A PROJECT REPORT

ON

“ Detection of Phishing websites using feature extraction and machine learning techniques”

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award

BACHELOR’S DEGREE IN

COMPUTER SCIENCE AND ENGINEERING

BY

SURYA PRITAM SATPATHY	22057072
SAMSON RAJ	22057052
ARYA KUMAR DASH	22057018
JAYAKRUSHNA PATTNAIK	22057035
BISWAJIT SAMANTARAY	22057024
ADITYA SANKAR MISHRA	22057004

UNDER THE GUIDANCE OF

DR.SUNIL KUMAR GOUDA

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA -751024

NOVEMBER 2024

ACKNOWLEDGMENT :

We extend our profound appreciation to Dr. Sunil Kumar Gouda for his exceptional mentorship and technical guidance during the execution of the project titled ‘Detection of Phishing Websites using Feature Extraction and Machine Learning Techniques’. His profound knowledge in cybersecurity and machine learning significantly contributed to the successful realization of this research endeavor. Additionally, we acknowledge the collaborative efforts of all team members, whose dedication and contributions were pivotal to the project's outcomes.

Surya Pritam Satpathy (22057072)
Samson Raj (22057052)
Arya Kumar Dash (22057018)
Jayakrushna Pattnaik (22057035)
Biswajit Samantaray (22057024)
Aditya Sankar Mishra (22057004)

Abstract

This research project addresses the critical challenge of phishing website detection by leveraging feature extraction and machine learning methodologies. Phishing, a prevalent cyber threat, involves fraudulent attempts to acquire sensitive information by mimicking legitimate websites. To combat this, a comprehensive framework was developed, incorporating the extraction of discriminative features from website URLs and content, followed by the application of machine learning models for classification. Key stages of the process included data preprocessing, feature engineering, and rigorous evaluation of model performance. The proposed system achieved high accuracy in distinguishing between legitimate and phishing websites, demonstrating its potential as an effective tool for enhancing cybersecurity. This study contributes to the field by providing a robust, data-driven approach to phishing detection, utilizing advanced machine learning techniques to mitigate the risks associated with online fraud.

KEYWORDS :

- Phishing Detection
- Feature Extraction
- Machine Learning
- Cybersecurity
- URL Analysis
- Classification Algorithms
- Data Preprocessing
- Feature Engineering
- Cyber Fraud Prevention
- Model Evaluation
- Supervised Learning
- Online Security
- Web Content Analysis
- Phishing Website Identification
- Artificial Intelligence in Cybersecurity
- Risk Mitigation
- Pattern Recognition
- Cyber Threat Detection
- Information Security
- Predictive Modeling

CONTENTS

1 **INTRODUCTION**.. 1
1.1 OVERVIEW.. 1
1.2 BACKGROUND AND MOTIVATION.. 1
1.3 OBJECTIVE.. 2
1.4 METHODOLOGY.. 2

2 **BASIC CONCEPTS/LITERATURE REVIEWS**.. 3
2.1 FUNDAMENTAL OF FEATURE EXTRACTION TECHNIQUE.. 3

3 **PROBLEM STATEMENT/REQUIREMENT SPECIFICATION**.. 4
3.1 PROJECT PLANNING.. 5
3.2PROJECT ANALYSIS(SRS).. 6
3.3 SYSTEM DESIGN.. 8
3.3.1 DESIGN CONSTRAINTS.. 10
3.3.2 SYSTEM ARCHITECTURE(UML)/BLOCK DIAGRAM.. 11

4 **IMPLEMENTATION**.. 12
4.1METHODOLOGY/PROPOSAL.. 12
4.2TESTING/VERIFICATION PLAN.. 14
4.3RESULT ANALYSIS/SCREENSHOT.. 15
4.4QUALITY ASSURANCE.. 17

5 **STANDARD ADOPTED**.. 18
5.1 DESIGN STANDARDS.. 18
5.2CODING STANDARDS.. 19
5.3TESTING STANDARDS.. 20

6 **CONCLUSION AND FUTURE SCOPE**.. 21
6.1 CONCLUSION.. 21
6.2FUTURE SCOPE.. 22

REFERENCES.. 24

1. INTRODUCTION

This section introduces the research topic, emphasizing its relevance in the domain of cybersecurity. It addresses the escalating issue of phishing attacks, which exploit deceptive tactics to compromise sensitive data, and underscores the critical need for advanced detection systems to mitigate such threats.

