Fake News Detection

Deep Learning

Name	Roll Number
Ishank Kumar	CS22B1043
Thallapally Nimisha	CS22B1082

B.Tech in Computer Science and Engineering
IIITDM Kancheepuram

Contents

1 Objective

The primary goal of this project is to develop and evaluate machine learning and deep learning models that can classify news articles as **fake** or **real**.

2 Dataset

We utilized the ISOT Fake News Dataset, which includes:

- Real news from Reuters (21,417 articles)
- Fake news from unreliable sources (23,481 articles)
- Fields: Title, text, label, date
- Focused on political and world news

Data Splitting

- Training: 27,366 samples
- Validation: 3,910 samples
- Testing: 7,820 samples
- Final dataset size after cleaning: 39,096 articles

3 Preprocessing

1. Merged real and fake datasets with labels (0 = fake, 1 = real)
2. Removed:
    - Duplicates and null entries
    - Articles with ≤ 6 words
    - URLs, special characters, extra whitespace
    - Dataset-specific patterns like datelines and image credits using regex
3. Created two dataset variants:
    (a) With repeated header/footer patterns
    (b) Without repeated patterns
4. Tokenized and padded sequences for deep learning models

4 Models Explored

4.1 Logistic Regression (Baseline)

A simple linear model used as a baseline. It captured basic textual patterns and showed reasonable performance on the clean dataset.

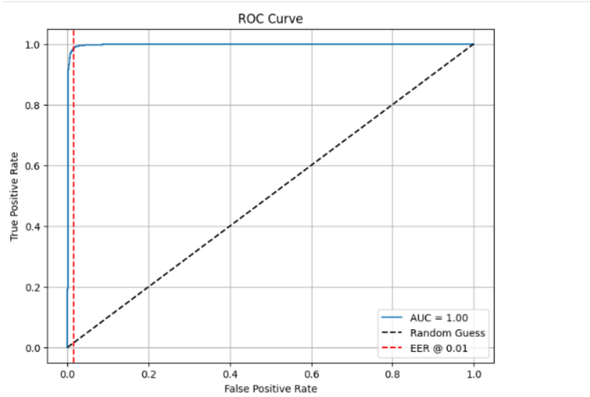

Figure 1: Confusion Matrix: Logistic Regression

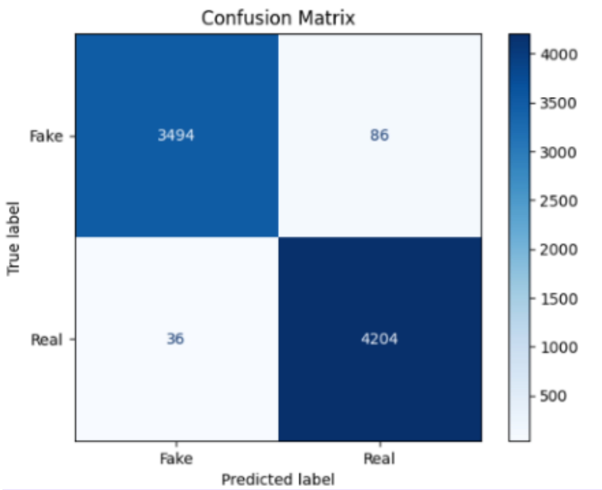

Figure 2: ROC Curve: Logistic Regression

```
Evaluation Metrics:
Accuracy:  0.9844
Precision: 0.9800
AUC:       0.9985
EER:       0.0144

Classification Report:
              precision    recall  f1-score   support

        Fake       0.99      0.98      0.98      3580
        Real       0.98      0.99      0.99      4240

    accuracy                           0.98      7820
   macro avg       0.98      0.98      0.98      7820
weighted avg       0.98      0.98      0.98      7820
```

Figure 3: Classification Report: Logistic Regression

4.2 Bi-LSTM

```
model = Sequential([
    Embedding(input_dim=10000, output_dim=128),
    Bidirectional(LSTM(64, return_sequences=False)),
    Dropout(0.5),
    Dense(32, activation='relu'),
    Dropout(0.3),
    Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['
   accuracy'])
model.summary()
```

Listing 1: Bi-LSTM Model Architecture

- **Embedding Layer** — 128-dimensional embedding vectors
- **Bidirectional LSTM Layer** — 64 units
- **Dropout** — 0.5 and 0.3
- **Dense Layers** — 32 units (ReLU), 1 unit (Sigmoid)
- **Optimizer:** Adam **Loss:** Binary Crossentropy **Metric:** Accuracy

