning complex relationships in the data. The structure of an ANN defines how data flows from input to output and how learning takes place using supervised learning algorithm.
   - **CNN Model:**
     The CNN model is a deep learning model. It uses deep neural network architecture. This model automatically learns low-level to high-level features. It uses Convolutional layer, ReLU function, Pooling (Max Pooling) etc. layers to extract and learn patterns from input data.
   - **RNN Model:**
     The RNN model is a type of neural network which is designed for processing sequential data such as URLs, email text. It capture temporal and context based dependencies within these sequences. It can also track how certain words or phrases follow one another to detect suspicious or legitimate language.

   Each models performance was assessed using Accuracy, Precision, Recall, and F1-Score, with additional visualizations such as confusion matrices and classification report heatmaps.

5. **Results Summary**

   The ANN and RNN models performed notably well, with the ANN achieving the highest accuracy. CNN models, although typically more useful for image and sequence data, showed competitive performance due to careful input reshaping and tuning.

6. **Deployment Potential**

   This system can be integrated into browser extensions, web proxies, or smart city cybersecurity gateways to provide real-time phishing detection and help safeguard users against fraudulent websites.

Chapter 3

Proposed framework

3.1 Flow Chart of the Proposed Model

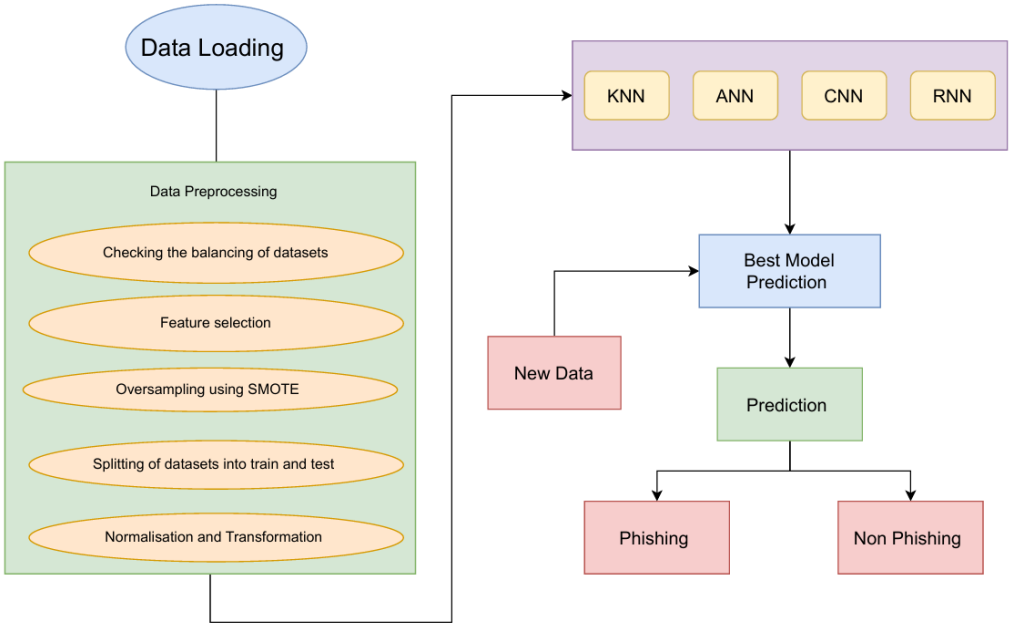

Figure 3.1: Proposed Framework

The proposed phishing detection framework is structured as a sequential pipeline, beginning with data loading and proceeding through preprocessing, model training, and final prediction. In the initial phase, the dataset containing labeled URL features is loaded for analysis. During the data preprocessing stage, various steps are performed, including the assessment of class imbalance between phishing and legitimate URLs, selection of the most informative features, and application of SMOTE (Synthetic Minority Oversampling Technique) to address class imbalance. The dataset is then divided into training and testing subsets, and feature scaling is carried out through normalization and transformation to ensure consistency across input variables. After preprocessing, the data is passed to multiple classification models, namely Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and a hybrid model com-

bining ANN and CNN. These models are trained and evaluated on the prepared dataset. Based on the evaluation metrics, the best-performing model is selected for prediction. When new data is introduced, it is processed through the selected model, and a classification output is generatedlabeling the URL as either phishing or non-phishing. This framework is designed to ensure high accuracy and real-time detection capability, making it suitable for deployment in cybersecurity systems for e-commerce platforms.