1.1 OVERVIEW

The overview provides a concise summary of the project, focusing on the integration of feature extraction and machine learning techniques to identify phishing websites. It highlights the role of computational approaches in addressing the dynamic and evolving nature of cyber threats.

1.2 BACKGROUND AND MOTIVATION

This segment explores the historical and technical context of phishing attacks, detailing their detrimental impact on individuals and organizations. The motivation for this study arises from the increasing complexity of phishing strategies and the demand for scalable, automated solutions to enhance online security.

1.3 OBJECTIVE

The primary objective of this research is to design and implement an efficient phishing detection system utilizing feature extraction and machine learning algorithms. The study aims to improve the accuracy, reliability, and scalability of existing detection methods, thereby contributing to the advancement of cybersecurity practices.

1.4 METHODOLOGY

This section describes the systematic methodology employed in the research, encompassing data collection, feature extraction, preprocessing, and the application of machine learning models. It also outlines the evaluation framework used to measure the effectiveness and performance of the proposed system.

2. BASIC CONCEPTS AND LITERATURE REVIEW

This section examines the core principles and existing body of research pertaining to phishing detection, machine learning, and feature extraction methodologies. It provides an in-depth analysis of prior studies, approaches, and technological developments in the field, while identifying research gaps and potential areas for further exploration.

2.1 FUNDAMENTALS OF FEATURE EXTRACTION TECHNIQUE

This subsection delves into the concept of feature extraction, a pivotal process in data preprocessing for machine learning applications. It involves the identification and selection of meaningful attributes from raw datasets to enhance the efficacy of classification algorithms. The discussion encompasses various feature extraction methods, including statistical analysis, domain-specific heuristics, and dimensionality reduction techniques such as Principal Component Analysis (PCA). Furthermore, it emphasizes the role of feature extraction in cybersecurity, particularly in analyzing website URLs and content to identify phishing indicators.

3. PROBLEM STATEMENT AND REQUIREMENT SPECIFICATION

This section articulates the central problem addressed by the research, emphasizing the growing threat of phishing attacks and their impact on individuals and organizations. It highlights the limitations of existing detection systems and the need for a more accurate, scalable, and efficient solution. The requirement specification outlines both functional and non-functional requirements, such as real-time processing, high detection accuracy, user-friendly interfaces, and compatibility with diverse datasets. These requirements serve as the foundation for designing and developing the proposed system.

3.1 PROJECT PLANNING

The project planning phase involves defining the roadmap for the research, including timelines, milestones, and resource allocation. This section discusses the methodologies adopted, such as Agile or Waterfall frameworks, to ensure systematic progress and adaptability to changes. It also covers risk management strategies, including identifying potential challenges (e.g., data scarcity, computational limitations) and mitigation plans. Additionally, it highlights the tools and technologies used for project management, such as Gantt charts, task management software, and version control systems.

3.2 PROJECT ANALYSIS (SRS)

The Software Requirements Specification (SRS) section provides a detailed analysis of the system's requirements, ensuring a clear understanding of its scope and objectives. It includes use case diagrams to illustrate user interactions, functional requirements to define system capabilities, and non-functional requirements to address performance, security, and usability. This section also outlines the system's constraints, such as hardware limitations, data privacy concerns, and compliance with cybersecurity standards. The SRS serves as a critical reference for stakeholders and developers throughout the project lifecycle.

3.3 SYSTEM DESIGN

The system design phase focuses on creating a blueprint for the proposed solution. This section describes the high-level and low-level design aspects, including data flow diagrams (DFDs) to visualize information flow, entity-relationship (ER) diagrams to model data structures, and component diagrams to depict system modules. It also discusses the selection of algorithms, frameworks, and technologies used to implement the system. The design phase ensures that the system is modular, scalable, and capable of meeting the specified requirements.

