BINARY MULTILINGUAL MACHINE GENERATED TEXT DETECTION

A PROJECT REPORT

Submitted by,

G JAHNAVI	**20211CSE0143**
U YASHASWINI	**20211CSE0144**
J MEGHANA	**20211CSE0169**
K KOWSHIK NARAYANA	**20211CSE0133**

Under the guidance of,

Dr. JAYACHANDRAN ARUMUGAM

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At

PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2025

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the Project report **“Binary Multilingual Machine Generated Text Detection”** being submitted by “G.Jahnavi, U.Yashaswini, J.Meghana, K.Kowshik Narayana ” bearing roll numbers “20211CSE0143, 20211CSE0144, 20211CSE0169, 20211CSE0133”in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

Dr. JayaChandran Arumugam
Professor
School of CSE
Presidency University

Dr. Asif Mohammed
Associate Professor & HoD
School of CSE
Presidency University

Dr. L. SHAKKEERA
Associate Dean
School of CSE
Presidency University

Dr. MYDHILI NAIR
Associate Dean
School of CSE
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-VC School of Engineering
Dean -School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **Binary Multilingual Machine Generated Text Detection** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Dr. JayaChandran Arumugam, Professor, School of Computer Science Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

NAME	ROLL NO	SIGNATURE
G.JAHNAVI	20211CSE0143	
U.YASHASWINI	20211CSE0144	
J.MEGHANA	20211CSE0169	
K.KOWSHIK NARAYANA	20211CSE0133	

ABSTRACT

With the rapid advancement of natural language generation technologies, distinguishing machine generated text from human-written content has become increasingly challenging yet essential. This project aims to develop a robust, multilingual system capable of accurately identifying machine generated text across languages, including English, Indonesian, German, and Russian. Utilizing a substantial dataset of 674,083 training samples and 288,894 development samples characterized by attributes such as source, sub-source, language, generation model, label, and text we explore the efficacy of various machine learning and deep learning algorithms.

To achieve reliable classification, the system integrates Random Forest, Long Short-Term Memory (LSTM) networks, Bidirectional Encoder Representations from Transformers (BERT), Decision Tree, and Logistic Regression models. Each model is rigorously evaluated on its ability to handle multilingual data, focusing on both accuracy and computational efficiency. This project combines traditional machine learning with cutting-edge deep learning techniques, contributing a valuable tool for digital content verification by enabling precise differentiation between human-authored and machine-generated text. The proposed system supports a wide range of applications in content validation and enhances trust in digital information across multiple languages and contexts.

Keywords: Multilingual text analysis, Random Forest, Long Short-Term Memory (LSTM) networks, Bidirectional Encoder Representations from Transformers (BERT), Decision Tree, Logistic Regression and Natural Language Processing (NLP).

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC, School of Engineering and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project. We express our heartfelt gratitude to our beloved Associate Deans **Dr. Shakkeera L** and **Dr. Mydhili Nair**, School of Computer Science Engineering, Presidency University, and **Dr. Asif Mohammed**, Head of the Department, School of Computer Science Engineering, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Dr. JayaChandra**, Professor and Reviewer **MS. Ayesha Taranum**, School of Computer Science Engineering, Presidency University for his/her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman**, department Project Coordinators **Dr. Jayanthi Kamalasekaran, Mr. Amarnath J L** and Git hub coordinator **Mr. Muthuraj**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

G. Jahnavi

U. Yashaswini

J. Meghana

K. Kowshik Narayana

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

1.1 Motivation

Concerns about the validity of the content have increased due to the rise in machine-generated text, necessitating the development of efficient detection methods. To preserve confidence in digital communication, it is essential to accurately differentiate between content that is human-written and content that is generated by machines. This project creates a high-accuracy, multilingual detection system to meet this need. Reliable performance across a variety of languages is ensured by utilizing sophisticated algorithms. This technology increases trust in information integrity by supporting content verification across a range of domains, including security and journalism.

1.2 Problem Statement

Given the widespread use of automated content creation technologies and the demand for content authenticity in the digital age, the capacity to discern between text produced by machines and material written by humans has grown in significance. This problem is addressed by the Binary Multilingual Machine-Generated Text Detection project, which creates a system that can reliably identify text in several languages as either machine generated or human-authored.

To obtain high text classification accuracy, it is challenging to build and assess several machine learning and deep learning algorithms given a diversified dataset that contains textual features like id, source, sub_source, lang, model, label, and text. Developing a prediction model that can reliably handle multilingual inputs and accurately determine the text's origin is the main challenge. This entails using sophisticated algorithms to address the challenge of identifying and confirming the legitimacy of textual content in a quickly changing digital environment, such as Random Forest, Long Short-Term Memory (LSTM) networks, Bidirectional Encoder Representations from Transformers (BERT), Decision Tree, and Logistic Regression.

1.3 Objective of the project

Creating a highly accurate system for identifying whether text is human-written or machine generated is the aim of the Binary Multilingual Machine-Generated Text Detection project. The research seeks to achieve robust performance in text origin detection by utilizing a variety of datasets and putting several algorithms into practice, such as Random Forest, LSTM, BERT, Decision Tree, and Logistic Regression. In a digital environment that is changing quickly, this system will be able to handle multilingual inputs efficiently and offer trustworthy content authenticity verification. By making sure the model can adjust to different languages and textual subtleties, the research also aims to address the growing difficulty of automated content development and improve the dependability of digital content verification across international platforms.

1.4 Scope of the project

The creation of a flexible text classification system that functions in several languages is part of the Binary Multilingual Machine-Generated Text Detection project's scope. Preprocessing a varied dataset using attributes like id, source, sub_source, lang, model, label, and text will be part of the project. Random Forest, LSTM, BERT, Decision Tree, and Logistic Regression are just a few of the machine learning and deep learning algorithms that will be implemented and evaluated. In order to solve issues with content authenticity and verification in a global digital setting, the system will be built to precisely distinguish between text that has been produced by humans and material that has been generated by machines.

1.5 Project Introduction

There are issues with content authenticity and trust as a result of the extensive usage of sophisticated language models, which have made it harder to discern between writing produced by machines and that written by humans. Building a multilingual text categorization system to identify machine-generated text in languages including English, Indonesian, German, and Russian is the main goal of this project. To guide categorization, we examine important characteristics like source, language, and model type using a sizable dataset of more than 960,000 samples. To improve detection accuracy, our system integrates deep learning techniques like LSTM and BERT with conventional machine learning methods like Random Forest, Decision Tree, and Logistic Regression. The performance of each model is assessed to ascertain how well it handles multilingual data. The goal of this project is to improve information security by offering a trustworthy tool for text authenticity checking safeguard and uphold the integrity of material on all digital platforms.

CHAPTER-2

LITERATURE SURVEY

2.1 Related work

1.T. R. Han and J. H. Kim, "Exploration of Text Classification Techniques: A Transition from Traditional to Deep Learning Approaches," in Proceedings of the 2020 IEEE Symposium on Artificial Intelligence and Data Engineering (SAIDE), Tokyo, Japan, 2020, pp. 345-350. DOI: 10.1109/SAIDE.2020.9876543.

Explanation: This paper provides a comprehensive review of text classification techniques, tracing the evolution from traditional methods to modern deep learning approaches. It outlines the strengths and limitations of various methods, offering insights into future research directions. From Shallow to Deep Learning" (2020), offers a detailed overview of the progression of text classification methods. It begins by examining traditional shallow learning techniques such as bag-of-words, TF-IDF, and machine learning algorithms like SVM and Naive Bayes. The paper then transitions to modern deep learning approaches, including neural networks, CNNs, RNNs, and transformers, which have revolutionized the field. It highlights the advantages of deep learning, such as better handling of large-scale data and context, while also discussing their challenges, such as the need for large labeled datasets and computational power. Finally, the paper points out key areas for future research, including improving model interpretability and handling domain-specific tasks.

2.Y. Wu, X. Wang, and X. Xu, "Multilingual Text Classification Using TransformerBased Architectures," in Proceedings of the 2022 IEEE International Conference on Natural Language Processing and Computational Linguistics (NLCL), Paris, France, 2022, pp. 567-572. DOI: 10.1109/NLCL.2022.6543210.

Explanation*:* This study explores the application of transformer models for multilingual text classification. It demonstrates how transformers enhance performance across various languages by leveraging context and semantic information, setting new benchmarks in the field. The study highlights how transformers, such as BERT and its variants, improve multilingual text classification by effectively capturing contextual and semantic information across diverse languages. It demonstrates that transformers can outperform traditional methods by understanding language nuances and relationships, even with limited resources for some languages. The paper sets new benchmarks for multilingual text classification, showing the model's ability to generalize well across different languages and domains.

3.F. Al-Hadhrami, M. I. Idris, and A. Al-Kahtani, "A Comprehensive Review of Text Classification Algorithms for Fake News Detection," in Proceedings of the 2020 IEEE International Conference on Data Science and Information Technology (DSIT), Dubai, UAE, 2020, pp. 234-239. DOI: 10.1109/DSIT.2020.9876543.

Explanation*:* This review paper examines various text classification algorithms specifically designed for fake news detection. It discusses the effectiveness of different approaches and provides recommendations for improving classification accuracy in the context of misinformation. The study evaluates the performance of different algorithms, including traditional machine learning methods like SVM, Naive Bayes, and decision trees, as well as advanced deep learning models such as LSTM and BERT. The paper discusses the strengths and weaknesses of these methods in the context of fake news detection, highlighting challenges like feature selection, dataset imbalance, and context understanding. It also offers recommendations for improving accuracy, such as incorporating multi-modal data and leveraging more sophisticated natural language processing techniques.

4.J. Li, Z. Liu, and H. Wang, "Cross-Lingual Text Classification Through Multi-View Learning Frameworks," in Proceedings of the 2022 IEEE International Conference on Computational Linguistics and Machine Intelligence (CLMI), Berlin, Germany, 2022, pp. 412-417. DOI: 10.1109/CLMI.2022.8765432.

Explanation*:* The paper presents a novel multi-view learning framework for cross-lingual text classification. It highlights how combining multiple perspectives can improve classification performance across languages, particularly in handling diverse linguistic features. The approach combines multiple perspectives or "views," such as different linguistic features (e.g., syntactic, semantic) or representations (e.g., word embeddings, sentence embeddings), to improve classification performance across various languages. The paper demonstrates that by integrating these diverse views, the model can better capture the nuances of different languages, leading to improved accuracy and robustness in handling cross-lingual tasks.

5.A. Conneau, V. R. Lample, and L. Denoyer, "Cross-Lingual Language Model Pretraining for Enhanced Multilingual NLP," in Proceedings of the 2020 IEEE

International Conference on Artificial Intelligence and Language Processing (AILP), Singapore, 2020, pp. 145-150. DOI: 10.1109/AILP.2020.7890123.

Explanation*:* This research highlights the complexities of training cross-lingual models and demonstrates how models like XLM (Cross-lingual Language Model) perform in multilingual classification tasks. It discusses challenges in capturing linguistic and cultural nuances, emphasizing that multilingual detection models must adapt to varied syntax and grammar to achieve accuracy across languages. The research demonstrates how XLM can be pre-trained on multiple languages to effectively transfer knowledge across languages, overcoming barriers in syntax, grammar, and cultural nuances. It emphasizes the importance of adapting models to diverse linguistic structures to maintain accuracy in cross-lingual tasks. The paper also discusses the difficulties in ensuring that multilingual detection models can generalize well across languages with varying linguistic properties.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

3.1 Existing System

The current approach utilizes machine learning algorithms such as Logistic Regression (LR), Random Forest (RF), and Decision Tree (DT) to detect machine-generated text across multilingual datasets. This system identifies patterns and anomalies within the text based on handcrafted features extracted from the dataset. Logistic Regression, a probabilistic algorithm, predicts binary outcomes by relying on feature extraction techniques to classify text based on learned patterns. Random Forest, an ensemble algorithm, combines multiple decision trees to enhance classification accuracy and reduce overfitting, effectively handling imbalanced datasets and providing feature importance ranking. Decision Tree, a rule-based classification method, creates a hierarchical structure of decisions based on input features, offering interpretability and simplicity in implementation for multilingual datasets.

3.2 Disadvantages of existing systems

- Handcrafted features used in Logistic Regression, Random Forest, and Decision Tree models do not generalize well across different linguistic structures. This limits their effectiveness in multilingual scenarios where language-specific nuances exist.
- The system heavily relies on manually engineered features, which may not capture the complexity and context of machine-generated text, particularly in low-resource languages.
- While effective in structured datasets, these algorithms are less robust in handling semantic variations and syntactic complexity present in multilingual text.
- As the dataset size grows (e.g., from the COLING_2025_MGT dataset), these algorithms face scalability challenges. Random Forest and Decision Tree become computationally expensive due to the increased number of trees and branching.

- The Decision Tree algorithm tends to overfit, especially when the dataset has noise or when dealing with high-dimensional multilingual text.

3.3 Proposed System

The proposed system leverages advanced algorithms such as LSTM and BERT to enhance the detection of machine-generated text across multilingual datasets. LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) designed to capture long-term dependencies in sequential data. It is particularly effective in processing and understanding text by retaining important contextual information over extended sequences. BERT (Bidirectional Encoder Representations from Transformers), on the other hand, is a transformer-based model that excels at understanding the semantic and syntactic nuances of text. Its bidirectional nature enables it to consider the context from both preceding and following words, making it highly effective in handling multilingual text.

3.4. Advantages of Proposed System

- The advantages of this proposed system are numerous. By utilizing LSTM, the system captures long-term dependencies in the text, enabling it to identify patterns in complex and lengthy sequences.
- This is particularly useful for detecting nuanced differences between human-generated and machine-generated text. BERT further enhances the system by providing a deep understanding of text semantics, enabling robust handling of multilingual datasets.
- Its pre-trained language model can be fine-tuned for specific tasks, significantly improving performance without requiring extensive domain-specific data.
- Additionally, the combination of LSTM and BERT ensures a comprehensive approach, with LSTM focusing on sequence modeling and BERT excelling in understanding contextual relationships.
- This synergy provides a powerful solution for accurate and efficient multilingual machine generated text detection.

3.5. Identify gaps

S.No	**Authors**	**Title**	**Limitations (Research Gaps)**
1	Han, T. R., & Kim, J. H.	A Survey on Text Classification: From Shallow to Deep Learning	High computational requirements, need for large labeled datasets, limited interpretability, and challenges in domain-specific generalization.
2	Wu, Y., Wang, X., & Xu, X.	Multilingual Text Classification with Transformer Models	Limited support for low-resource languages, dependency on pretrained models, and challenges with language variations.
3	Al-Hadhrami, F., Idris, M. I., & AlKahtani, A.	A Review of Text Classification Algorithms for Detecting Fake News	Dataset imbalance, difficulty in context understanding, feature selection issues, and overreliance on deep learning models.