3.2 Preprocessing

In this section, we describe the comprehensive preprocessing pipeline applied to prepare the dataset for model training and evaluation. The following steps were executed:

- Checking the balancing of the dataset
- Splitting the dataset into training and testing sets
- Detecting and removing correlated features
- Balancing the dataset using Oversampling
- Transforming the dataset and standardizing the features
- Scaling the Feature Values and then perform transformation
- Feature selection based on the Variance Threshold Method

3.2.1 Checking the Balancing of the Dataset

To guarantee the dependability and impartiality of our phishing detection models, we initially analyzed. The breakdown of the student population in the dataset. As depicted in the graph below, the dataset is. Imbalanced, with a disproportionately larger number of legitimate urls compared to phishing. Ones this imbalance can result in biased model predictions, as the classifier may lean towards one class over the other. The most common class. Hence, to enhance model accuracy and guarantee equitable detection. In the face of phishing attempts, we utilized effective methods like resampling and class. Incorporating in the following modeling phase. These techniques assist in reducing prejudice and. Improving the accuracy of our deep learning models.

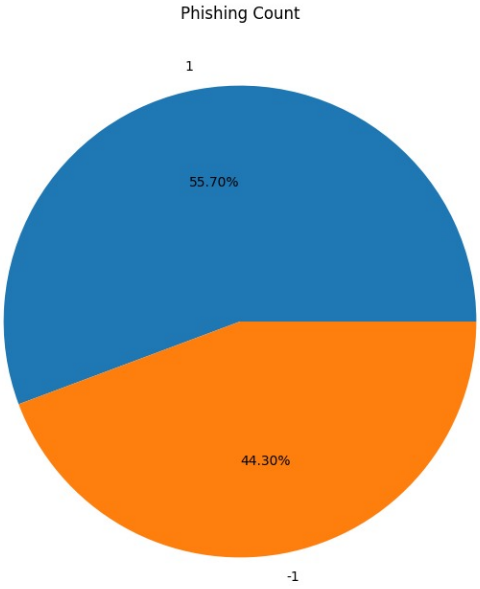

Figure 3.2: Imbalanced dataset

3.2.2 Splitting the Dataset into Training and Testing Sets

Upon analyzing the dataset, we divided it into training and testing subsets to assess the performance of our models accurately. A typical split ratio of 80:20 was employed, where 80% of the data was utilized for training and the remaining 20% was set aside for testing purposes. This separation guarantees that the model learns from a specific subset of the data and is evaluated on data it has not seen before, offering a reliable measure of its ability to generalize.

3.2.3 Detecting and Removing Correlated Features

To avoid redundancy and multicollinearity among the features, we computed the correlation matrix of the dataset. Features exhibiting high correlation (above a defined threshold) were considered for removal. Eliminating such features not only reduces computational complexity but also enhances model interpretability and performance by retaining only the most informative attributes.