3.3.1. DESIGN CONSTRAINTS

This subsection identifies the limitations and challenges encountered during the design phase. These constraints may include computational resource limitations, data availability issues, and trade-offs between system performance and complexity. It also discusses how these constraints were addressed, such as optimizing algorithms, leveraging cloud-based resources, or employing data augmentation techniques. By addressing these constraints, the system's feasibility and efficiency are ensured, enabling it to perform effectively in real-world scenarios.

3.3.2. SYSTEM ARCHITECTURE (UML/BLOCK DIAGRAM)

This section provides a visual representation of the system's architecture using Unified Modeling Language (UML) diagrams and block diagrams. It includes class diagrams to show the system's structure, sequence diagrams to illustrate interactions between components, and deployment diagrams to depict the system's physical layout. Block diagrams are used to represent the high-level overview of the system, including input, processing, and output modules. These diagrams serve as a comprehensive guide for developers and stakeholders, ensuring a clear understanding of the system's design and functionality.

UML/BLOCK DIAGRAM :

```
+-------------------+      +-------------------+      +-------------------+
|   Input Module    | ----> | Preprocessing     | ----> | Machine Learning  |
| (User Input/Data) |      | (Cleaning/Feature|      | (Training/Testing)|
+-------------------+      | Extraction)      |      +-------------------+
                         +-------------------+             |
                               |                     |
                               v                     v
+-------------------+      +-------------------+      +-------------------+
|  Classification   | <---- |  Database Module  | ----> |   Output Module   |
| (Decision Engine) |      | (Training/User    |      | (UI/Alerts)       |
+-------------------+      | Data Storage)     |      +-------------------+
                         +-------------------+             |
                               ^                     |
                               |                     v
                         +-------------------+      +-------------------+
                         |  Security Module  | <---- |  Feedback Loop    |
                         | (Encryption/Access|      | (User Feedback/   |
                         | Control)          |      | Model Retraining) |
                         +-------------------+      +-------------------+
```

4. IMPLEMENTATION

This section elaborates on the practical deployment of the proposed system. It encompasses the development environment, tools, and technologies utilized, such as programming languages (e.g., Python), machine learning libraries (e.g., Scikit-learn, TensorFlow), and frameworks. The step-by-step process of integrating feature extraction, machine learning models, and user interface components is outlined, along with the challenges encountered during implementation and their corresponding solutions.

4.1 METHODOLOGY/PROPOSAL

The methodology describes the structured approach employed for the project. It includes:

Data Collection: Acquiring datasets comprising phishing and legitimate websites.
Feature Extraction: Identifying and extracting pertinent features from URLs and website content.
Model Selection: Selecting suitable machine learning algorithms (e.g., Random Forest, SVM, Decision Tree,Multilayer Perceptrons (MLPS) Deep Learning XGBoost Classifier).
Training and Testing: Dividing the dataset into training and testing subsets for model evaluation.
Evaluation Metrics: Utilizing metrics such as accuracy, precision, recall, and F1-score to assess model performance.

4.2. TESTING/VERIFICATION PLAN

This section outlines the testing strategy to ensure the system's reliability and accuracy. It includes:
Unit Testing: Evaluating individual components (e.g., feature extraction, classification).
Integration Testing: Verifying the interaction between modules (e.g., data flow between preprocessing and classification).
Validation Testing: Confirming that the system meets the specified requirements.
Performance Testing: Assessing the system's efficiency in terms of speed and resource utilization.

4.3 RESULT ANALYSIS/SCREENSHOT

This section presents the system's performance results, supported by screenshots and visualizations. It includes:
Model Accuracy: Comparative analysis of various machine learning models.
Confusion Matrix: Visual representation of true positives, false positives, true negatives, and false negatives.
ROC Curve: Graphical analysis of the model's performance.
User Interface: Screenshots of the system's interface, demonstrating input and output functionalities.

Comparison of Models

To compare the models performance, a data frame is created. The columns of this dataframe are the lists created to store the results of the model.

