FAKE NEWS DETECTION

DEEP LEARNING PROJECT

PRESENTED BY
CS22B1043 Ishank Kumar
CS22B1082 Thallapally Nimisha

OVERVIEW

Objective:

Develop and evaluate models to classify news articles as fake or real.

Dataset:

- Text-based news dataset with labeled samples (fake.csv & real.csv)
- Identified common structural patterns like newswire headers and image credits, and experimented with:
  - Keeping these patterns to observe potential influence on model behavior.
  - Removing them to prevent the model from learning non-semantic, superficial cues.

Data Preprocessing Approaches:

- Standard NLP cleaning (lowercasing, punctuation removal, etc.)
- Tokenization & padding for deep learning models
- Two variations: with and without the common line

OVERVIEW

Models Explored:

- Logistic Regression (baseline)
- BiLSTM (Bidirectional Long Short-Term Memory)
- BiGRU (Bidirectional Gated Recurrent Unit)
- Transformer-based model (Custom Transformer)

Key Goals:

- Compare traditional vs. deep learning approaches
- Evaluate the impact of pre-processing variations
- Identify the most effective model for fake news classification

DATASET SUMMARY

ISOT Fake News Dataset

- Purpose: Classify news articles as real or fake.
- Source:
  - Real news: Collected from Reuters.com (~12,600 articles).
  - Fake news: Gathered from unreliable sources flagged by Politifact and Wikipedia (~12,600 articles).
- Timeframe: Articles primarily from 2016–2017.
- Content Fields: Each article includes title, text, label, and date.
- Key Focus Areas: Majority of articles are on politics and world news.

News Type	Total Articles	Major Subjects
Real	21,417	World, Government, Middle-East
Fake	23,481	US Politics, Left-News, General Politics

Note: Original punctuation and errors in fake news articles were retained to preserve real-world characteristics.

DATA CLEANING AND PRE-PROCESSING

Initial Steps:

- Merged fake.csv and real.csv datasets
- Labelled data: 0 = Fake, 1 = Real
- Identified a common repeated line in both files

Cleaning Steps:

- Removed URLs, special characters, and extra whitespace
- Lowercased all text
- Tokenized and padded sequences for DL models

Two Variants Created:

1. With the header and footer patterns
2. Without the header and footer patterns

DATA CLEANING AND PRE-PROCESSING

1. Initial Cleaning
    - Removed Duplicates
        - Ensured unique articles in the dataset to avoid biased learning.
    - Removed Null Entries
        - Eliminated rows with missing values to maintain consistency and avoid runtime errors during training.

DATA CLEANING AND PRE-PROCESSING

2. Pattern-Based Cleaning

- Start Pattern: Datelines & Attribution Headers
  - Many articles started with newswire headers like:
  - "WASHINGTON (Reuters) – ..."
  - These do not contribute to content semantics and were removed.
- End Pattern: Image References
  - Articles often ended with metadata or image credits such as:
  - "...featured image via Shutterstock."
  - Regex used:
  - r'([^.]*?(images|image|capture|featured|via)[^.]*\.)\s*$'
  - Last sentence removed if it matched this pattern.

- Issue:
  - These patterns were dataset-specific and did not carry semantic information.
  - Risk: Model learns these superficial cues instead of meaningful content.

DATA CLEANING AND PRE-PROCESSING

3. Removed Very Short Articles
   - Dropped texts with length ≤ 6 words
     - These were often headlines, broken entries, or noise.
     - Such short texts lack context and meaningful features for training an LSTM model.

4. Final Output
   - Cleaned dataset with semantically rich and unique text.
   - Prevented the model from overfitting on structural noise (like headers or captions).
   - Ensures model focuses on actual content rather than formatting artifacts.
   - Improved the model's ability to learn from semantic content, not superficial cues.

DATASET SPLIT OVERVIEW

Split	Count	True	Fake	Percentage
Train	27366	14836	12530	70
Test	7820	4240	3580	20
Validation	3910	2120	1790	10

Total Samples:
44898 (True - 21417 , Fake - 23481)
After dropping NAN and duplicates:
39105
After removing common patterns:
39105
Removes lines with <= 6 words:
39096
Final dataset size:
39096 (True - 21196 , Fake - 17900)