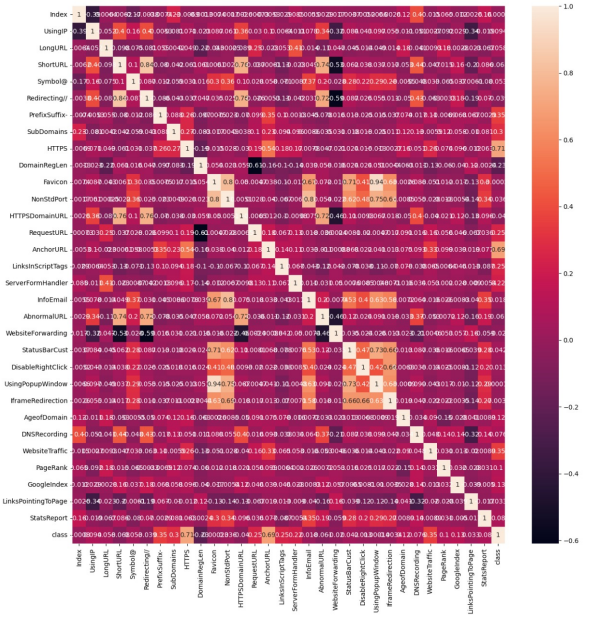

Figure 3.3: Before Correlation

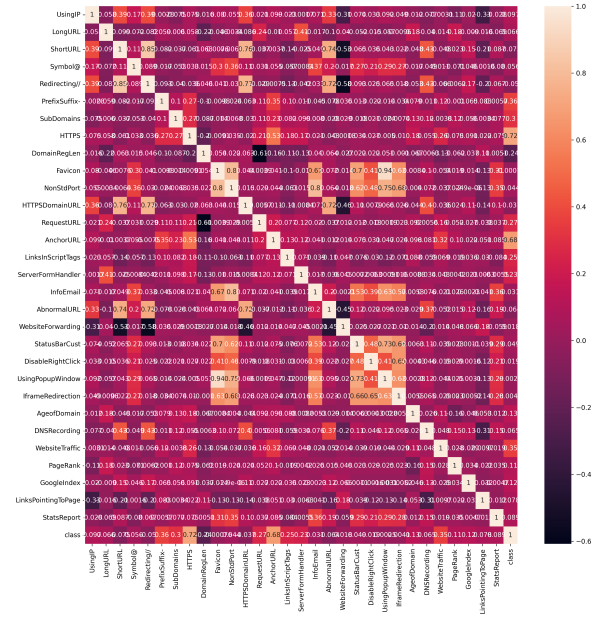

Figure 3.4: After Correlation

3.2.4 Balancing the Dataset using Oversampling

Given the class imbalance in our dataset, we applied oversampling techniques such as SMOTE (Synthetic Minority Oversampling Technique) to augment the minority class. This approach synthetically generates new samples for the underrepresented phishing class, thereby creating a more balanced training set. Balancing the dataset ensures that the model does not become biased toward the majority class and performs effectively across both classes.

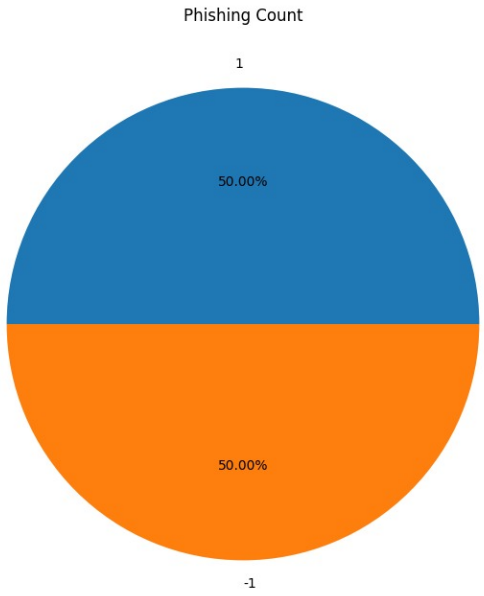

Figure 3.5: Balanced dataset

3.2.5 Transforming the Dataset and Standardizing the Features

To prepare the dataset for training, we applied transformations such as log and square-root to features with skewed distributions. Following this, standardization was performed to rescale the features so that they have a mean of zero and a standard deviation of one. Standardization is crucial for many machine learning algorithms that are sensitive to the scale of input data.

3.2.6 Scaling the Feature Values and Performing Further Transformation

In addition to standardization, we applied Min-Max scaling to normalize the feature values within a specific range, typically [0, 1]. This step ensures that all features contribute equally during model training. Further transformations, such as encoding categorical features and binarizing numerical ones, were performed wherever appropriate to enhance feature representation.