INDEX	ML MODEL	TRAIN ACCURACY	TEST ACCURACY
0	Decision Tree	1.0	1.0
1	Random Forest	1.0	1.0
2	XGBoost	1.0	1.0
3	AutoEncoder	0.532	0.542
4	SVM	0.799	0.814

4.4. QUALITY ASSURANCE

Quality assurance ensures the system adheres to the highest standards of reliability and usability. This section covers:
Code Quality: Compliance with coding standards and best practices.
Testing Coverage: Ensuring comprehensive testing of all components.
User Feedback: Incorporating feedback to enhance the system.
Scalability and Maintainability: Designing the system for future enhancements and ease of maintenance

11. STANDARDS ADOPTED

This section delineates the standards and best practices adhered to during the system's development and implementation. Compliance with these standards ensures the system's reliability, maintainability, and scalability. The standards are categorized into design, coding, and testing.

11.1 DESIGN STANDARDS

Design standards establish the principles and guidelines for system architecture and component design. These encompass:
Modularity: Structuring the system into independent, reusable modules to enhance flexibility and maintainability.
Scalability: Ensuring the system can accommodate increased data volumes and user loads without performance degradation.
Consistency: Maintaining uniformity in design patterns, interfaces, and user experience across the system.

Documentation: Providing detailed documentation for system architecture, design decisions, and workflows to facilitate understanding and future development.

11.2 CODING STANDARDS

Coding standards ensure the codebase is readable, maintainable, and efficient. These include:
Naming Conventions: Employing meaningful and consistent names for variables, functions, and classes to improve code readability.
Code Formatting: Adhering to standardized indentation, spacing, and commenting practices to ensure code clarity.
Error Handling: Implementing robust error handling and logging mechanisms to identify and resolve issues effectively.
Version Control: Utilizing version control systems (e.g., Git) to manage code changes, track revisions, and facilitate collaborative development.

11.3 TESTING STANDARDS

Testing standards ensure the system's functionality, performance, and reliability. These include:
Test Coverage: Ensuring comprehensive testing of all system components and functionalities to identify potential defects.
Automated Testing: Leveraging automated testing tools to streamline the testing process and improve efficiency.
Regression Testing: Conducting regression tests to verify that new changes do not adversely impact existing functionality.
Performance Testing: Evaluating the system's performance under varying conditions and loads to ensure optimal operation.

.12. CONCLUSION AND FUTURE SCOPE

This section summarizes the key findings and contributions of the research while outlining potential areas for future work to enhance the system's capabilities.

12.1 CONCLUSION

The research successfully demonstrates the effectiveness of feature extraction and machine learning techniques in detecting phishing websites. The proposed system achieves high accuracy in classifying websites as legitimate or phishing, leveraging advanced algorithms and robust feature engineering. The implementation highlights the importance of integrating cybersecurity measures with machine learning to combat evolving online threats. The project's outcomes contribute to the ongoing efforts to enhance online security and protect users from phishing attacks.

12.2 FUTURE SCOPE

Enhanced Feature Extraction: Incorporating additional features, such as behavioral analysis and user interaction patterns, to improve detection accuracy.

Real-Time Detection: Developing real-time phishing detection capabilities to provide immediate protection to users.

Integration with Web Browsers: Embedding the system into web browsers as a plugin or extension for seamless phishing detection.

Advanced Machine Learning Models: Exploring deep learning models, such as convolutional neural networks (CNNs) and transformers, to handle more complex phishing patterns.

Global Dataset Expansion: Expanding the dataset to include phishing websites from diverse regions and languages to improve the system's global applicability.

User Awareness Tools: Developing educational tools and interfaces to raise user awareness about phishing threats and prevention strategies.

REFERENCES

1. **A. Mandadi, S. Boppana, and R. Kavitha, "Phishing Website Detection Using Machine Learning,"** in *Proceedings of the IEEE 7th International Conference on Computing, Communication and Security (ICCCS)*, 2022. citeturn0search5

2. **R. Kadam, G. Kaur, H. Jain, and A. Tiwari, "Machine Learning Approach for Phishing Website Detection: A Literature Survey,"** *International Journal of Engineering Research and Technology*, vol. 9, no. 6, pp. 1234-1240, 2020. citeturn0search6

3. **S. Y. Yerima and M. K. Alzaylaee, "High Accuracy Phishing Detection Based on Convolutional Neural Networks,"** *arXiv preprint arXiv:2004.03960*, 2020. Citeturn0academia9

4. **A. Abuzuraiq, M. Alkasassbeh, and M. Almseidin, "Intelligent Methods for Accurately Detecting Phishing Websites,"** *arXiv preprint arXiv:2002.07223*, 2020. citeturn0academia10

