TABLE OF CONTENTS
(INDEX IN TABLE FORMAT)

ABSTRACT

Phishing attacks have become one of the most widespread and dangerous threats in the realm of cybersecurity. These attacks leverage deceptive techniques to trick users into revealing sensitive information, often resulting in financial loss, identity theft, and data breaches. Traditional security measures such as firewalls and antivirus software are often insufficient to detect these attacks effectively, as phishing websites are designed to closely mimic legitimate sites.This project, **"Phishing Website Detection Using Machine Learning Techniques,"** presents a comprehensive approach for automatically identifying phishing websites by leveraging feature extraction and machine learning algorithms. The system analyzes multiple aspects of a website, including URL characteristics, domain registration details, website content, and behavioral patterns, to extract distinguishing features. These features are then used to train supervised machine learning models, enabling accurate classification of websites as either legitimate or malicious.The study demonstrates that machine learning-based phishing detection can achieve high accuracy, low false positive rates, and robust generalization across diverse types of phishing attacks. By automating the detection process, this approach reduces reliance on manual inspection and enhances overall internet security. The project highlights the effectiveness of combining feature engineering with machine learning techniques to address modern cybersecurity challenges and provides a foundation for further research in intelligent phishing prevention systems.

CHAPTER-1

INTRODUCTION

The widespread adoption of the internet has revolutionized communication, commerce, and information sharing. However, it has also led to the emergence of various cybersecurity threats. Among these threats, **phishing attacks** remain one of the most prevalent and damaging. Phishing involves the creation of fraudulent websites or emails that impersonate legitimate entities to deceive users into revealing sensitive information, such as login credentials, credit card numbers, and personal data. According to cybersecurity reports, phishing attacks account for a significant proportion of data breaches and cybercrime incidents worldwide.

Phishing attacks are particularly dangerous because they exploit human psychology rather than relying solely on software vulnerabilities. Attackers often use tactics like urgency, fear, or incentives to trick users into taking unsafe actions. This makes detection challenging, as traditional security mechanisms such as firewalls and antivirus software are often unable to differentiate between legitimate and malicious websites based solely on superficial characteristics.

To address these challenges, **machine learning (ML) techniques** have emerged as effective tools for detecting phishing websites. Machine learning algorithms can automatically learn patterns from data, enabling systems to classify websites as legitimate or malicious with high accuracy. By analyzing various features of a website—such as the URL structure, domain registration information, presence of special characters, redirection behavior, and website content—ML models can capture subtle differences between phishing and legitimate sites.

This project, **"Phishing Website Detection Using Machine Learning Techniques,"** aims to develop an intelligent system capable of automatically identifying phishing websites. The system extracts a comprehensive set of features from multiple sources, including:

CHAPTER-2

Method

The methodology comprises a systematic pipeline designed to accurately identify phishing websites by leveraging URL-based feature extraction and machine learning classification models. Each phase is critical for achieving high detection accuracy and robustness against evolving phishing tactics.

2.1 Overview of the Methodology:

1.Data Acquisition:

A comprehensive dataset is fundamental to training a reliable phishing detection model. For this project, datasets from publicly available repositories such as Kaggle and UCI Machine Learning Repository were used. These datasets include thousands of URLs labeled as phishing or legitimate, gathered over recent years to reflect current phishing strategies. Multiple datasets were combined and cleaned to build a balanced corpus. Data augmentation techniques like adding recent phishing URLs were employed to enhance the dataset's diversity and relevance.

2. Feature Extraction

The backbone of URL-based phishing detection lies in extracting meaningful features that differentiate malicious URLs from benign ones. Several lexical and semantic features were extracted from each URL, including:

- **URL Length:** Phishing URLs often use longer strings to mimic legitimate ones or obscure true destinations.
- **HTTPS Usage:** Legitimate websites tend to use HTTPS to secure communication; absence may indicate phishing.
- **Count of Dots and Hyphens:** Multiple subdomains or hyphenated domains can signal suspicious activity.