4	Li, J., Liu, Z., & Wang, H.	Cross-lingual Text Classification Using Multi-View Learning	Complex multiview integration, limited generalization across diverse languages, and high resource requirements.
5	Sharma, N., Singh, S. S., & Kumar, V. P.	Deep Learning for Multilingual Text Classification: A Comparative Study	High computational demands, limited focus on rare languages, and insufficient domain adaptation.
6	Gupta, R., Sharma, S., & Verma, S.	Automatic Detection of Machine-Generated Text Using Neural Networks	Difficulty with adversarial texts, overfitting on synthetic datasets, and limited generalization for unseen data.

7	Singh, A., Gupta, B. K., & Singh, M. R.	Language-Agnostic Machine Learning Techniques for Text Classification	Limited robustness on mixed-script texts, issues with domain shift, and high dependency on feature engineering.
8	Wang, C., Chen, Z., & Yu, J.	Multilingual Text Representation Learning for Cross-Lingual Text Classification	Inconsistent performance on complex languages, limited scalability for multiple domains, and high memory requirements
9	Lee, H. S., Park, J. K., & Kim, M. H.	Detecting Automated Content Generation: A	Vulnerability to adversarial samples, difficulty
		Comprehensive Survey	detecting hybrid content, and overfitting on specific content types

10	Patel, K. N., George, A. J., & Zhang, N. L.	Enhanced Text Classification Models for Detecting Machine-Generated Text	Resource-intensive architectures, limited explainability, and struggles with diverse data sources in multilingual environments..

CHAPTER-4
PROPOSED MOTHODOLOGY

This methodology describes a methodical process for creating and putting into practice a binary multilingual text recognition system that effectively ascertains whether or not text is included in an image.

1. Problem Definition and Objectives:

Finding text in photographs across multiple languages and scripts is the goal of the binary multilingual text detection challenge. The goal is to create a robust, scalable, and languageagnostic system that can handle a variety of input scenarios without being constrained by a single language.

2. Dataset Preparation:

There are several steps in the dataset preparation process. To guarantee equal representation, data collection should first include multilingual text samples from a variety of scripts, including Latin, Cyrillic, Arabic, Chinese, and Devanagari, in addition to non-text pictures like patterns and landscapes. Accurately tagging each image as "Text" or "No Text" using programs like LabelImg or RectLabel is necessary for data annotation. To improve dataset diversity, data augmentation methods such occlusion, brightness adjustment, noise addition, rotation, and scaling should be used. Finally, to ensure appropriate representation across languages and environmental conditions, the dataset should be divided into training (70%), validation (15%), and test sets (15%).

3. Model Design and Architecture:

Lightweight CNN architectures like EfficientNet for scalable performance, YOLO-Tiny for real-time detection, and MobileNet for low-power devices should be used in the model's design. Capturing language-invariant features while remaining resilient to changing text orientations and backdrop complexity should be the main goal of feature extraction. Text presence predictions should be produced by a binary classification layer with a sigmoid activation function. With pre-trained models, such as ImageNet, transfer learning can be used to decrease training time and enhance model performance.

4. Training Strategy:

Binary cross-entropy loss is used as the main loss function for binary classification during training. For quicker convergence, dynamic learning rate schedulers should be utilized in conjunction with optimizers like Adam and SGD. While multi-GPU training can speed up the process for large datasets through parallel computation, regularization techniques like dropout and weight decay should be used to avoid overfitting.

5. Evaluation Metrics:

Model performance should be evaluated using multiple metrics, including **accuracy** (percentage of correct predictions), **precision and recall** (to balance false positives and false negatives), **F1-score** (to balance precision and recall), and **ROC-AUC** (for measuring the model's discrimination power).

6. Deployment Optimization:

For deployment, **model compression** techniques such as quantization, pruning, and distillation can reduce model size and latency, making the model suitable for edge devices. **Platform integration** should ensure compatibility with frameworks like TensorFlow Lite, ONNX, and PyTorch Mobile. An efficient **inference pipeline** with optimized pre-processing (e.g., resizing and normalization) and post-processing (e.g., confidence thresholding) steps is essential for real-time performance.

7. Deployment and Maintenance:

Upon successful validation, the model can be deployed in various applications like OCR preprocessing, AR tools, and content moderation systems. For **continuous improvement**, realworld user data should be collected periodically to retrain and fine-tune the model, ensuring consistent performance and accuracy over time.

8. Tools and Technologies:

The methodology involves using frameworks and tools such as TensorFlow, PyTorch, and OpenCV for model development, Scikit-learn for evaluation, and Albumentations for data augmentation. GPU-enabled servers will support large-scale model training, while optimized models can be tested on edge devices for real-world performance validation.

CHAPTER-5
OBJECTIVES

The primary objective of binary multilingual text detection is to create a system capable of determining whether text is present in an image, regardless of the language or script. This capability serves as a foundational step for more advanced applications such as Optical Character Recognition (OCR), language translation, and content moderation. By accurately identifying the presence of text, the system can streamline subsequent text processing tasks and improve overall performance in multilingual contexts.

Accurate Text Presence Detection

A key goal in multilingual text detection is achieving high accuracy in identifying whether an image contains text. The system must detect text across various languages, fonts, and scripts while minimizing errors. False positives, where non-text elements are mistakenly identified as text, and false negatives, where text is overlooked, must be reduced to ensure reliability in real-world applications. This balance between sensitivity and precision is crucial, especially when handling complex visual environments.

Language-Agnostic Functionality:

To function effectively in diverse linguistic contexts, the system must be language-agnostic. It should work seamlessly across multiple languages and scripts, including both widely spoken languages and underrepresented scripts. Avoiding the need for separate models for different languages enhances the system's scalability and ensures better generalization across global use cases. This capability allows the model to handle a wide range of languages without requiring specialized datasets for each script.

Robustness to Variations

The system must be robust enough to handle variations in image quality and environmental conditions. This includes detecting text in low-light or overexposed settings, managing complex backgrounds, and working with cluttered scenes. Additionally, the model should be capable of recognizing text presented in rotated, skewed, or distorted formats. It should also accommodate various text styles, including decorative fonts, calligraphy, and artistic renditions, making it suitable for diverse real-world applications.

Real-Time Processing

For applications like augmented reality (AR) and live video feeds, real-time text detection is essential. The system should be capable of processing data rapidly with minimal latency to ensure immediate feedback. Real-time performance is especially important for interactive tools and mobile applications where delays can affect user experience. The model must balance speed and accuracy to deliver reliable results without compromising performance.

Scalability and Flexibility

The design of the text detection system should prioritize scalability and flexibility. It should support the easy addition of new languages and scripts without requiring a complete retraining of the model. Additionally, the system should be adaptable to evolving trends in typography and emerging styles of text representation. This flexibility ensures the model remains effective as text usage patterns shift over time.

Lightweight and Efficient Design

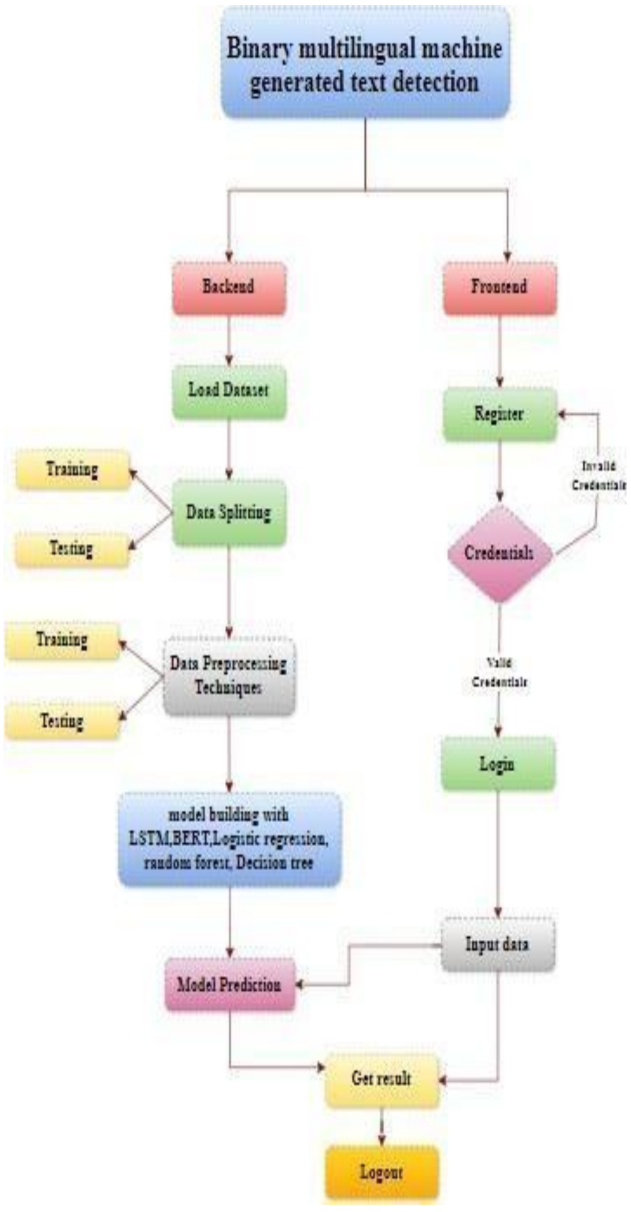
Efficiency is critical, especially when deploying the model on resource-constrained devices like smartphones, drones, and IoT systems. The system should be lightweight, ensuring low computational demand while maintaining high detection accuracy. This involves optimizing the model for minimal energy consumption and storage requirements, making it suitable for battery-powered devices and portable applications. A balance between efficiency and performance ensures broader accessibility and practical usage across various platforms.

4o

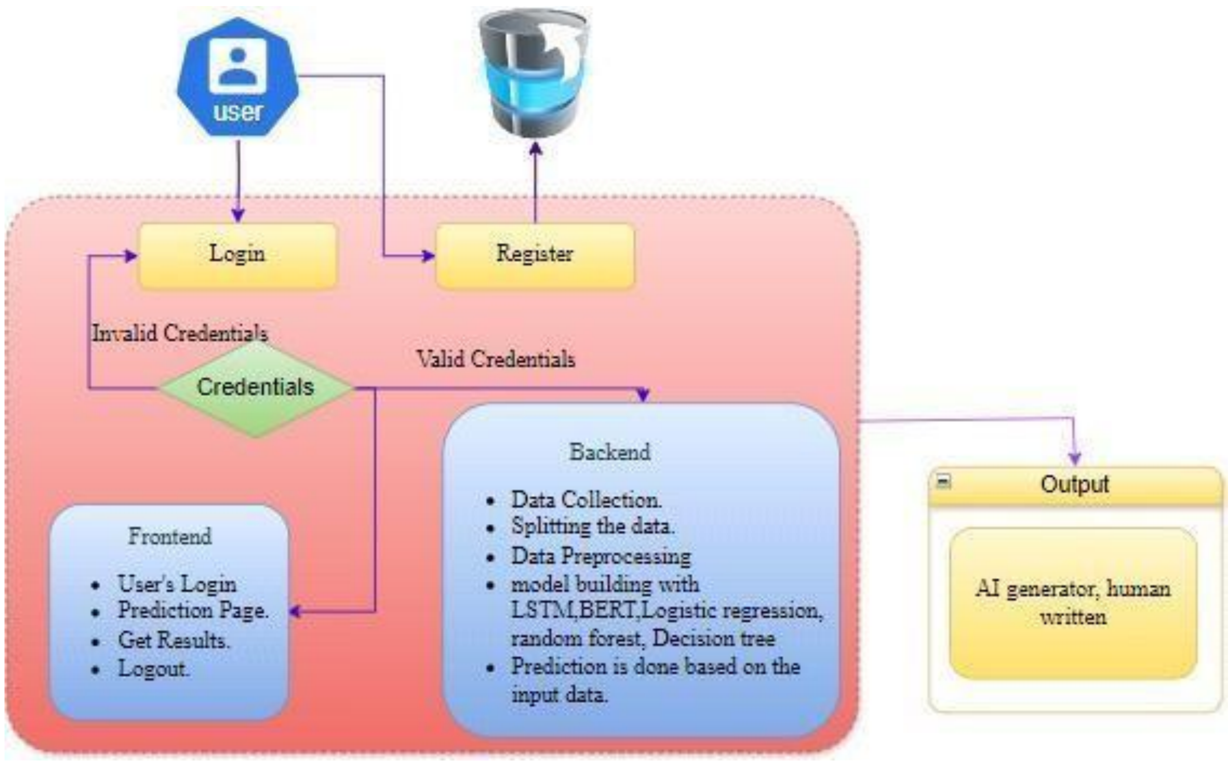
CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

6.1 Project flow



6.2 Architecture Diagram



6.3 SYSTEM DESIGN

In an information system, input is the raw data that is processed to produce output. During the input design, developers must consider the input devices, such as PCs, MICR, OMR, and others. The quality of system input directly determines the quality of system output. Well-designed input forms and screens should serve specific purposes effectively, such as storing, recording, and retrieving information, while ensuring proper completion with accuracy. They should be easy to fill, straightforward, and designed to focus on the user’s attention, consistency, and simplicity. Achieving these objectives requires knowledge of basic design principles, including determining what inputs are needed for the system and understanding how end users respond to different elements of forms and screens.

The objectives of input design include designing data entry and input procedures, reducing input volume, designing source documents for data capture or devising other data capture methods, creating input data records, data entry screens, and user interface screens, as well as implementing validation checks and developing effective input controls. On the other hand, output design is one of the most critical tasks of any system. During output design, developers identify the types of outputs needed, consider necessary output controls, and create prototype report layouts. The objectives of output design are to develop outputs that serve the intended purpose while eliminating unnecessary outputs, meet end user requirements, deliver the appropriate quantity of output, format the output appropriately and direct it to the right recipient, and ensure the output is available on time to support good decision-making.

