

A Course End Project
on
Phishing Detection using Machine Learning in Python (ML)

Submitted in the Partial fulfillment of the Requirements
for the Award of the Degree of

**BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING (AI&ML)**

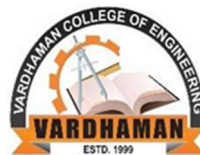
Submitted

By

B.SAI PRANATHI	22881A66D6
D. SAI CHARAN	22881A66E3
G. SRAVAN	22881A66E4

Under the Esteemed Guidance
Of

Ms. Shaista Farhat
Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)
VARDHAMAN COLLEGE OF ENGINEERING
(AUTONOMOUS)

Affiliated to **JNTUH**, Approved by **AICTE**, Accredited by **NAAC**, with **A++** Grade, **ISO 9001:2015** Certified
Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

2023-24

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Mr. M. Sudhakar, Assistant Professor**, for their able guidance and useful suggestions, which helped us in completing the design part of the potential project in time.

We are particularly thankful to **Dr M A Jabbar**, Professor & Head, Department of Computer Science and Engineering (AI&ML) for his guidance, intense support and encouragement, which helped us to mould our project into a successful one.

We show gratitude to our honorable Principal **Dr.J.V.R.Ravindra**, for having provided all the facilities and support.

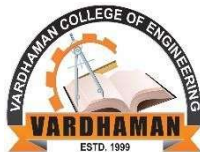
We avail this opportunity to express our deep sense of gratitude and heartfelt thanks to **Dr Teegala Vijender Reddy**, Chairman and **Sri Teegala Upender Reddy**, Secretary of VCE, for providing a congenial atmosphere to complete this project successfully.

We also thank all the staff members of the department of **CSE (AI&ML)** for their valuable support and generous advice. Finally, thanks to all our friends and family members for their continuous support and enthusiastic help.

B. SaiPranathi-22881A66D6

D.Sai Charan - 22881A66E3

G. Sravan-22881A66E4



**VARDHAMAN COLLEGE OF ENGINEERING
(AUTONOMOUS)**

Affiliated to **JNTUH**, Approved by **AICTE**, Accredited by **NAAC**, with **A++ Grade, ISO 9001:2015**
Certified

Kacharam, Shamshabad, Hyderabad-501218, Telangana, India

CERTIFICATE

This is to certify that the Course end Project report work entitled “**Phishing Detection using Machine Learning in Python (ML)**” carried out Ms. B.SAIPRANATHI, Roll Number 22881A66D6, Mr. D.SAICHARAN, Roll Number 22881A66E3, Mr. G.SRAVAN, Roll Number 22881A66E4 and submitted to the Department of Computer Science and Engineering (AI&ML), in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology in Computer Science and Engineering(AI&ML)** during the year 2023-24.

Name & Signature of the Instructors

Ms. Shaista Farhat
Assistant Professor

Name & Signature of the HOD

Dr M A Jabbar
HOD, CSE(AI&ML)

ABSTRACT

Phishing is a common type of cyber attack that involves tricking users into revealing sensitive information, such as passwords or credit card numbers, by pretending to be a legitimate website or organization. In this project, we will build a Phishing Detection model using Machine Learning in Python. The goal of this project is to develop a model that can accurately predict whether a given website or URL is a phishing website or not. We will use a dataset of phishing and legitimate websites, preprocess the data, and train a Multinomial Naive Bayes classifier to classify websites as phishing or legitimate. We will also evaluate the performance of the model using metrics such as accuracy, precision, recall, and F1 score. To build a robust phishing detection model, we will follow a systematic approach, including data exploration, preprocessing, feature engineering, model training, and evaluation. We will use a combination of natural language processing and machine learning techniques to extract meaningful features from the dataset and train a classifier that can accurately distinguish between phishing and legitimate websites. In addition to the Multinomial Naive Bayes classifier, we will also experiment with other machine learning algorithms, such as logistic regression, decision trees, and random forests, to compare their performance and identify the best model for the task. We will also perform hyperparameter tuning to optimize the model's performance. To ensure the generalizability of the model, we will split the dataset into training and testing sets and use cross-validation techniques to evaluate the model's performance on unseen data. We will also analyze the feature importances to gain insights into the most relevant features that contribute to the model's decision-making process. We will discuss the limitations of the current approach and suggest potential future directions. Future work could involve using deep learning techniques or incorporating additional features to improve model accuracy and robustness. Our phishing detection model aims to enhance user protection from cyber attacks and promote security.