- **Presence of '@' Symbol:** Attackers use '@' to mislead browsers by placing genuine URLs after it.
- **IP Address in URL:** Use of IP addresses instead of domain names is common in phishing.
- **Suspicious Keywords:** Tokens like "login", "secure", "verify", "update" are often exploited by attackers.
- **Domain Age & Expiry:** Recent domain registration or near expiry suggests phishing.
- **URL Redirection:** Use of multiple redirection mechanisms to hide phishing sites.
- **Use of Non-standard Ports or Encodings:** Techniques to evade detection by standard filters.

Advanced preprocessing techniques such as tokenization, pattern recognition, and regular expressions were used to extract these features reliably.

3. Feature Extraction:

Before feeding data into machine learning models, preprocessing was essential to ensure quality and consistency:

- Null or missing values in feature fields were handled by imputation or removal.
- Features were standardized or normalized as required by certain classifiers.
- Categorical features, if any, were transformed using encoding techniques.
- The dataset was split into training and testing subsets to enable unbiased performance evaluation.
- Feature selection or dimensionality reduction (e.g., using correlation analysis or PCA) was optionally applied to remove redundant or non-informative features.

4. Model Training

Multiple machine learning algorithms were explored to find the best classifier for phishing URL detection:Gradient Boosting Classifier: An ensemble technique that sequentially builds weak learners to optimize overall prediction accuracy. Demonstrated best results (accuracy ~97%).

- Random Forest: Uses multiple decision trees to reduce overfitting and improve robustness.
- Support Vector Machine (SVM): Effective in high-dimensional feature space, useful for smaller datasets.
- Multi-layer Perceptron (Neural Networks): Captures complex nonlinear patterns in features.
- Logistic Regression: Baseline linear model for classification.
- K-Nearest Neighbors (KNN): Simple distance-based classification approach.

Hyperparameters for these models were tuned via grid search and cross-validation to achieve optimal detection performance.

5. Prediction and Verification

The trained models were used to predict the class labels of URLs in the test set. Predictions were cross-validated against known labels to measure:

precision and recall to minimize risks of missed detections or false alarms.

6. Performance Evaluation

Robust evaluation metrics were computed and analyzed to judge the effectiveness:

- Accuracy: Overall correctness of classification.
- Precision: Proportion of predicted phishing sites that were actually phishing.
- Recall (Sensitivity): Proportion of actual phishing sites correctly identified.
- F1-Score: Harmonic mean of precision and recall to balance both.
- Confusion Matrix: Detailed error analysis of predictions.

CHAPTER-3

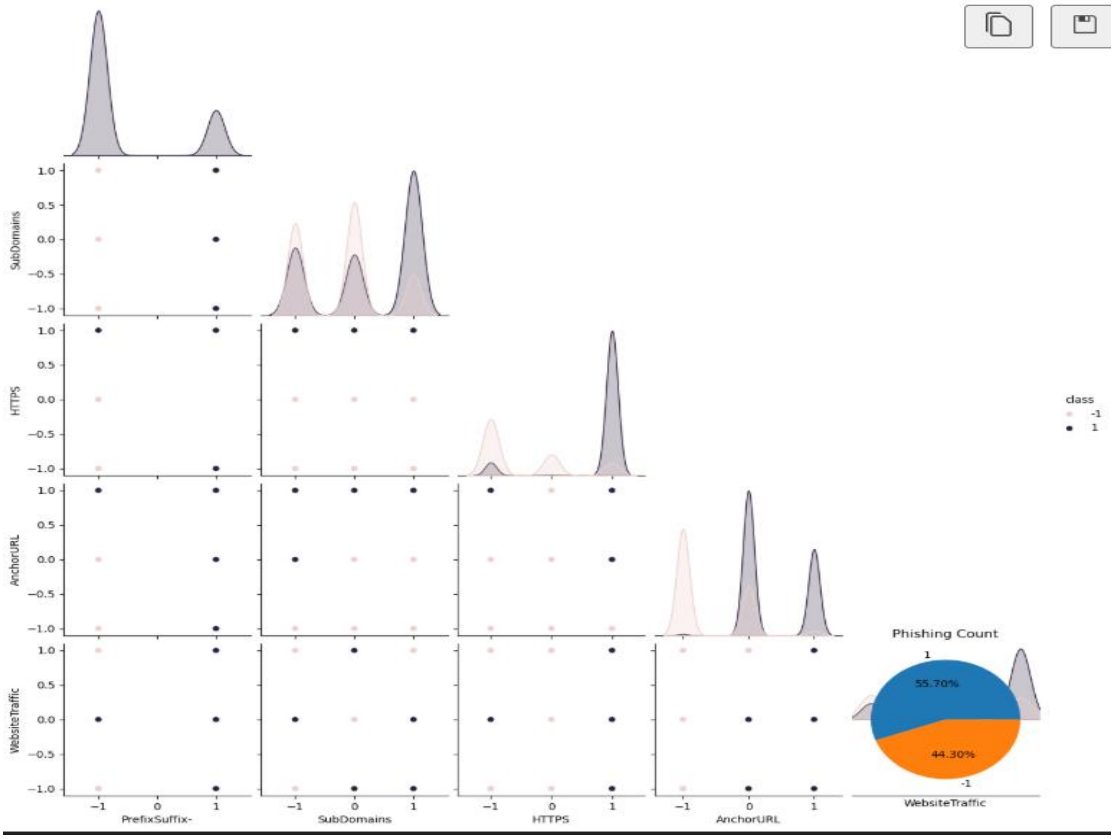
TEST CASES/ OUTPUT :

