TABLE OF CONTENTS

ABSTRACT

In the modern digital landscape, the rapid dissemination of misinformation and fake news has become a critical concern, impacting public perception, trust, and decision-making. Traditional fake news detection systems predominantly rely on text-based analysis, which often overlooks the deceptive power of manipulated images and visual content accompanying news stories. To address these limitations, this project presents a multimodal fake news detection system that integrates NLP and CV techniques for more accurate and robust identification of false information.

The system accepts both textual and visual inputs, enabling users to submit news in the form of text, images, or both. For textual analysis, the proposed model employs a hybrid deep learning architecture combining BERT and BiLSTM. BERT is used for its contextual language understanding capabilities, while BiLSTM captures the sequential flow and relationships between words. This hybrid model enhances the system's ability to detect subtle patterns and context-dependent nuances in deceptive news content.

For image analysis, a pre-trained ResNet-101 Convolutional Neural Network is utilized to classify images as real or fake based on visual features. This deep CNN is capable of identifying manipulated, altered, or misleading images that are commonly used to support false narratives. By combining the outputs from both the text and image classifiers, the system produces a comprehensive credibility score, providing a more holistic evaluation of the news content.

An innovative feature of this system is the user credibility scoring mechanism, which calculates a trust score based on the historical accuracy of content posted by each user. This not only helps in identifying repeat offenders but also promotes digital accountability and responsible content sharing.

The system is deployed as a Flask-based web application with a responsive user interface and integrated SQLite database for user and submission tracking. Designed with modularity in mind, the architecture supports real-time predictions, easy maintenance, and future upgrades such as enhanced scoring logic, model re-training, and cloud integration.

Overall, the proposed multimodal fake news detection system provides a scalable, user-friendly, and intelligent platform for combating misinformation, empowering users and communities to make informed decisions in the face of digital deception.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACRONYM	FULL FORM
AI	Artificial Intelligence
ML	Machine Learning
NLP	Natural Language Processing
DL	Deep Learning
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
BERT	Bidirectional Encoder Representations from Transformers
API	Application Programming Interface
GPU	Graphics Processing Unit
OCR	Optical Character Recognition
SQL	Structured Query Language
HTML	HyperText Markup Language
URL	Uniform Resource Locator
HTTP	HyperText Transfer Protocol
BiLSTM	Bidirectional Long Short-Term Memory
ResNet	Residual Network

CHAPTER-I

INTRODUCTION

1.1 OVERVIEW

Fake news refers to the deliberate spread of misinformation or hoaxes under the guise of legitimate news. With the rapid proliferation of digital content through social media and online platforms, fake news has become a serious threat to the authenticity of information and public trust. It can manipulate public opinion, influence political outcomes, and distort understanding of critical topics such as health, science, and current events.

The traditional manual methods of fact-checking, although accurate, are time-consuming and cannot keep pace with the volume of content generated online every second. To address this challenge, automated fake news detection systems using advanced machine learning and deep learning techniques are gaining momentum.

In the proposed system, a hybrid deep learning approach is adopted to tackle fake news detection more effectively. For analyzing textual content, the system integrates BERT (Bidirectional Encoder Representations from Transformers) and BiLSTM (Bidirectional Long Short-Term Memory). BERT excels at understanding the context of words within a sentence, enabling deeper semantic analysis. When combined with BiLSTM, which captures long-range dependencies in both directions, the system can detect subtle patterns and cues in text that are often present in deceptive news articles. This enhances the ability to identify nuanced linguistic characteristics indicative of fake news.

For image-based fake news detection, the system incorporates ResNet-101, a powerful convolutional neural network architecture known for its residual learning capabilities. ResNet-101 is effective in image classification tasks and is capable of detecting anomalies or manipulations in images that may suggest misinformation. This visual analysis complements the textual analysis by identifying tampered or misleading visuals accompanying news content.

By combining both textual and visual modalities, the system offers a multi-modal fake news detection framework that is robust, scalable, and accurate. This comprehensive approach ensures more reliable detection of fake news across various formats, thereby contributing to the fight against misinformation in the digital age.

1.2 OBJECTIVE

The primary objective of this project is to develop a multi-modal fake news detection system that can accurately identify and classify misleading or false information present in both textual and visual formats. By leveraging advanced deep learning models such as BERT and BiLSTM for text analysis and ResNet-101 for image analysis, the system aims to automate the process of detecting fake news, thereby improving scalability, speed, and accuracy compared to manual fact-checking methods. The ultimate goal is to support the fight against misinformation by providing a reliable, intelligent tool that enhances public awareness, ensures information integrity, and helps maintain trust in digital content.

1.3 MOTIVATION

Social media facilitates the creation and sharing of information that uses computer-mediated technologies. This media changed the way groups of people interact and communicate. It allows low cost, simple access and fast dissemination of information to them. The majority of people search and consume news from social media rather than traditional news organizations these days. On one side, where social media have become a powerful source of information and bringing people together, on the other side it also put a negative impact on society.

Look at some examples herewith; Facebook Inc’s popular messaging service, WhatsApp became a political battle-platform in Brazil’s election. False rumours, manipulated photos, de-contextualized videos, and audio jokes were used for campaigning. These kinds of stuff went viral on the digital platform without monitoring their origin or reach. A nationwide block on major social media and messaging sites including Facebook and Instagram was done in Sri Lanka after multiple terrorist attacks in the year 2019. The government claimed that “false news reports” were circulating online. This is evident in the challenges the world's most powerful tech companies face in reducing the spread of misinformation.

Such examples show that Social Media enables the widespread use of “fake news” as well. The news disseminated on social media platforms may be of low quality carrying misleading information intentionally. This sacrifices the credibility of the information. Millions of news articles are being circulated every day on the Internet – how one can trust which is real and which is fake? Thus incredible or fake news is one of the biggest challenges in our digitally connected world.

Fake news detection on social media has recently become an emerging research domain. The domain focuses on dealing with the sensitive issue of preventing the spread of fake news on social media. Fake news identification on social media faces several challenges. Firstly, it is difficult to collect fake news data. Furthermore, it is difficult to label fake news manually. Since they are intentionally written to mislead readers, it is difficult to detect them simply based on news content. Furthermore, Facebook, Whatsapp, and Twitter are closed messaging apps.

The misinformation disseminated by trusted news outlets or their friends and family is therefore difficult to be considered as fake. It is not easy to verify the credibility of newly emerging and time-bound news as they are not sufficient to train the application dataset. Significant approaches to differentiate credible users, extract useful news features and develop authentic information dissemination systems are some useful domains of research and need further investigations. If we can't control the spread of fake news, the trust in the system will collapse. There will be widespread distrust among people. There will be nothing left that can be objectively used. It means the destruction of political and social coherence. We wanted to build some sort of web-based system that can fight this nightmare scenario. And we made some significant progress towards that goal.

Machine learning (ML) is a type of Artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. The extensive spread of news can have a significant negative impact on individuals and society. Understanding the truth of new and message with news detection can create positive impact on the society.

1.4 ORGANIZATION OF THE CHAPTERS

Chapter I: Gives a general introduction of the project, the existing systems, objective and motivation of the study.

Chapter II: Literature survey that is done to support the project and various machine learning algorithms has been discussed.

Chapter III: Explains the existing system architecture, modules, description for the individual modules involved in the work and implementations.

Chapter IV: Explains the proposed system, it consists of the framework design, modules, descriptions and implementation.

Chapter V: The Result and output of the proposed system screenshots has been displayed.

Chapter VI: Describes the conclusion of the work done and presents some future enhancement work that can be carried over.

CHAPTER II

LITERATURE REVIEW

2.1 TECHNIQUES

2.1.1 SUPERVISED LEARNING:

Supervised learning is a machine learning method where a model is trained on labeled data to make predictions. It learns the mapping between inputs and outputs by analyzing training data and is later tested with unseen data to evaluate its performance. This approach mimics a teacher-student model of learning and is used in tasks like spam detection and fake news classification.

Supervised learning can be grouped further in two categories of algorithms:

1. Classification
2. Regression

1. Classification :

Classification is a type of supervised learning that is used to predict categorical values, such as whether a customer will churn or not, whether an email is spam or not, or whether a medical image shows a tumor or not. Classification algorithms learn a function that maps from the input features to a probability distribution over the output classes.

- Logistic Regression
- Support Vector Machines
- Decision Trees
- Random Forests
- Naive Baye

2. Regression :

Regression is a type of supervised learning that is used to predict continuous values, such as house prices, stock prices, or customer churn. Regression algorithms learn a function that maps from the input features to the output value.

- Linear Regression
- Polynomial Regression
- Support Vector Machine Regression
- Decision Tree Regression
- Random Forest Regression

2.1.2 DEEP LEARNING TECHNIQUES

Deep learning is a subset of machine learning that focuses on algorithms inspired by the structure and function of the brain called artificial neural networks. In recent years, deep learning has significantly advanced the field of fake news detection due to its capability to automatically learn complex patterns from large-scale datasets. The following deep learning models are particularly prominent in text-based fake news detection:

1. LSTM :

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) designed to learn long-term dependencies in sequence data. It uses memory cells to retain information over time, making it effective for natural language tasks. In fake news detection, LSTMs capture contextual relationships in text, helping identify patterns of misinformation across word sequences.

2. BiLSTM:

BiLSTM enhances standard LSTM by processing input sequences in both forward and backward directions. This allows the model to capture context from both past and future words, improving its understanding of text semantics. Such bidirectional context is especially valuable in fake news detection, where meaning often depends on surrounding words. BiLSTM has shown improved accuracy in text classification tasks like fake news and sentiment analysis.

3. BERT:

BERT is a powerful language model by Google that uses a transformer-based architecture with bidirectional training, allowing it to understand the context of words by looking at both sides of a sentence. Unlike traditional models that read text sequentially, BERT processes the entire sentence at once, capturing deeper word relationships. In fake news detection, BERT excels at identifying deceptive language patterns and can be fine-tuned for accurate text classification tasks.

2.1.3 IMAGE-BASED DETECTION TECHNIQUES

Fake news is not confined to textual information alone; images and videos are often used to mislead readers or lend credibility to false claims. Image-based fake news detection aims to analyze and classify visual content to determine its authenticity. Convolutional Neural Networks (CNNs) are the backbone of most image-based detection systems due to their proficiency in visual pattern recognition. The following are key CNN-based approaches used in this domain:

1. CNNs:

CNNs are deep learning models designed to process data with a grid-like topology, such as images. They consist of layers that apply convolutional filters to detect features like edges, textures, shapes, and objects. In fake news detection, CNNs are trained to identify visual inconsistencies that may suggest image manipulation, such as cloning, splicing, or retouching. The extracted features are then used to classify the image as real or fake. CNNs offer high accuracy in identifying anomalies in images and are scalable to large datasets.

2. ResNet:

ResNet is a powerful CNN architecture introduced to address the vanishing gradient problem in deep neural networks. It uses skip connections or residual connections that allow gradients to bypass certain layers, enabling the successful training of very deep networks. This design helps in learning highly complex and detailed visual patterns.

- **ResNet-50:** A 50-layer deep residual network that balances performance and computational cost. It is commonly used in image classification tasks, including detecting fake or tampered images.
- **ResNet-101:** An extended version of ResNet-50 with 101 layers, capable of learning even more intricate visual features. It is particularly effective in tasks that require distinguishing between subtle manipulations, such as detecting forged faces, altered scenes, or photoshopped elements.

2.2 SURVEY OF THE RELATED WORK

S.NO	PAPER NAME	TECHNIQUES USED	DATA SET USED	EXISTING SYSTEMS	LIMITATIONS
1	A Hybrid Transformer-Based Model for Optimizing Fake News Detection.	BERT with bi-directional deep learning layers such as BiLSTM and BiGRU.	WELFake dataset	focus on the distribution of negative news but struggle to provide sufficient context or logic behind the prediction.	Inability to Analyze Visual Content. Limited Contextual Understanding.
2	MEFaND: A Multimodel Framework for Early Fake News Detection	Multimodel ensemble framework for early-stage fake news detection	Early-stage fake news datasets	Single-model detection approaches	Potential computational overhead due to multiple model integration
3	A Multi-Kernel Optimized Convolutional Neural Network With Urdu Word Embedding to Detect Fake News	MOCNN model combines deep learning Ensemble learning techniques	UFN (Urdu Fake News): BET (Benchmark for Evaluating Truthfulness):	The MOCNN model leverages multiple variable-length kernels in the convolutional layer. Parameters of the MOCNN model are optimized using grid search techniques	Overfitting due to extensive hyper-parameter tuning and dropout regularization.
4	Enhancing Fake News Detection by Multi-Feature Classification	global, spatial, and temporal features CNN+ Bi LSTM + FLN	ISOT: A dataset containing	Spatial features are obtained through a convolutional	Tested only on English datasets, future focus on non-English datasets.