Model Architecture

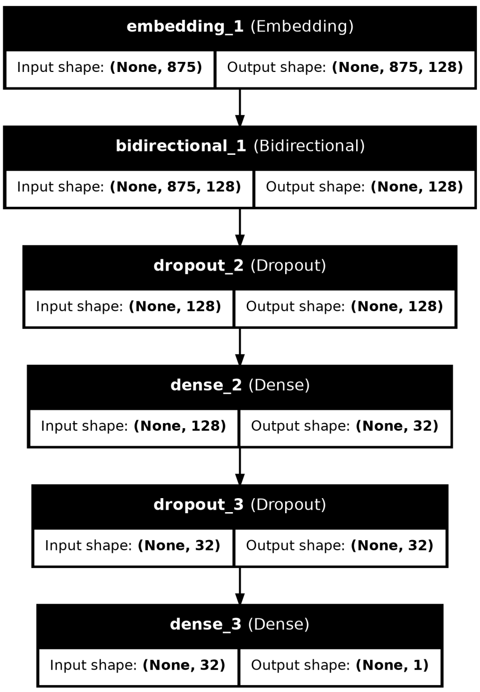

Figure 4: Model Architecture: Bi-LSTM

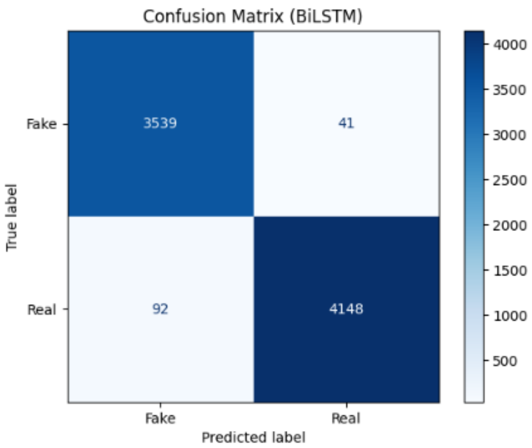

Figure 5: Confusion Matrix: Bi-LSTM

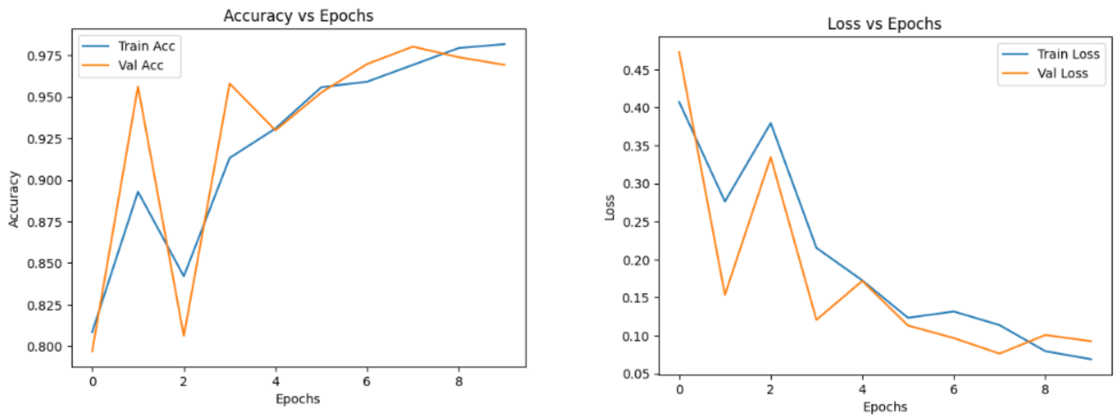

Figure 6: Training and Validation Accuracy / Loss: Bi-LSTM

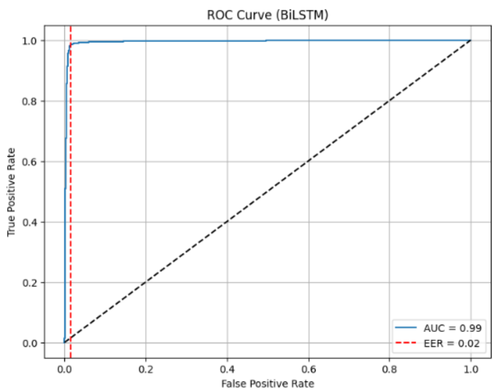

Figure 7: ROC Curve: Bi-LSTM

Classification Report:

	precision	recall	f1-score	support
Fake	0.97	0.99	0.98	3580
Real	0.99	0.98	0.98	4240
accuracy			0.98	7820
macro avg	0.98	0.98	0.98	7820
weighted avg	0.98	0.98	0.98	7820