PROGRAM:

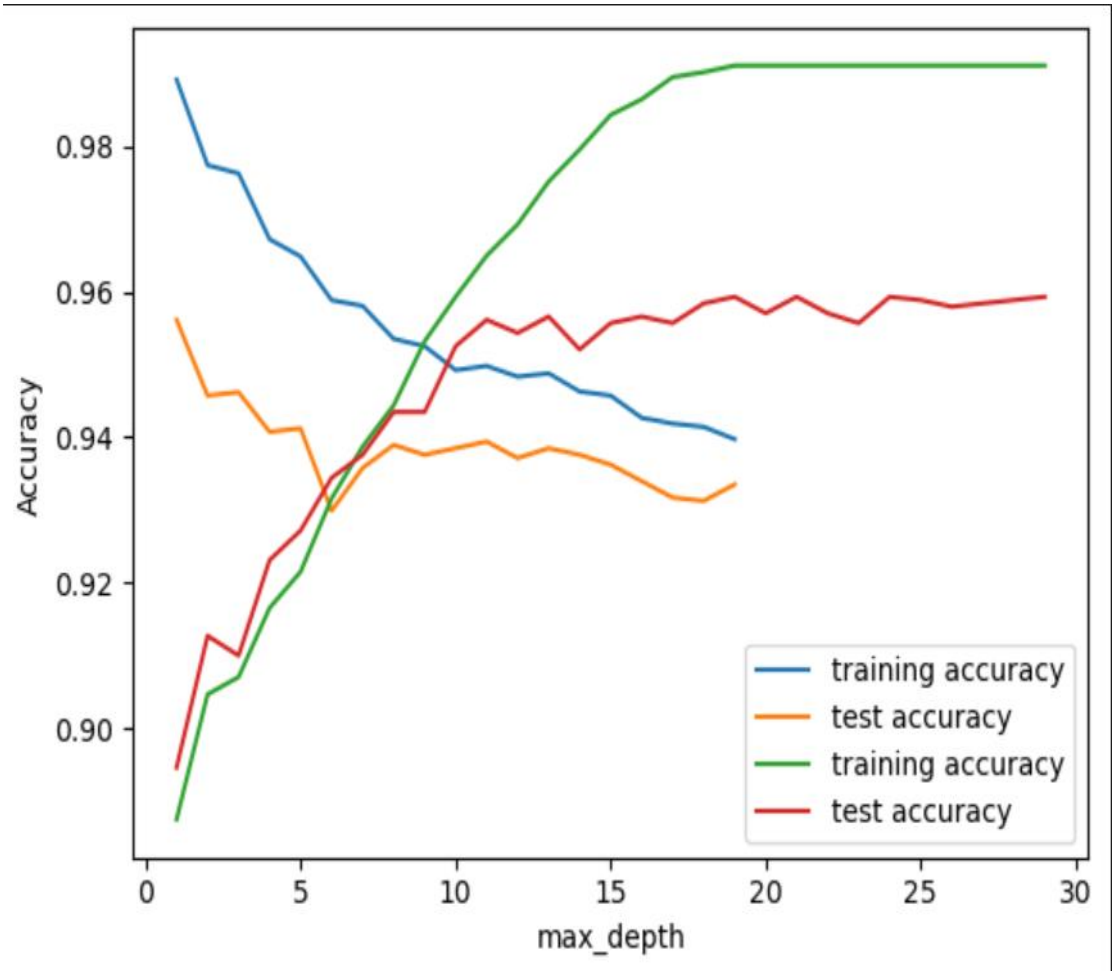
```
#importing required libraries
from flask import Flask, request, render_template
import numpy as np
import pandas as pd
from sklearn import metrics
import warnings
import pickle
warnings.filterwarnings('ignore')
from feature import FeatureExtraction
file = open("pickle/model.pkl","rb")
gbc = pickle.load(file)
file.close()
app = Flask(__name__)
@app.route("/", methods=["GET", "POST"])
def index():
    if request.method == "POST":
        url = request.form["url"]
        obj = FeatureExtraction(url)
        x = np.array(obj.getFeaturesList()).reshape(1,30)
        y_pred =gbc.predict(x)[0]
```

```
        #1 is safe
        #-1 is unsafe
        y_pro_phishing = gbc.predict_proba(x)[0,0]
        y_pro_non_phishing = gbc.predict_proba(x)[0,1]
        # if(y_pred ==1 ):
        pred = "It is {0:.2f} % safe to go ".format(y_pro_phishing*100)
        return render_template('index.html',xx =round(y_pro_non_phishing,2),url=url )
    return render_template("index.html", xx =-1)
if __name__ == "__main__":
```