		(Convolutional Neural Network + Bi LSTM + Fast Learning Network)	labelled fake news articles. FA-KES: Another dataset used for comparison.	neural network (CNN). Temporal features are captured using bi-directional long short-term memory (Bi LSTM).	
5	A Comprehensive Review on Fake News Detection With Deep Learning	Deep learning techniques like Attention, GANs, and BERT. Leveraging neural network models for fake news detection. Feature extraction and selection for fake news detection	One such dataset contains 100,000 news articles categorized into ten classes (one real news class and various fake news classes).	It emphasizes the importance of accurate detection due to the widespread impact of misinformation. Researchers explore datasets, NLP techniques, and evaluation metrics to enhance detection mechanisms.	Lack of focus on data quality assessment in fake news detection. Limited exploration of network-based patterns for fake news detection.

2.3 SURVEY CONCLUSION

The integration of advanced machine learning and deep learning techniques in fake news detection has been a prominent area of recent research [1][2][4]. The literature highlights the significant role of supervised learning methods, particularly classification algorithms like Logistic Regression, Support Vector Machines, and Random Forests, in enabling early-stage detection of misinformation by learning from labeled datasets [1][4].

Several studies emphasize the superiority of deep learning approaches such as LSTM, BiLSTM, CNN, and transformer-based models like BERT in capturing the contextual, semantic, and temporal nuances of deceptive content [1][2]. These models are particularly effective in handling large volumes of text data and have shown improved accuracy over traditional machine learning models. However, challenges remain in fine-tuning these models to handle multilingual and domain-specific content effectively [2][5].

The survey also underscores the importance of incorporating image-based analysis for detecting manipulated or misleading visual content associated with fake news. Techniques such as CNNs and ResNet variants (ResNet-50, ResNet-101) have proven effective in identifying inconsistencies or tampered elements in images [4]. Despite their strengths, these models are computationally intensive and require large, labeled image datasets for optimal performance.

Furthermore, ensemble and hybrid models combining textual and visual features have emerged as promising solutions for improving detection robustness and reducing false positives [2][3]. Nevertheless, issues such as overfitting, lack of explainability, and computational overhead remain prevalent across several existing frameworks [3][5].

In conclusion, while the current literature presents a strong foundation for fake news detection using both textual and visual cues, future research must address the limitations related to dataset diversity, model interpretability, and cross-modal analysis. Emphasis should also be placed on lightweight architectures and real-time detection systems that can scale effectively across social media platforms and multiple languages.

CHAPTER III

EXISTING WORK

3.1 ARCHITECTURE

The majority of fake news detection systems focus on textual content analysis, leveraging advanced deep learning models like BERT, Bi-LSTM, and Bi-GRU to classify news articles as real or fake. BERT (Bidirectional Encoder Representations from Transformers) is particularly effective in understanding the context of text by tokenizing the input and using bidirectional attention mechanisms. This allows BERT to grasp the meaning of words based on the surrounding context, enhancing its performance in tasks like fake news detection. The use of Bi-LSTM (Bidirectional Long Short-Term Memory) and Bi-GRU (Bidirectional Gated Recurrent Units) further complements BERT by processing sequential data in both forward and backward directions. This capability is crucial for capturing temporal dependencies in text, making these models particularly effective for analyzing narrative structures, detecting inconsistencies in storytelling, and identifying linguistic patterns indicative of misinformation.

Despite the impressive results these models achieve in detecting fake news from textual content, there are several significant limitations in their application. One of the primary challenges is their inability to detect multimedia-based misinformation, specifically fake images and videos. The growing prevalence of multimedia in news articles means that an effective fake news detection system must be capable of handling not only textual content but also visual information. The current systems, however, are predominantly focused on text and lack the necessary mechanisms for analyzing visual media, which limits their effectiveness in an increasingly multimedia-driven information landscape.

Moreover, these text-based models do not support multimodal analysis, which is critical for comprehensively evaluating fake news that combines both textual and visual content. In many cases, fake news is not just about the narrative but also about how images are manipulated to support misleading claims. Without the ability to analyze both the text and images together, these systems miss out on detecting more complex forms of misinformation that leverage the power of both mediums.

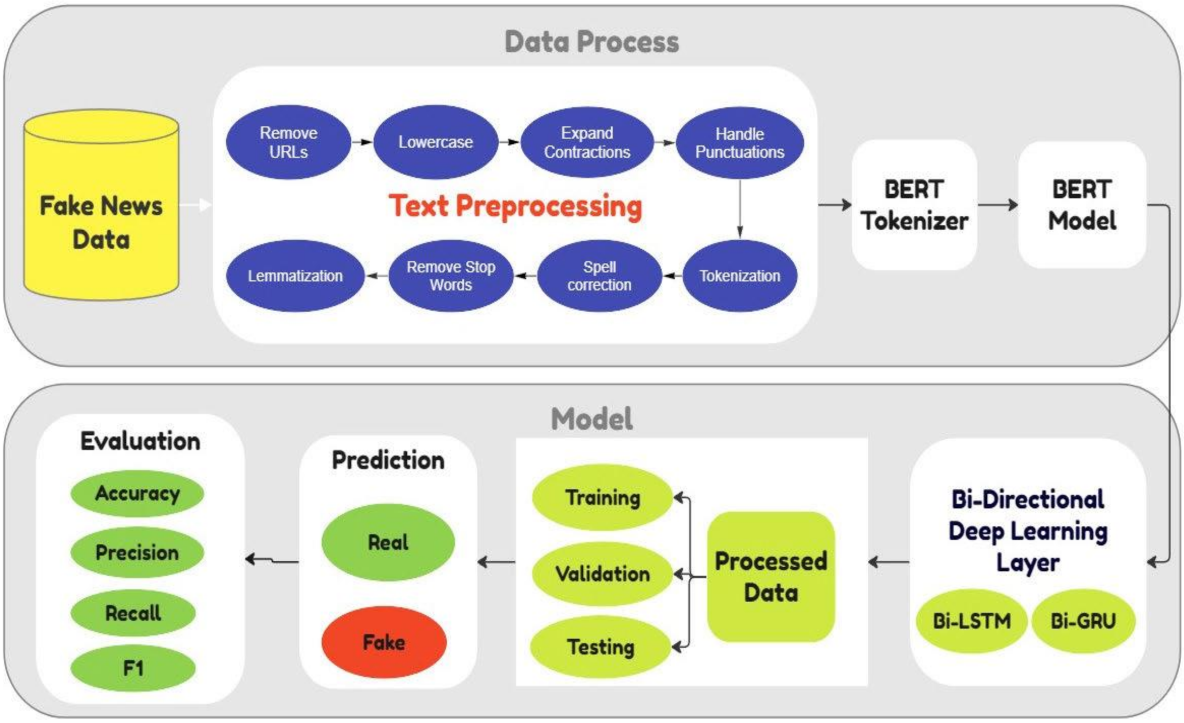


Fig 3.1. Architecture of Existing System

3.2 ALGORITHMS AND WORKING

3.2.1 BERT:

BERT is a powerful transformer-based model specifically designed to understand the context of words in a sentence, making it highly effective for natural language understanding tasks such as fake news detection. At the heart of BERT lies an attention mechanism that enables the model to focus on all parts of the input text simultaneously, rather than processing it sequentially as traditional RNNs or LSTMs do. This non-sequential, bidirectional approach allows BERT to derive rich, contextual relationships between words, irrespective of their position in the sentence.

BERT's architecture is composed primarily of transformer encoders, which are responsible for encoding the input text into dense vector representations. These encoders process entire sequences of text at once using self-attention mechanisms, making the model bidirectional—capable of understanding both left and right context simultaneously. Unlike traditional models that read text from left to right or right to left, BERT reads the entire sequence in both directions, enabling it to learn more meaningful word representations.

Before being fed into the model, the input text is tokenized and formatted appropriately. The special classification token [CLS] is inserted at the beginning of each input sequence, and a separation token [SEP] is used to distinguish between different segments when needed. Each token is then converted into embeddings which are passed through multiple layers of transformer encoders. Attention masks are used to indicate which tokens should be attended to, ensuring that the model focuses on meaningful parts of the text.

The output of each encoder layer is a set of fixed-length vectors, with the final hidden state of the [CLS] token used as the aggregated representation of the entire input sequence. This vector is then passed to a classifier (typically a dense layer) to perform the final classification task—whether the news is real or fake.

In the context of fake news detection, BERT proves especially useful due to its ability to capture subtle semantic nuances. Fake news often disguises itself using language that mimics legitimate news sources, making it challenging to detect with simple keyword matching or syntactic analysis. BERT's dynamic word embeddings, which adjust based on surrounding context, help distinguish between superficially similar but semantically different statements. This ability to understand the intent and deeper meaning of words makes BERT highly suited for identifying misleading or deceptive content in news articles.

By learning complex contextual and semantic relationships, BERT enhances the detection of linguistic manipulations often present in fake news—such as ambiguous statements, emotional language, or the misuse of technical terminology. Its robust representation of language thus provides a solid foundation for more accurate and context-aware fake news classification models.

3.2.2 Bi-Directional Deep Learning

Bi-directional deep learning models such as and Bi-GRU play a pivotal role in enhancing the accuracy and contextual understanding required for fake news detection. These models are designed to process text data in both forward and backward directions, enabling them to capture dependencies and relationships that are not easily detectable through traditional, unidirectional models. In the realm of misinformation, where the meaning of a statement often hinges on subtle contextual cues and the sequencing of information, this dual-processing capability becomes highly advantageous.

1. Bi-LSTM

Bidirectional Long Short-Term Memory (Bi-LSTM) networks are an advanced type of Recurrent Neural Network (RNN) that can retain information from both past and future contexts. This is made possible by having two LSTM layers—one that processes the sequence from the beginning to the end and another that processes it in reverse. This two-way information flow allows the model to understand not just the immediate neighborhood of a word but also the broader context in which it appears.

Bi-LSTM is particularly effective at modeling long-range dependencies in text, which is essential for detecting fake news that may rely on inconsistencies or misleading statements scattered throughout an article. For instance, a misleading headline might only be revealed as false when compared with statements deeper in the content. Bi-LSTM's memory mechanism enables it to handle such scenarios by keeping track of relevant information over extended text spans.

Moreover, Bi-LSTM is adept at identifying subtle patterns and semantic variations that are often indicative of fake news. These might include unusual phrasing, emotionally charged language, or contradictory statements. By learning these linguistic cues, the model can more accurately classify whether a news item is real or fake. When used in combination with pre-trained contextual embeddings like BERT, Bi-LSTM further enhances the model's robustness by providing additional layers of sequence-aware interpretation.

2. Bi-GRU

Bidirectional Gated Recurrent Units (Bi-GRU) offer a computationally efficient alternative to Bi-LSTM while retaining most of its capabilities. Like Bi-LSTM, Bi-GRU processes input data in both directions, allowing the model to understand complex dependencies and relationships within the text. What sets GRUs apart is their simplified architecture, which requires fewer parameters and consumes less memory, thus making them more suitable for scenarios where computational resources are limited or fast inference is required.

Bi-GRU models incorporate gating mechanisms to regulate the flow of information, helping the network retain important information over longer sequences while mitigating the vanishing gradient problem often faced by deep RNNs. This makes Bi-GRU especially effective in capturing the flow of information across entire news articles or multi-sentence headlines, which is often critical in the detection of misinformation.