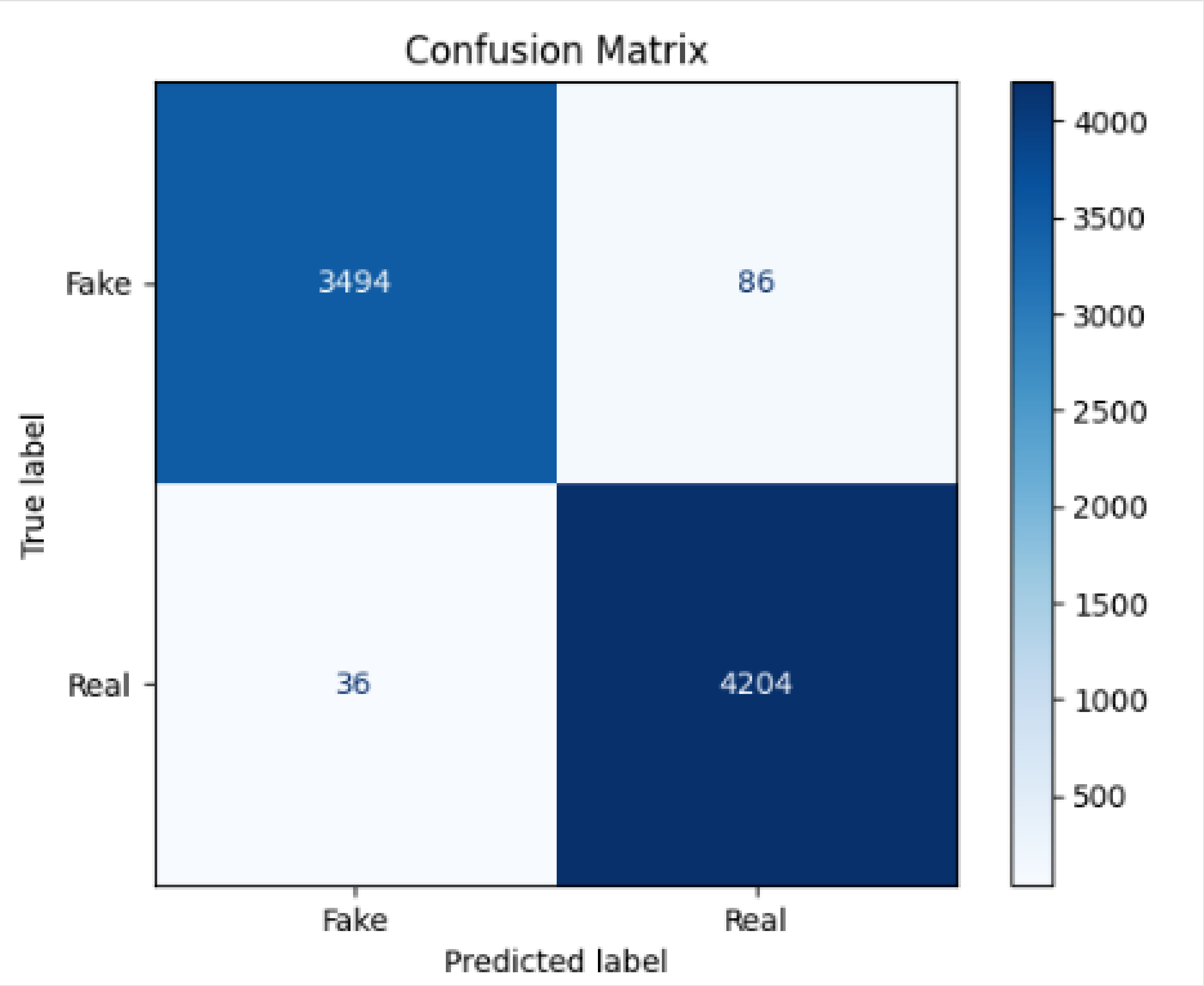
RESULT

LOGISTIC REGRESSION

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

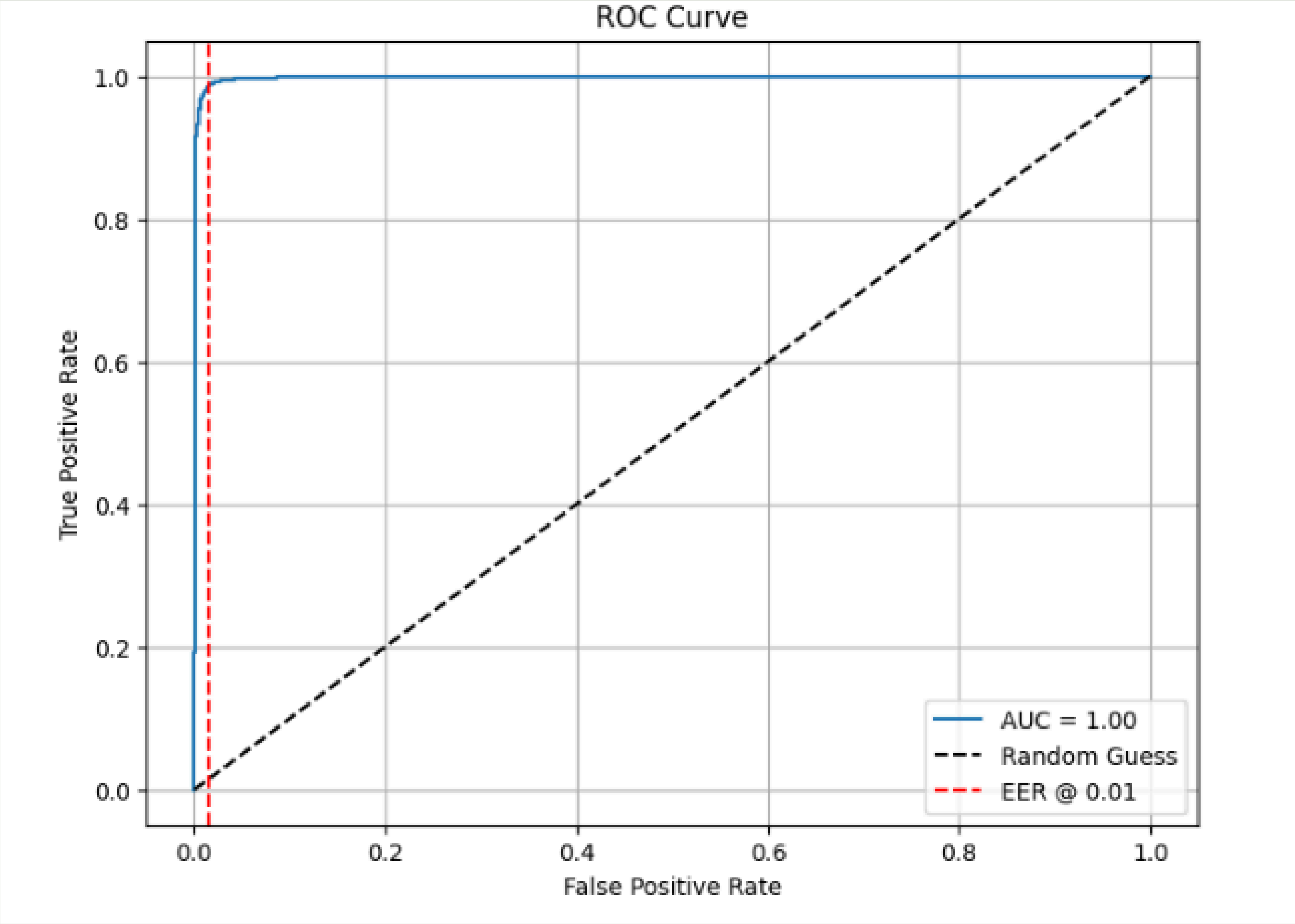
Classification Report:

	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

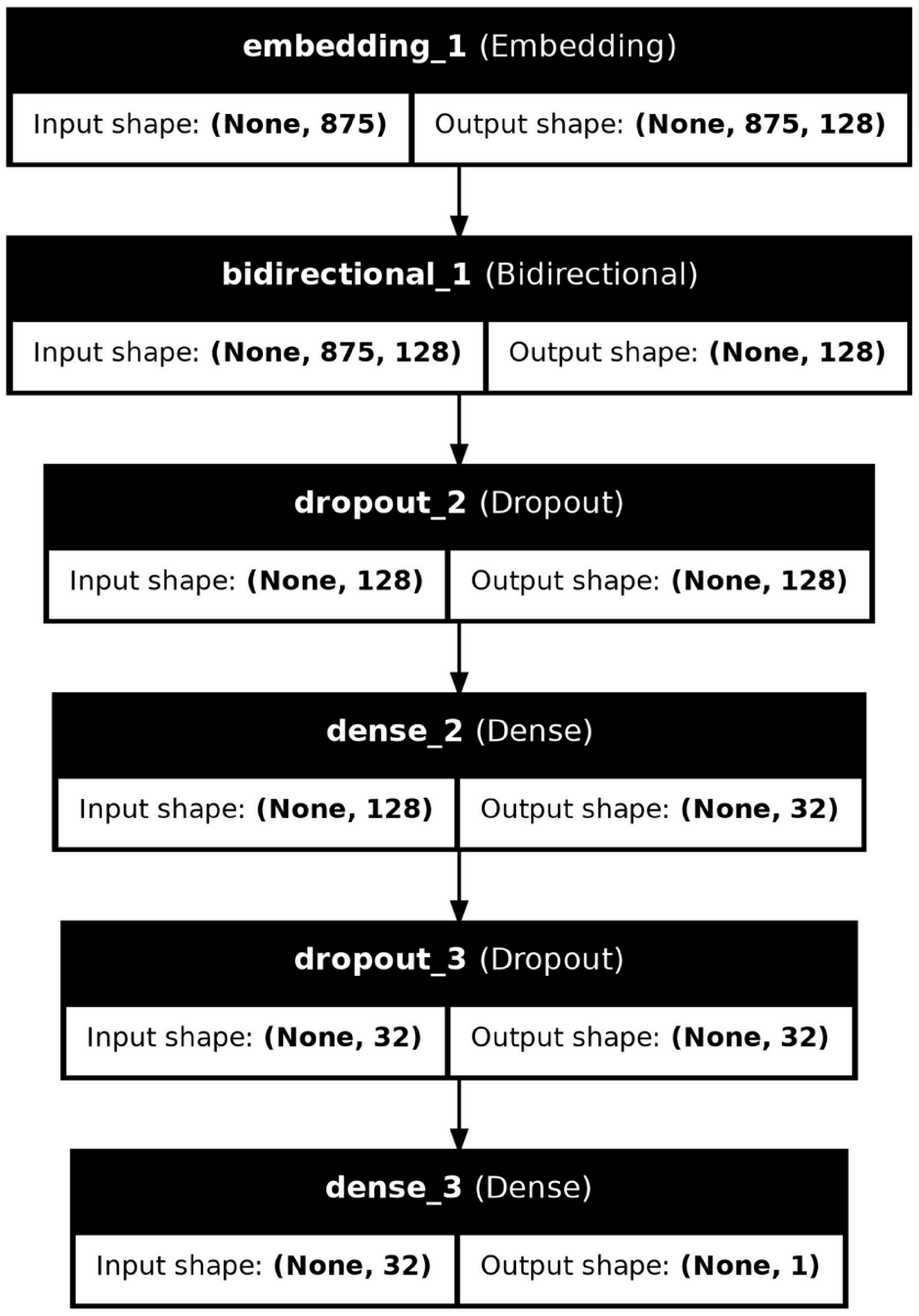


*False higher performance (Dataset specific)