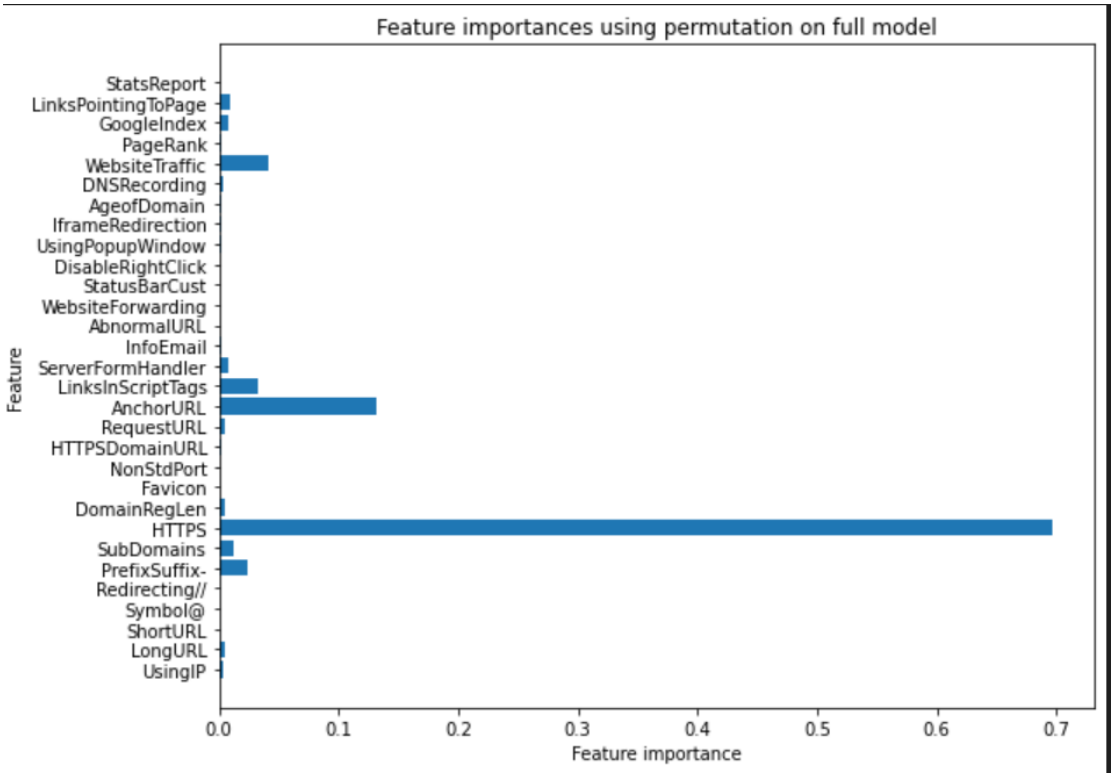
app.run(debug=True)