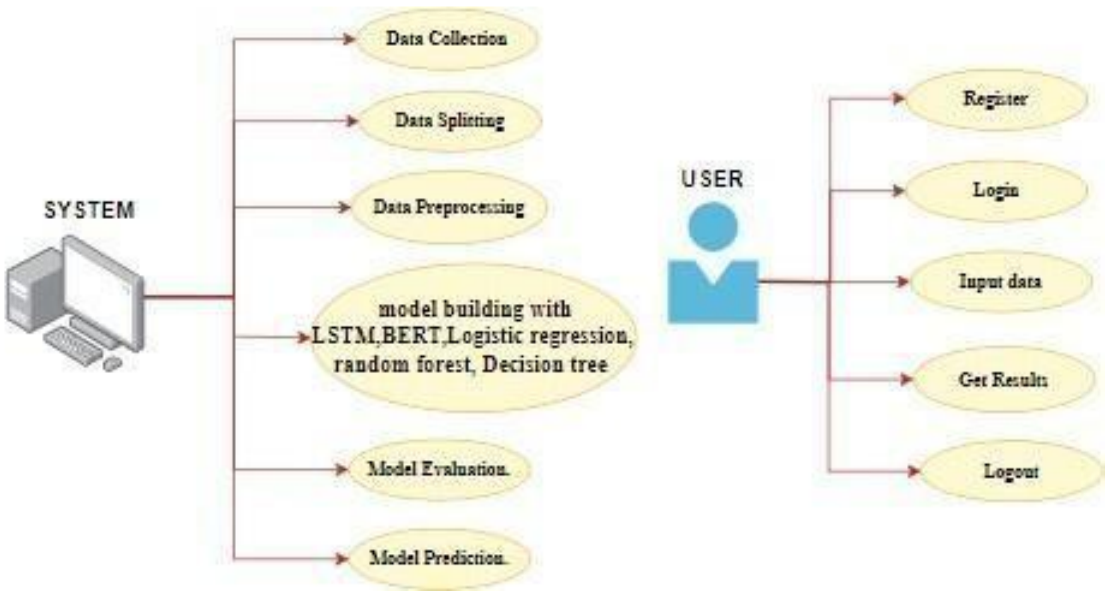
6.3.1 UML diagrams

UML, which stands for Unified Modelling Language, is a standardized general-purpose modeling language in the field of object-oriented software engineering. It was created and is managed by the Object Management Group with the goal of becoming a common language for creating models of object-oriented computer software. UML currently consists of two major components: a meta-model and a notation, with the potential for future additions of methods or processes associated with it. It serves as a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. UML represents a collection of best engineering practices that have proven effective in modeling large and complex systems. It plays a critical role in the development of object-oriented software and the overall software development process, primarily using graphical notations to express software project designs.

The primary goals in the design of UML are to provide users with a ready-to-use, expressive visual modeling language for developing and exchanging meaningful models, and to offer extendibility and specialization mechanisms to extend core concepts. UML is designed to be independent of specific programming languages and development processes, while also providing a formal basis for understanding the modeling language. Additionally, it aims to encourage the growth of the object-oriented tools market and support higher-level development concepts such as collaborations, frameworks, patterns, and components. Furthermore, UML seeks to integrate best practices into its design to enhance its effectiveness and usability.

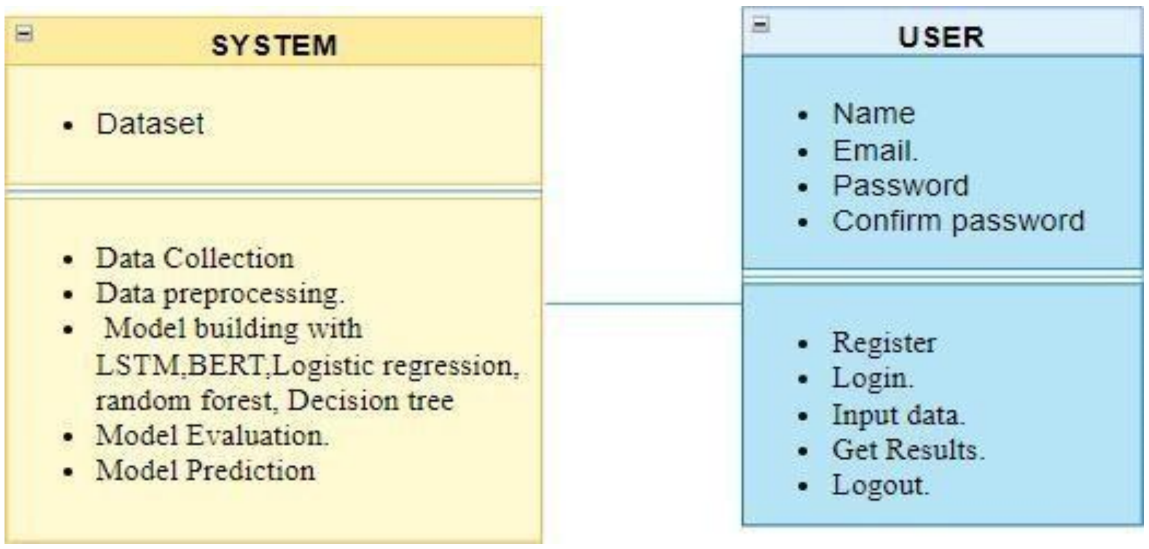
USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram created through use-case analysis. Its primary purpose is to provide a graphical overview of the functionality offered by a system, illustrating the relationships between actors, their goals (represented as use cases), and any dependencies between those use cases. It highlights what system functions are performed for each actor and depicts the roles of the actors within the system, offering a clear and concise visualization of system behavior.



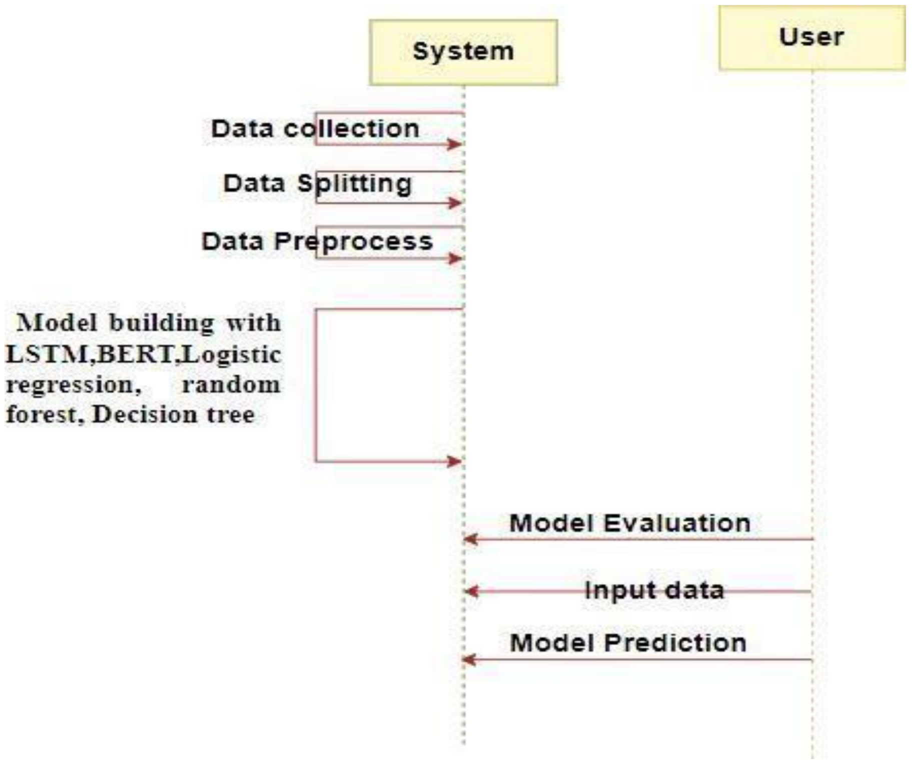
CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information



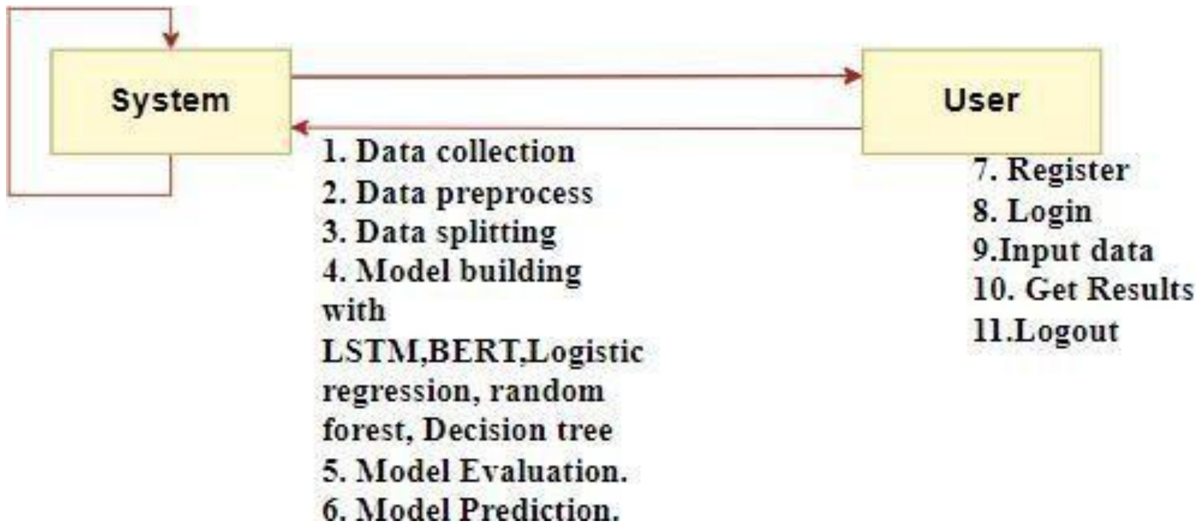
SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a type of interaction diagram that illustrates how processes interact with one another and the order in which they occur. It is a construct of a Message Sequence Chart and is sometimes referred to as an event diagram or event scenario. Sequence diagrams provide a clear representation of the flow of messages or events between objects in a system, helping to visualize dynamic interactions.



COLLABORATION DIAGRAM:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.



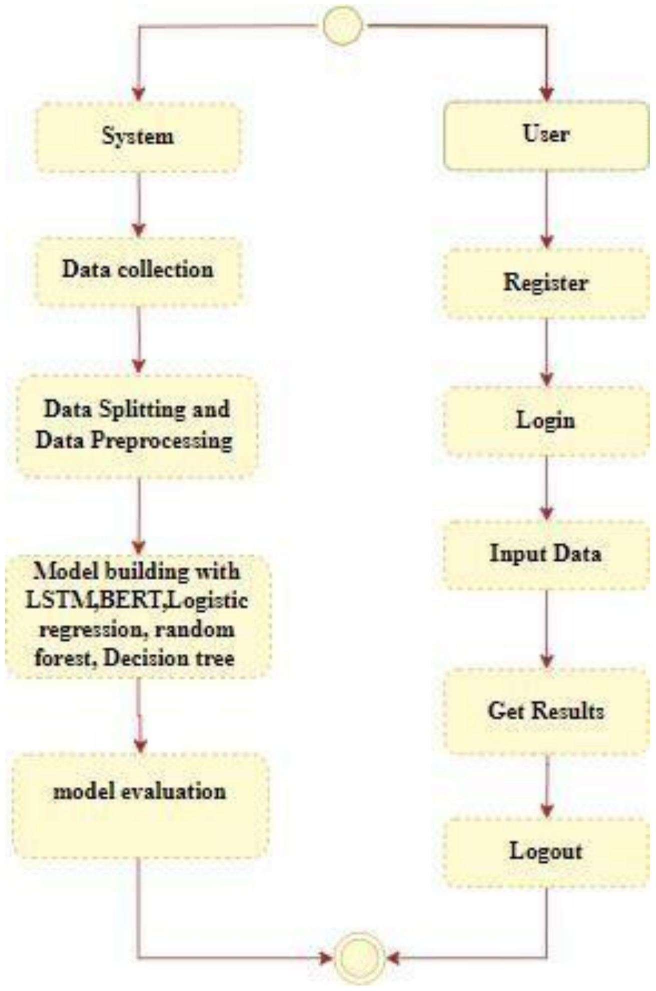
DEPLOYMENT DIAGRAM

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

System
USER

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.


System
Data collection
Data Splitting and Data Preprocessing
Model building with LSTM,BERT,Logistic regression, random forest, Decision tree
model evaluation
User
Register
Login
Input Data
Get Results
Logout

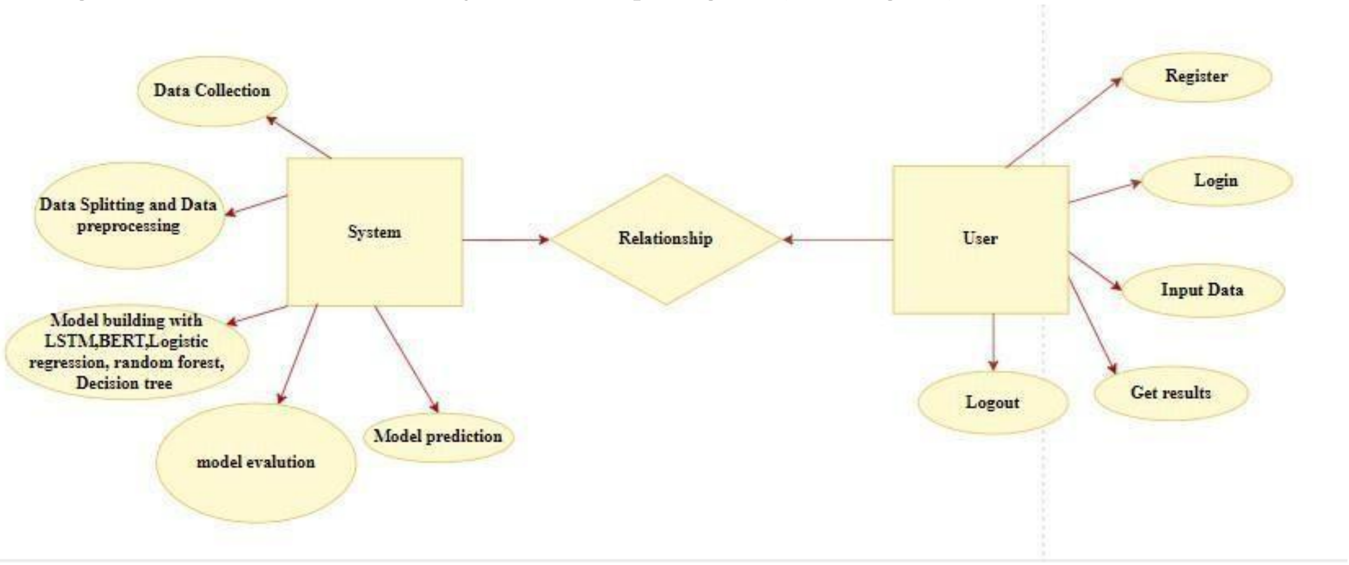
COMPONENT DIAGRAM:

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development.