RESULT
LOGISTIC REGRESSION



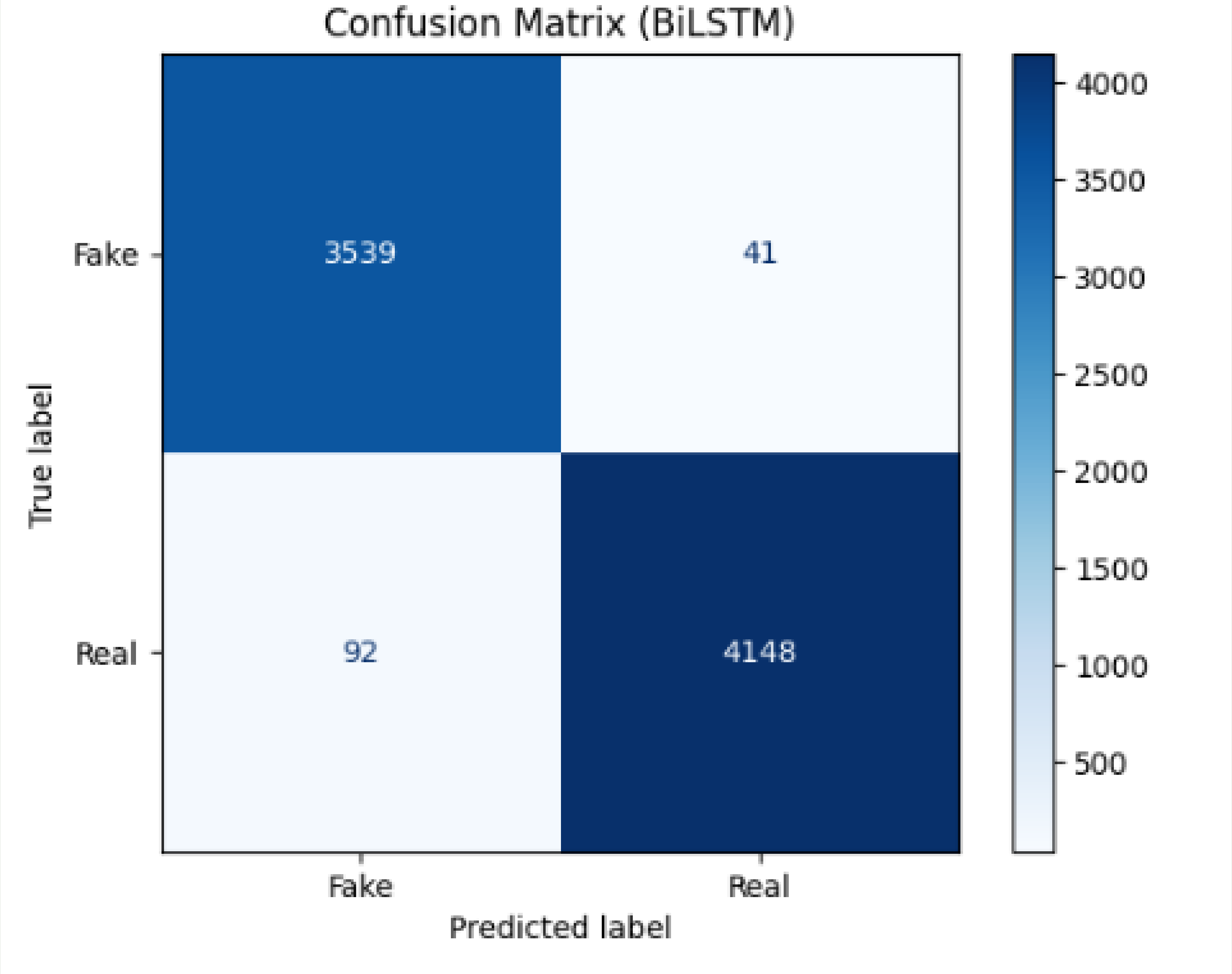
BILSTM ARCHITECTURE



RESULT

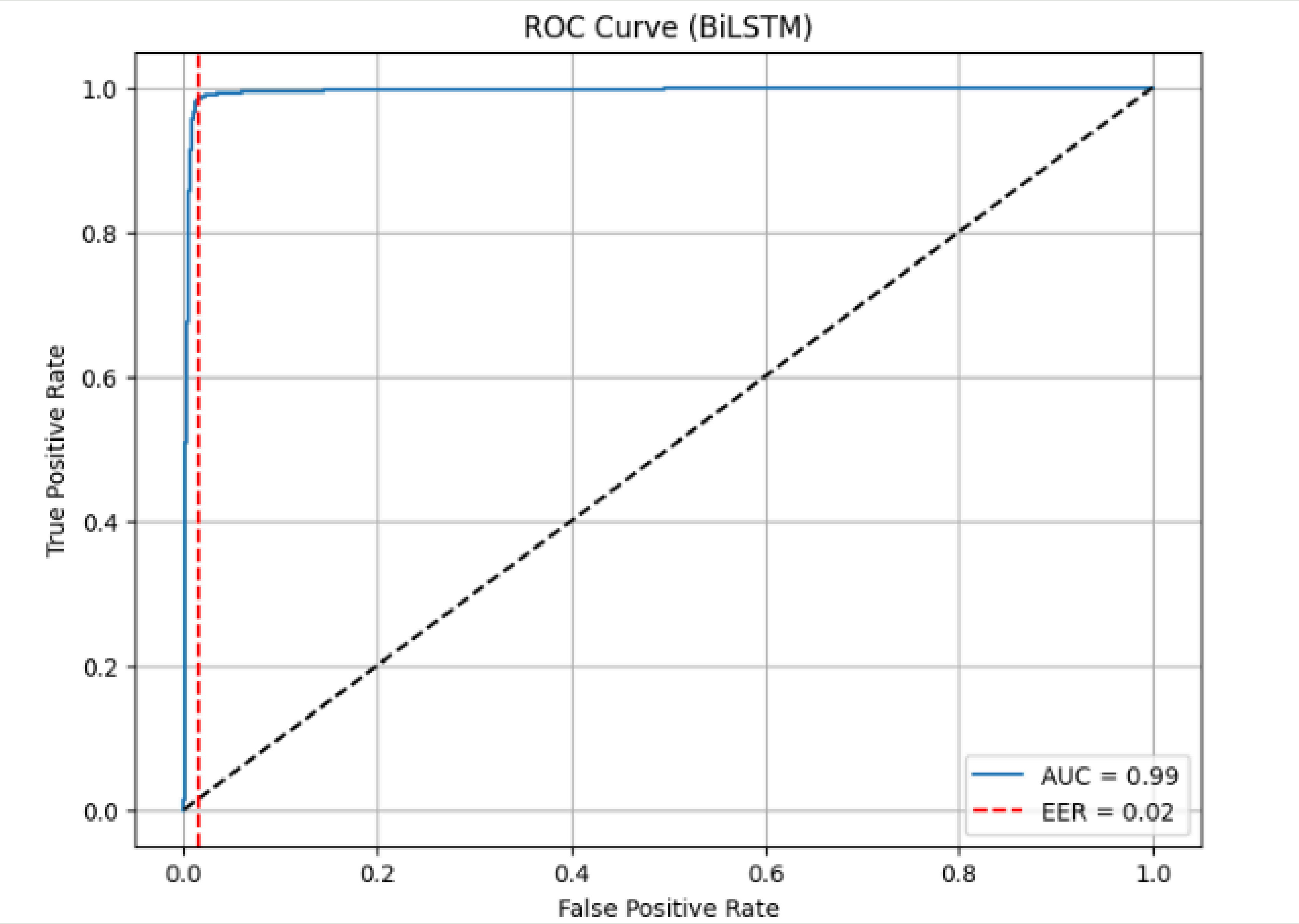
BI-LSTM

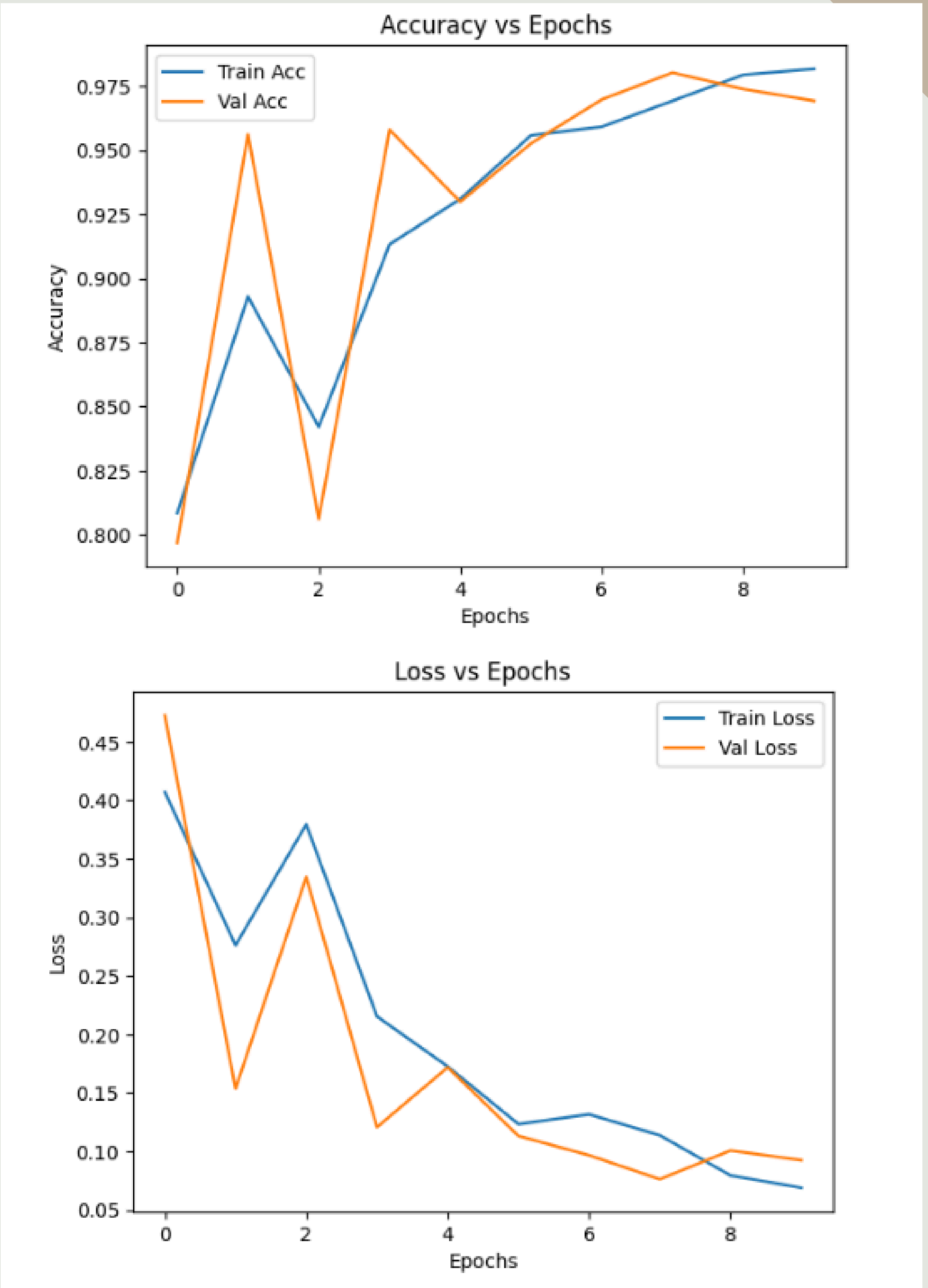
```
Evaluation Metrics:
Accuracy:  0.9830
Precision: 0.9902
AUC:       0.9948
EER:       0.0160

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
```