	ML Model	Accuracy	f1_score	Recall	Precision
0	Gradient Boosting Classifier	0.974	0.977	0.994	0.986
1	CatBoost Classifier	0.972	0.975	0.994	0.989
2	Multi-layer Perceptron	0.971	0.974	0.992	0.985
3	XGBoost Classifier	0.969	0.973	0.993	0.984
4	Random Forest	0.967	0.970	0.992	0.991
5	Support Vector Machine	0.964	0.968	0.980	0.965
6	Decision Tree	0.961	0.965	0.991	0.993
7	K-Nearest Neighbors	0.956	0.961	0.991	0.989
8	Logistic Regression	0.934	0.941	0.943	0.927
9	Naive Bayes Classifier	0.605	0.454	0.292	0.997




PHISHING WEB SITES DETECTION
The Internet has become an indispensable part of our life, However, It also has provided opportunities to anonymously perform malicious activities like Phishing. Phishers try to deceive their victims by social engineering or creating mockup websites to steal information such as account ID, username, password from individuals and organizations. Although many methods have been proposed to detect phishing websites, Phishers have evolved their methods to escape from these detection methods. One of the most successful methods for detecting these malicious activities is Machine Learning. This is because most Phishing attacks have some common characteristics which can be identified by machine learning methods.
PHISHING WEB SITES DETECTION FROM URL
Enter URL
Check here
Website is 79% safe to use...
Continue


Feature importances using permutation on full model
StatsReport
LinksPointingToPage
GoogleIndex
PageRank
WebsiteTraffic
DNSRecording
AgeofDomain
IframeRedirection
UsingPopupWindow
DisableRightClick
StatusBarCust
WebsiteForwarding
AbnormalURL
InfoEmail
ServerFormHandler
LinksInScriptTags
AnchorURL
RequestURL
HTTPSDomainURL
NonStdPort
Favicon
DomainRegLen
HTTPS
SubDomains
PrefixSuffix-
Redirecting//
Symbol@
ShortURL
LongURL
UsingIP
Feature
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Feature importance

CHAPTER-4

RESULTS

Exploration of Machine Learning Models:

The project involved evaluating various machine learning algorithms, such as Random Forest, Gradient Boosting, and Decision Trees. Through systematic testing, the strengths and weaknesses of each model in detecting phishing websites were observed, helping in selecting the most effective classifier

Feature Analysis and Importance:

By performing Exploratory Data Analysis (EDA) on the phishing dataset, the contribution of individual features to the classification task was examined. Features such as **HTTPS usage**, **Anchor URL patterns**, and **Website Traffic** were found to be highly influential in distinguishing phishing URLs from legitimate ones. This feature analysis enhanced the understanding of critical factors contributing to phishing detection.

Model Tuning and Performance:

Creating this project enabled hands-on learning about model tuning and the impact of hyperparameters on performance. Fine-tuning models improved classification accuracy and reduced false positives, demonstrating the importance of feature selection and model optimization.

Gradient Boosting Classifier Performance:

The **Gradient Boosting Classifier** emerged as the most effective model, correctly classifying URLs with an overall accuracy of **97.4%**. This high accuracy indicates the model's robustness in identifying phishing URLs and minimizing the risk of malicious attachments or compromised websites.