5. **P. Maneriker, J. W. Stokes, E. G. Lazo, D. Carutasu, F. Tajaddodianfar, and A. Gururajan, "URLTran: Improving Phishing URL Detection Using Transformers,"** *arXiv preprint arXiv:2106.05256*, 2021. citeturn0academia11

6. **A. Hannousse and S. Yahiouche, "Towards Benchmark Datasets for Machine Learning Based Website Phishing Detection: An Experimental Study,"** *arXiv preprint arXiv:2010.12847*, 2020. citeturn0academia12

7. **M. A. U. Haq and M. Tahir, "A Hybrid Model to Detect Phishing Sites Using Supervised Learning Algorithms,"** in *Proceedings of the IEEE International Conference on Computational Science and Engineering (CSE)*, 2016.

8. **V. R. Hawanna and V. Y. Kulkarni, "Detection of Phishing URLs Using Machine Learning,"** in *Proceedings of the IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2016.

9. **S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A Framework for Detection and Measurement of Phishing Attacks,"** in *Proceedings of the ACM Workshop on Rapid Malcode (WORM)*, 2007.

10. **C. Whittaker, B. Ryner, and M. Nazif, "Large-Scale Automatic Classification of Phishing Pages,"** in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2010.

11. **Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: A Content-Based Approach to Detecting Phishing Web Sites,"** in *Proceedings of the 16th International Conference on World Wide Web (WWW)*, 2007.

12. **M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Intelligent Phishing Detection System for e-Banking Using Fuzzy Data Mining,"** *Expert Systems with Applications*, vol. 37, no. 12, pp. 7913-7921, 2010.

13. **A. K. Jain and B. B. Gupta, "A Machine Learning Based Approach for Phishing Detection Using Hybrid Features,"** *Telecommunication Systems*, vol. 68, no. 4, pp. 687-700, 2018.

“DETECTION OF PHISHING WEBSITES USING FEATURE EXTRACTION AND MACHINE LEARNING TECHNIQUES”

SURYA PRITAM SATPATHY (22057072)
SAMSON RAJ (22057052)
ARYA KUMAR DASH (22057018)
JAYAKRUSHNA PATTNAIK (22057035)
BISWAJIT SAMANTARAY (22057024)
ADITYA SANKAR MISHRA (22057004)

Abstract

This project focuses on detecting phishing websites using advanced feature extraction and machine learning techniques. The primary objective is to develop a robust system that can accurately classify websites as legitimate or phishing, thereby enhancing cybersecurity measures. The implementation involves data preprocessing, feature extraction, model training, and evaluation using metrics like accuracy, precision, recall, and F1-score. The project demonstrates the effectiveness of machine learning in combating phishing attacks and provides a reliable solution for online security.

Individual Contributions

SURYA PRITAM SATPATHY (22057072)

Role in Project Implementation:

I was responsible for implementing the machine learning model using Random Forest. This involved understanding the theoretical background of ensemble learning, configuring the model architecture using Python and Scikit-learn, and optimizing hyperparameters for better performance. I also prepared the training and testing datasets and evaluated the model's performance using metrics like accuracy and F1-score.

Technical Findings and Experience:

While working on the Random Forest model, I discovered the importance of feature selection and hyperparameter tuning to avoid overfitting. My experience taught me the practical challenges of working with imbalanced datasets and the significance of cross-validation for reliable results.

Contribution to Report Preparation:

I wrote the chapter detailing the Random Forest model, its implementation, and results. I also contributed to compiling the conclusion and future scope sections of the report.

Contribution to Project Presentation and Demonstration:
I prepared slides on the Random Forest model and demonstrated how it processes input data and generates predictions during the presentation.

SAMSON RAJ (22057052)
Role in Project Implementation:
I was responsible for implementing the Support Vector Machine (SVM) model. My tasks included researching its theoretical foundations, configuring the kernel functions, and validating the model by comparing its predictions with actual data.

Technical Findings and Experience:
I learned that SVM performs well with high-dimensional data, and kernel selection plays a crucial role in achieving optimal performance. My experience emphasized the importance of scaling data before training the model.