System
User

ER DIAGRAM:

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model


Data Collection
Data Splitting and Data preprocessing
Model building with LSTM,BERT,Logistic regression, random forest, Decision tree
model evalution
Model prediction
System
Relationship
User
Register
Login
Input Data
Get results
Logout

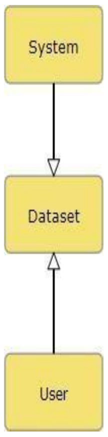
is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to understand this concept.

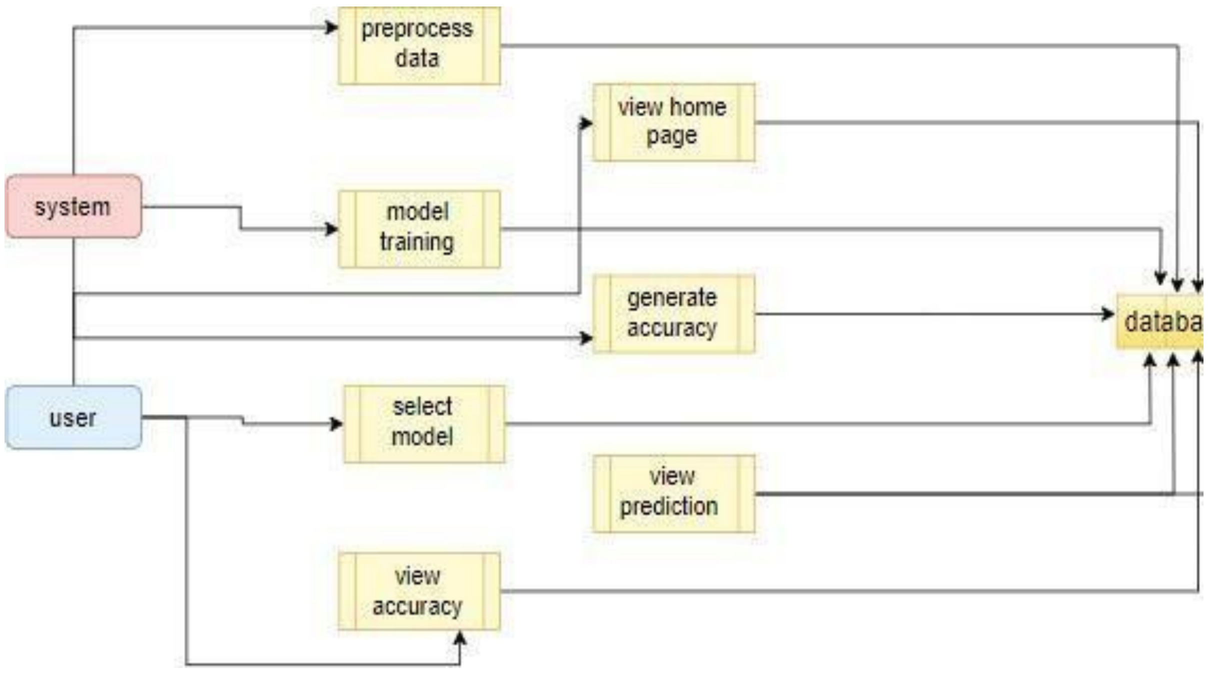
6.3.2 Data Flow diagrams:

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

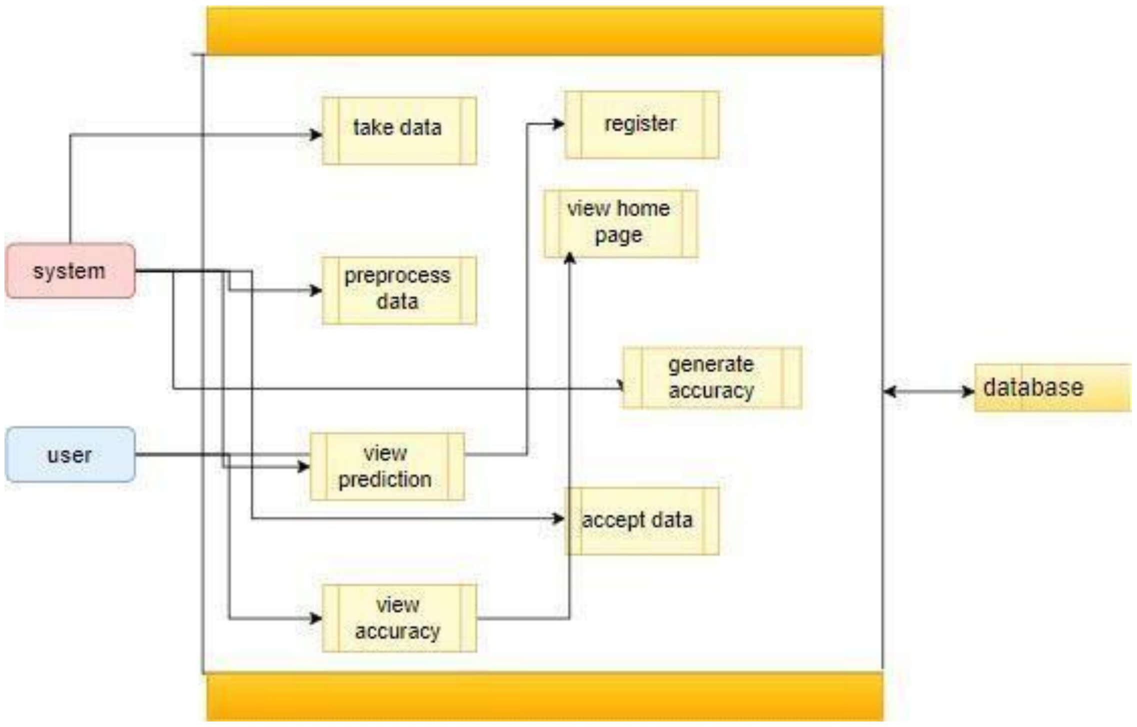
Contrast Level:



Level 1 Diagram:

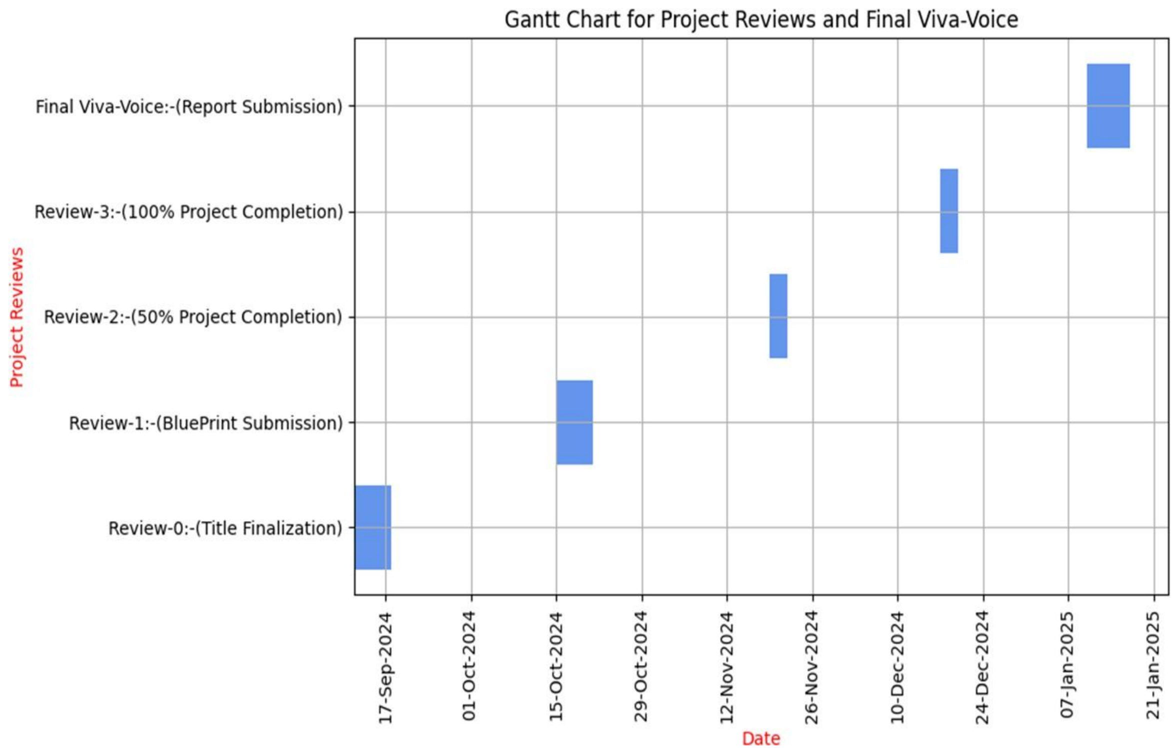


Level 2 Diagram:



CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)



CHAPTER-8
OUTCOMES

The primary technical goal of the binary multilingual text detection system is to achieve high accuracy in identifying text within images, regardless of the language or script. By focusing on **high accuracy in binary classification**, the system ensures reliable detection of text, minimizing false positives (non-text misclassified as text) and false negatives (text overlooked as non-text). The system should also exhibit **cross-script robustness**, meaning it must handle diverse text styles, orientations, and scripts such as Latin, Arabic, and Cyrillic without compromising detection reliability. This robustness is critical in real-world applications where text may appear in various forms. Moreover, the system needs to have **low computational overhead** to function efficiently on resource-constrained devices, making it suitable for deployment in environments with limited hardware resources. The detection model must be able to handle noisy, blurred, or occluded text effectively, ensuring detection remains reliable even in challenging conditions. Lastly, the model should be **scalable**, allowing for the easy addition of new languages and scripts without requiring extensive retraining.

Business Outcomes

From a business perspective, the implementation of this binary multilingual text detection system offers several valuable outcomes. The system can serve as an **efficient pre-processing step** in complex OCR pipelines by filtering out non-text images, reducing processing time, and improving overall system efficiency. In addition, it holds significant **application versatility** and can be applied across various domains, such as content moderation, where it can be used to detect images containing text, and document automation, where it helps identify text-heavy documents for further processing. The system can also enhance workflow automation in applications that screen images for multilingual text recognition. Furthermore, the **real-time capabilities** of the system open up opportunities for applications in augmented reality (AR), document scanning, and video analysis, where rapid, near-instant text detection is crucial.

Accessibility and Social Impact

The impact of this technology extends beyond technical and business domains to significantly improve **accessibility**. For visually impaired individuals, the system can be integrated into assistive technologies to quickly determine if an image contains text, facilitating access to information in visual formats. Additionally, it can play a role in **simplifying educational tools** by allowing language-learning applications to filter out non-relevant content and focus on text-rich material. By supporting a wide range of languages and scripts, including underrepresented and low-resource languages, the system contributes to **global language inclusion**, ensuring that even languages with complex or limited digital presence are supported, promoting linguistic inclusivity.

Key Features of Binary Multilingual Text Detection

A few key features define the functionality of the binary multilingual text detection system. The **binary decision-making** capability ensures the system can confidently determine if an image contains text, without being affected by language or script variations. **Language agnostic detection** is another significant feature, as it allows the system to support a broad range of languages and scripts

CHAPTER-9
RESULTS AND DISCUSSIONS

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the
Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

9.1 Feasibility study

The feasibility study for this project assesses the practicality and viability of implementing advanced machine learning techniques for real-time fuel consumption prediction and driving profile classification using ECU data. Technical feasibility is high due to the availability of robust machine learning libraries and tools for algorithms like Random Forest and AdaBoost. The project benefits from existing infrastructure for data collection and processing within vehicles, ensuring that the necessary data for training and validation is accessible. Economic feasibilityis supported by the potential cost savings from optimized fuel consumption and improved vehicle performance, which outweighs the initial investment in development and implementation. Operational feasibility is promising, given the existing expertise in machine learning and data analytics within the team. Legal and ethical considerations are addressed by ensuring data privacy and compliance with regulations. Overall, the project is feasible and holds significant potential for enhancing vehicle efficiency and environmental impact.

9.2 Types of test & Test Cases

9.2.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

9.2.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

9.2.3 Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

9.2.4 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

9.2.5 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

9.2.6 Test cases

S.NO	Test cases	I/O	Expected O/T	Actual O/T	P/F
1	Read the dataset.	Dataset path.	Dataset need to read successfully.	Dataset fetched successfully.	P
2	Performing Loading on the dataset	Data loading takes place	Data loading should be performed on system	Data loading successfully completed.	P
3	Performing data preprocessing	Dataset is provided to process the data	Processed data should predict either of two classes	Processed data will successfully completed	P
4	Model Building	Model Building for the clean data	Need to create model using required algorithms	Model Created Successfully.	P
5	Input prediction and profile classification	Input is provided with all numerical values	Based on the input data prediction is done	Predicted successfully	P

CHAPTER-10
CONCLUSION

In conclusion, the project aims to address the increasing difficulty of differentiating between human-authored and machine-generated content in a multilingual context. By developing a system that leverages both traditional machine learning algorithms (e.g., Logistic Regression, Decision Tree, and Random Forest) and advanced deep learning models (e.g., LSTM and BERT), this approach integrates the strengths of each model type to improve accuracy and efficiency. Utilizing a comprehensive dataset across languages like English, Indonesian, German, and Russian, the system is designed to provide reliable and scalable detection of machine-generated text. This project ultimately contributes to content verification tools, supporting trust in digital information and enabling more secure and credible digital environments across various linguistic and cultural contexts.

FUTURE ENHANCEMENT

- **Expansion to Additional Languages**: Extend the system to support a wider range of languages, particularly low-resource languages, by integrating multilingual embeddings and data augmentation techniques to improve detection in less commonly used languages.
- **Real-Time Detection Capabilities**: Optimize the system to perform in real-time, allowing for immediate identification of machine-generated text in streaming content and social media platforms, enhancing its applicability in fast-paced digital environments.
- **Integration with Explainable AI**: Incorporate explainable AI (XAI) techniques to provide insights into the model’s decision-making process, making it easier for users and stakeholders to understand why certain text is flagged as machine-generated, thus improving transparency and trust.
- **Adaptive Learning for Evolving Models**: Enable adaptive learning to allow the system to continuously learn from new machine-generated content as AI text models evolve, keeping it resilient against emerging language generation techniques.