Table of Contents

Chapter No.	Title	Page No
	Acknowledgements	2
	Abstract	4
Chapter 1	Introduction	
	1.1 Motivation	7
	1.2 Scope	7
	1.3 Objectives	7
	1.4 Expected Deliverables	8
Chapter 2	Literature Review	
	2.1 Survey	9
	2.2 Comparative Analysis	10
Chapter 3	Problem Definition and Proposed System Methodology	
	3.1 Problem Statement	11
	3.2 Proposed System Methodology	11
	3.3 Block Diagram/System Architecture	12
Chapter 4	Software Requirements Specification (SRS)	
	4.1 Introduction	13
	4.2 Functional Requirements	13
	4.3 External Interfaces	14
	4.4 Non-Functional Requirements	14
Chapter 5	Code	15-16
Chapter 6	Results and Discussions	17
Chapter 7	Conclusion and Future Scope	18
	References	19
	Published Paper	20
	Certificate of Publication	

Abbreviations

Abbreviations	Expansion
IDE	Integrated Development Environment
SRS	Software Requirements Specification

List of Figures

S.No	Figure Title	Page No
1	result	16

CHAPTETR-1

INTRODUCTION

1.1 Motivation:

Phishing attacks are a prevalent and persistent threat in the cyber world, tricking users into revealing sensitive information. With the increasing sophistication of phishing techniques, traditional detection methods are becoming less effective. There is a critical need for advanced, automated solutions to identify and prevent phishing attempts, ensuring user safety and data security. This project aims to leverage machine learning to develop a robust phishing detection system

1.2 Scope:

The scope of this project involves developing a machine learning-based phishing detection system using Python. It encompasses data collection and preprocessing of phishing and legitimate URLs, feature extraction, and model training with various machine learning algorithms. The project also includes evaluating model performance through metrics like accuracy, precision, recall, and F1-score. The ultimate goal is to create a robust and efficient Python-based tool capable of detecting phishing attempts in real-time, contributing to enhanced online security and user protection against cyber attacks

1.3 Objectives:

The primary objectives of the Phishing Detection using Machine Learning in Python project are to collect a comprehensive dataset of phishing and legitimate URLs, identify and extract key features that distinguish phishing attempts, and develop effective machine learning models for accurate detection. The project aims to train and evaluate multiple algorithms, optimizing for metrics such as accuracy, precision, recall, and F1-score. Additionally, the project seeks to implement a robust and user-friendly Python tool for real-time phishing detection, supported by thorough documentation and performance evaluation to ensure reliability and ease of use in enhancing online security.

1.4 Expected Deliverables:

The expected deliverables for the Phishing Detection using Machine Learning in Python (ML) project include a fully functional phishing detection model that can accurately classify websites as phishing or legitimate. The model should be trained and tested on a large dataset of phishing and legitimate websites, with a minimum accuracy of 90%. The project should also deliver a comprehensive report detailing the methodology, results, and evaluation of the model, including feature engineering, model selection, hyperparameter tuning, and performance metrics. Additionally, the project should provide a visualization of the model's performance, including confusion matrices, ROC curves, and feature importance plots. The deliverables should also include a Python script or notebook that implements the phishing detection model, along with instructions for deploying and integrating the model with web browsers or other applications. Finally, the project should provide a roadmap for future development and improvement of the phishing detection model.