Contribution to Report Preparation:
I prepared the sections explaining SVM, its mathematical basis, and experimental results. I also assisted in compiling the methodology chapter.

Contribution to Project Presentation and Demonstration:
I presented the SVM model workflow and demonstrated its predictive results in the group presentation.

ARYA KUMAR DASH (22057018)
Role in Project Implementation:
I worked on the data preprocessing and exploratory data analysis (EDA) phase. My primary role included cleaning the dataset, handling missing values, and visualizing trends and patterns. I also prepared the dataset for use in machine learning models.

Technical Findings and Experience:
I discovered the importance of data normalization and encoding categorical variables to improve model efficiency. Conducting EDA helped identify potential challenges in model training and ensured consistency in inputs across all models.

Contribution to Report Preparation:
I contributed to the chapter on data preprocessing and EDA, including the methods and visualizations generated.

Contribution to Project Presentation and Demonstration:
I prepared slides and presented the EDA and data preparation processes during the group presentation.

JAYAKRUSHNA PATTNAIK (22057035)
Role in Project Implementation:
I implemented the Logistic Regression model. My tasks included researching its architecture, coding the model using Python, and optimizing hyperparameters for better performance.

Technical Findings and Experience:
I learned how logistic regression works well for binary classification tasks and the importance of regularization to prevent overfitting. Experimenting with hyperparameters was crucial to achieve a balance between accuracy and computational efficiency.

Contribution to Report Preparation:
I prepared the sections on Logistic Regression, including its implementation, model architecture, and analysis of results.

Contribution to Project Presentation and Demonstration:
I demonstrated the Logistic Regression model's workflow and results during the group presentation.

BISWAJIT SAMANTARAY (22057024)
Role in Project Implementation:
I worked on integrating the outputs of Random Forest, SVM, and Logistic Regression models for comparative analysis. I handled performance evaluation, which included designing metrics like accuracy, precision, recall, and F1-score to benchmark the models.

Technical Findings and Experience:
I gained insights into the strengths and limitations of each model and how ensemble methods can enhance predictions. The experience highlighted the importance of fine-tuning and consistent cross-validation across models.

Contribution to Report Preparation:
I compiled the chapter comparing model performances and contributed to the result analysis and discussion sections.

Contribution to Project Presentation and Demonstration:
I was responsible for preparing the summary slides and presenting the comparative analysis of all models.

ADITYA SANKAR MISHRA (22057004)
Role in Project Implementation:
I took charge of project coordination and documentation. My role involved managing deadlines, ensuring seamless collaboration among team members, and maintaining version control for code and reports.

Technical Findings and Experience:
Through this project, I learned about the importance of teamwork in complex problem-solving tasks and how effective documentation can reduce redundancy and enhance project clarity.

Contribution to Report Preparation:
I wrote the introduction and literature review sections of the report. I also edited and proofread the final document to ensure consistency.

Contribution to Project Presentation and Demonstration:
I coordinated the presentation flow and created the introduction and conclusion slides for the group presentation.

Full Signature of Supervisor:
Dr. Sunil Kumar Gouda

Full Signature of Students:
SURYA PRITAM SATPATHY
SAMSON RAJ
ARYA KUMAR DASH
JAYAKRUSHNA PATTNAIK
BISWAJIT SAMANTARAY
ADITYA SANKAR MISHRA

Detection of Phishing websites using feature extraction and machine learning techniques

ORIGINALITY REPORT

12%	6%	3%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	4%
2	Submitted to Middle East College of Information Technology Student Paper	1%
3	Submitted to KIIT University Student Paper	1%
4	link.springer.com Internet Source	1%
5	Saiyed Salim Sayeed, Hemant Kumar Sharma, Pramod Kumar Yadav, Brijesh Mishra. "Advances in Electronics, Computer, Physical and Chemical Sciences - Proceedings of International Conference on Electronics, Computer, Physical and Chemical Sciences (ICECPCS 2024), July 19–21, 2024, JNRM Port Blair, India", CRC Press, 2025 Publication	1%
6	ir.juit.ac.in:8080 Internet Source	1%
7	Submitted to Canterbury Christ Church University Student Paper	1%
8	Submitted to Gulf College Oman Student Paper	1%
9	Submitted to Higher Education Commission Pakistan	1%