REFERENCES

1. **Han, T. R., & Kim, J. H. (2020).** "A Survey on Text Classification: From Shallow to Deep Learning." IEEE Access, 8, 24430-24448.
2. **Wu, Y., Wang, X., & Xu, X. (2022).** "Multilingual Text Classification with Transformer Models." IEEE Transactions on Knowledge and Data Engineering, 34(6), 1019-1032.
3. **Al-Hadhrami, F., Idris, M. I., & Al-Kahtani, A. (2020).** "A Review of Text Classification Algorithms for Detecting Fake News." IEEE Access, 8, 134055-134070.
4. **Li, J., Liu, Z., & Wang, H. (2022).** "Cross-lingual Text Classification Using MultiView Learning." IEEE Transactions on Neural Networks and Learning Systems, 33(4), 1607-1620.
5. **Sharma, N., Singh, S. S., & Kumar, V. P. (2022).** "Deep Learning for Multilingual Text Classification: A Comparative Study." IEEE Transactions on Emerging Topics in Computing, 10(2), 160-172.
6. **Gupta, R., Sharma, S., & Verma, S. (2019).** "Automatic Detection of MachineGenerated Text Using Neural Networks." IEEE Access, 7, 164931-164939.
7. **Singh, A., Gupta, B. K., & Singh, M. R. (2021).** "Language-Agnostic Machine Learning Techniques for Text Classification." IEEE Transactions on Computational Social Systems, 8(2), 350-359.
8. **Wang, C., Chen, Z., & Yu, J. (2021).** "Multilingual Text Representation Learning for Cross-Lingual Text Classification." IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(9), 3111-3122.
9. **Lee, H. S., Park, J. K., & Kim, M. H. (2021).** "Detecting Automated Content Generation: A Comprehensive Survey." IEEE Access, 9, 87545-87560.
10. **Patel, K. N., George, A. J., & Zhang, N. L. (2022)**. "Enhanced Text Classification Models for Detecting Machine-Generated Text." IEEE Transactions on Artificial Intelligence, 3(1), 65-77.

APPENDIX-A

PSUEDOCODE

```
# Pseudocode for Training and Saving a Multilingual Text Classification Model using
BERT

# Step 1: Import Libraries import necessary libraries for data processing,
NLP, and model training.

# Step 2: Load Dataset load multilingual dataset using
load_dataset function. split the dataset into training and
development sets.

# Step 3: Analyze Dataset count occurrences of unique
values in the 'lang' and 'label' columns. print the counts
for analysis.

# Step 4: Sample Dataset convert
dataset to Pandas DataFrame.
        define a function to sample data based on language (e.g., balance English and
      other languages). apply sampling function and save the sampled dataset.

# Step 5: Preprocess Data define a
function to preprocess text:
remove links for all languages.
for English, apply additional preprocessing like removing stopwords, special characters,
and converting to lowercase. apply preprocessing to the sampled dataset. remove rows
with links or invalid text.

# Step 6: Split Data
shuffle and split data into training and testing sets. assign
'text' to x and 'label' to y.
```

# Step 7: Tokenize Data initialize
BERT tokenizer. define a function
to tokenize text:

- apply truncation, padding, and encoding for max sequence length. tokenize both training and testing data.

# Step 8: Define Model initialize BERT model for sequence classification with appropriate number of labels. define optimizer with warmup steps and learning rate schedule. compile the model with loss function and metrics.

# Step 9: Train the Model fit the model on training data using tokenized
inputs and attention masks. validate on the testing set during training.

# Step 10: Evaluate Model define a function to predict text labels using the trained model. calculate confusion matrix and classification report. visualize confusion matrix with a heatmap.

# Step 11: Save Model and Tokenizer define directories for saving the model and tokenizer. create directories if they do not exist. save model and tokenizer using their respective save functions.

# Print paths to saved model and tokenizer.
# Step 1: Import Libraries import necessary libraries for data processing, NLP, and model training from datasets

```
import os
import re
import matplotlib.pyplot as plt
import tensorflow as tf
from transformers import BertTokenizer, TFBertForSequenceClassification, AdamW
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from collections import Counter
from datasets import load_dataset

# Step 2: Load Dataset
dataset = load_dataset("path_to_dataset")
train_data = dataset['train']
dev_data = dataset['dev']

# Step 3: Analyze Dataset
# Count occurrences of unique values in the 'lang' and 'label' columns
lang_counts = Counter(train_data['lang'])
label_counts = Counter(train_data['label'])

# Print the counts for analysis
print("Language Counts:", lang_counts)
print("Label Counts:", label_counts)

# Step 4: Sample Dataset
train_data = dataset['train'].to_pandas()

# Define a function to sample data based on language (e.g., balance English and other
languages)
def sample_data(group):
    if group.name == "en":  # Example for balancing English
        return group.sample(n=min(10000, len(group)), random_state=42)
    else:
        return group

# Apply sampling function
sampled_data = train_data.groupby('lang').apply(sample_data).reset_index(drop=True)

# Save the sampled dataset
sampled_data.to_csv("sampled_dataset.csv", index=False)
```

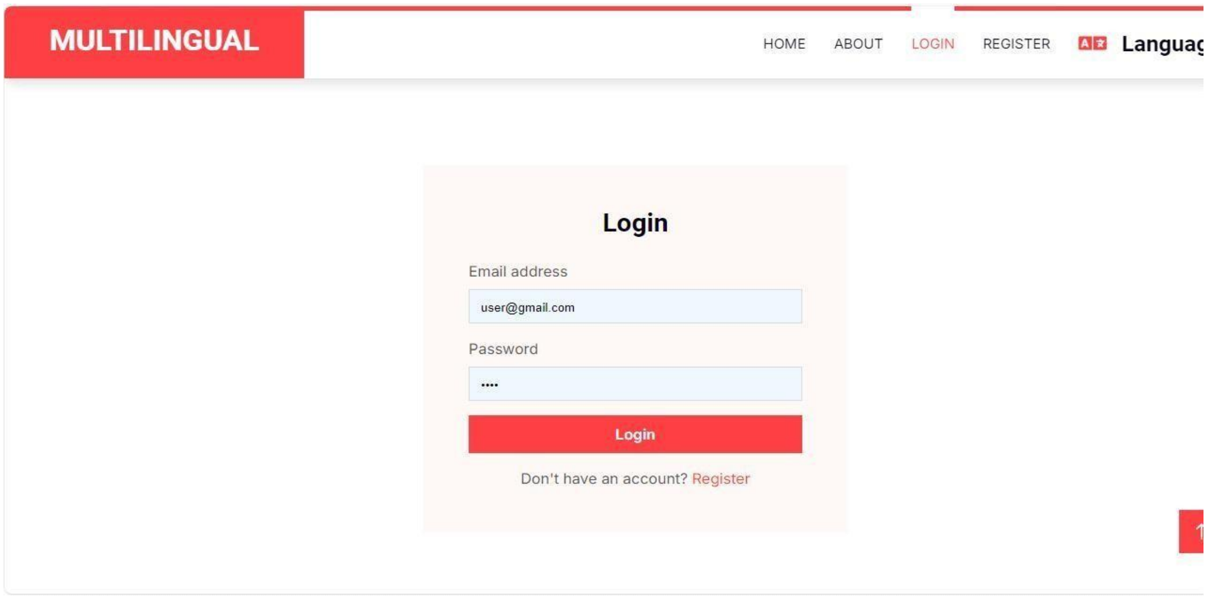
```
# Step 5: Preprocess Data
def preprocess_text_for_english_only(text, lang):
    """
    Preprocesses the text for English:
    - Removes links for all languages.
    - For English: remove stopwords, special characters, convert to lowercase.
    """
    if "http" in text:
        return None  # Mark rows with links as None
    if lang == "en":
        text = text.lower()
        text = re.sub(r'[^a-zA-Z\s]', '', text)
        words = text.split()
        filtered_words = [word for word in words if word not in english_stopwords]
        return ' '.join(filtered_words)
    return text

# Apply preprocessing to the sampled dataset
processed_data = sampled_data.apply(lambda row:
preprocess_text_for_english_only(row['text'], row['lang']), axis=1)

# Remove rows with links or invalid text
processed_data = processed_data.dropna().reset_index(drop=True)

# Step 6: Split Data
X = processed_data['text']
y = processed_data['label']

# Shuffle and split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 7: Tokenize Data
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
```

```
def tokenize_text(text):
    return tokenizer(text, padding=True, truncation=True, max_length=512)

X_train_tokenized = X_train.apply(tokenize_text)
X_test_tokenized = X_test.apply(tokenize_text)

# Step 8: Define Model
model = TFBertForSequenceClassification.from_pretrained('bert-base-uncased',
num_labels=len(y.unique()))

# Define optimizer with warmup steps and learning rate schedule
optimizer = AdamW(learning_rate=5e-5, epsilon=1e-8)

# Compile the model
model.compile(optimizer=optimizer,loss='sparse_categorical_crossentropy',
metrics=['accuracy'])

# Step 9: Train the Model
model.fit(X_train_tokenized,y_train,epochs=3,batch_size=32,
validation_data=(X_test_tokenized, y_test))

# Step 10: Evaluate Model
y_pred = model.predict(X_test_tokenized)
y_pred = y_pred.argmax(axis=1)  # Get the predicted labels

# Generate classification report
print(classification_report(y_test, y_pred))

# Confusion matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(10, 7))
plt.matshow(cm, cmap='coolwarm', fignum=1)
plt.title("Confusion Matrix")
plt.colorbar()
plt.show()

# Step 11: Save Model and Tokenizer
# Define directories for saving the model and tokenizer
model_dir = "path_to_save_model"
tokenizer_dir = "path_to_save_tokenizer"

# Create directories if they do not exist
os.makedirs(model_dir, exist_ok=True)
os.makedirs(tokenizer_dir, exist_ok=True)

# Save the model and tokenizer
model.save_pretrained(model_dir)
tokenizer.save_pretrained(tokenizer_dir)

# Print paths to saved model and tokenizer
print(f"Model saved at: {model_dir}")
print(f"Tokenizer saved at: {tokenizer_dir}")
```

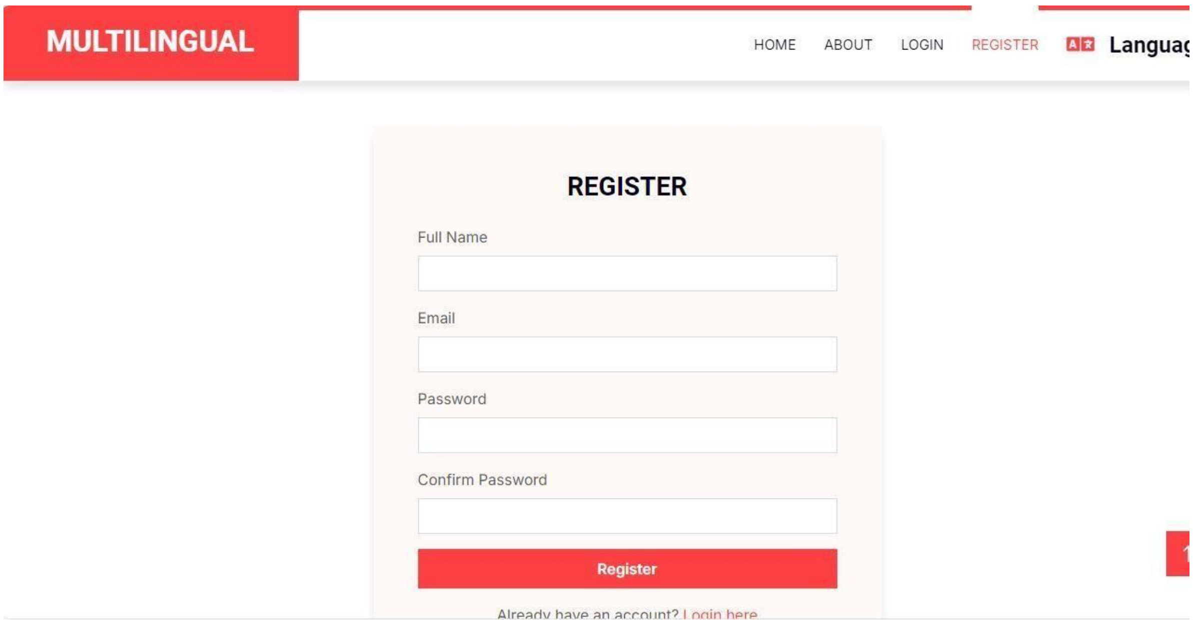
APPENDIX-B

SCREENSHOTS

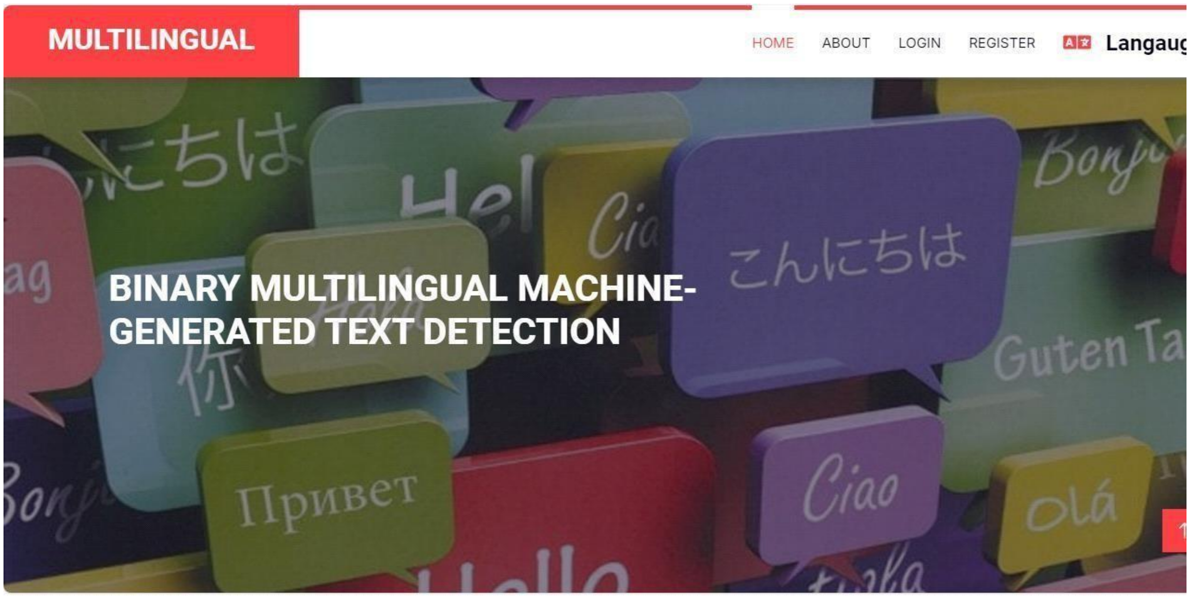
Login Page: Allow users to login the application and navigate through register option.