In the context of fake news detection, Bi-GRU is highly beneficial for identifying linguistic strategies used to deceive readers, such as subtle manipulation of facts, omission of critical details, or misleading summaries. Its dual-directional analysis helps uncover contradictions and inconsistencies that may only become apparent when considering the full context of the article.

Furthermore, Bi-GRU is known for offering a good trade-off between performance and computational efficiency. This makes it particularly useful in dynamic or time-sensitive applications—such as monitoring social media platforms—where real-time or near-real-time detection of fake news is essential to preventing the spread of misinformation.

3.3 MODULES DESCRIPTION

3.3.1 Data Loading and Preprocessing:

The dataset used is "WELFake_Dataset.csv", containing news titles, texts, and labels indicating real or fake news. A new column, content, is created by combining the title and text to form the main input. Only content and label columns are selected. Missing values are removed, and labels are converted to integers (0 for fake, 1 for real). The dataset is split into training and validation sets in an 80:20 ratio using train_test_split().

3.3.2 Tokenization using BERT:

The bert-base-uncased tokenizer is used to convert input text into token IDs and attention masks. It handles padding and truncation to maintain a uniform sequence length of 512 tokens. The tokenized outputs are stored in dictionaries for both training and validation sets.

3.3.3 Dataset and DataLoader Creation:

A custom PyTorch Dataset class, FakeNewsDataset, manages the tokenized inputs and labels. These are wrapped in DataLoader objects with a batch size of 16 to enable efficient batch processing and shuffling during training.

3.3.4 Model Architecture:

Two model variants are built using pre-trained BERT embeddings. The first uses a BiLSTM layer with a hidden size of 256, while the second uses a BiGRU layer. Both are followed by a dropout layer and a fully connected output layer with a sigmoid activation function that predicts the probability of the news being real.

3.3.5 Model Training:

The models are trained using Binary Cross-Entropy Loss and the Adam optimizer with a learning rate of 2e-5. A learning rate scheduler reduces the rate every 5 epochs. During training, loss and accuracy are tracked. BERT parameters are frozen to reduce computational load and focus learning on the RNN layers. Results are visualized using Matplotlib.

3.3.6 Evaluation:

Evaluation is done using the validation set with the model in eval() mode. Metrics such as accuracy, precision, recall, and F1-score are computed using sklearn.metrics. Predictions are thresholded at 0.5 to classify as real or fake.

3.3.7 Visualization:

A training plot is generated showing the loss and accuracy trends over the training epochs. This visualization helps monitor the model’s learning progress and identify overfitting or underfitting trends.

3.4 EXISTING SYSTEM IMPLEMENTATION

3.4.1 Loading and Preprocessing

```
df = pd.read_csv("WELFake_Dataset.csv")
df['content'] = df['title'] + " " + df['text']
df = df[['content', 'label']].dropna()
df['label'] = df['label'].astype(int)

train_texts, val_texts, train_labels, val_labels = train_test_split(
    df['content'].tolist(), df['label'].tolist(), test_size=0.2, random_state=42)

tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')

train_encodings = tokenizer(train_texts, truncation=True, padding=True, max_length=512)
val_encodings = tokenizer(val_texts, truncation=True, padding=True, max_length=512)
```

3.4.2 Converting to PYTorch Dataset

```
# Convert to PyTorch Dataset
class FakeNewsDataset(Dataset):
    def __init__(self, encodings, labels):
        self.encodings = encodings
        self.labels = labels

    def __len__(self):
        return len(self.labels)

    def __getitem__(self, idx):
        item = {key: torch.tensor(val[idx]) for key, val in self.encodings.items()}
        item['labels'] = torch.tensor(self.labels[idx])
        return item

train_dataset = FakeNewsDataset(train_encodings, train_labels)
val_dataset = FakeNewsDataset(val_encodings, val_labels)

train_loader = DataLoader(train_dataset, batch_size=16, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=16, shuffle=False)
```

3.4.3 Model Building

```python
# Define BERT + Bi-LSTM Model
class BertBiLSTM(nn.Module):
    def __init__(self):
        super(BertBiLSTM, self).__init__()
        self.bert = BertModel.from_pretrained('bert-base-uncased')
        self.lstm = nn.LSTM(768, 256, bidirectional=True, batch_first=True)
        self.dropout = nn.Dropout(0.3)
        self.fc = nn.Linear(512, 1)
        self.sigmoid = nn.Sigmoid()

    def forward(self, input_ids, attention_mask):
        with torch.no_grad():  # Freeze BERT during LSTM training
            bert_output = self.bert(input_ids=input_ids, attention_mask=attention_mask)
        lstm_output, _ = self.lstm(bert_output.last_hidden_state)
        lstm_output = self.dropout(lstm_output[:, -1, :])  # Take last output
        output = self.fc(lstm_output)
        return self.sigmoid(output)

# Initialize Model
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = BertBiLSTM().to(device)

# Define Loss and Optimizer
criterion = nn.BCELoss()  # Binary Cross Entropy Loss
optimizer = torch.optim.Adam(model.parameters(), lr=2e-5)
```

```python
# Define BERT + Bi-GRU Model
class BertBiGRU(nn.Module):
    def __init__(self):
        super(BertBiGRU, self).__init__()
        self.bert = BertModel.from_pretrained('bert-base-uncased')
        self.gru = nn.GRU(768, 256, bidirectional=True, batch_first=True)
        self.dropout = nn.Dropout(0.3)
        self.fc = nn.Linear(512, 1)
        self.sigmoid = nn.Sigmoid()

    def forward(self, input_ids, attention_mask):
        with torch.no_grad():  # Freeze BERT during GRU training
            bert_output = self.bert(input_ids=input_ids, attention_mask=attention_mask)
        gru_output, _ = self.gru(bert_output.last_hidden_state)
        gru_output = self.dropout(gru_output[:, -1, :])  # Take last output
        output = self.fc(gru_output)
        return self.sigmoid(output)

# Initialize Model
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = BertBiGRU().to(device)

# Define Loss and Optimizer
criterion = nn.BCELoss()  # Binary Cross Entropy Loss
optimizer = torch.optim.Adam(model.parameters(), lr=2e-5)
```

3.4.4 Model Training

```
def train_model(model, train_loader, val_loader, epochs=5):
    model.train()
    scheduler = torch.optim.lr_scheduler.StepLR(optimizer, step_size=5, gamma=0.5)

    train_losses = []
    train_accuracies = []

    for epoch in range(epochs):
        total_loss = 0
        correct_predictions = 0
        total_predictions = 0

        for batch in train_loader:
            input_ids = batch['input_ids'].to(device)
            attention_mask = batch['attention_mask'].to(device)
            labels = batch['labels'].float().to(device)

            optimizer.zero_grad()
            outputs = model(input_ids, attention_mask).squeeze()

            preds = torch.sigmoid(outputs)
            preds = (preds > 0.5).float()

            loss = criterion(outputs, labels)
            loss.backward()
            optimizer.step()

            correct_predictions += (preds == labels).sum().item()
            total_predictions += labels.size(0)
            total_loss += loss.item()

        avg_loss = total_loss / len(train_loader)
        train_accuracy = (correct_predictions / total_predictions) * 100

        train_losses.append(avg_loss)
        train_accuracies.append(train_accuracy)

        print(f"Epoch {epoch+1}: Loss = {avg_loss:.4f}, Accuracy = {train_accuracy:.2f}%")

        scheduler.step()
    # Plot after training
    plot_training(train_losses, train_accuracies)

# Train the model
train_model(model, train_loader, val_loader, epochs=3)
```

3.4.5 Evaluating Model

```
# Evaluate the Model
def evaluate_model(model, val_loader):
    model.eval()
    predictions, true_labels = [], []

    with torch.no_grad():
        for batch in val_loader:
            input_ids = batch['input_ids'].to(device)
            attention_mask = batch['attention_mask'].to(device)
            labels = batch['labels'].to(device)

            outputs = model(input_ids, attention_mask).squeeze()
            preds = (outputs > 0.5).long().cpu().numpy()
            predictions.extend(preds)
            true_labels.extend(labels.cpu().numpy())

    accuracy = accuracy_score(true_labels, predictions)
    print(f"Validation Accuracy: {accuracy:.4f}")
    print(classification_report(true_labels, predictions))

# Run Evaluation
evaluate_model(model, val_loader)
```

CHAPTER IV

PROPOSED WORK

4.1 ARCHITECTURE

The proposed system is a multi-modal fake news detection web application developed to identify misinformation using both textual and visual data. Unlike conventional systems that focus solely on text analysis, this system integrates Natural Language Processing (NLP) and Computer Vision (CV) techniques to enhance the accuracy and robustness of fake news classification.

The application allows users to submit news content in the form of text, images, or both. For textual analysis, the system employs a hybrid deep learning model that combines BERT and BiLSTM This architecture captures both contextual semantics and sequential dependencies in the text, ensuring a more nuanced understanding of content.

For image analysis, the system incorporates a deep Convolutional Neural Network (CNN) model (e.g., ResNet-101 or similar), which is trained to detect signs of visual manipulation, doctored images, or misleading visual content. These two modes of analysis operate in parallel and their outputs are fused to generate a comprehensive prediction.

The web application is built using the Flask web framework, with a responsive front end for user interaction and a SQLite database for user management and content tracking. The architecture is modular, allowing each component—such as preprocessing, classification, credibility scoring, and storage—to function independently, enabling scalability and future upgrades.

A key innovation in this system is the introduction of a credibility scoring mechanism. This feature maintains a dynamic reliability score for each user based on their posting history. Every submission—whether classified as fake or real—is logged in the database along with the user's unique ID. The credibility score is computed as:

This score reflects the user's overall trustworthiness and is displayed publicly alongside their posts. Users who frequently post fake news will experience a decline in their score, signaling reduced credibility to others and potential moderators. This scoring system fosters a feedback loop that encourages responsible content sharing, making the application not only a fake news detector but also a tool for promoting digital accountability and media literacy.

All predictions are generated in real-time, and the system's design ensures that future improvements—such as model retraining, enhanced scoring algorithms, or migration to cloud databases—can be implemented with minimal disruption.

The ultimate goal is to deliver a scalable, user-friendly, and intelligent platform that aids individuals and communities in identifying and mitigating the spread of fake news.

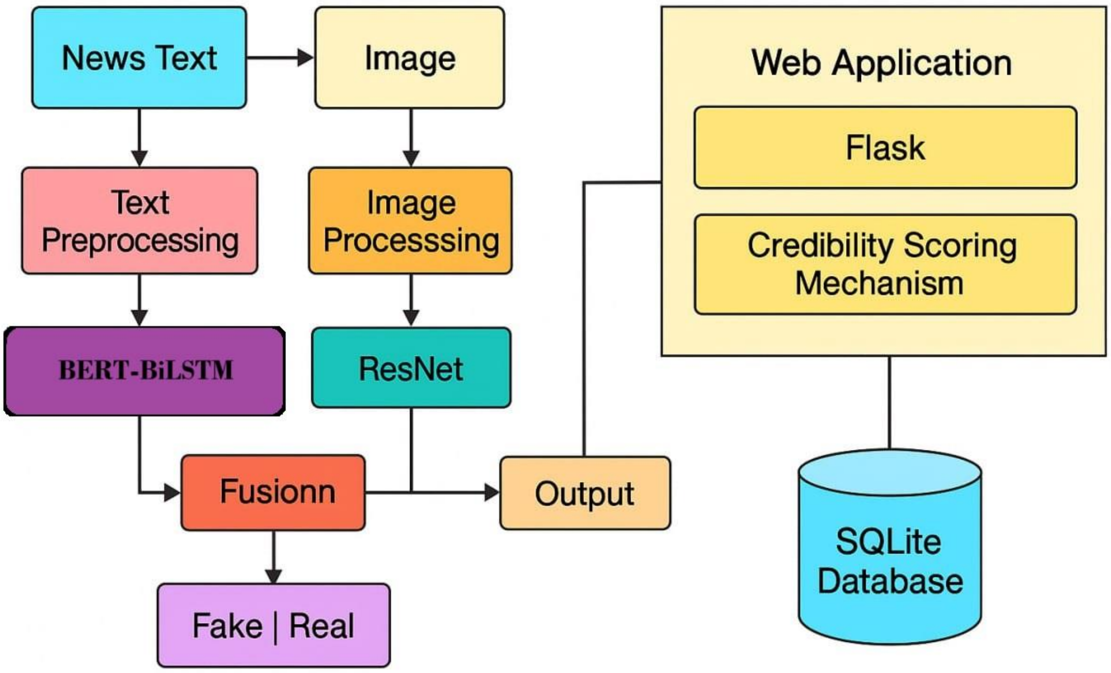


Fig 4.1. Architecture of Proposed Work

4.2 ALGORITHMS AND WORKING

4.2.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained deep learning model developed by Google in 2018. It is based on the Transformer architecture and is designed to understand the context of a word by looking at both its left and right surroundings simultaneously, allowing it to grasp complex language nuances. The model comprises multiple layers of bidirectional Transformer encoders, where each layer uses self-attention mechanisms to capture the interrelationships between words, regardless of their position in the input sequence. BERT is pre-trained on large-scale datasets through tasks such as Masked Language Modeling and Next Sentence Prediction, enabling it to learn deep semantic representations. In the proposed system, BERT is utilized to generate contextual embeddings for the input textual data. These embeddings provide rich semantic features that are passed to subsequent neural layers, forming the foundation for accurate text classification in the fake news detection pipeline.