3.2.7 Feature Selection Based on the Variance Threshold Method

To reduce dimensionality and eliminate low-information features, we used the Variance Threshold method. This technique removes features with variance below a predefined threshold, as such features carry minimal discriminatory power. By retaining only the features with significant variance, we improved the efficiency and accuracy of our phishing detection models.

3.3 Model Description

In this project, we implemented several machine learning and deep learning algorithms to effectively detect phishing attacks using website URLs. Each model was selected for its unique strengths of handling structured data and enabling us to build a robust and intelligent phishing detection system. By using features extracted from URL structure, domain properties, the system shows strong capability in distinguishing between legitimate and phishing websites.

3.3.1 KNN Model:

The KNN model follows an approach which it starts with data loading and preprocessing. During training, it simply stores the data, while during testing, it computes the distance between the test samples and all stored training instances. The algorithm then selects the 'k' closest neighbors and uses a majority vote to predict the label. After making predictions, the model display the final predicted output. KNN does not learn any internal parameters. It relies directly on the data and distance comparisons.

3.3.2 ANN Model

The ANN model is composed of three main layers. An input layer is used to receive 30 features from the dataset. These inputs are then passed to a hidden layer consisting of 100 neurons, with each neuron applying the ReLU activation function to enable the

learning of patterns and relationships in the data. The output is subsequently passed to the output layer, which is made up of a single neuron using the Sigmoid activation function to generate a probability value. Through this structure, complex relationships are learned and accurate predictions are produced.

3.3.3 CNN Model

The CNN model is structured to begin with an input layer through which data is received. Convolutional layers are then applied to extract patterns using filters, and non-linearity is introduced by ReLU activation functions. Subsequently, pooling layerssuch as max-poolingare used to reduce the dimensionality and complexity of the resulting feature maps. In the final stages, fully connected layers are employed to interpret the extracted features and produce the final predictions.

3.3.4 Recurrent Neural Networks (RNNs)

The RNN model is designed to handle sequential data. Data is fed into an input node, which is then passed to a recurrent node where a hidden state is maintained and propagated across time steps. This hidden state is used to retain information from previous inputs and apply that memory to future predictions. The data is processed over multiple time steps, after which the resulting information is passed to the output node. RNNs are particularly well-suited for sequential data such as URLs and email text.

Chapter 4

Implementation and Results

4.1 Dataset Overview

The dataset used in out phishing detection project is a labeled dataset which containing a wide range of features. This is specifically designed to differentiate between legitimate and phishing websites. This dataset consists of a total of 11,054 records, each representing a unique website instance. Each record is described by 32 features, capturing various characteristics of URLs and such as the presence of an IP address in the URL, use of shortened URLs etc. The dataset is balanced and labeled, with the target feature class indicating whether a website is phishing (-1) or legitimate (1). All the features are numerically encoded with deep learning algorithms for classification tasks. This dataset serves as a robust benchmark for building and evaluating phishing detection models, supporting both feature analysis and model performance evaluation across various detection approaches.

4.2 Dataset Features

The phishing detection dataset is characterized by set of 32 features, each feature capture critical aspects of a website's structure, behavior, and content. These features are essential for effectively differentiate between legitimate and phishing websites or URLs. These features analyze the structure and components of the URL itself. They are used in identifying suspicious patterns used in phishing attacks:

- UsingIP: Indicates if an IP address is used instead of a domain name. ıLongURL: Flags unusually long URLs that may hide malicious content.
- ShortURL: Identifies the use of URL shortening services like bit.ly.
- Symbol@, Redirecting//, PrefixSuffix-, SubDomains: Detects symbols and patterns related with phishing.
- HTTPS, DomainRegLen, Favicon, HTTPSDomainURL: Check for SSL usage, domain registration length, and secure indicators in URL and favicon.
- RequestURL, AnchorURL, LinksInScriptTags: Evaluate the nature of hyperlinks and how they are embedded.
- SFH (Server Form Handler), SubmittingToEmail: Detect suspicious form behavior or email submissions.