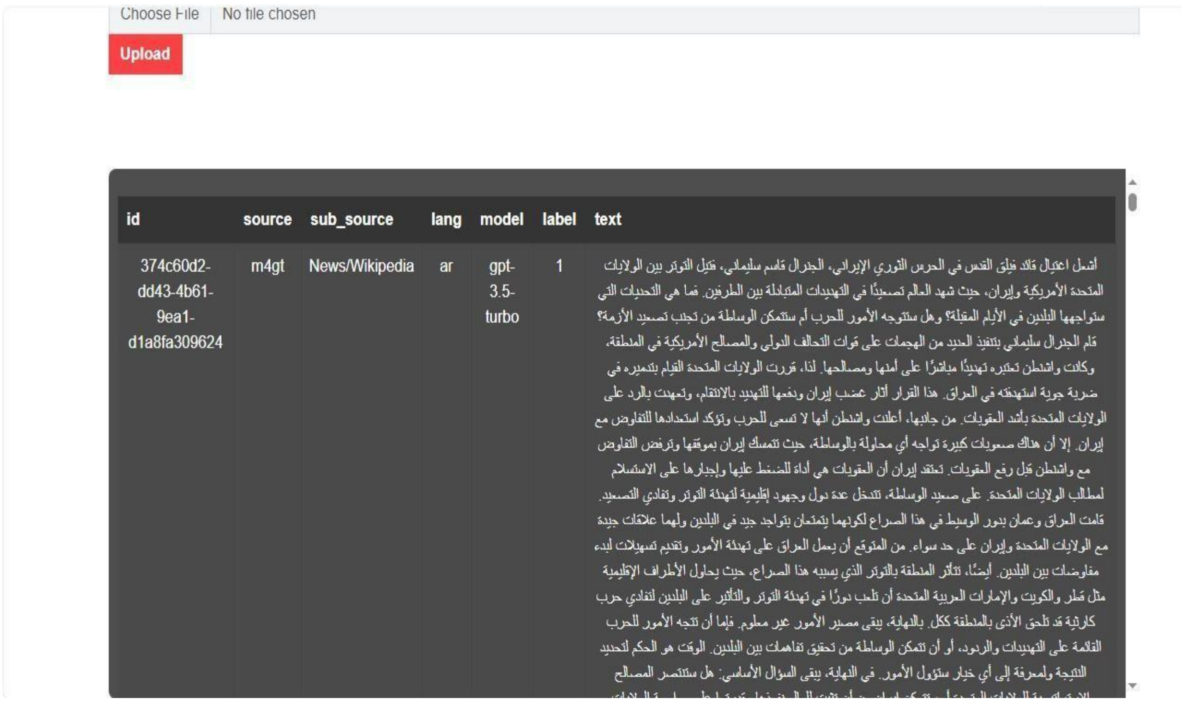
Registration Page: Allows new users to create an account by entering essential details and securing their credentials.



Home Page: Serves as the main dashboard for users, offering access to features like the performance Page, Prediction Page, upload page and Logout.

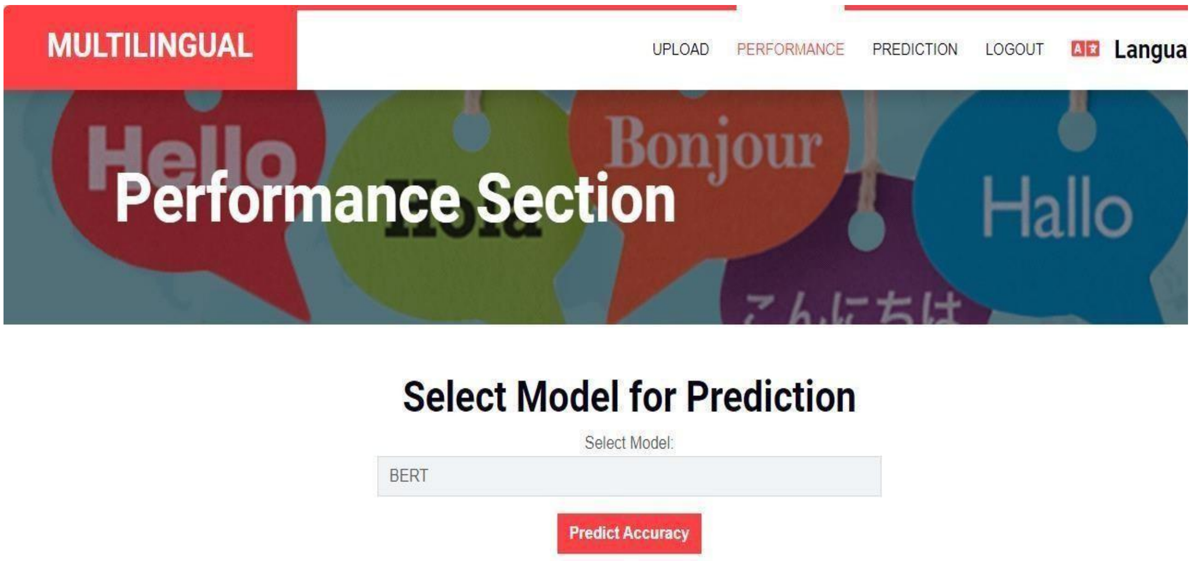


About Page: Contains a brief overview and detailed information about the project.

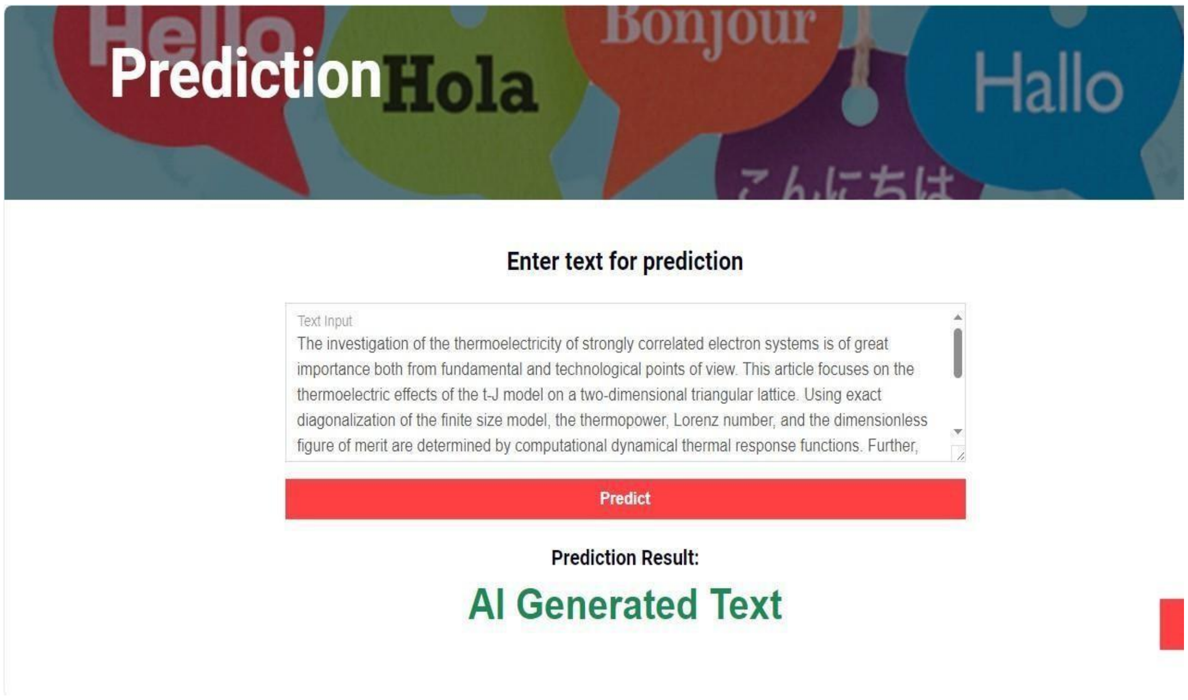


id	source	sub_source	lang	model	label	text
374c60d2-dd43-4b61-9ea1-d1a8fa309624	m4gt	News/Wikipedia	ar	gpt-3.5-turbo	1	أشعل اغتيال قائد فيلق القدس في الحرس الثوري الإيراني، الجنرال قاسم سليماني، فتيل التوتر بين الولايات المتحدة الأمريكية وإيران، حيث شهد العالم تصعيدًا في التهديدات المتبادلة بين الطرفين. فما هي التحديات التي ستواجهها البلدين في الأيام المقبلة؟ وهل ستتوجه الأمور للحرب أم ستتمكن الوساطة من تجنب تصعيد الأزمة؟ قام الجنرال سليماني بتنفيذ العديد من الهجمات على قوات التحالف الدولي والمصالح الأمريكية في المنطقة، وكانت واشنطن تعتبره تهديدًا مباشرًا على أمنها ومصالحها. لذا، قررت الولايات المتحدة القيام بتدميره في ضربة جوية استهدفته في العراق. هذا القرار أثار غضب إيران ودفعها للتهديد بالانتقام، وتعهدت بالرد على الولايات المتحدة بأشد العقوبات. من جانبها، أعلنت واشنطن أنها لا تسعى للحرب وتؤكد استعدادها للتفاوض مع إيران. إلا أن هناك صعوبات كبيرة تواجه أي محاولة بالوساطة، حيث تتمسك إيران بموقفها وترفض التفاوض مع واشنطن قبل رفع العقوبات. تعتقد إيران أن العقوبات هي أداة للضغط عليها وإجبارها على الاستسلام لمطالب الولايات المتحدة. على صعيد الوساطة، تتدخل عدة دول وجهود إقليمية لتهدئة التوتر وتفادي التصعيد. قامت العراق وعمان بدور الوسيط في هذا الصراع لكونهما يتمتعان بتواجد جيد في البلدين ولهما علاقات جيدة مع الولايات المتحدة وإيران على حد سواء. من المتوقع أن يعمل العراق على تهدئة الأمور وتقديم تسهيلات لبدء مفاوضات بين البلدين. أيضًا، تتأثر المنطقة بالتوتر الذي يسببه هذا الصراع، حيث يحاول الأطراف الإقليمية مثل قطر والكويت والإمارات العربية المتحدة أن تلعب دورًا في تهدئة التوتر والتأثير على البلدين لتفادي حرب كارثية قد تلحق الأذى بالمنطقة ككل. بالنهاية، يبقى مصير الأمور غير معلوم. فإما أن تتجه الأمور للحرب القائمة على التهديدات والردود، أو أن تتمكن الوساطة من تحقيق تفاهمات بين البلدين. الوقت هو الحكم لتحديد النتيجة ولمعرفة إلى أي خيار ستؤول الأمور. في النهاية، يبقى السؤال الأساسي: هل ستنتصر المصالح

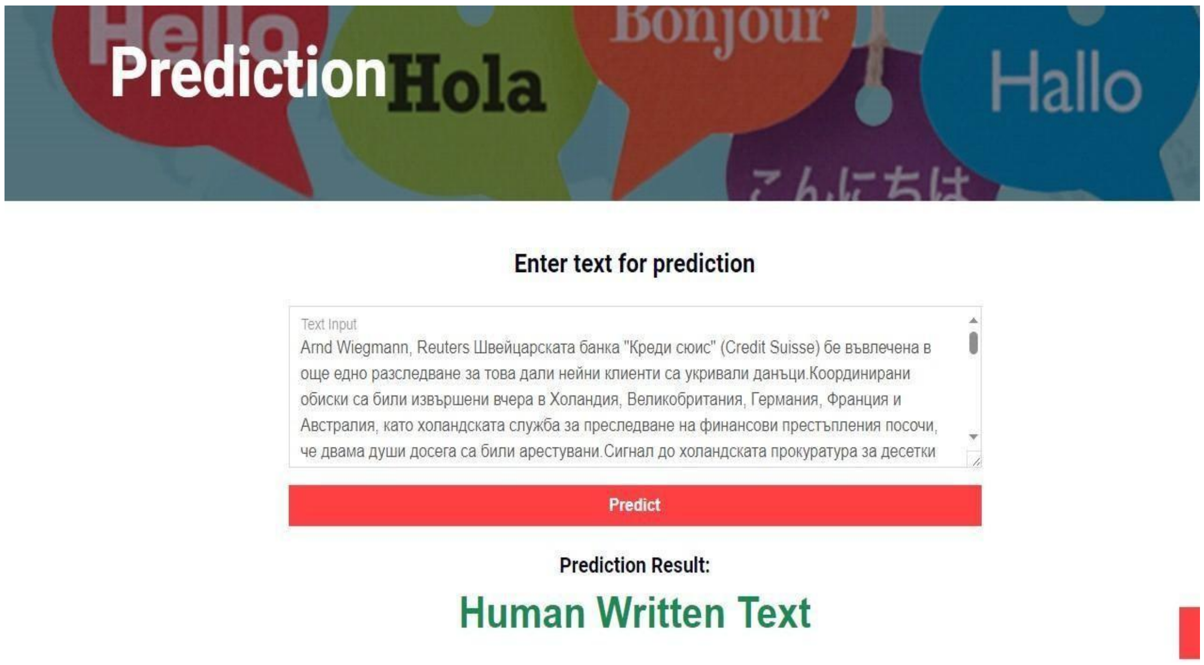
Performance Page: Displays performance metrics of the predictive models, including accuracy and algorithm name.



Prediction Page: Provides accurate predictions based on the input features given by the user.