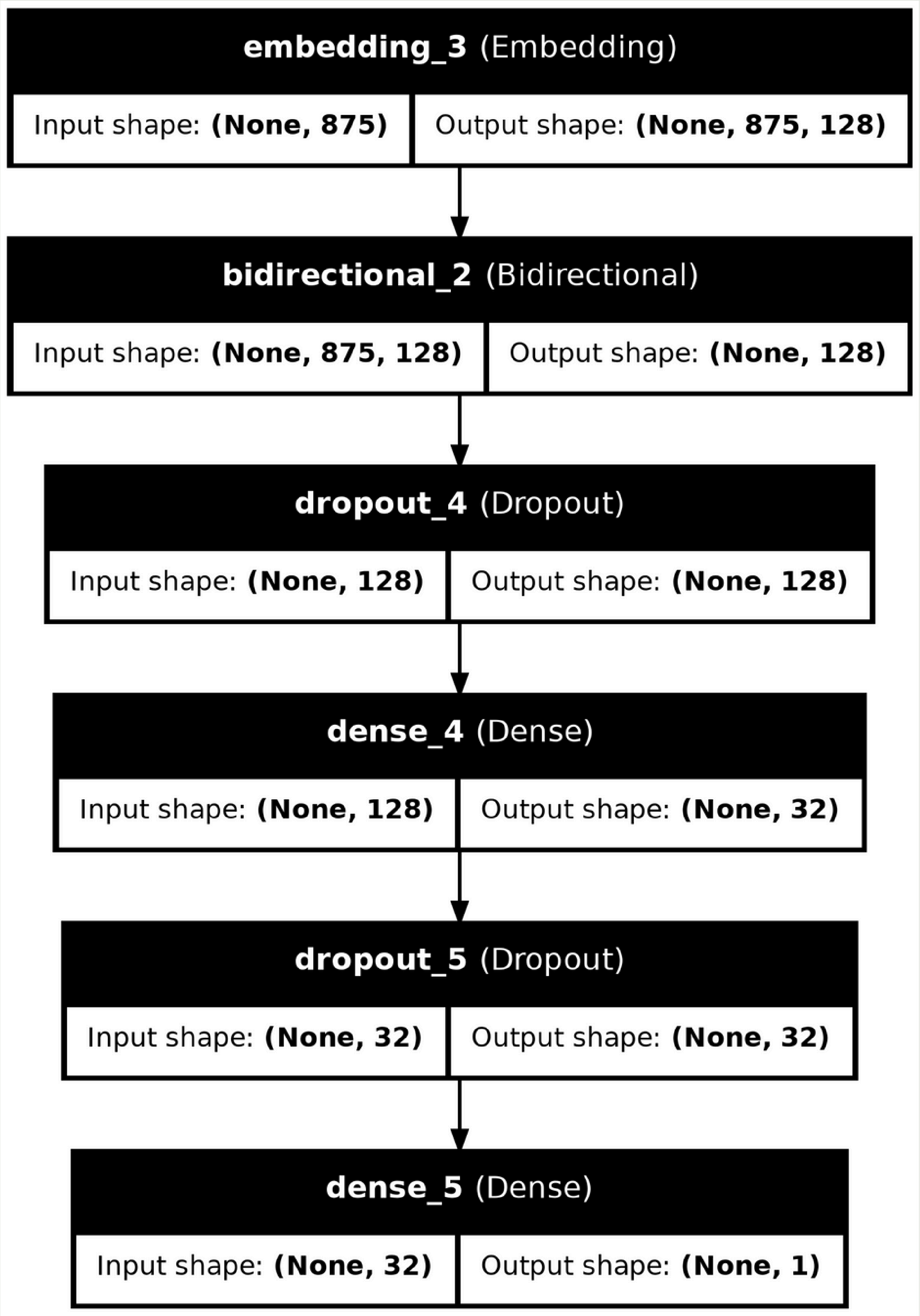


RESULT

BI-LSTM







BIGRU ARCHITECTURE

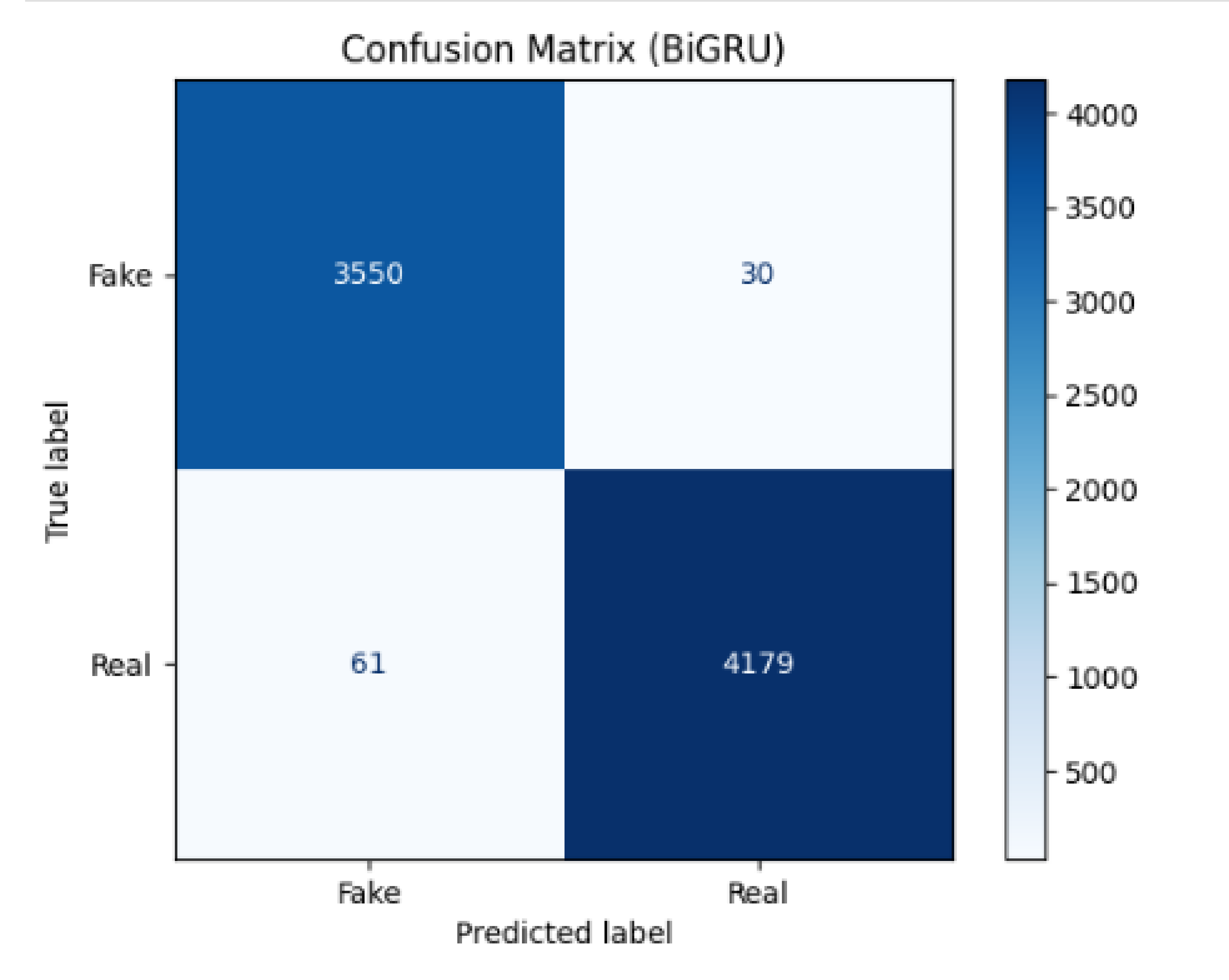
RESULT

BI-GRU

Evaluation Metrics:
Accuracy: 0.9884
Precision: 0.9929
AUC: 0.9994
EER: 0.0108

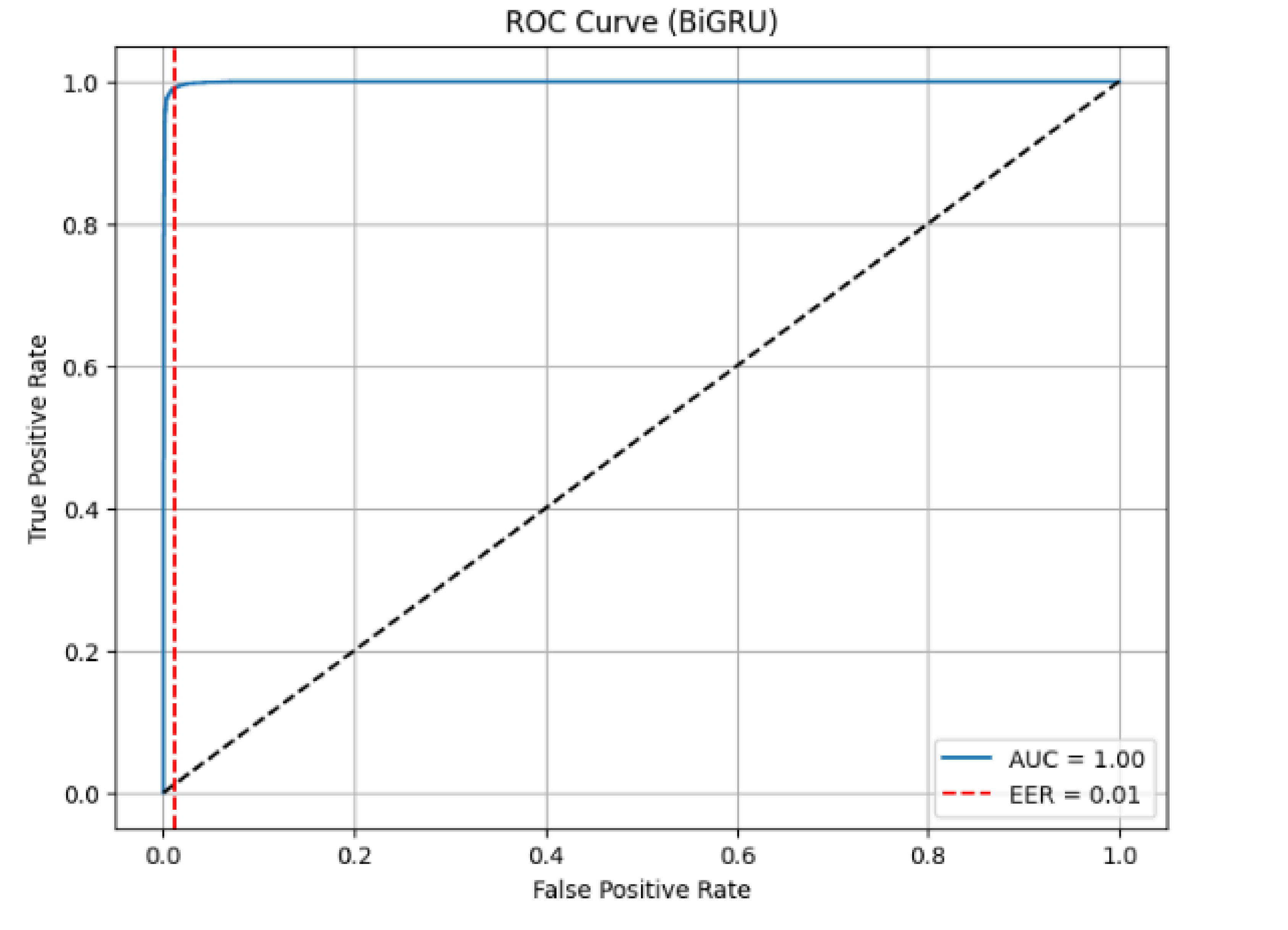