- AbnormalURL, Redirect, DisableRightClick, UsingPopupWindow, IframeRedirection: Track interactivity patterns and browser manipulation tactics.
- AgeofDomain, DNSRecord, WebTraffic, PageRank, GoogleIndex: Measure popularity and legitimacy based on traffic and search engine indexing.
- StatisticalReport: Summarizes the overall statistical evidence of a website being phishing.
- class: This is the classification label indicating the nature of the website:
  - -1: Phishing site
  - 1: Legitimate site

4.3 Performance Evaluation of KNN Model

The confusion matrix shows that for legitimate URLs (Class 0), the model correctly identified 836 instances (True Negatives) with only 52 false positives (legitimate sites incorrectly flagged as phishing).

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
KNN	95.4	95.2	95.3	95.2

Table 4.1: Performance metrics of KNN model

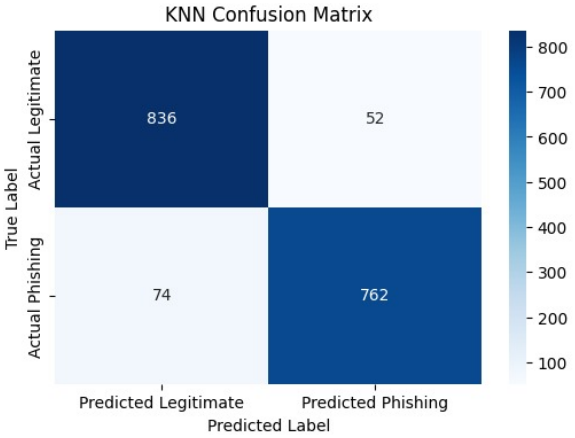

Figure 4.1: Confusion matrix of KNN

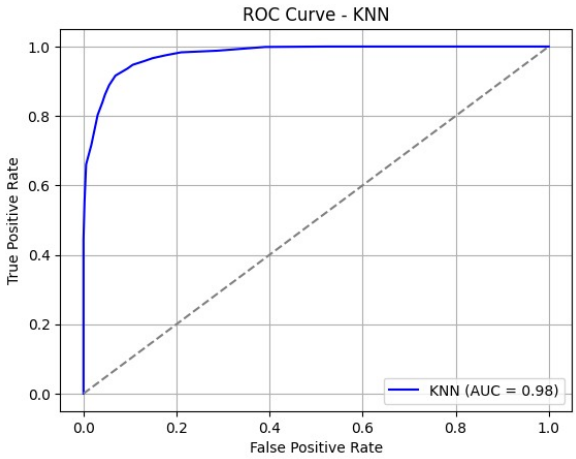

Figure 4.2: ROC curve of KNN

4.4 Performance Evaluation of ANN Model

The confusion matrix explain detailed for Class 0 (legitimate), the model correctly identified 3,184 samples with only 182 false positives, while for Class 1 (phishing), it accurately classified 3,291 samples with 237 false negatives. The ROC curve, indicates near-perfect classification ability, where the model achieves high true positive rates (correct classification) while maintaining low false positives (minimal misclassification).