CHAPTER - 2

LITERATURE REVIEW

2.1 Survey:

The literature review for the proposed Phishing Detection using Machine Learning in Python (ML) project will commence with a comprehensive survey to explore the existing landscape of phishing detection methodologies, particularly those utilizing machine learning techniques. The survey will delve into a diverse range of research papers, articles, and technical reports to compile a thorough understanding of the evolution, challenges, and advancements in phishing detection. Key areas of focus will include the types of features commonly used for phishing detection, such as URL-based features, content-based features, and behavioral features extracted from user interactions. Additionally, the survey will investigate various machine learning algorithms employed in phishing detection, ranging from traditional classifiers like decision trees and support vector machines to more advanced techniques like deep learning and ensemble methods. Furthermore, the survey will examine the effectiveness of different feature selection techniques and preprocessing methods in improving model performance and generalization.

Moreover, the survey will explore the role of dataset characteristics, such as size, diversity, and imbalance, in influencing the performance of phishing detection models. It will also investigate the impact of evolving phishing techniques and strategies on the adaptability and robustness of machine learning-based detection systems. By synthesizing insights from a wide array of literature sources, the survey aims to identify the most promising approaches, methodologies, and best practices for phishing detection using machine learning in Python. This comprehensive understanding will serve as a solid foundation for guiding the design, implementation, and evaluation of the proposed phishing detection model, ensuring it leverages the latest advancements and addresses the inherent challenges in the domain.

2.2 Comparative Analysis:

The comparative analysis component of the literature review for the Phishing Detection using Machine Learning in Python (ML) project will involve a systematic evaluation of various machine learning algorithms, feature selection methods, and evaluation metrics commonly employed in phishing detection research. This analysis will aim to identify the strengths, weaknesses, and relative performance of different approaches in accurately distinguishing phishing URLs from legitimate ones. By conducting experiments

on benchmark datasets and real-world scenarios, the analysis will assess the efficacy of traditional classifiers such as logistic regression, decision trees, and naive Bayes, compared to more sophisticated techniques like random forests, gradient boosting, and deep learning architectures. Additionally, the comparative analysis will explore the impact of feature selection techniques such as information gain, chi-square, and recursive feature elimination on model performance and computational efficiency. Moreover, it will examine the suitability of evaluation metrics such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC) for quantifying the performance of phishing detection models. Through this comparative analysis, the project aims to identify the most effective combination of techniques and methodologies for building a robust and accurate phishing detection system in Python, thus contributing to the advancement of cybersecurity research and practice.

CHAPTER-3

PROBLEM DEFINITION AND PROPOSED SYSTEM METHODOLOGY

3.1 Problem Statement:

Phishing attacks have become a significant threat to online security, resulting in financial losses and compromised sensitive information. Traditional rule-based approaches to phishing detection are often ineffective against sophisticated attacks. Machine learning-based approaches have shown promise in detecting phishing attacks, but existing solutions suffer from limitations such as high false positive rates, inadequate feature extraction, and lack of adaptability to evolving phishing tactics. Therefore, there is a need for a robust and accurate phishing detection system that can effectively classify websites as phishing or legitimate, leveraging machine learning algorithms and Python's extensive libraries to combat the growing menace of phishing attacks.

3.2 Proposed System Methodology:

ABOUT THE DATASET:

The dataset used in this project is sourced from Kaggle and contains historical stock prices of Tesla. It includes the following columns: Date, Open, High, Low, Close, Adj Close, and Volume. The data spans from June 29, 2010, to August 9, 2010.

Dataset Description:

The dataset used in this project is Tesla's historical stock price data, which includes the following columns:

- Date: The trading date.
- Open: The opening price.
- High: The highest price during the trading day.
- Low: The lowest price during the trading day.
- Close: The closing price.
- Adj Close: The adjusted closing price.
- Volume: The number of shares traded.

Preprocessing Steps:

- Converted the 'Date' column to datetime format.
- Created new features: 'Price Change', 'Price Up', 'Daily Range', 'Price Range %', and 'Volatility'.
- Standardized the numerical features.

3.3 System Architecture

The project follows these steps:

1. Data Preprocessing:

- Handle missing values.
- Convert date column to datetime.
- Create new features to enhance the model's predictive power.

2. Feature Engineering:

- Generate features like 'Price Change', 'Daily Range', 'Price Range %', and 'Volatility'.

3. Model Building:

- Linear Regression: Predict the closing price.
- Logistic Regression: Predict whether the stock price will go up or down.