Classification Report:

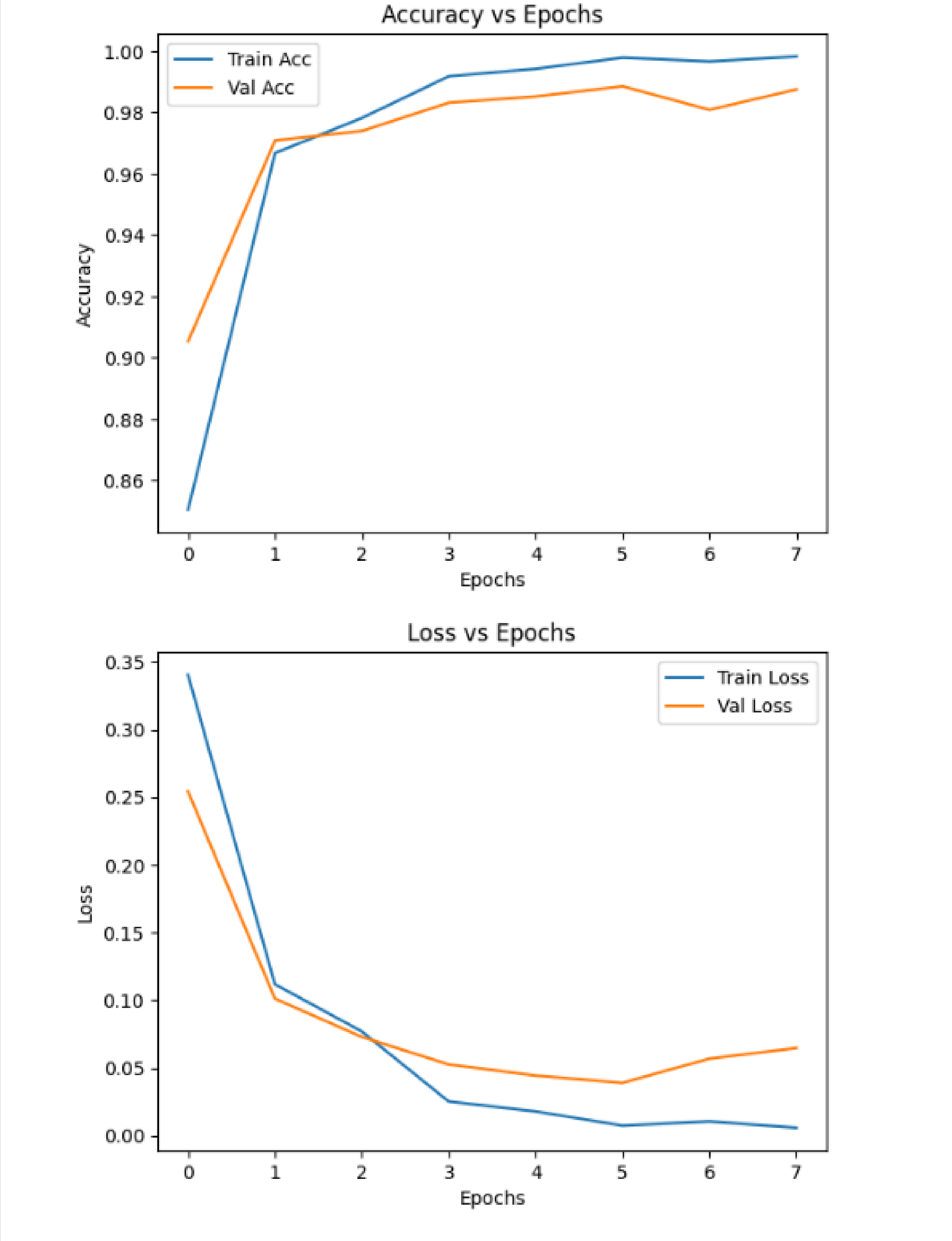
	precision	recall	f1-score	support
Fake	0.98	0.99	0.99	3580
Real	0.99	0.99	0.99	4240
accuracy			0.99	7820
macro avg	0.99	0.99	0.99	7820
weighted avg	0.99	0.99	0.99	7820