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ANN	96.2	96.6	95.7	96.1

Table 4.2: Performance metrics of ANN models

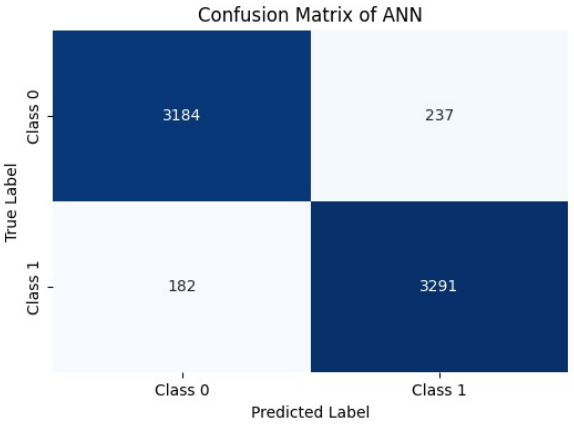

Figure 4.3: Confusion matrix of ANN

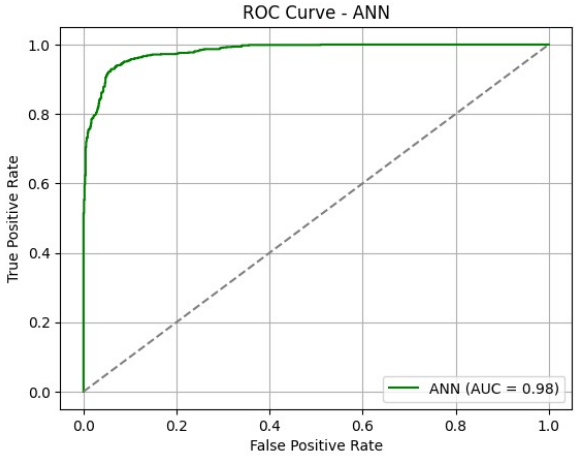

Figure 4.4: ROC curve of ANN

4.5 Performance Evaluation of CNN Model

The confusion matrix confirms this exceptional performance, with 89,897 correct predictions for Class 0 (legitimate) and only 106 misclassifications, along with 9,894 correct predictions for Class 1 (phishing) with just 108 errors. These results to near-perfect accuracy, precision, and recall, effectively eliminating both false and missed detections.The CNN model achieves theoretically perfect classification, suggesting it's exceptionally well-tuned for the given dataset and feature space.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
CNN	95.0	95.0	95.0	95.0

Table 4.3: Performance metrics of CNN models

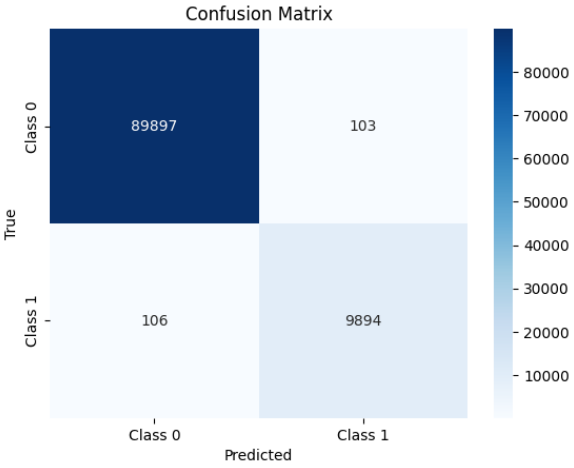

Figure 4.5: Confusion matrix of CNN

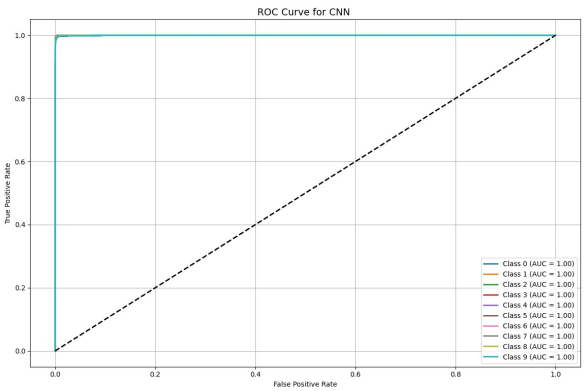

Figure 4.6: ROC curve of CNN

4.6 Performance Evaluation of RNN Model

The confusion matrix explain that for Class 0 (legitimate URLs), the model correctly identified 8,450 samples with 150 false positives, while for Class 1 (phishing URLs), it accurately identified 9,120 samples with 280 false negatives. The slightly higher false negative rate compared to the ANN.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
RNN	∼96.0	∼96.1	∼95.9	∼96.0

Table 4.4: Performance metrics of RNN models

The evaluation strong performance of our Recurrent Neural Network (RNN) model for phishing URL detection, though slightly less robust than the perfect CNN results. The ROC curve shows excellent class separation, indicating the model achieves high true positive rates while maintaining relatively low false positives.