Figure 8: Classification Report: Bi-LSTM

4.3 Bi-GRU

```
model = Sequential([
    Embedding(input_dim=10000, output_dim=128),
    Bidirectional(GRU(64, return_sequences=False)),
    Dropout(0.5),
    Dense(32, activation='relu'),
    Dropout(0.3),
    Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['
   accuracy'])
model.summary()
```

Listing 2: Bi-GRU Model Architecture

- **Embedding Layer** — 128-dimensional embedding vectors
- **Bidirectional GRU Layer** — 64 units
- **Dropout** — 0.5 and 0.3
- **Dense Layers** — 32 units (ReLU), 1 unit (Sigmoid)
- **Optimizer:** Adam / RMSProp **Loss:** Binary Crossentropy **Metric:** Accuracy

Model Architecture

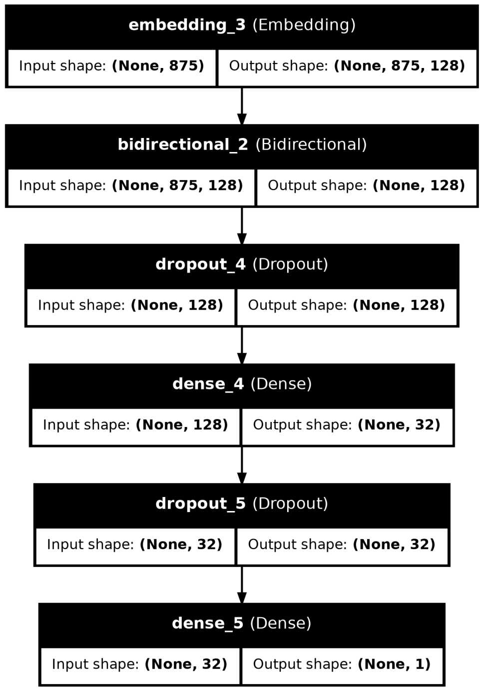

Figure 9: Model Architecture: Bi-GRU

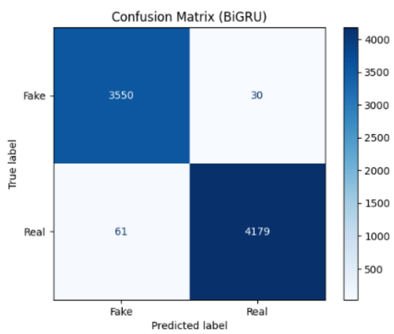

Figure 10: Confusion Matrix: Bi-GRU

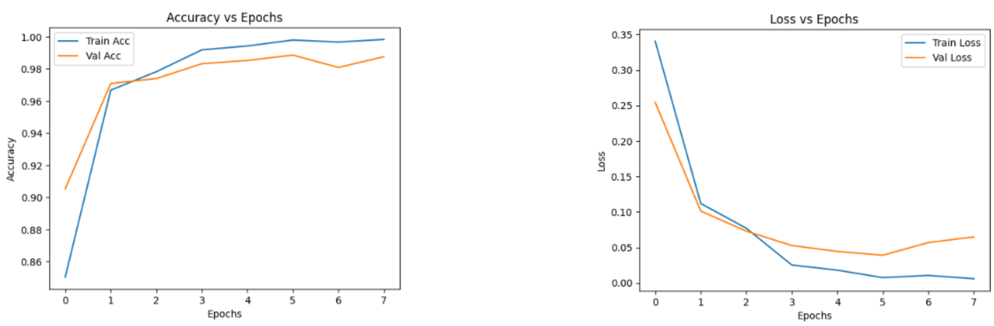

Figure 11: Training and Validation Accuracy / Loss: Bi-GRU

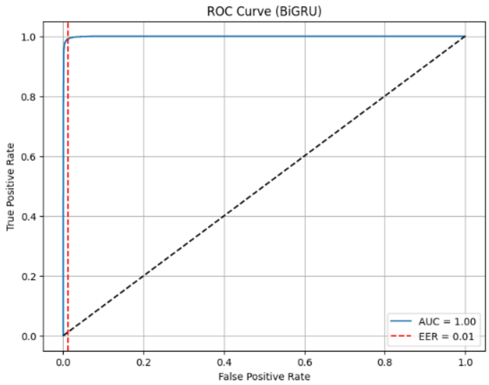

Figure 12: ROC Curve: Bi-GRU

Classification Report:

	precision	recall	f1-score	support
Fake	0.98	0.99	0.99	3580
Real	0.99	0.99	0.99	4240
accuracy			0.99	7820
macro avg	0.99	0.99	0.99	7820
weighted avg	0.99	0.99	0.99	7820