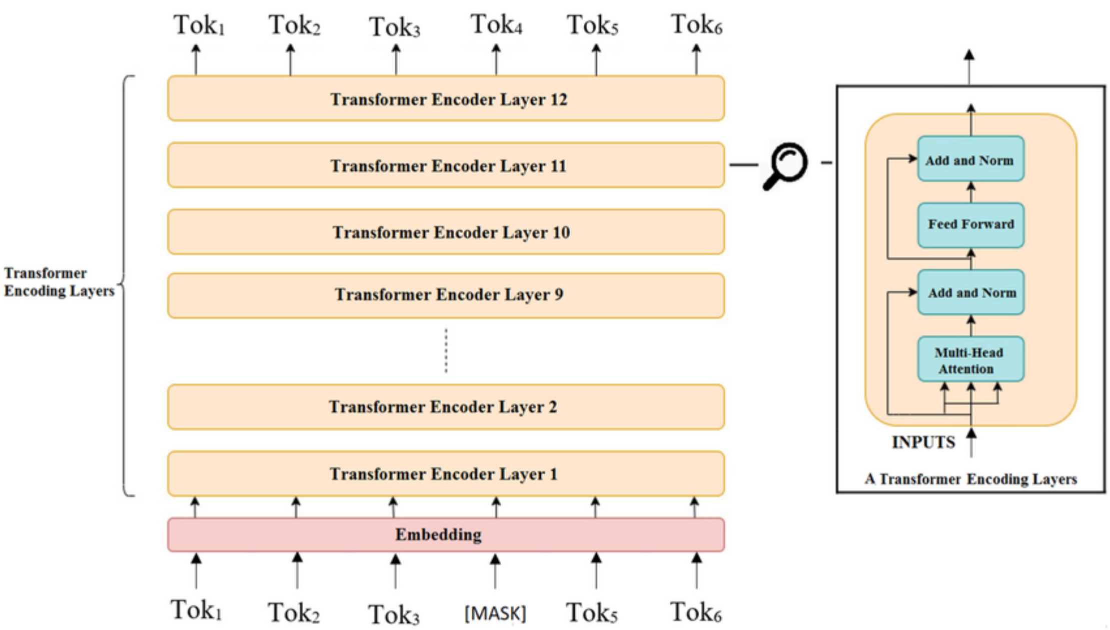


Fig 4.2.1 BERT model Architecture

4.2.2 BiLSTM

BiLSTM is an advanced variant of the traditional Long Short-Term Memory (LSTM) network, designed to effectively capture long-term dependencies in sequential data by processing it in both forward and backward directions. It consists of two parallel LSTM layers—one that reads the input sequence from start to end and another that reads it in reverse. This bidirectional structure enables the model to incorporate both past and future context, thereby improving the understanding of each word in relation to the entire sentence. In the proposed system, after the contextual embeddings are generated by BERT, the BiLSTM layer is applied to further capture the sequential dependencies and sentence structure. This layered combination allows the model to enhance semantic comprehension and improves the accuracy of the text classification task in the fake news detection pipeline.

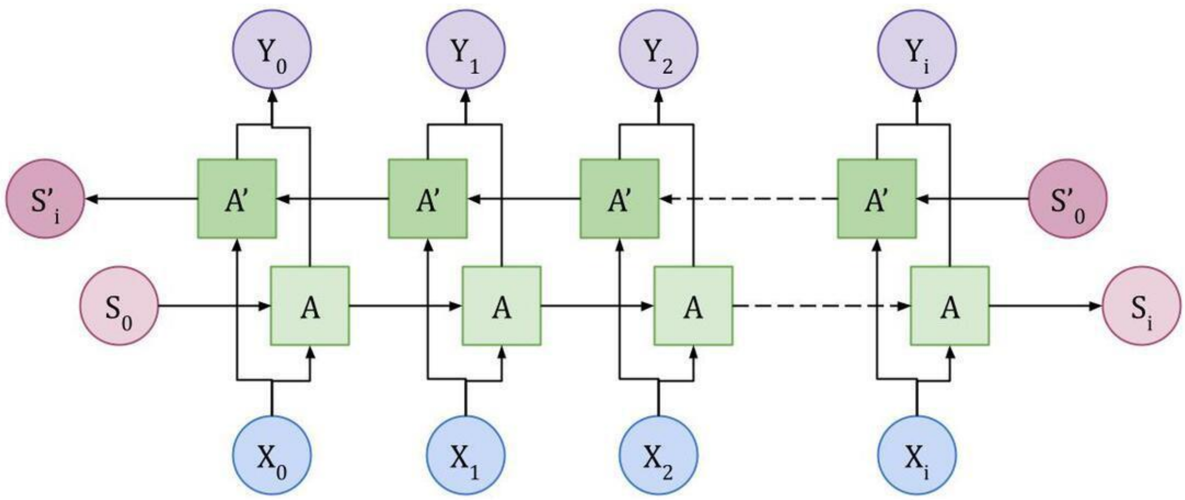


Fig 4.2.2 Bidirectional LSTM layer Architecture

4.2.3 ResNet-101

ResNet-101 is a deep convolutional neural network comprising 101 layers and is specifically designed to overcome the vanishing gradient problem often encountered in very deep architectures. It introduces residual connections—also known as skip connections—that enable gradients to flow through identity mappings, making it easier to train deeper networks without degradation in performance. The architecture consists of multiple residual blocks, each containing convolutional layers and shortcut paths that allow the input of a block to bypass one or more layers and be added directly to the output. This design allows the model to learn identity functions more efficiently and supports deeper network construction. In the proposed system, ResNet-101 is utilized to analyze the image component of the news content. It extracts high-level visual features and representations from the input image, enabling the detection of visual patterns, inconsistencies, or manipulations typically associated with fake or misleading content. The rich visual features extracted are then used for binary classification—predicting whether the image supports genuine or fake news.

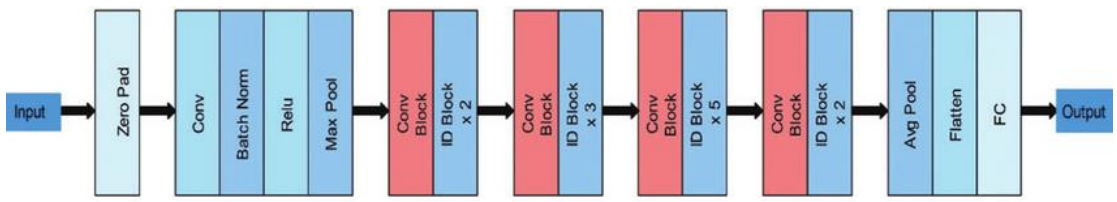


Fig 4.2.3 Architecture of ResNet 101

4.3 MODULE DESCRIPTION

4.3.1 User Authentication Module

The user authentication module enables secure access control within the application. It allows users to register, log in, and log out of the system. During registration, users are required to provide their name, mobile number, email, and password, which are stored in the database. Upon login, the system verifies the provided credentials against the stored values. If authentication is successful, a session is created to manage the user's state throughout their interaction with the platform. This module ensures that only registered users can access the fake news detection functionalities and the content feed.

4.3.2 Text Analysis Module

The text analysis module is responsible for detecting fake news based on the textual content submitted by users. It utilizes a pre-trained deep learning model built with Keras, stored as sentiment_model.h5. The textual input is processed using a tokenizer (tokenizer.pkl) that converts the input sentence into a sequence of tokens, which is then padded to a fixed maximum sequence length. The model predicts the probability of the content being real or fake. If the probability score is below 0.5, the system classifies the input as fake news; otherwise, it is considered real. This module is designed to be lightweight and fast, enabling real-time predictions for short or long textual inputs.

4.3.3 Image Analysis Module

The image analysis module detects fake news by analyzing images uploaded by the user. It employs a convolutional neural network model (model_v1.h5) that has been trained to classify images as either real or fake. The image is first resized to a standard size of 128x128 pixels and converted into a numerical array using Keras image preprocessing utilities. The processed image is then passed through the model to obtain a probability score. If the score exceeds 0.5, the image is labeled as fake; otherwise, it is real. This module works in conjunction with the text analysis module, and either can be used independently or together to make a prediction.

4.3.4 Feed Management Module

The feed management module facilitates the submission and display of content by users. Authenticated users can post text, images, or both, which are then processed by the respective prediction modules. After classification, the results are stored in the database along with the content, image file path, timestamp, prediction outcome, and the user's ID. The feed page displays all posts made by all users in reverse chronological order, showing details such as the content, image (if any), predicted label (real or fake), and associated user information. This module provides an interactive and informative space where users can view and evaluate the credibility of shared content.

4.3.5 Credibility Scoring Module

The credibility scoring module computes and displays the credibility of each user based on their posting history. It calculates the percentage of real posts made by a user out of their total submissions. The formula used is:

Credibility = (Number of Real Posts / Total Posts) × 100

This credibility score is displayed alongside each post, providing additional context to users about the reliability of the content source. By highlighting users with high credibility, this module promotes trustworthiness and encourages responsible sharing behavior.

4.3.6 Database Management Module

This module handles all database-related operations using SQLite. It manages two primary tables: the user table and the feed table. The user table stores user profile information such as ID, name, mobile number, email, and password, while the feed table records every submission with fields for content, image filename, prediction outcome, timestamp, and the ID of the submitting user. The database connection is established through a helper function get_db(), which also ensures that results are returned in a format suitable for easy access and rendering in templates.