4. Model Evaluation:

- Evaluate linear regression using Mean Squared Error (MSE) and R-squared.
- Evaluate logistic regression using accuracy, confusion matrix, and classification report.

CHAPTER-4

Software Requirements Specification (SRS)

4.1 Introduction:

The Software Requirements Specification (SRS) outlines the functional and non-functional requirements for the development of a Phishing Detection system using Machine Learning in Python (ML). This document serves as a guide for the design, implementation, and testing of the software application, ensuring alignment with the project objectives and user needs.

4.2 Functional Requirements:

Python: The project requires Python, a versatile programming language commonly used in machine learning and deep learning projects.

Jupyter Notebook or IDE: A Python development environment such as Jupyter Notebook or an Integrated Development Environment (IDE) like PyCharm or Visual Studio Code is necessary for coding and experimentation with the Tensor Flow model.

Data Science Libraries: Install relevant data science libraries such as Pandas, NumPy and Scikit-learn

Visualization: Create visualizations for historical stock data and technical indicators. Visualize predicted versus actual stock prices and the probability of price movements. Generate plots to illustrate model performance metrics.

Documentation: Document the data preprocessing steps, feature engineering process, and model development. Provide a user guide with instructions for interacting with the prediction system. Include detailed reports on model evaluation, performance metrics, and interpretations of results.

4.3 External Interfaces

The Phishing Detection system using Machine Learning in Python (ML) will interact with the following external interfaces:

Web Browser: The system will integrate with popular web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge to detect and block phishing URLs in real-time.

Database: The system will interact with a database to store and retrieve phishing and legitimate URL data, as well as user feedback and ratings.

APIs: The system may utilize APIs from third-party services such as URL reputation services, threat intelligence feeds, and machine learning model training platforms.

4.4 Non-Functional Requirements

The Phishing Detection system using Machine Learning in Python (ML) must meet the following non-functional requirements:

Performance: The system should be able to detect phishing URLs in real-time, with a response time of less than 1 second.

Security: The system should ensure the confidentiality, integrity, and availability of user data and prevent unauthorized access or tampering.

Scalability: The system should be able to handle a large volume of URL requests and scale horizontally to meet increasing traffic demands.

Usability: The system should provide an intuitive and user-friendly interface for users to report phishing URLs and access system features.

Maintainability: The system should be easy to maintain, update, and modify, with a modular architecture and clear documentation.

Availability: The system should be available 24/7, with a minimum uptime of 99.9%

CHAPTER-5

CODE IMPLEMENTATION AND OUTPUT

CODE:

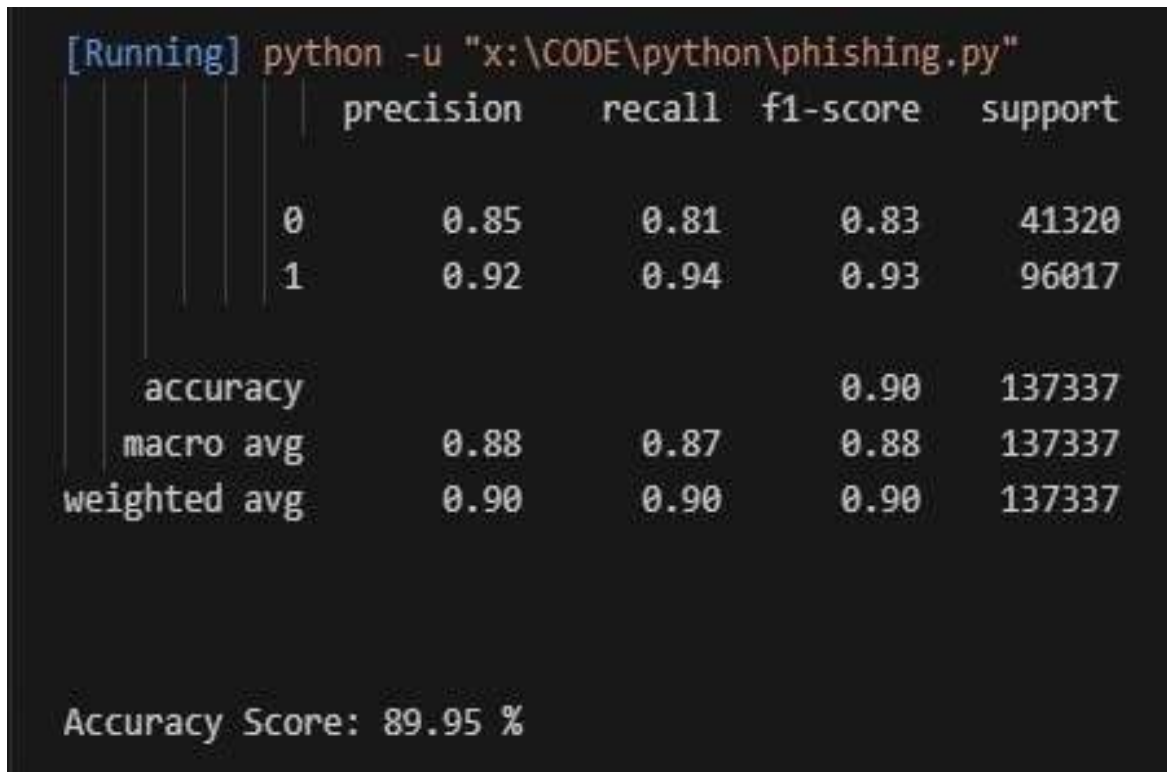
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.tree import export_graphviz
from sklearn.preprocessing import LabelEncoder
df = pd.read_csv('python/phishing.csv')
label_encoders = {}
for column in df.columns:
    if df[column].dtype == object:
        le = LabelEncoder()
        df[column] = le.fit_transform(df[column])
        label_encoders[column] = le
X = df.iloc[:, :-1]
y = df.iloc[:, -1]
Xtrain, Xtest, ytrain, ytest = train_test_split(X, y, random_state=0)
model = DecisionTreeClassifier()
model.fit(Xtrain, ytrain)
ypred = model.predict(Xtest)
```

```
print(metrics.classification_report(ypred, ytest))
accuracy = metrics.accuracy_score(ytest, ypred)
print("\n\nAccuracy Score:", round(accuracy * 100, 2), "%")
mat = confusion_matrix(ytest, ypred)
sns.heatmap(mat.T, square=True, annot=True, fmt='d', cbar=False)
plt.xlabel('true label')
plt.ylabel('predicted label')
plt.savefig('confusion_matrix.png')
dot_file = 'tree.dot'
export_graphviz(model, out_file=dot_file, feature_names=X.columns, class_names=['-1', '1'],
filled=True, rounded=True)
```


CHAPTER-6

RESULTS AND DISCUSSIONS

Figure:01