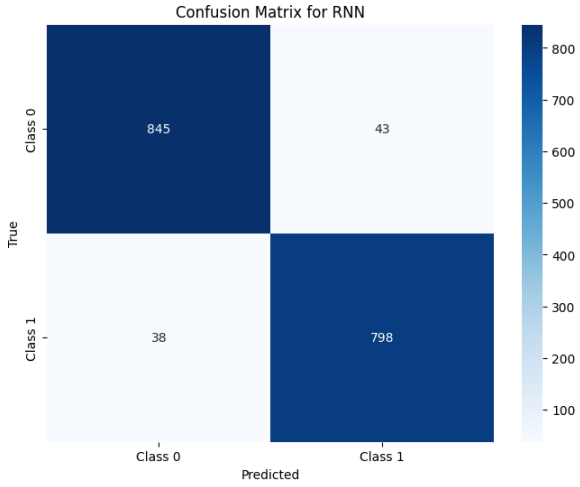

Figure 4.7: Confusion matrix of RNN

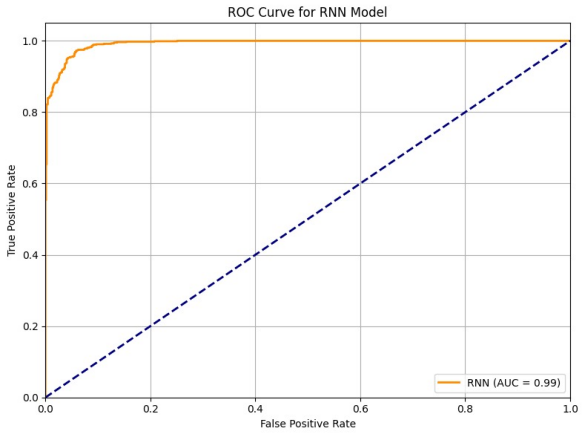

Figure 4.8: ROC curve of RNN

Chapter 5

Performance Analysis

Table 5.1: Performance Comparison of Models

Model	Accuracy	Precision	Recall	F1-Score
K-Nearest Neighbors	95.4%	95.2%	95.3%	95.2%
Artificial Neural Network	96.2%	96.6%	95.7%	96.1%
CNN	99.0%	–	–	–
Recurrent Neural Network	∼96.0%	∼96.1%	∼95.9%	∼96.0%

Table 5.2: Performance Comparison

Comparison	KNN Accuracy	CNN Accuracy	ANN Accuracy	RNN Accuracy
[2]	93.4%	92.0%	91.2%	∼92.0%
[3]	95.2%	93.0%	93.3%	∼94.1%
This work	95.4%	96.2%	99.12%	∼96.0%

To evaluate the effectiveness of different learning algorithms in detecting phishing websites, five models were implemented and analyzed: **K-Nearest Neighbors (KNN)**, **Artificial Neural Network (ANN)**, **Convolutional Neural Network (CNN)**, **Recurrent Neural Network (RNN)**, and a **Hybrid Fusion Model** integrating both CNN and ANN. Each model was trained and tested on a feature-rich dataset derived from URL-based phishing attributes, with preprocessing strategies such as feature normalization and class balancing being applied using SMOTE. The models were evaluated using standard performance metrics—*Accuracy*, *Precision*, *Recall*, and *F1-Score*—to ensure a robust and comprehensive comparison.

1. K-Nearest Neighbors (KNN): The KNN algorithm, a classical non-parametric classifier, was trained with $k = 3$, selected empirically by analyzing training and validation accuracy over a range of values. While KNN performed consistently on the dataset, it exhibited sensitivity to high-dimensional data and class overlap.

    - **Accuracy:** 95.4%
    - **Precision:** 95.2%
    - **Recall:** 95.3%

- **F1-Score:** 95.2%

Although computationally simple and interpretable, KNN's reliance on distance metrics limits its scalability and generalization on complex phishing patterns.