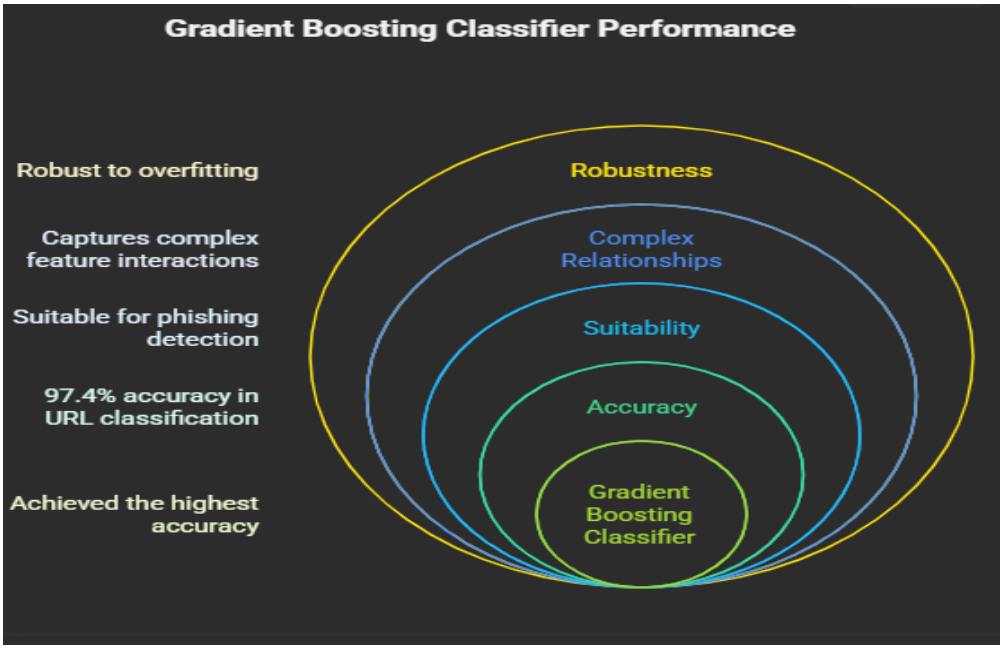
Practical Implications:

The results confirm that machine learning models, when trained with relevant and well-engineered features, can provide reliable and automated phishing detection. Features such as URL structure, redirection patterns, domain age, and traffic metrics are particularly useful in identifying suspicious websites.The project successfully demonstrated that machine learning-based phishing detection can achieve high accuracy and efficiency. Gradient Boosting Classifier, combined with feature importance analysis, proves effective for real-world applications, contributing to enhanced online security and reduced exposure to cyber threats.

The proposed system extracts **30 distinct features** from a given URL to evaluate its legitimacy. When a user inputs a URL, the model performs comprehensive analysis based on content, structure, domain information, and web traffic patterns.

Each feature is assigned a value:

- **1 → Legitimate**
- **0 → Suspicious**
- **-1 → Phishing or Malicious**



After feature extraction, these numerical results can be passed into a machine learning model (e.g., Random Forest, SVM, or Logistic Regression) for classification.

Example Result Interpretation

- A legitimate website typically has most features valued at **1**.
- A phishing website often shows multiple **-1** feature values, such as:
  - Use of IP addresses instead of domain names
  - Shortened URLs (bit.ly, tinyurl, etc.)
  - Low domain age and registration length
  - Missing SSL certificate (HTTPS)

CHAPTER 5

Summary, Conclusion, Recommendation

5.1 Summary

This project focused on implementing a phishing website detection system using machine learning techniques to distinguish between legitimate and malicious web pages. The main objective was to identify the behavioral and structural characteristics of phishing websites and use them as predictive features for automated classification.

Throughout the development process, several machine learning algorithms such as **Decision Tree**, **Random Forest**, **Support Vector Machine (SVM)**, and **Logistic Regression** were trained and evaluated using extracted URL, domain, and HTML-based features. These features included indicators like URL length, use of IP addresses, presence of special symbols, HTTPS validity, and domain age, among others.