4.4 PROPOSED WORK IMPLEMENTATION:

4.4.1 User Registration and Login

```
@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        name = request.form['name']
        mobile = request.form['mobile']
        email = request.form['email']
        password = request.form['password']

        db = get_db()
        user_id = db.execute("SELECT MAX(id) FROM user").fetchone()[0]
        next_id = (user_id + 1) if user_id else 1

        db.execute("INSERT INTO user (id, name, mobile, email, password) VALUES (?, ?, ?, ?, ?)",
                   (next_id, name, mobile, email, password))
        db.commit()
        return redirect('/login')
    return render_template('register.html')

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        db = get_db()
        user = db.execute("SELECT * FROM user WHERE email=? AND password=?", (email, password)).fetchone()
        if user:
            session['user_id'] = user['id']
            return redirect('/feed')
        flash("Invalid credentials")
    return render_template('login.html')
```

4.4.2 Text Prediction

```
def predict_text(text):
    seq = tokenizer.texts_to_sequences([text])
    padded = pad_sequences(seq, maxlen=max_sequence_length)
    pred = text_model.predict(padded)[0][0]
    return "Fake" if pred < 0.5 else "Real"
```

4.4.3 Image Prediction

```python
def predict_image(img_path):
    my_image = load_img(img_path, target_size=(img_height, img_width))
    my_image = img_to_array(my_image)
    my_image = np.expand_dims(my_image, 0)
    out = np.round(image_model.predict(my_image)[0][0], 2)
    return "Fake" if out > 0.5 else "Real"
```

4.4.4 Feed Submission and Storage

```python
@app.route('/feed', methods=['GET', 'POST'])
def feed():
    if 'user_id' not in session:
        return redirect('/login')

    db = get_db()
    prediction_data = None

    if request.method == 'POST':
        content = request.form.get('content', '').strip()
        image = request.files.get('image')

        if not content and (not image or image.filename == ''):
            flash("Please provide either text or an image.")
            return redirect('/feed')

        text_label = None
        image_label = None
        filename = None
        is_fake = 0

        if content:
            text_label = predict_text(content)
            if text_label == "Fake":
                is_fake = 1

        if image and image.filename != '':
            filename = secure_filename(image.filename)
            filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
            image.save(filepath)

            image_label = predict_image(filepath)
            if image_label == "Fake":
                is_fake = 1

        feed_id = db.execute("SELECT MAX(fid) FROM feed").fetchone()[0]
        next_fid = (feed_id + 1) if feed_id else 1

        db.execute(
            "INSERT INTO feed (fid, content, images, fake, trans, uid) VALUES (?, ?, ?, ?, ?, ?)",
            (next_fid, content, filename, is_fake, datetime.now(), session['user_id'])
        )
        db.commit()
```

4.4.5 Displaying Feed and Credibility Score

```python
# Calculate credibility for each user
post_data_with_credibility = []
for post in posts:
    user_id = post['uid']
    user_posts = db.execute(
        "SELECT * FROM feed WHERE uid = ?", (user_id,)
    ).fetchall()
    total_posts = len(user_posts)
    fake_posts = sum(1 for p in user_posts if p['fake'] == 1)
    credibility = round(100 * (total_posts - fake_posts) / total_posts, 2) if total_posts > 0 else 100

    post_data_with_credibility.append({
        **post,
        "credibility": credibility
    })

return render_template('feed.html', prediction=prediction_data, posts=post_data_with_credibility)
```

4.4.6 Database Schema Overview

User Table Schema:

```sql
CREATE TABLE user (
    id INTEGER PRIMARY KEY,
    name TEXT,
    mobile TEXT,
    email TEXT,
    password TEXT
);
```

Feed Table Schema:

```sql
CREATE TABLE feed (
    fid INTEGER PRIMARY KEY,
    content TEXT,
    images TEXT,
    fake INTEGER,
    trans TIMESTAMP,
    uid INTEGER,
    FOREIGN KEY(uid) REFERENCES user(id)
);
```

4.4.7 Model Training for Text Analysis

```python
def train_model(model, train_loader, val_loader, epochs=5):
    model.train()
    scheduler = torch.optim.lr_scheduler.StepLR(optimizer, step_size=5, gamma=0.5)

    train_losses = []
    train_accuracies = []

    for epoch in range(epochs):
        total_loss = 0
        correct_predictions = 0
        total_predictions = 0

        for batch in train_loader:
            input_ids = batch['input_ids'].to(device)
            attention_mask = batch['attention_mask'].to(device)
            labels = batch['labels'].float().to(device)

            optimizer.zero_grad()
            outputs = model(input_ids, attention_mask).squeeze()

            preds = torch.sigmoid(outputs)
            preds = (preds > 0.5).float()

            loss = criterion(outputs, labels)
            loss.backward()
            optimizer.step()

            correct_predictions += (preds == labels).sum().item()
            total_predictions += labels.size(0)
            total_loss += loss.item()

        avg_loss = total_loss / len(train_loader)
        train_accuracy = (correct_predictions / total_predictions) * 100

        train_losses.append(avg_loss)
        train_accuracies.append(train_accuracy)

        print(f"Epoch {epoch+1}: Loss = {avg_loss:.4f}, Accuracy = {train_accuracy:.2f}%")

        scheduler.step()
    # Plot after training
    plot_training(train_losses, train_accuracies)

# Train the model
train_model(model, train_loader, val_loader, epochs=3)
```

4.4.8 Model Training for Image Analysis

```
import tensorflow as tf
from tensorflow.keras.applications import ResNet101
from tensorflow.keras import layers, models

num_classes = 2

base_model = ResNet101(include_top=False, weights='imagenet', input_shape=(img_height, img_width, 3))

base_model.trainable = False

model = models.Sequential([
    tf.keras.layers.InputLayer(input_shape=(img_height, img_width, 3)),
    tf.keras.layers.Rescaling(1./255),

    base_model,

    tf.keras.layers.Conv2D(64, 3, activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D(),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.5),

    tf.keras.layers.Dense(num_classes, activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

earlystop_callback = tf.keras.callbacks.EarlyStopping(monitor='val_accuracy',
                                                      min_delta=0.0001,
                                                      patience=5,
                                                      restore_best_weights=True)

history = model.fit(train_ds,
                    validation_data=val_ds,
                    epochs=10,
                    callbacks=[earlystop_callback])
```

CHAPTER V

SIMULATION RESULTS/EXPERIMENTAL RESULTS

5.1 DATASET DESCRIPTION

5.1.1 Text Dataset – WELFake (Kaggle)

- **Source**: [Kaggle - Fake News Classification Dataset]()
- **Format**: CSV (Comma-Separated Values)
- **Description**: The WELFake dataset consists of news articles labeled as either "fake" or "real." It includes both the text content of the articles and corresponding metadata such as titles and labels. This dataset is particularly suitable for training Natural Language Processing (NLP) models for binary classification tasks.
- **Features**:
    - title: Title of the news article.
    - text: Main body of the article.
    - label: Binary indicator — 0 for Real, 1 for Fake.

- **Purpose in Project**: The dataset is used to train the text-based classification model using a tokenization pipeline followed by deep learning (e.g., LSTM/BiLSTM or BERT).

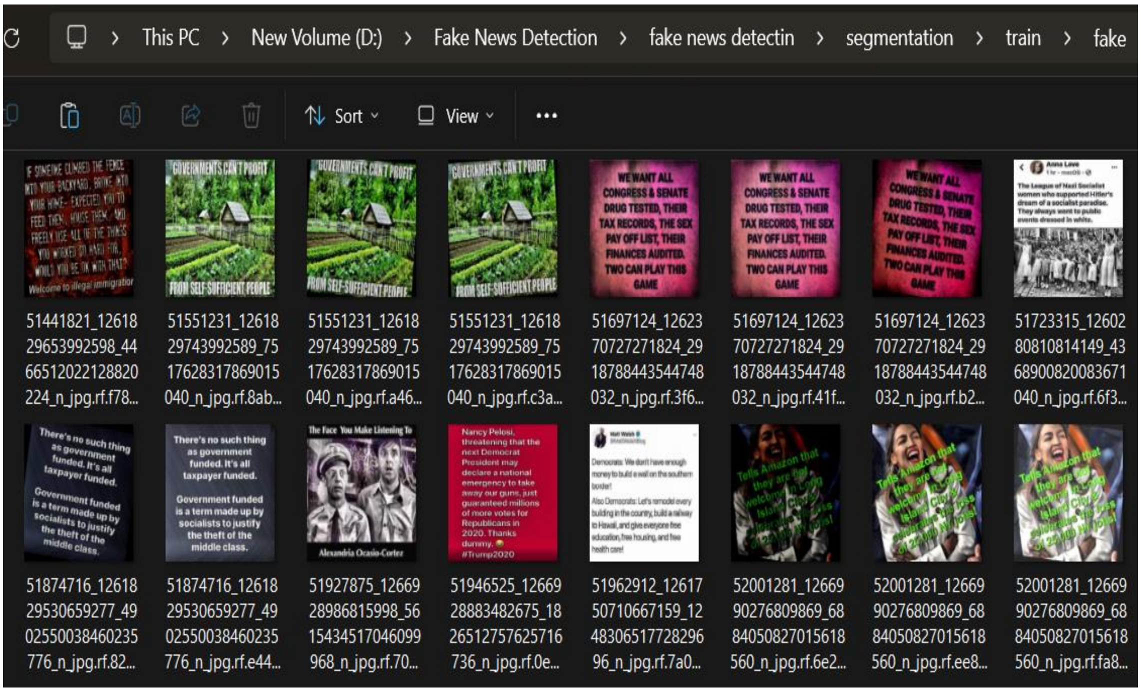
A	B	C	D
	title	text	label
0	LAW ENFORCEMENT ON HIGH ALERT Following Threats Aga	No comment is expected from Barack Obama Members of the #FYF911 or #FukYoFlag and #BlackLivesMatter movements called for th	1
1		Did they post their votes for Hillary already?	1
2	UNBELIEVABLE! OBAMAâ€™S ATTORNEY GENERAL SAYS M	Now, most of the demonstrators gathered last night were exercising their constitutional and protected right to peaceful protest in ord	1
3	Bobby Jindal, raised Hindu, uses story of Christian conversio	A dozen politically active pastors came here for a private dinner Friday night to hear a conversion story unique in the context of	0
4	SATAN 2: Russia unvelis an image of its terrifying new â€˜SU	The RS-28 Sarmat missile, dubbed Satan 2, will replace the SS-18 Flies at 4.3 miles (7km) per sec and with a range of 6,213 miles	1
5	About Time! Christian Group Sues Amazon and SPLC for Des	All we can say on this one is it s about time someone sued the Southern Poverty Law Center!On Tuesday, D. James Kennedy Ministries	1
6	DR BEN CARSON TARGETED BY THE IRS: â€œI never had an	DR. BEN CARSON TELLS THE STORY OF WHAT HAPPENED WHEN HE SPOKE OUT AGAINST OBAMA:	1
7	HOUSE INTEL CHAIR On Trump-Russia Fake Story: â€œNo e		1
8	Sports Bar Owner Bans NFL Gamesâ€¦Will Show Only â€˜Tr	The owner of the Ringling Bar, located south of White Sulphur Springs, is standing behind his Facebook post that criticizes NFL players w	1
9	Latest Pipeline Leak Underscores Dangers Of Dakota Access	FILE â€“ In this Sept. 15, 2005 file photo, the marker that welcomes commuters to Cushing, Okla. is seen. (AP Photo/The Oklahoman,	1
10	GOP Senator Just Smacked Down The Most Punchable Alt-R	The most punchable Alt-Right Nazi on the internet just got a thorough beatdown from Sen. Ben Sasse (R-Neb.) on Twitter during an epi	1
11	May Brexit offer would hurt, cost EU citizens - EU parliamen	BRUSSELS (Reuters) - British Prime Minister Theresa May s offer of settled status for EU residents is flawed and will leave them with f	0
12	Schumer calls on Trump to appoint official to oversee Puert	WASHINGTON (Reuters) - Charles Schumer, the top Democrat in the U.S. Senate, called on President Donald Trump on Sunday to name	0
13	WATCH: HILARIOUS AD Calls Into Question Health Of Aging	After watching this telling video, you ll wonder if instead of working so hard to get back into the White House, Hillary s time would be b	1
14	No Change Expected for ESPN Political Agenda Despite Hug	As more and more sports fans turn off ESPN to protest the networkâ€™s social and political agenda, parent company Disneyâ€™s dec	0
15	Billionaire Odebrecht in Brazil scandal released to house arr	RIO DE JANEIRO/SAO PAULO (Reuters) - Billionaire Marcelo Odebrecht, the highest-profile executive imprisoned in Brazil s massive gra	0
16	BRITISH WOMAN LOSES VIRGINITY To Asylum Seeking Rapi	Europe is likely not going to be a top destination for families with young daughters, and they have no one to blame but themselves for	1
17	U.N. seeks humanitarian pause in Sanaa where streets "bat	GENEVA (Reuters) - The United Nations called on Monday for a humanitarian pause in the Yemeni capital of Sanaa on Tuesday to allov	0
18	MAJOR LIBERAL RAG RELUCTANTLY PUBLISHES Article On F	The Atlantic, a publication that wouldn t know unbiased journalism if it bit them in the a$$ published what appears to be a reluctant pie	1
19	Second judge says Clinton email setup may have been in 'ba	NEW YORK (Reuters) - A second federal judge has taken the rare step of allowing a group suing for records from Hillary Clintonâ€™s tir	0
20	America gives Grand Piano to horse	Wednesday 9 November 2016 by Lucas Wilde America gives Grand Piano to horse	1
21	Hillaryâ€™s crime family: End of days for the U.S.A	Hillaryâ€™s crime family: End of days for the U.S.A Based on the foregoing, SOMEONE got to Comeyâ€”somewhere along the line By	1
22	Sean Spicer Baffles Reporters, Claims Trump Isnâ€™t Respo	On Tuesday, White House Propaganda Minister Sean Spicer once again baffled reporters and other thinking individuals by once again sa	1
23	UNHOLY ALLIANCE: Hillary Clintonâ€™s Saudi Sponsors Sup	21st Century Wire says Amid the tossing and turning of media hit pieces and partisan mud slinging in advance of the US presidential vo	1
24	Even Trumpâ€™s Best Friend Joe Scarborough Canâ€™t Ha	Recently, Joe Scarborough has found himself the subject of, shall we say, criticism because of his insanely pro-Trump bias something t	1
25	BOOM! Danish Government Considers Seizing Migrantâ€™s	Is the European gravy train finally coming to an end?The Danish parliament is considering a bill to seize migrants valuables to pay for tl	1
26	Supreme Court Wonâ€™t Hear Appeal From Texas on Voter	WASHINGTON â€” The Supreme Court rejected on Monday an appeal from Texas officials seeking to restore the stateâ€™s strict vot	0