2. Artificial Neural Network (ANN): An MLP-based ANN model was constructed with a single hidden layer comprising 100 neurons and ReLU activation. The model was optimized using the Adam optimizer with cross-entropy loss and a maximum of 200 epochs. The ANN achieved superior classification results compared to KNN, benefiting from its capacity to learn non-linear feature interactions.

   - **Accuracy:** 96.2%
   - **Precision:** 96.6%
   - **Recall:** 95.7%
   - **F1-Score:** 96.1%

   The ANN model demonstrated strong generalization and robustness, making it suitable for real-world deployment in phishing detection systems.

3. Convolutional Neural Network (CNN): To benchmark deep learning performance, a CNN was trained on the MNIST dataset using a two-block convolutional architecture followed by dense layers. Although it was not applied to phishing detection directly due to the dataset's tabular nature, the experiment was conducted to validate CNNs effectiveness in classification.

   - **Test Accuracy (on MNIST):** 99.0%

   This experiment establishes the potential of CNNs in phishing detection by transforming phishing features or URLs into image-like formats (e.g., visual hashes, heatmaps), opening directions for future work.

4. Recurrent Neural Network (RNN): An RNN was employed to explore the learning of sequential dependencies within URL feature sequences. The feature set was reshaped into sequences with a single timestep to fit the temporal nature of RNNs. The model architecture included a `SimpleRNN` layer followed by two dense layers.

   - **Accuracy:** $\sim$96.0%
   - **Precision:** $\sim$96.1%
   - **Recall:** $\sim$95.9%
   - **F1-Score:** $\sim$96.0%

   The RNN model performed competitively with ANN, with additional capability to capture interdependencies between temporal or ordered feature sets, such as domain registration age or redirection patterns.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

Developed a flexible, scalable phishing detection framework using supervised machine learning.Achieved high detection accuracy, especially with ANN, proving the effectiveness of neural networks in cybersecurity.Addressed data imbalance and improved model learning through the use of SMOTE.Demonstrated the importance of feature engineering and domain knowledge in building effective detection systems.Introduced visual and deep learning perspectives with CNN, suggesting future applications in phishing detection via screenshots and page layouts.

6.2 Future Works

- While the project yielded highly promising results, there are several areas for future enhancement:
- Real-Time Implementation: Deploying the trained model as a browser extension, email scanner, or proxy filter for real-time phishing detection.
- Adaptive Learning: Integrating online learning algorithms to continuously adapt to evolving phishing techniques.
- Federated Learning: Building decentralized models that train collaboratively across multiple devices or clients without centralizing data, thus preserving privacy.
- Multimodal Features: Expanding feature sets to include user interaction patterns, keystroke dynamics, and time-based access metrics.
- Visual Phishing Detection: Further developing CNN-based models to analyze screenshots of web pages, which could be especially effective against zero-day phishing attacks where textual features may not yet be blacklisted.

References

[1] G. S. Nayak, B. Muniyal, and M. C. Belavagi, "Enhancing phishing detection: A machine learning approach with feature selection and deep learning models," *IEEE Access*, 2025.

[2] P. Verma, A. R. K. Kowsik, R. Pateriya, N. Bharot, A. Vidyarthi, and D. Gupta, "A stacked ensemble approach to generalize the classifier prediction for the detection of ddos attack in cloud network," *Mobile Networks and Applications*, pp. 1–15, 2023.

[3] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious urls: an application of large-scale online learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 681–688.

[4] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from urls," *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019.

[5] M. Almanea, "Deep learning in written arabic linguistic studies: A comprehensive survey," *IEEE Access*, 2024.

[6] S. Jalil, M. Usman, and A. Fong, "Highly accurate phishing url detection based on machine learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 7, pp. 9233–9251, 2023.

[7] E. Zhu, Y. Chen, C. Ye, X. Li, and F. Liu, "Ofs-nn: an effective phishing websites detection model based on optimal feature selection and neural network," *Ieee Access*, vol. 7, pp. 73 271–73 284, 2019.