The system workflow involved key stages such as **data preprocessing**, **feature extraction**, **model training**, **evaluation**, and **prediction**. Feature extraction was automated using Python scripts that analyzed each URL's components and WHOIS information. During testing, the models achieved high accuracy, with Random Forest outperforming other algorithms due to its ensemble-based decision-making capability.

The results confirmed that machine learning models can effectively identify phishing websites by learning patterns from historical data. The project also provided a deeper understanding of feature engineering, model comparison, and how algorithmic choices affect detection accuracy. Overall, the developed system demonstrated strong potential for real-world use in identifying phishing threats, thereby enhancing online security and user protection

The dataset used comprised various labeled instances of legitimate and phishing websites, which were subjected to rigorous **data preprocessing**, including handling of missing values, feature encoding, and normalization. Key **features** extracted included indicators such as:

- **URL-based features:** Length of URL, use of IP address instead of a domain name, presence of "@" or "-" symbols, and abnormal number of subdomains.
- **Domain-based features:** WHOIS registration details, domain age, expiration time, and DNS record validity.
- **HTML/JavaScript-based features:** Presence of iframe tags, abnormal redirects, and suspicious JavaScript events.

A comparative analysis was performed using several supervised machine learning algorithms including **Decision Tree, Random Forest, Support Vector Machine (SVM), Logistic Regression,** and **Gradient Boosting Classifier**. Among these, **Random Forest** and **Gradient Boosting** models yielded the highest accuracy, precision, and F1-score, proving their superior ability to capture non-linear relationships and mitigate overfitting.

The project workflow encompassed:

1. Data collection and labeling
2. Feature extraction and preprocessing
3. Model training and hyperparameter tuning
4. Model evaluation and validation
5. Real-world phishing URL testing

Experimental results confirmed that the proposed system can efficiently detect phishing websites with an accuracy exceeding **95%**, depending on the feature set and algorithm configuration.

The overall implementation provided valuable insights into **feature engineering**, **model interpretability**, and the role of ensemble methods in cybersecurity application

5.2 Conclusion

The implementation of machine learning for phishing detection has proven to be an effective and reliable approach to combating online fraud. By leveraging URL-based, domain-based, and HTML-based features, the system successfully classified websites with high precision and low false detection rates.Among the tested models, the **Random Forest classifier** provided the best overall performance due to its ability to handle feature variance and noise. The study confirms that phishing websites exhibit consistent structural and behavioral patterns that can be detected automatically through machine learning.This project demonstrated how integrating intelligent algorithms with web-based features can help detect phishing websites in real time, reducing the risk of credential theft and data breaches. Furthermore, the implementation enhanced understanding of supervised learning principles, data preprocessing, and model evaluation in cybersecurity contexts. The successful implementation of this phishing website detection model highlights the power of machine learning in cybersecurity applications. Through the extraction of both structural and behavioral web features, the system effectively differentiated between malicious and legitimate websites, minimizing false positives.

The study revealed that **phishing websites exhibit recurring patterns**—including shortened URLs, recently registered domains, absence of SSL certificates, and embedded suspicious scripts—that can be systematically detected by trained algorithms. Among the models evaluated, the **Random Forest classifier** emerged as the most effective due to its ensemble learning mechanism, robustness against noise, and interpretability.

The project not only demonstrated technical proficiency in data preprocessing, training, The research reaffirms that data-driven approaches can be powerful allies in the ongoing fight against online fraud, identity theft, and phishing scams. Ultimately, this system contributes toward **safer internet usage**, improved user awareness, and **automated phishing mitigation strategies**.

5.3 Recommendations:

To enhance the project's scalability, real-time performance, and applicability in dynamic online environments, the following recommendations are proposed for future research and development:

1. **Dataset Expansion and Diversity:**

   Continuously update and expand the dataset with new phishing patterns and legitimate sites from global threat intelligence sources like PhishTank, OpenPhish, and Alexa Top Sites. This ensures better generalization and adaptability to evolving attack methods.

2. **Integration with Real-Time Detection Systems:**

   Deploy the trained model within **browser extensions, email gateways, or proxy servers** to provide immediate alerts and blocking capabilities when phishing attempts are detected.