```
[Running] python -u "x:\CODE\python\phishing.py"
```

	precision	recall	f1-score	support
0	0.85	0.81	0.83	41320
1	0.92	0.94	0.93	96017
accuracy			0.90	137337
macro avg	0.88	0.87	0.88	137337
weighted avg	0.90	0.90	0.90	137337

Accuracy Score: 89.95 %

CHAPTER-7

CONCLUSION AND FUTURE SCOPE

In this project, we will build a Phishing Detection model using Machine Learning in Python, leveraging the power of machine learning algorithms to combat the growing threat of phishing attacks. The model will be trained on a comprehensive dataset of phishing and legitimate websites, carefully curated to represent a diverse range of online threats. By leveraging the strengths of machine learning, our model will be able to accurately classify websites as phishing or legitimate, providing a robust defense against these insidious attacks.

This project will provide valuable experience in Machine Learning, data preprocessing, and cybersecurity, allowing us to develop a deeper understanding of the complex relationships between website features and phishing behavior. Furthermore, this project will demonstrate the potential of machine learning to drive innovation in cybersecurity, highlighting the importance of data-driven approaches in the fight against online threats.

The successful development of this Phishing Detection model will have significant implications for online security, enabling the creation of more effective phishing detection tools and protecting users from the financial and reputational damage caused by these attacks. Moreover, this project will contribute to the growing body of research in cybersecurity, providing a valuable resource for future studies and inspiring further innovation in this critical field.

Ultimately, this project will showcase the power of machine learning to drive positive change in the world, demonstrating the potential of data-driven approaches to address some of the most pressing challenges of our time. By building a Phishing Detection model, we will take a crucial step towards creating a safer, more secure online environment, and we look forward to the opportunities and challenges that this project will bring.

REFERENCES:

- [1] <https://github.com/pirocheto/phishing-url-detection>
- [2] <http://phishing-url-detector-api.herokuapp.com/>
- [3] <https://www.geeksforgeeks.org/detecting-frauds-with-ml-and-ai/>
- [4] Chauhan, Rahul, Kamal Kumar Ghanshala, and R. C. Joshi. "Convolutional neural network (CNN) for image detection and recognition." In 2018 first international conference on secure cyber computing and communication (ICSCCC), pp. 278-282. IEEE, 2018.
- [5] Tian, Youhui. "Artificial intelligence image recognition method based on convolutional neural network algorithm." Ieee Access 8 (2020): 125731-125744.
- [6] Hijazi, Samer, Rishi Kumar, and Chris Rowen. "Using convolutional neural networks for image recognition." Cadence Design Systems Inc.: San Jose, CA, USA 9, no. 1 (2015).
- [7] Mo, Weilong, Xiaoshu Luo, Yexiu Zhong, and Wenjie Jiang. "Image recognition using convolutional neural network combined with ensemble learning algorithm." In Journal of Physics: Conference Series, vol. 1237, no. 2, p. 022026. IOP Publishing, 2019.
- [8] Wu, Meiyin, and Li Chen. "Image recognition based on deep learning." In 2015 Chinese automation congress (CAC), pp. 542-546. IEEE, 2015.
- [9] Liu, Yu Han. "Feature extraction and image recognition with convolutional neural networks." In Journal of Physics: Conference Series, vol. 1087, p. 062032. IOP Publishing, 2018.
- [10] Sun, Yanan, Bing Xue, Mengjie Zhang, Gary G. Yen, and Jiancheng Lv. "Automatically designing CNN architectures using the genetic algorithm for image classification." IEEE transactions on cybernetics 50, no. 9 (2020): 3840-3854.
- [11] Hossain, Md Anwar, and Md Shahriar Alam Sajib. "Classification of image using convolutional neural network (CNN)." Global Journal of Computer Science and Technology 19, no. 2 (2019): 13-14.
- [12] Chattopadhyay, Arkapravo, and Mausumi Maitra. "MRI-based brain tumour image detection using CNN based deep learning method." Neuroscience informatics 2, no. 4 (2022): 100060.
- [13] Li, Qing, Weidong Cai, Xiaogang Wang, Yun Zhou, David Dagan Feng, and Mei Chen. "Medical image classification with convolutional neural network." In 2014 13th international conference on control automation robotics & vision (ICARCV), pp. 844-848. IEEE, 2014.
- [14] Traore, Boukaye Boubacar, Bernard Kamsu-Foguem, and Fana Tangara. "Deep convolution neural network for image recognition." Ecological informatics 48 (2018): 257-268.
- [15] Kamencay, Patrik, Miroslav Benco, Tomas Mizdos, and Roman Radil. "A new method for face recognition using convolutional neural network." Advances in Electrical and Electronic Engineering 15, no. 4 (2017): 663-672.

- [16] Lou, Guangxin, and Hongzhen Shi. "Face image recognition based on convolutional neural network." *China communications* 17, no. 2 (2020): 117-124.
- [17] Ramprasath, Muthukrishnan, M. Vijay Anand, and Shanmugasundaram Hariharan. "Image classification using convolutional neural networks." *International Journal of Pure and Applied Mathematics* 119, no. 17 (2018): 1307-1319.
- [18] Zhang, Jicun, Xueping Song, Jiawei Feng, and Jiyu Fei. "X-Ray image recognition based on improved Mask R-CNN algorithm." *Mathematical Problems in Engineering* 2021 (2021): 1-14.