WELFake_Dataset

Fig 5.1.1 Text Dataset

5.1.2 Image Dataset – Fake News Image Classifier (Roboflow)

- **Source**: [Roboflow - Fake News Image Classifier]
- **Format**: Image dataset with class-labeled directories
- **Description**: This dataset contains **images** labeled as "Fake" or "Real," categorized into separate folders. It has been curated for the task of identifying manipulated or misleading visual content that is often used in fake news. The images vary in content, format, and manipulation techniques, helping the model generalize better.
- **Structure**:
  - `/Fake`: Contains fake news images.
  - `/Real`: Contains real news images.
- **Purpose in Project**:
  Used for training the **image classification model** (e.g., ResNet101), which predicts the likelihood of an image being fake or real based on visual patterns.



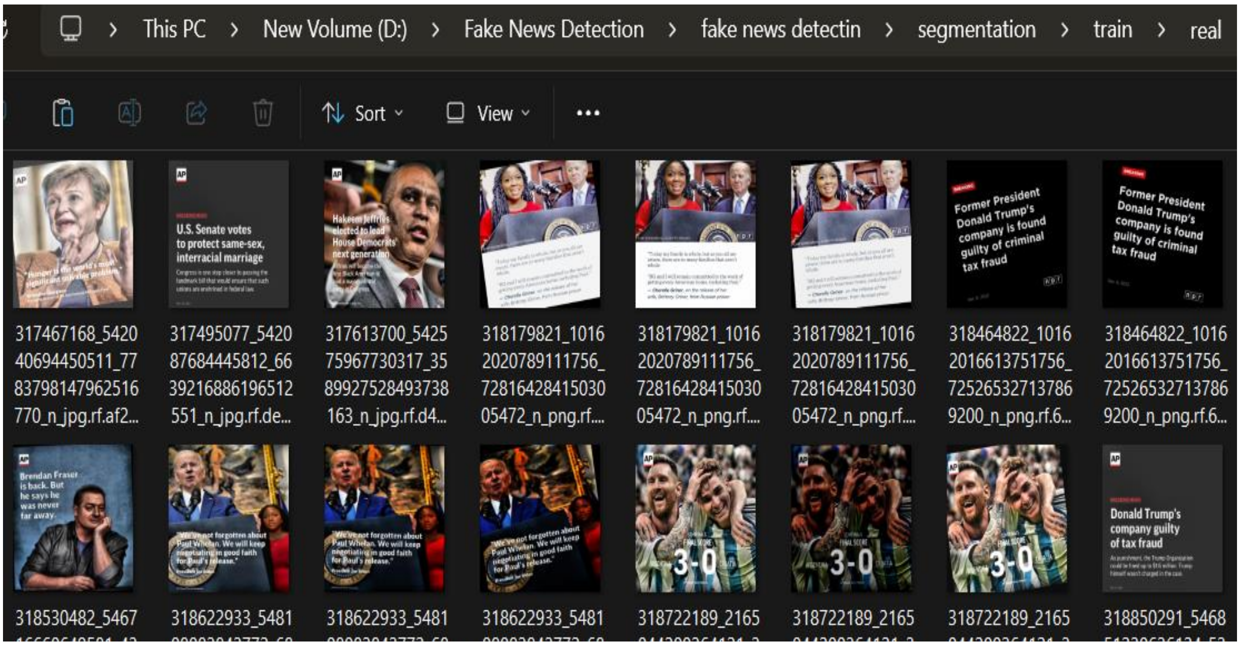


Fig 5.1.1 Image Dataset

5.2 EXPERIMENTAL SETUP

5.2.1 Hardware Requirements:

1. **Processor:** Minimum: Intel(R) Core(TM) i5 – 10th Gen
2. **Memory (RAM):** Minimum: 4 GB or above
3. **Processor Speed:** Minimum: 2.0 GHz
4. **Hard Disk:** Minimum: 256 GB SSD or above

5.2.2 Software Requirements:

1. **Operating System:** Windows 10 or above
2. **Programming Language:** Python 3.7 or later
3. **Front End:** HTML, CSS, JavaScript
4. **Back End:** SQLite
5. **Development Frameworks and Libraries:** TensorFlow