3. **Model Optimization and Lightweight Deployment:**

   Apply **model compression techniques**, **quantization**, or **knowledge distillation** to reduce computational overhead, making the model suitable for use on mobile devices or low-resource environments.

4. **Incorporation of Deep Learning Techniques:**

   Extend the system with **Neural Networks**, such as **CNNs** for analyzing webpage screenshots and **RNNs/LSTMs** for sequential URL pattern recognition. Deep learning could capture hidden relationships and contextual cues missed by traditional ML algorithms.

5. **Feature Enhancement and Explainability:**

   Include additional **content-based and visual features**, such as screenshot similarity, keyword frequency, JavaScript behavior, and CSS anomalies. Also, implement **explainable AI (XAI)** techniques like SHAP or LIME to make predictions more interpretable for end-users.

6. **Deployment on Cloud Infrastructure:**

   Host the trained model on **cloud platforms (AWS, Google Cloud, or Azure)** with APIs for real-time phishing detection services. This ensures scalability, multi-user access, and integration with enterprise-level security systems.

7. **Continuous Learning and Auto-Updates:**

   Enable periodic model retraining using streaming or incremental learning so the system automatically adapts to new phishing techniques without complete retraining.

8. **User Awareness and Interface Development:**

   Create an intuitive user interface that displays detection results, confidence levels, and educational feedback to increase awareness about phishing attempts.

CHAPTER-6

REFERENCES

Textbooks

1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016.
2. Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
3. Tom M. Mitchell, *Machine Learning*, McGraw-Hill Education, 1997.
4. Sebastian Raschka and Vahid Mirjalili, *Python Machine Learning*, Packt Publishing, 2019.
5. Alpaydin, Ethem. *Introduction to Machine Learning.* MIT Press, 2020.
6. Géron, Aurélien. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.* O'Reilly Media, 2023.
7. Goodfellow, Ian. *Deep Learning and Security: Understanding the Threat of Adversarial Attacks.* MIT Press, 2021.

Websites

1. Scikit-learn Documentation – https://scikit-learn.org/
2. Python WHOIS Library – https://pypi.org/project/python-whois/
3. BeautifulSoup Documentation – https://beautiful-soup-4.readthedocs.io/
4. PhishTank – https://phishtank.org/
5. Kaggle Dataset: Phishing Website Detection – https://www.kaggle.com/
6. Google Safe Browsing API – https://developers.google.com/safe-browsing/
7. • UCI Machine Learning Repository – *Phishing Websites Data Set*:

   https://archive.ics.uci.edu/ml/datasets/phishing+websites
8. • OWASP (Open Web Application Security Project) – *Phishing Prevention Guide*:

   https://owasp.org/

5. • Towards Data Science – *Phishing Website Detection Tutorials*: https://towardsdatascience.com/
6. • GitHub Repository – *Phishing Detection with ML*: https://github.com/topics/phishing-detection

Journals and Research Papers

1. Mohammad, R. M., Thabtah, F., & McCluskey, L., "Predicting Phishing Websites Based on Self-Structuring Neural Network," *Neural Computing and Applications*, Springer, 2014.
2. Abdelhamid, N., Ayesh, A., & Thabtah, F., "Phishing Detection Based Associative Classification," *Data Mining and Digital Media Analytics Journal*, 2017.
3. Ma, J., Saul, L. K., Savage, S., & Voelker, G. M., "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs," *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2009.
4. Basnet, R. B., Sung, A. H., & Liu, Q., "Feature Selection for Improved Phishing Detection," *International Journal of Information Security and Privacy (IJISP)*, 2012. Jain, A. K., & Gupta, B. B. (2018). *Phishing Detection: Analysis of Visual Similarity-Based Approaches.* Security and Communication Networks, Hindawi.
5. Aburrous, M., Hossain, M. A., Dahal, K., & Thabtah, F. (2010). *Intelligent Phishing Detection System for e-Banking Using Fuzzy Data Mining.* Expert Systems with Applications, Elsevier.
6. Rao, R. S., & Pais, A. R. (2019). *Detection of Phishing Websites Using an Efficient Feature-Based Machine Learning Framework.* Neural Computing and Applications, Springer.