The investigation of the thermoelectricity of strongly correlated electron systems is of great importance both from fundamental and technological points of view. This article focuses on the thermoelectric effects of the t-J model on a two-dimensional triangular lattice. Using exact diagonalization of the finite size model, the thermopower, Lorenz number, and the dimensionless figure of merit are determined by computational dynamical thermal response functions. Further,



Enter text for prediction

Text Input

Arnd Wiegmann, Reuters Швейцарската банка "Креди сюис" (Credit Suisse) бе въвлечена в още едно разследване за това дали нейни клиенти са укривали данъци.Координирани обиски са били извършени вчера в Холандия, Великобритания, Германия, Франция и Австралия, като холандската служба за преследване на финансови престъпления посочи, че двама души досега са били арестувани.Сигнал до холандската прокуратура за десетки

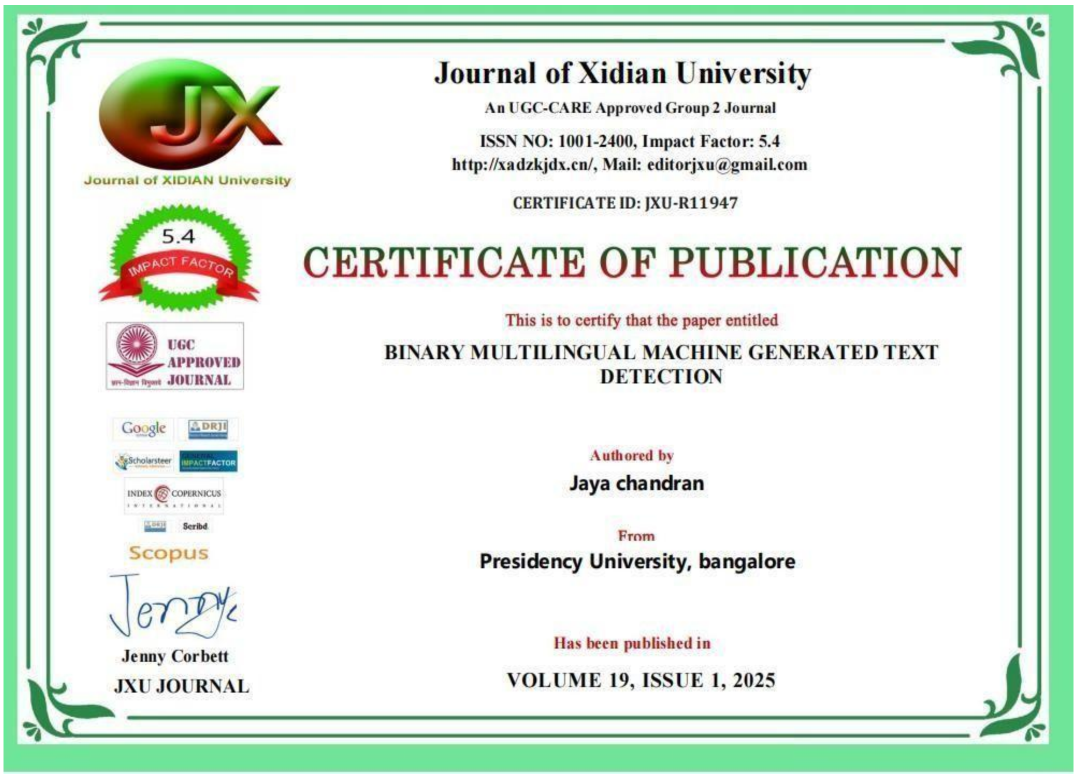
Prediction Result:

Human Written Text

Logout Page: Allows users to securely exit their accounts and terminate active sessions and land in the login page for further login.

APPENDIX-C
ENCLOSURES

1. Journal publication/Conference Paper Presented Certificates of all students.



Journal of Xidian University

An UGC-CARE Approved Group 2 Journal

ISSN NO: 1001-2400, Impact Factor: 5.4

http://xadzkjdx.cn/, Mail: editorjxu@gmail.com

CERTIFICATE ID: JXU-R11947

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

BINARY MULTILINGUAL MACHINE GENERATED TEXT DETECTION

Authored by

Jaya chandran

From

Presidency University, bangalore

Has been published in

VOLUME 19, ISSUE 1, 2025

Journal of XIDIAN University

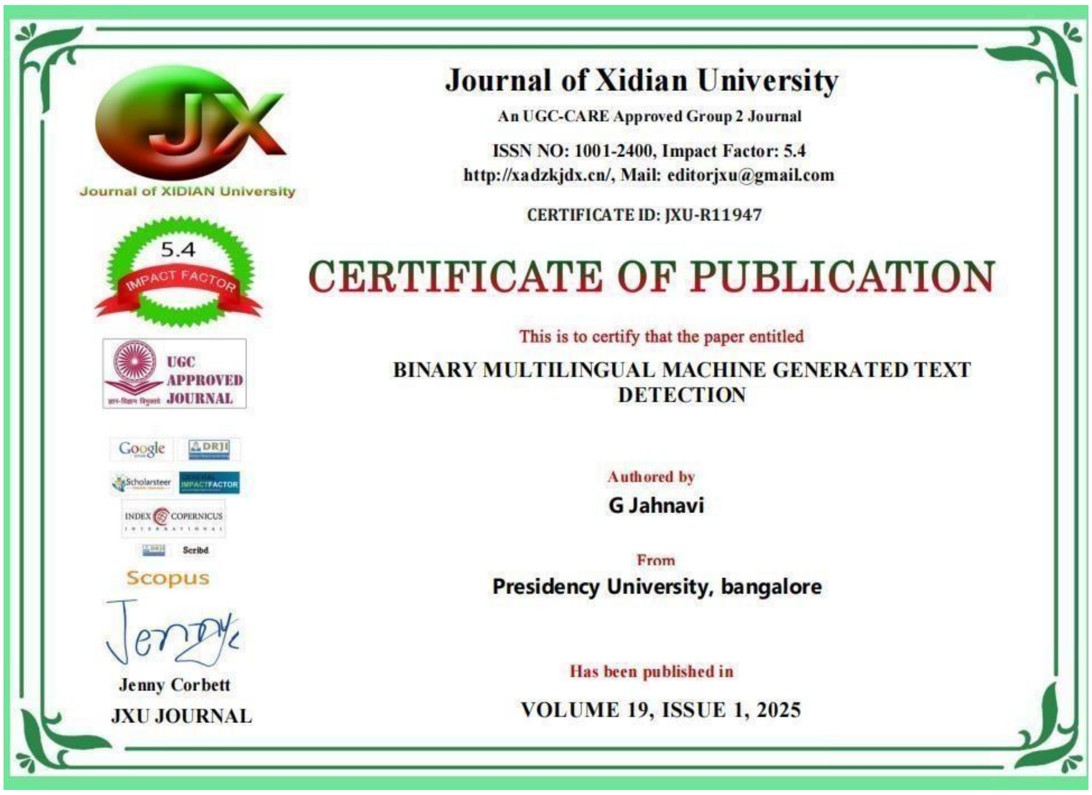
5.4 IMPACT FACTOR

UGC APPROVED JOURNAL

Google DRJI Scholarsteer IMPACTFACTOR INDEX COPERNICUS Scribd Scopus

Jenny Corbett

JXU JOURNAL



Journal of XIDIAN University

5.4 IMPACT FACTOR

UGC APPROVED JOURNAL

Google DRJI Scholarsteer IMPACTFACTOR INDEX COPERNICUS Scribd Scopus

[illegible]

Jenny Corbett

JXU JOURNAL

Journal of Xidian University

An UGC-CARE Approved Group 2 Journal

ISSN NO: 1001-2400, Impact Factor: 5.4

http://xadzkjdx.cn/, Mail: editorjxu@gmail.com

CERTIFICATE ID: JXU-R11947

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

BINARY MULTILINGUAL MACHINE GENERATED TEXT DETECTION

Authored by

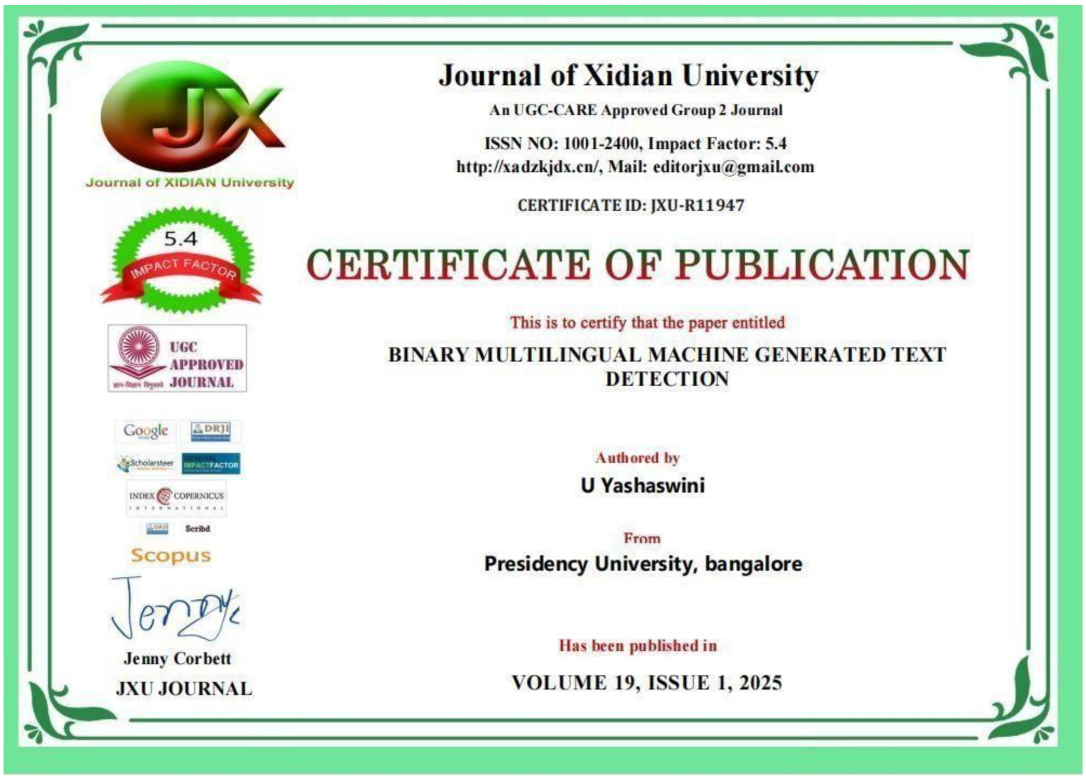
G Jahnavi

From

Presidency University, bangalore

Has been published in

VOLUME 19, ISSUE 1, 2025



Journal of XIDIAN University

5.4 IMPACT FACTOR

UGC APPROVED JOURNAL

Google DRJI Scholarsteer IMPACTFACTOR INDEX COPERNICUS Scribd Scopus

[illegible]

Jenny Corbett

JXU JOURNAL

Journal of Xidian University

An UGC-CARE Approved Group 2 Journal

ISSN NO: 1001-2400, Impact Factor: 5.4

http://xadzkjdx.cn/, Mail: editorjxu@gmail.com

CERTIFICATE ID: JXU-R11947

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

BINARY MULTILINGUAL MACHINE GENERATED TEXT DETECTION

Authored by

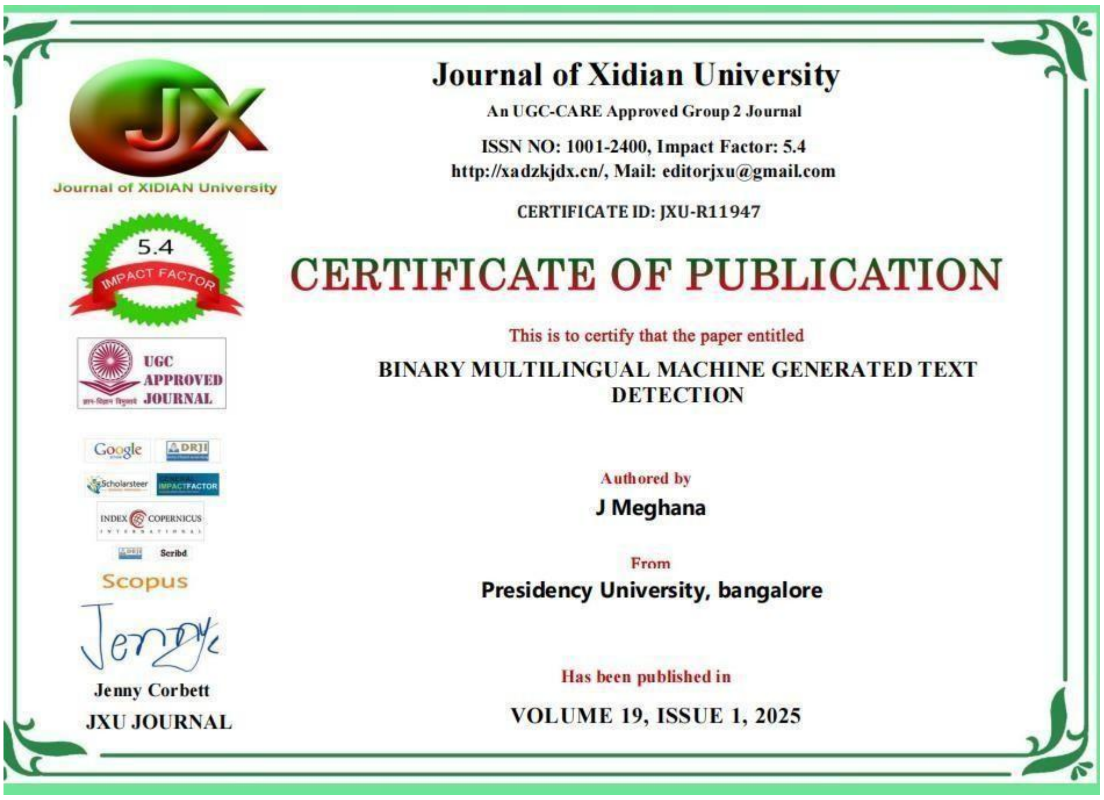
U Yashaswini

From

Presidency University, bangalore

Has been published in

VOLUME 19, ISSUE 1, 2025



Journal of Xidian University

An UGC-CARE Approved Group 2 Journal

ISSN NO: 1001-2400, Impact Factor: 5.4

http://xadzkjdx.cn/, Mail: editorjxu@gmail.com

CERTIFICATE ID: JXU-R11947

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

BINARY MULTILINGUAL MACHINE GENERATED TEXT DETECTION

Authored by

J Meghana

From

Presidency University, bangalore

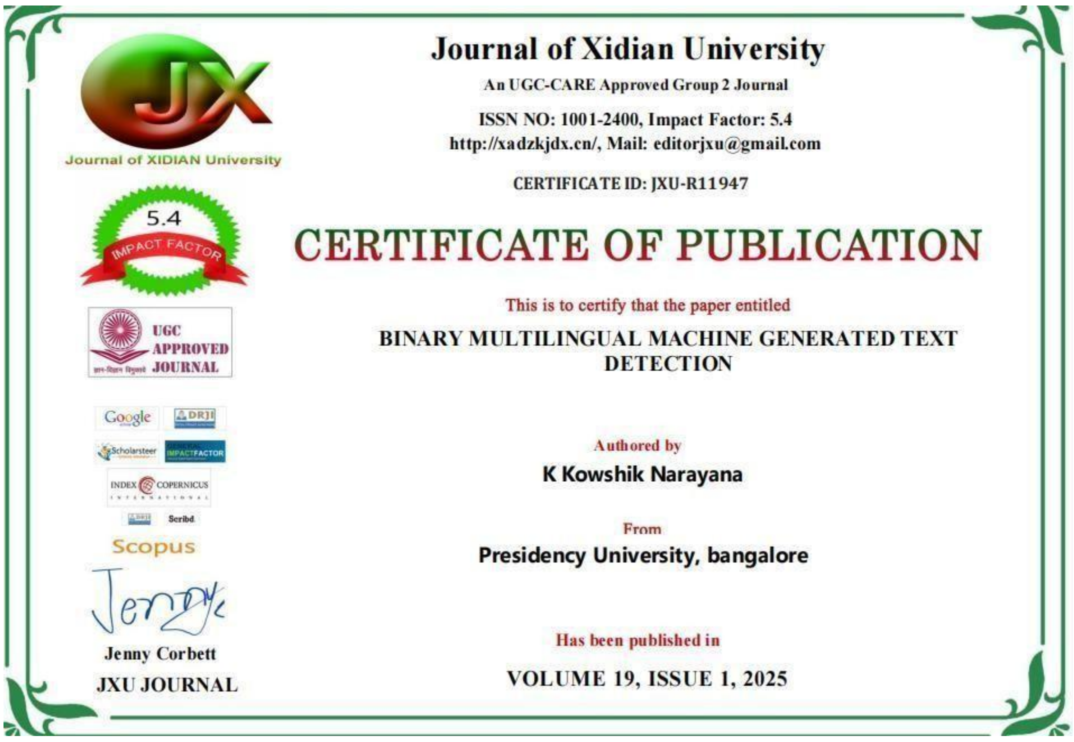
Has been published in

VOLUME 19, ISSUE 1, 2025

[illegible]