Figure 13: Classification Report: Bi-GRU

4.4 Transformer-based Model

```
inputs = Input(shape=(max_len,))
embedding = Embedding(input_dim=10000, output_dim=128)(inputs)

attention_output = MultiHeadAttention(num_heads=4, key_dim=128)(embedding,
    embedding)
attention_output = LayerNormalization()(attention_output)
attention_output = Dropout(0.3)(attention_output)

pooling_output = GlobalAveragePooling1D()(attention_output)

dense_1 = Dense(64, activation='relu')(pooling_output)
dropout_1 = Dropout(0.5)(dense_1)
dense_2 = Dense(32, activation='relu')(dropout_1)
dropout_2 = Dropout(0.3)(dense_2)
output = Dense(1, activation='sigmoid')(dropout_2)

model = Model(inputs=inputs, outputs=output)
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['
    accuracy'])
model.summary()
```

Listing 3: Transformer Model Architecture

Model Architecture

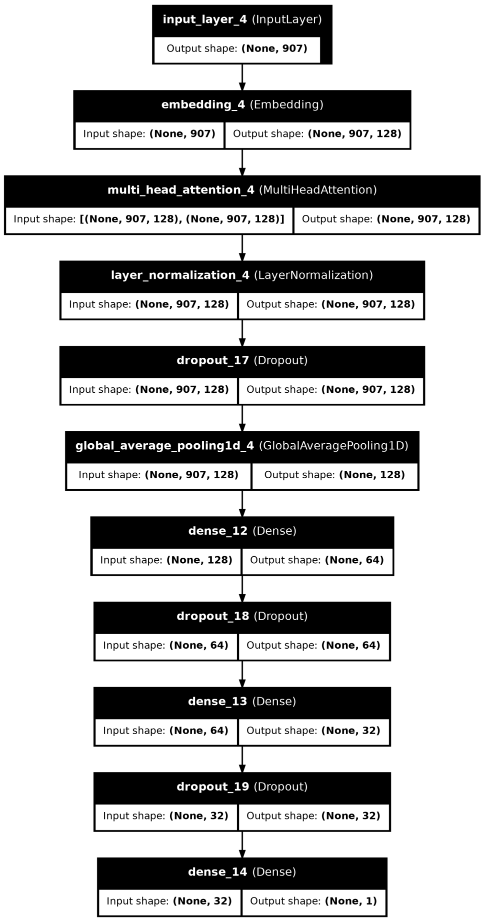

Figure 14: Model Architecture: Transformer

- **Multi-Head Attention** — 4 heads, key dim 128
- **Dense Layers** — 64 and 32 units (ReLU), final 1 unit (Sigmoid)
- **Dropout:** 0.3, 0.5, 0.3 **Pooling:** Global Average Pooling