5.3 RESULT ANALYSIS:

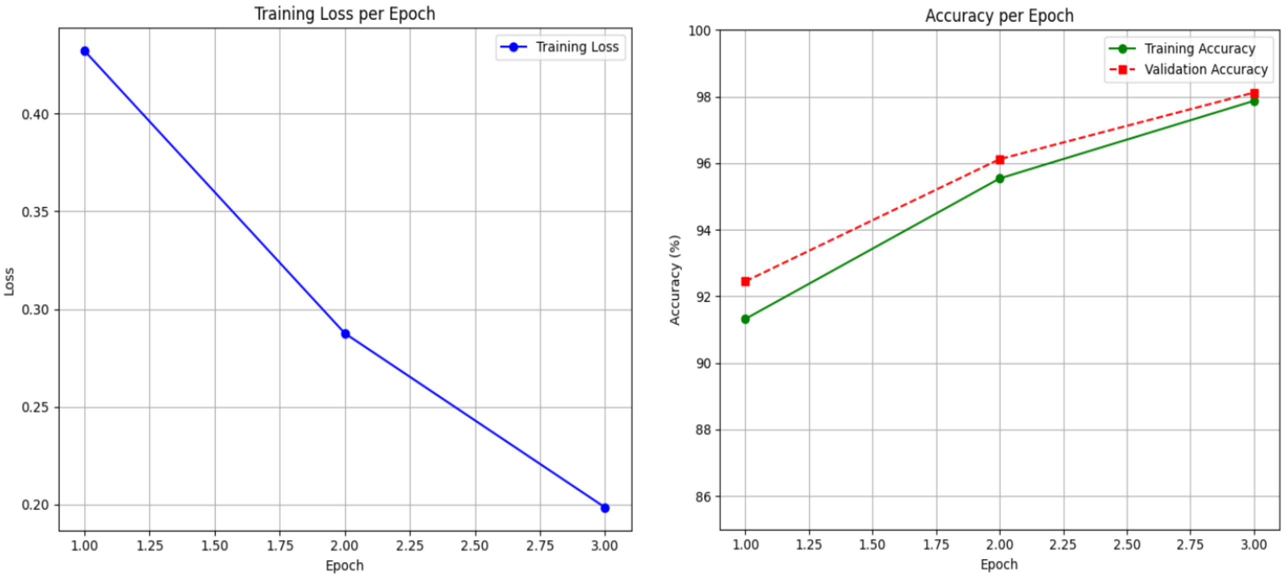


Fig 5.3.1 Training Loss and Training Accuracy for Text data

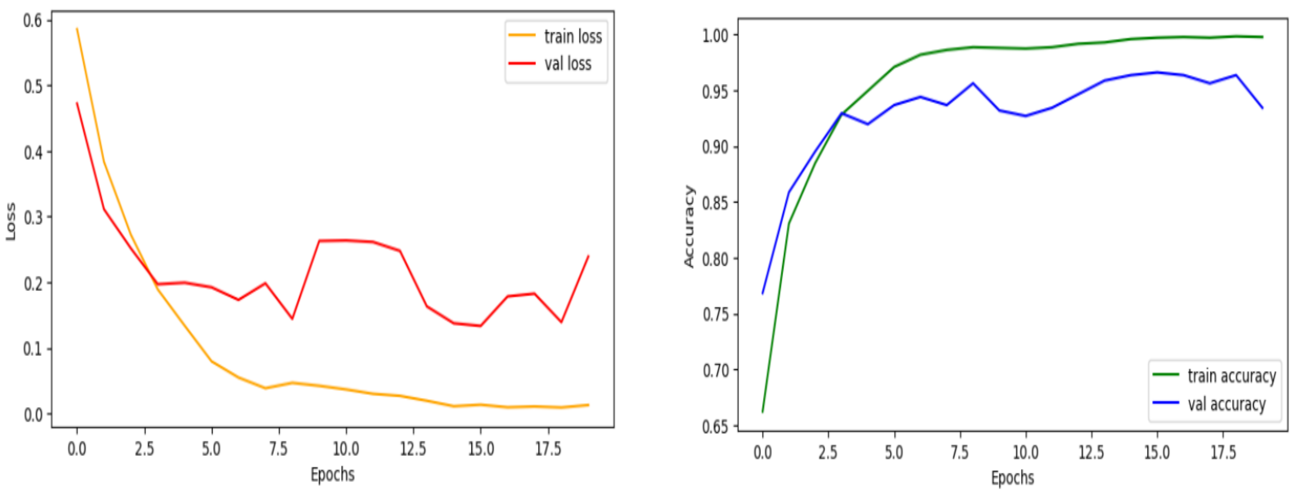


Fig 5.3.2 Training Loss and Training Accuracy for Image data

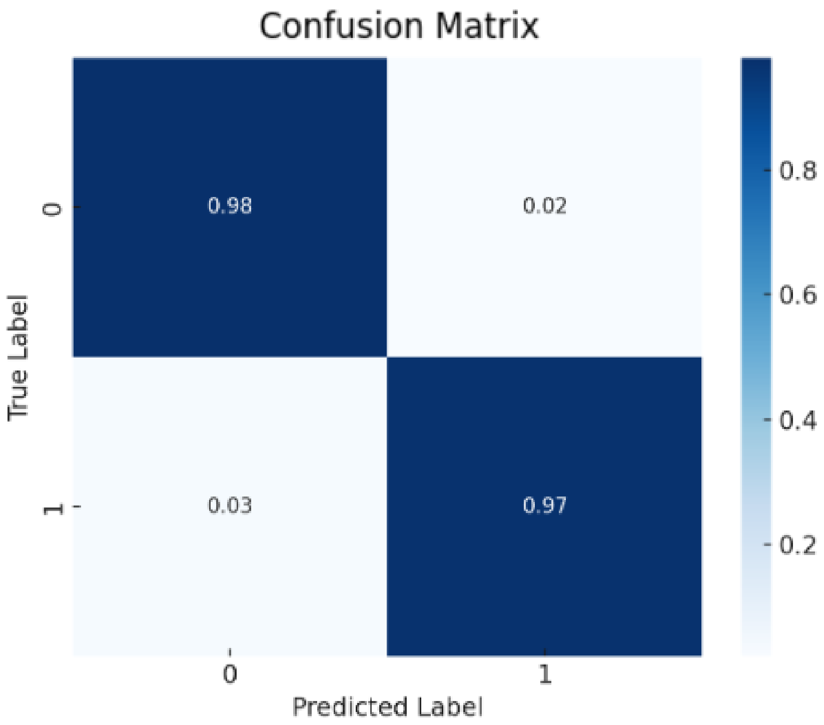


Fig 5.3.3 Confusion Matrix for Text

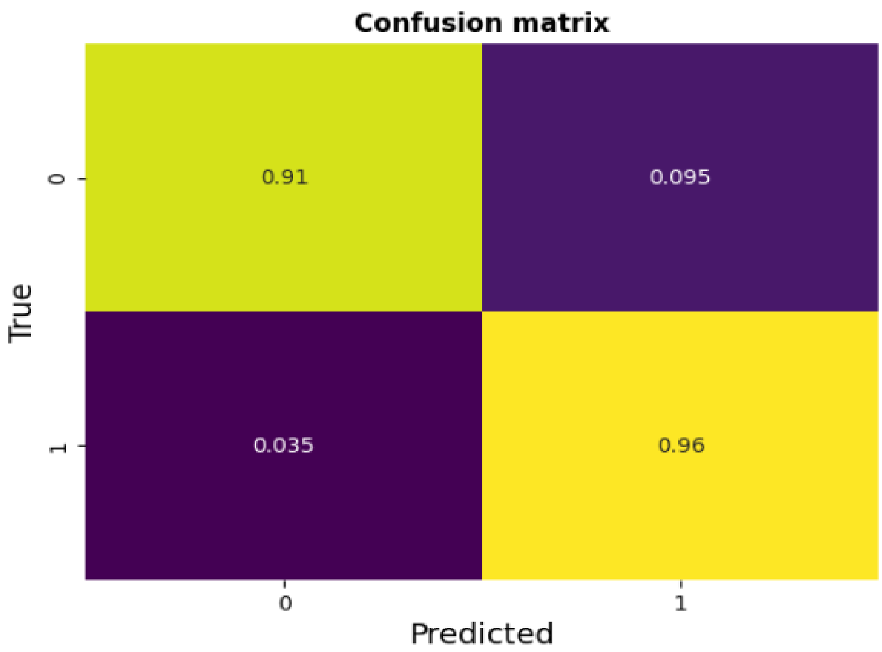


Fig 5.3.4 Confusion Matrix for Image

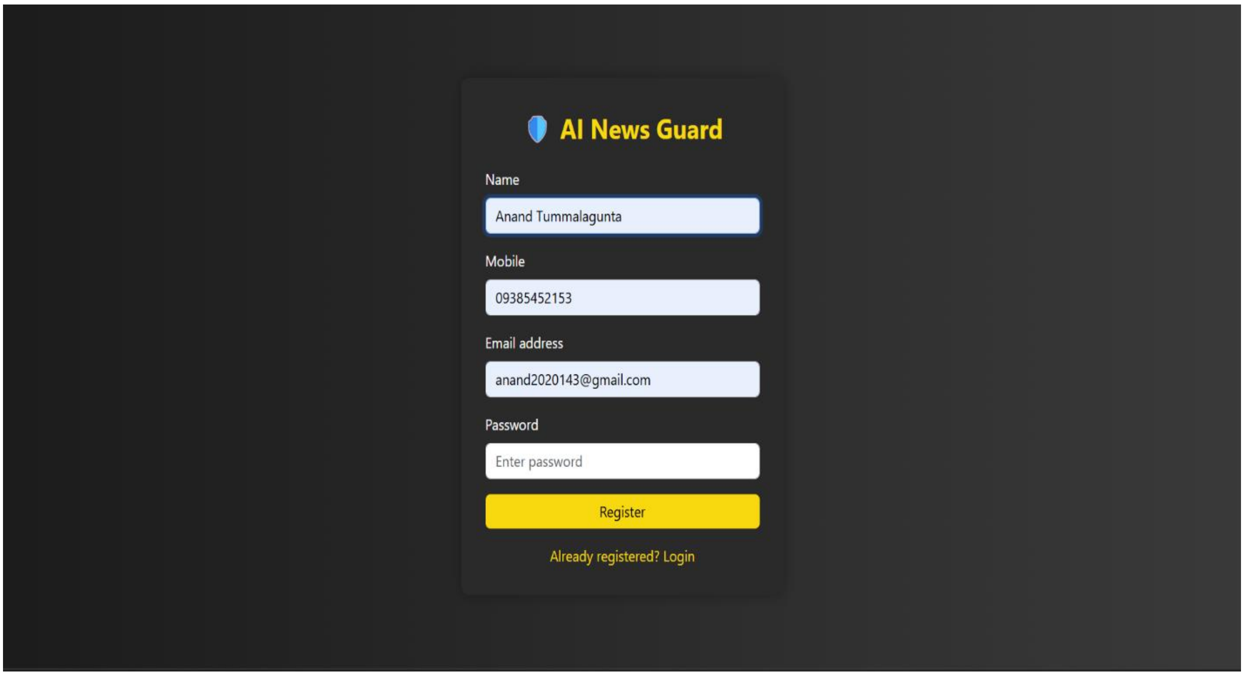


Fig 5.3.5 Registration Page

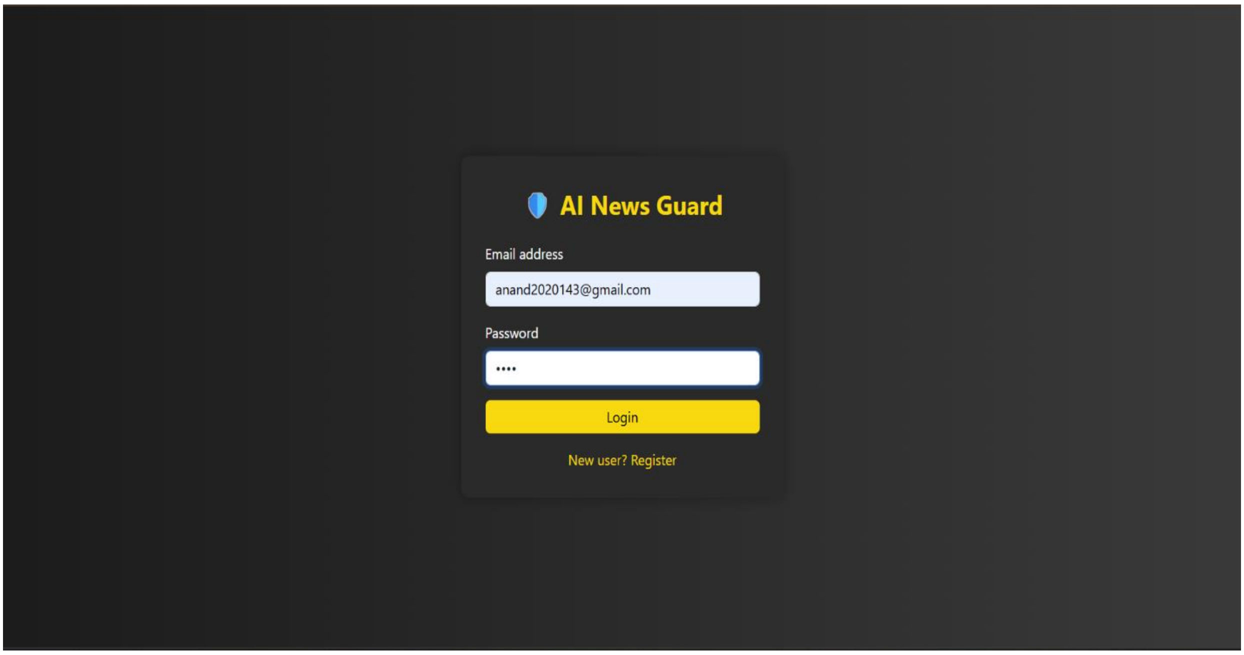


Fig 5.3.6 Login Page

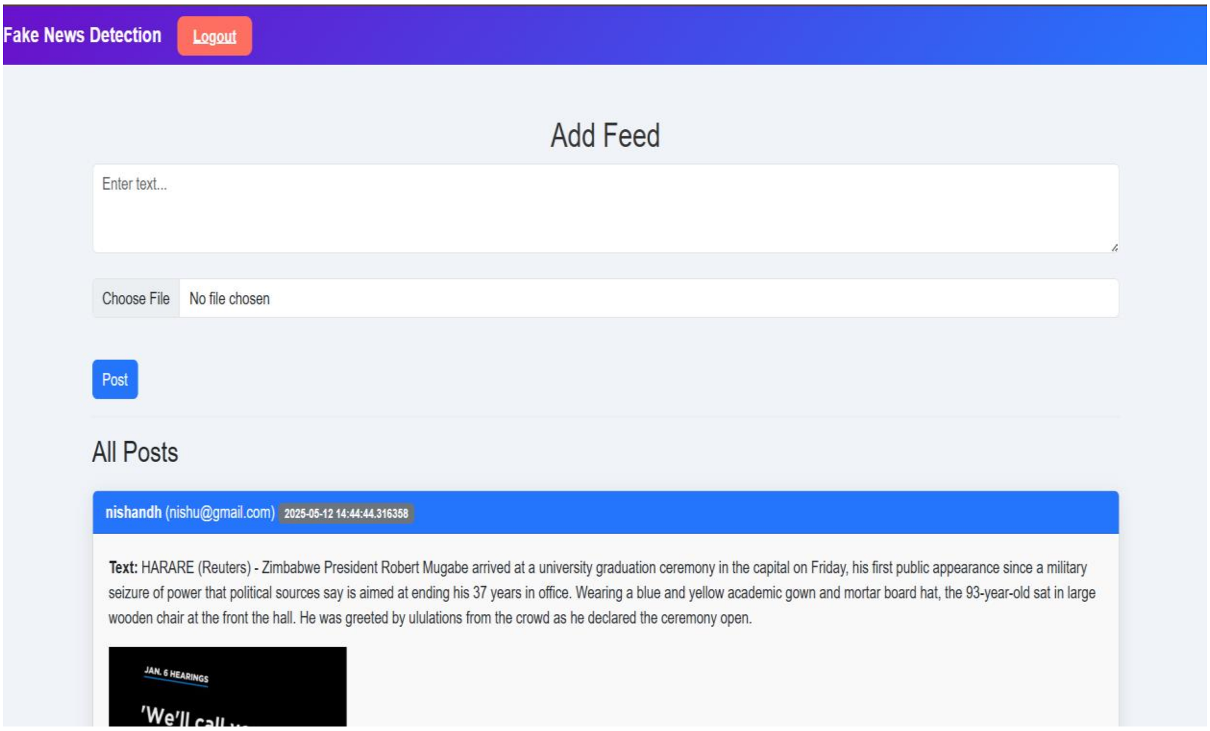


Fig 5.3.7 Home Page

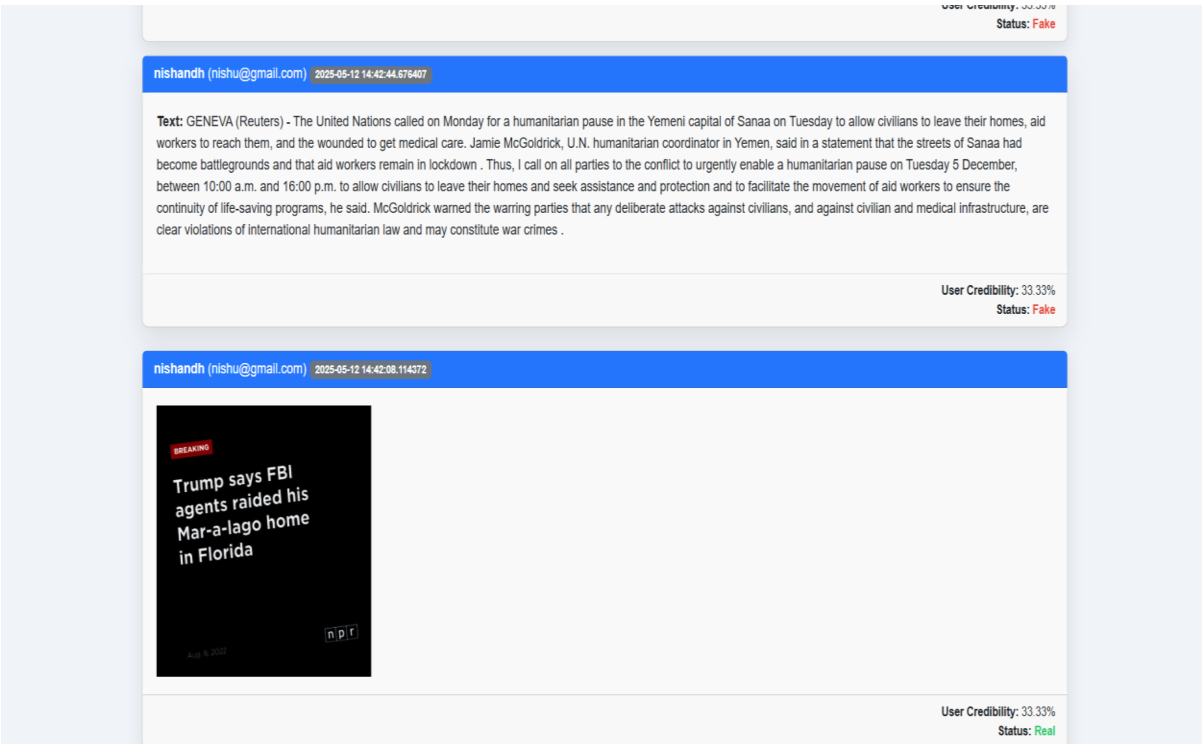
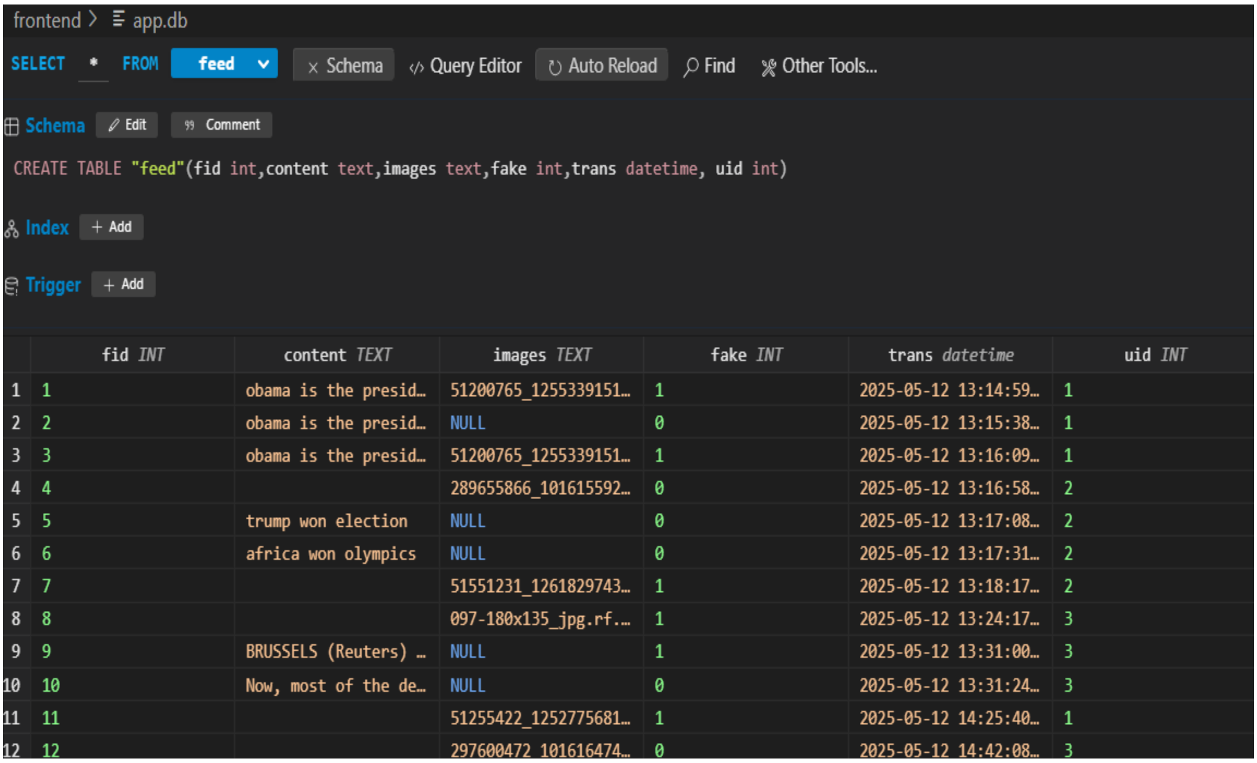


Fig 5.3.8 Feed Page

Fig 5.3.9 Prediction Result



	fid INT	content TEXT	images TEXT	fake INT	trans datetime	uid INT
1	1	obama is the presid…	51200765_1255339151…	1	2025-05-12 13:14:59…	1
2	2	obama is the presid…	NULL	0	2025-05-12 13:15:38…	1
3	3	obama is the presid…	51200765_1255339151…	1	2025-05-12 13:16:09…	1
4	4		289655866_101615592…	0	2025-05-12 13:16:58…	2
5	5	trump won election	NULL	0	2025-05-12 13:17:08…	2
6	6	africa won olympics	NULL	0	2025-05-12 13:17:31…	2
7	7		51551231_1261829743…	1	2025-05-12 13:18:17…	2
8	8		097-180x135_jpg.rf.…	1	2025-05-12 13:24:17…	3
9	9	BRUSSELS (Reuters) …	NULL	1	2025-05-12 13:31:00…	3
10	10	Now, most of the de…	NULL	0	2025-05-12 13:31:24…	3
11	11		51255422_1252775681…	1	2025-05-12 14:25:40…	1
12	12		297600472_101616474…	0	2025-05-12 14:42:08…	3

Fig 5.3.10 Feed Database

CHAPTER VI

CONCLUSION AND FUTURE ENHANCEMENTS

6.1 CONCLUSION

In today's digital era, the widespread dissemination of misinformation poses a serious threat to societies around the world. Fake news has the power to mislead public opinion, influence political decisions, and cause widespread panic. With the increasing use of social media platforms as primary sources of news and information, the need for robust and intelligent systems to detect fake news has become more urgent than ever. This project, titled "**Multimodal Fake News Detection System**", was conceptualized and developed to address this challenge through the integration of artificial intelligence techniques applied to both textual and visual information.

The project proposes and implements a solution that utilizes deep learning-based models to identify fake news using two modalities—text and image. The text-based detection model employs a Bidirectional Long Short-Term Memory (BiLSTM) network, which is adept at understanding the contextual relationships in sentences, thereby enhancing the accuracy of fake news classification. The BiLSTM model was trained on the WELFake dataset, a comprehensive corpus of labeled news articles sourced from reputable and misleading sources. The preprocessing phase included tokenization, stop-word removal, and sequence padding, which prepared the data for robust model training. With the tokenizer saved and reused during prediction, the system ensures consistency and reliability across all inputs.

In parallel, the visual analysis of fake news was conducted using a Convolutional Neural Network (CNN) based on the ResNet architecture. The model was trained on an image dataset curated from Roboflow, which consisted of fake and real news images labeled accurately for supervised learning. The image preprocessing involved resizing, normalization, and augmentation to improve generalization. The CNN was able to capture subtle features and visual patterns that typically distinguish fake images from genuine ones, further improving the credibility of the system.

A key contribution of this project is the integration of both models into a single platform using the Flask web framework. The web application provides an intuitive and user-friendly interface that allows users to register, log in, and submit text and/or images for evaluation. Each submission is processed through the respective trained models, and the result (Fake or Real) is displayed to the user. Additionally, each user's submission history is stored in a SQLite database, and their credibility score is calculated based on the proportion of their fake to real submissions. This approach not only facilitates fake news detection but also encourages responsible content sharing by promoting user accountability.

The application includes modules for secure session handling, error detection, and result visualization. It also maintains scalability by structuring the backend in a modular fashion. Furthermore, the database schema is designed to support long-term storage and retrieval of posts, allowing for extended analysis and audit trails.