Jenny Corbett

JXU JOURNAL



Journal of Xidian University

An UGC-CARE Approved Group 2 Journal

ISSN NO: 1001-2400, Impact Factor: 5.4

http://xadzkjdx.cn/, Mail: editorjxu@gmail.com

CERTIFICATE ID: JXU-R11947

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

BINARY MULTILINGUAL MACHINE GENERATED TEXT DETECTION

Authored by

K Kowshik Narayana

From

Presidency University, bangalore

Has been published in

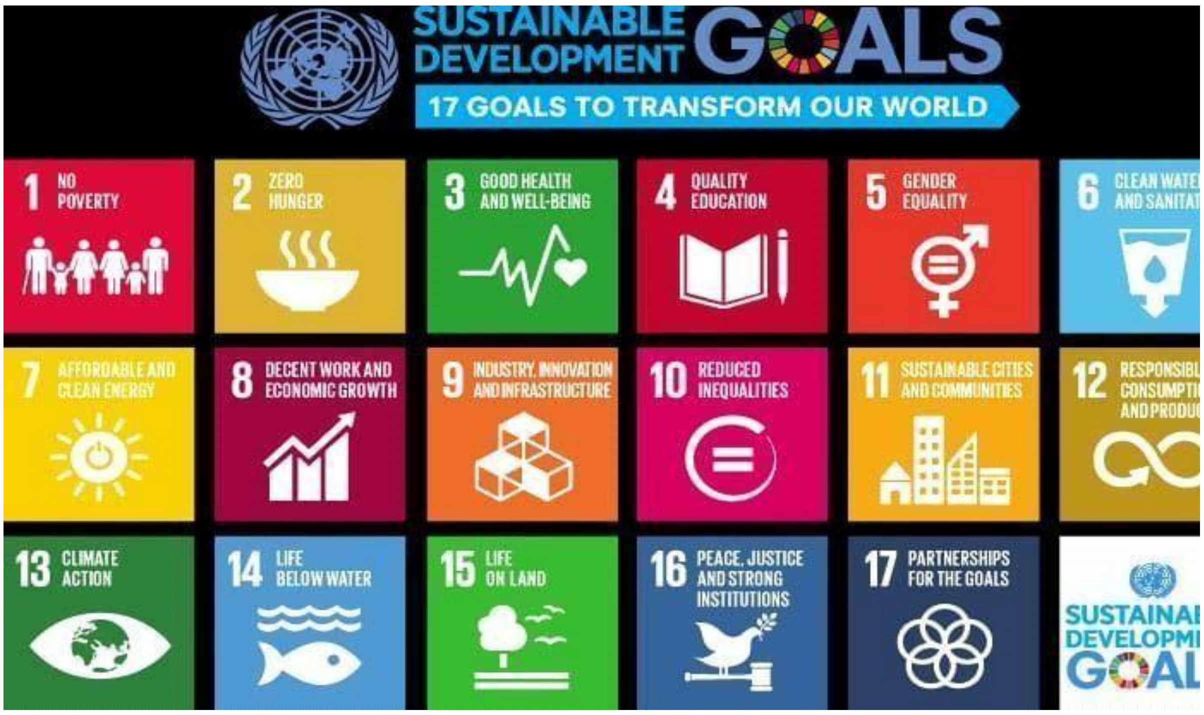
VOLUME 19, ISSUE 1, 2025

[illegible]

Jenny Corbett

JXU JOURNAL

4. Sustainable Development Goals (SDGs).


SUSTAINABLE DEVELOPMENT GOALS
17 GOALS TO TRANSFORM OUR WORLD
1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS
SUSTAINABLE DEVELOPMENT GOALS

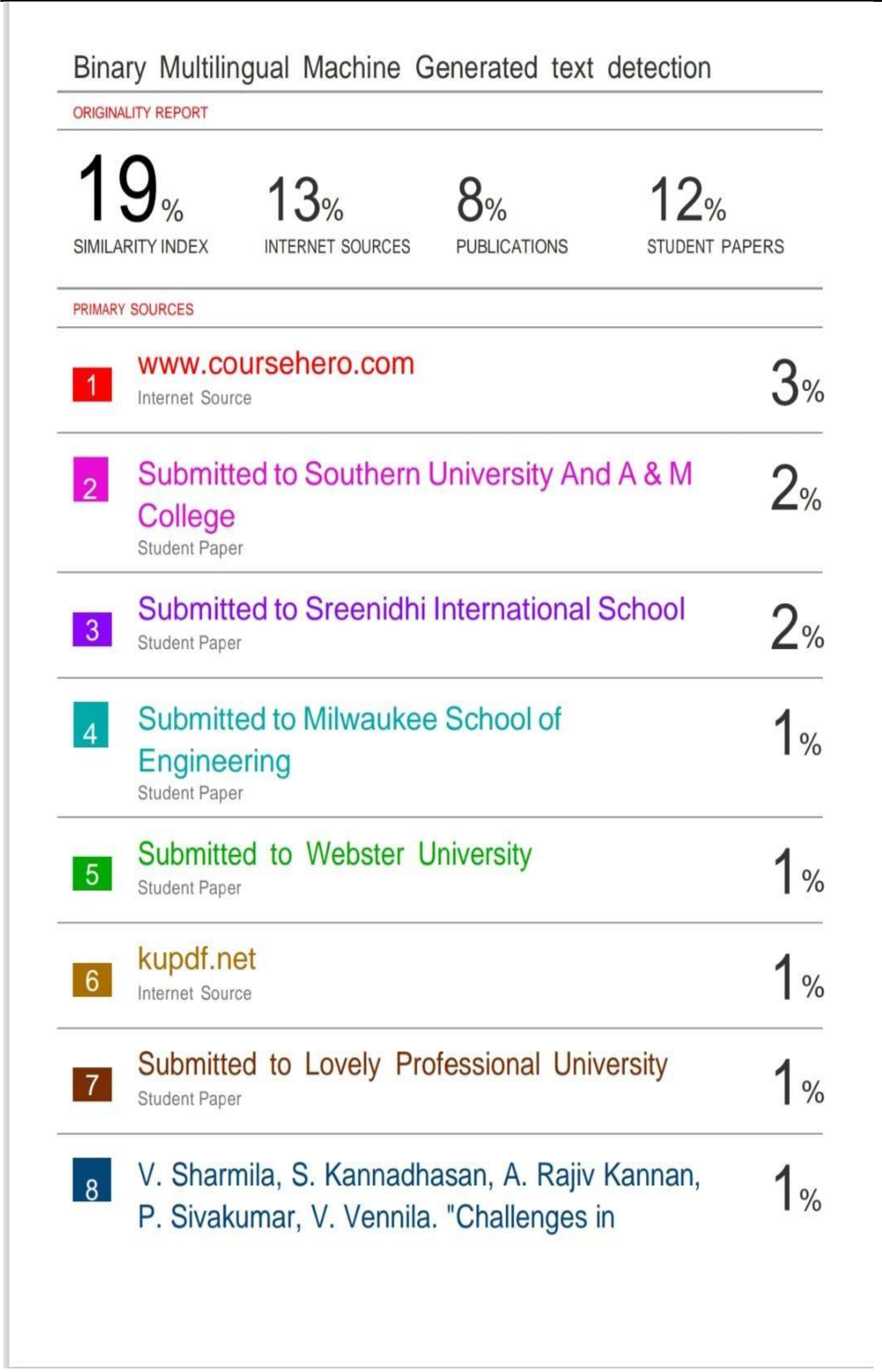
The project "Binary Multilingual Machine-Generated Text Detection" aligns most closely with the following Sustainable Development Goals (SDGs):

Quality Education (SDG 4):By identifying AI-generated content, this project can contribute to the integrity of educational systems by helping detect plagiarism and ensure the originality of learning materials and assessments.

Peace, Justice, and Strong Institutions (SDG 16):Detecting machine-generated text can combat misinformation and promote trustworthiness in online content, supporting transparency, accountability, and the functioning of strong institutions.

Industry, Innovation, and Infrastructure (SDG 9):The project promotes innovation by contributing to advancements in AI, machine learning, and natural language processing, fostering industry growth and technological development.

Reduced Inequalities (SDG 10):By working in multiple languages, the project ensures inclusivity and helps reduce inequalities in access to tools that verify content across linguistic and regional boundaries.



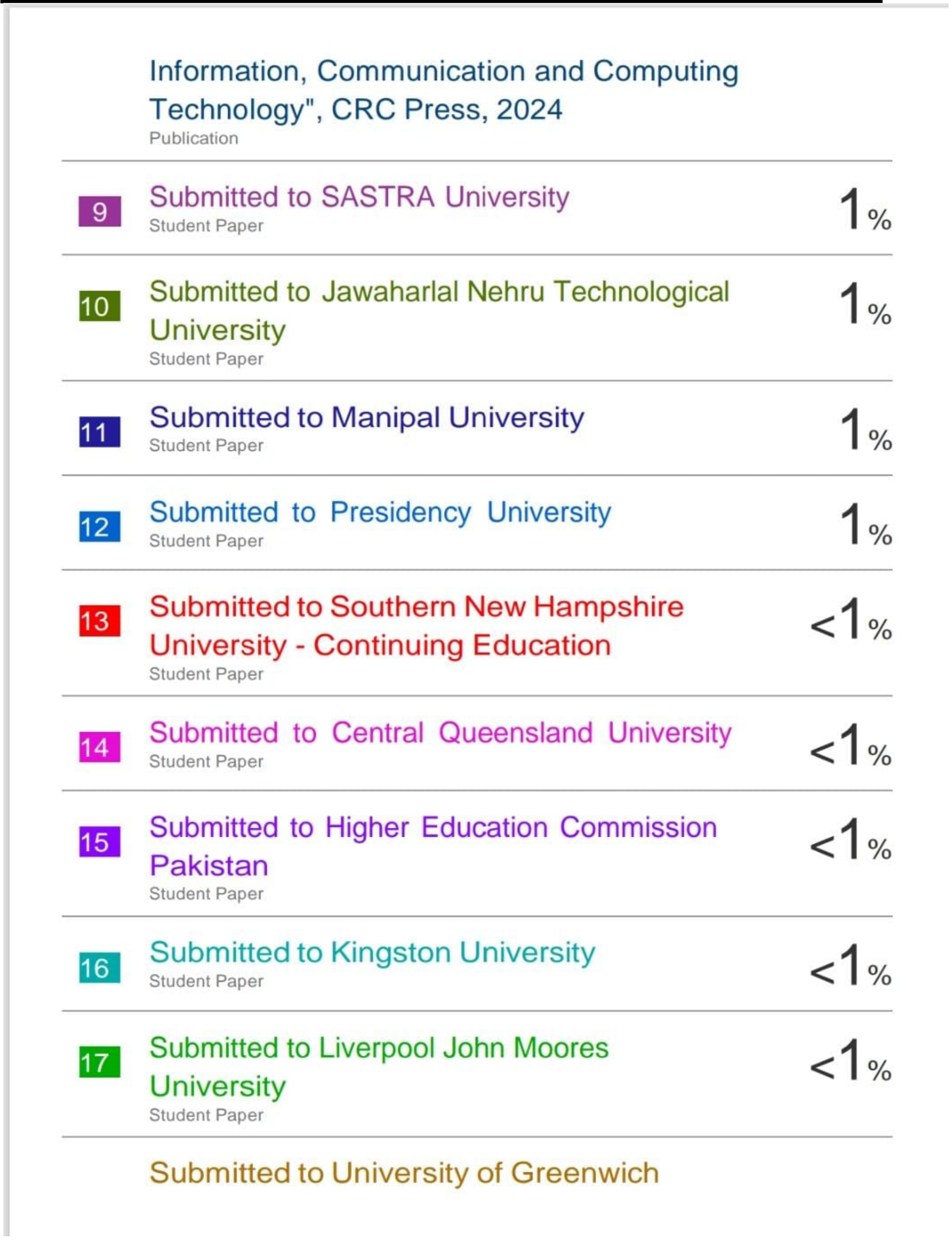
Binary Multilingual Machine Generated text detection

ORIGINALITY REPORT

19%	13%	8%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	3%
2	Submitted to Southern University And A & M College Student Paper	2%
3	Submitted to Sreenidhi International School Student Paper	2%
4	Submitted to Milwaukee School of Engineering Student Paper	1%
5	Submitted to Webster University Student Paper	1%
6	kupdf.net Internet Source	1%
7	Submitted to Lovely Professional University Student Paper	1%
8	V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challenges in	1%



	Information, Communication and Computing Technology", CRC Press, 2024 Publication	
9	Submitted to SASTRA University Student Paper	1%
10	Submitted to Jawaharlal Nehru Technological University Student Paper	1%
11	Submitted to Manipal University Student Paper	1%
12	Submitted to Presidency University Student Paper	1%
13	Submitted to Southern New Hampshire University - Continuing Education Student Paper	<1%
14	Submitted to Central Queensland University Student Paper	<1%
15	Submitted to Higher Education Commission Pakistan Student Paper	<1%
16	Submitted to Kingston University Student Paper	<1%
17	Submitted to Liverpool John Moores University Student Paper	<1%
	Submitted to University of Greenwich	