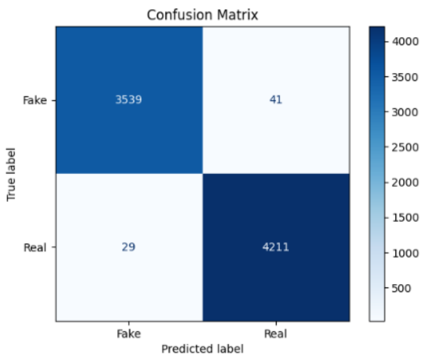

Figure 15: Confusion Matrix: Transformer

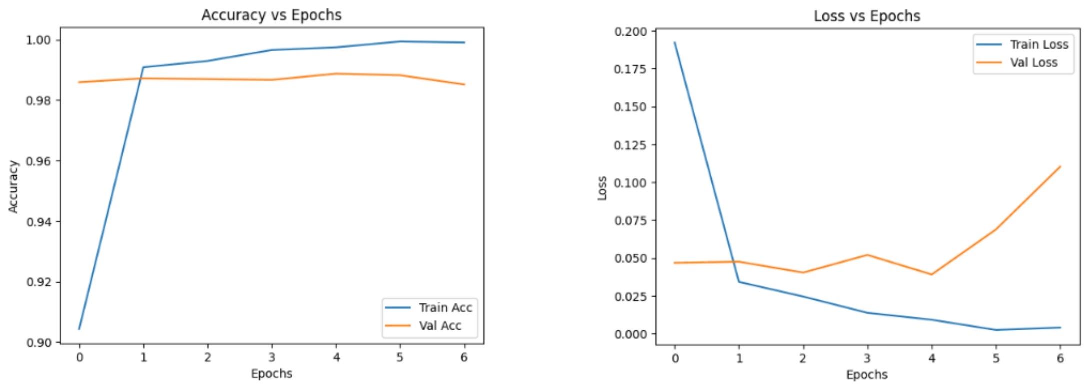

Figure 16: Training and Validation Accuracy / Loss: Transformer

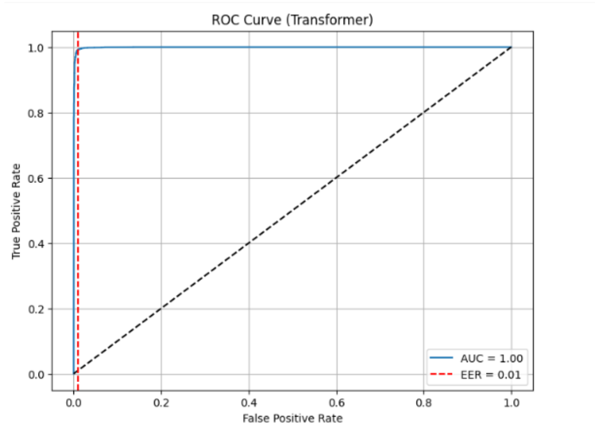

Figure 17: ROC Curve: Transformer

```
Classification Report:
              precision    recall  f1-score   support

        Fake       0.99      0.99      0.99      3580
        Real       0.99      0.99      0.99      4240

    accuracy                           0.99      7820
   macro avg       0.99      0.99      0.99      7820
weighted avg       0.99      0.99      0.99      7820
```

Figure 18: Classification Report: Transformer

5 Evaluation Metrics

- Accuracy, Precision, F1-Score
- AUC (Area Under ROC Curve)
- EER (Equal Error Rate)
- Confusion Matrix

6 Results

Model Performance Summary

Metric	Logistic Reg.	Bi-LSTM	Bi-GRU	Transformer
Accuracy	0.9844	0.9830	0.9884	0.9884
Precision	0.9800	0.9902	0.9929	0.9904
AUC	0.9985	0.9948	0.9994	0.9992
EER	0.0144	0.0160	0.0108	0.0094
F1-Score	0.98	0.98	0.99	0.99

7 Hyperparameter Tuning

We experimented with different batch sizes and optimizers to identify optimal configurations for each model. Evaluation was based on EER and AUC.

Table 1: Hyperparameter Tuning Results

Model	Batch Size / Length	Optimizer	EER	AUC
Bi-LSTM	32	Adam	0.0278	0.9923
	64	Adam	0.0160	0.9948
Bi-GRU	32	Adam	0.0132	0.9986
	64	Adam	0.0108	0.9994
	128	Adam	0.0281	0.9922
	32	RMSProp	0.0158	0.9980
	64	RMSProp	0.0111	0.9983
Transformer	64 (len=907)	Adam	0.0094	0.9992
	64 (len=500)	Adam	0.0108	0.9987
	32	RMSProp	0.0125	0.9993

Confusion Matrix Breakdown

- **Transformer:** TP = 3539, FP = 49, FN = 21, TN = 4211
- **Bi-GRU:** TP = 3550, FP = 30, FN = 61, TN = 4179
- **Bi-LSTM:** TP = 3539, FP = 41, FN = 92, TN = 4148
- **Logistic Regression:** TP = 3494, FP = 86, FN = 36, TN = 4204

8 Conclusion

- **Bi-GRU and Transformer** emerged as the top-performing models with an impressive F1-score of 0.99.
- **Transformer** achieved the lowest Equal Error Rate (EER) and false negatives, making it ideal for minimizing the risk of missing real news.
- **Bi-GRU** recorded the highest AUC and precision, making it particularly suitable for applications where false positives carry a high cost.

- **Logistic Regression** captures surface-level patterns but lacks semantic understanding, leading to good performance on this specific dataset but may not generalize well.
- **Bi-LSTM** showed slightly higher false negatives, making it less optimal for reliable real news detection.

Recommendation: For balanced, high-stakes scenarios, **Transformer** is recommended. In cases where high precision is crucial, such as legal or medical news screening, **Bi-GRU** is the better choice.

This project explores various deep learning models in detecting fake news with high accuracy. By evaluating models such as Bi-GRU, Transformer, Bi-LSTM, and Logistic Regression, we provide a comparative analysis of their strengths and trade-offs based on key performance metrics like F1-score, AUC, precision, and Equal Error Rate (EER).

The findings highlight that while traditional models like Logistic Regression can perform decently on this specific dataset, deep learning models—particularly Transformer and Bi-GRU—excel in terms of robustness and predictive reliability.

— *End of Report* —