The results of the system were promising. The text model achieved good accuracy in classifying news headlines and content, while the image model showed effectiveness in identifying manipulated or misleading visual content. When used together, the multimodal approach provided a higher level of confidence and accuracy compared to using text or image analysis alone.

In conclusion, this project highlights the potential of multimodal deep learning techniques in combating the spread of fake news. It combines technical efficiency with practical usability to deliver a system that is both accurate and user-oriented. The implementation demonstrates how AI can be effectively leveraged to tackle one of the most pressing challenges in modern communication. With further enhancements and real-time deployment, such systems can play a crucial role in ensuring the integrity of digital information and fostering a well-informed society.

6.2 FUTURE ENHANCEMENTS

While the current implementation of the Multimodal Fake News Detection System provides a reliable platform for detecting fake news based on text and image content, there is significant scope for future enhancements to improve accuracy, user experience, scalability, and overall effectiveness.

The following are some potential directions for future work:

1. Incorporation of Video Content Analysis

As misinformation is not limited to text and images, future versions of the system can include support for analyzing video content. By integrating video frame extraction and applying deep learning models (e.g., 3D CNNs or RNNs with visual embeddings), the system can detect misleading visual cues or doctored video content.

2. Multilingual Support

Currently, the system primarily works on English datasets. By incorporating Natural Language Processing (NLP) tools for multiple languages and training on diverse linguistic datasets, the system can cater to a broader user base, especially in multilingual countries.

3. Real-Time News Scraping and Detection

Future updates can enable real-time scraping of news articles from various online platforms and social media, followed by automated analysis and flagging of potential fake content. This can help prevent the viral spread of fake news before it causes significant damage...

4. Blockchain for Content Traceability

Future versions may explore the use of blockchain technology to trace the origin of news content. This can help verify source authenticity and prevent tampering or unauthorized modifications.

REFERENCES

1. F. Al-Quayed, D. Javed, N. Z. Jhanjhi, M. Humayun, and T. S. Alnusairi, "A Hybrid Transformer-Based Model for Optimizing Fake News Detection," in IEEE Access, 2024, doi: 10.1109/ACCESS.2024.3476432.
2. A. Sormeily, S. Dadkhah, X. Zhang and A. A. Ghorbani, "MEFaND: A Multimodel Framework for Early Fake News Detection," in IEEE Transactions on the Computational Social Systems, vol. 11, no. 4, pp. 5337-5352, 2024.
3. R. Kozik, A. Pawlicka, M. Pawlicki, M. Choras, W. Mazurczyk and K. Cabaj, "A Meta-Analysis of State-of-the-Art Automated Fake News Detection Methods," in IEEE Transactions on Computational Social Systems, vol. 11, no. 4, pp. 5219-5234, 2024.
4. A. H. J. Almarashy, M. -R. Feizi-Derakhshi and P. Salehpour, "Enhancing Fake News Detection by Multi-Feature Classification," in IEEE Access, vol. 11, pp. 139601-139613, 2023.
5. K. Zaheer, M. R. Talib, M. K. Hanif and M. U. Sarwar, "A Multi-Kernel Optimized Convolutional Neural Network With Urdu Word Embedding to Detect Fake News," in IEEE Access, vol. 11, pp. 142371-142382, 2023.H. L. Gururaj, H. Lakshmi, B. C. Soundarya, F. Flammini and V. Janhavi, "Machine Learning-Based Approach for Fake News Detection," in Journal of ICT Standardization, vol. 10, no. 4, pp. 509-530, 2022.
6. F. Khan, R. Alturki, G. Srivastava, F. Gazzawe, S. T. U. Shah and S. Mastorakis, "Explainable Detection of Fake News on Social Media Using Pyramidal Co-Attention Network," in IEEE Transactions on Computational Social Systems, vol. 11, no. 4, pp. 4574-4589, 2024.
7. D. Wang, W. Zhang, W. Wu and X. Guo, "Soft-Label for Multi-Domain Fake News Detection," in IEEE Access, vol. 11, pp. 98596-98611, 2023.
8. H. Hammouchi and M. Ghogho, "Evidence-Aware Multilingual Fake News Detection," in IEEE Access, vol. 10, pp. 116808-116818, 2022.
9. M. F. Mridha, A. J. Keya, M. A. Hamid, M. M. Monowar and M. S. Rahman, "A Comprehensive Review on Fake News Detection With Deep Learning," in IEEE Access, vol. 9, pp. 156151-156170, 2021.
10. L. Sun and H. Wang, "Topic-Aware Fake News Detection Based on Heterogeneous Graph," in IEEE Access, vol. 11, pp. 103743-103752, 2023, doi: 10.1109/ACCESS.2023.3318483.