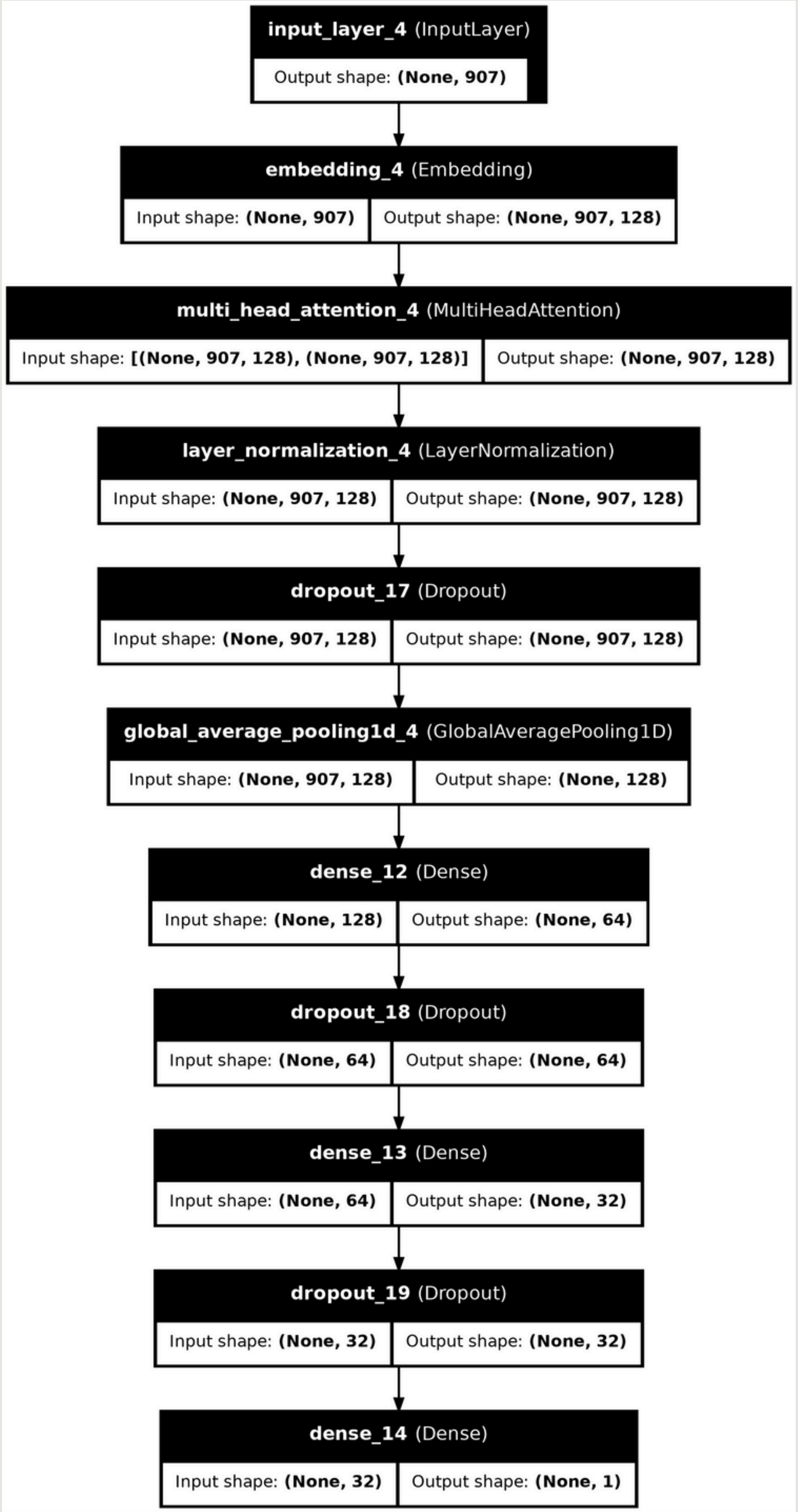


RESULT

BI-GRU







TRANSFORMER ARCHITECTURE

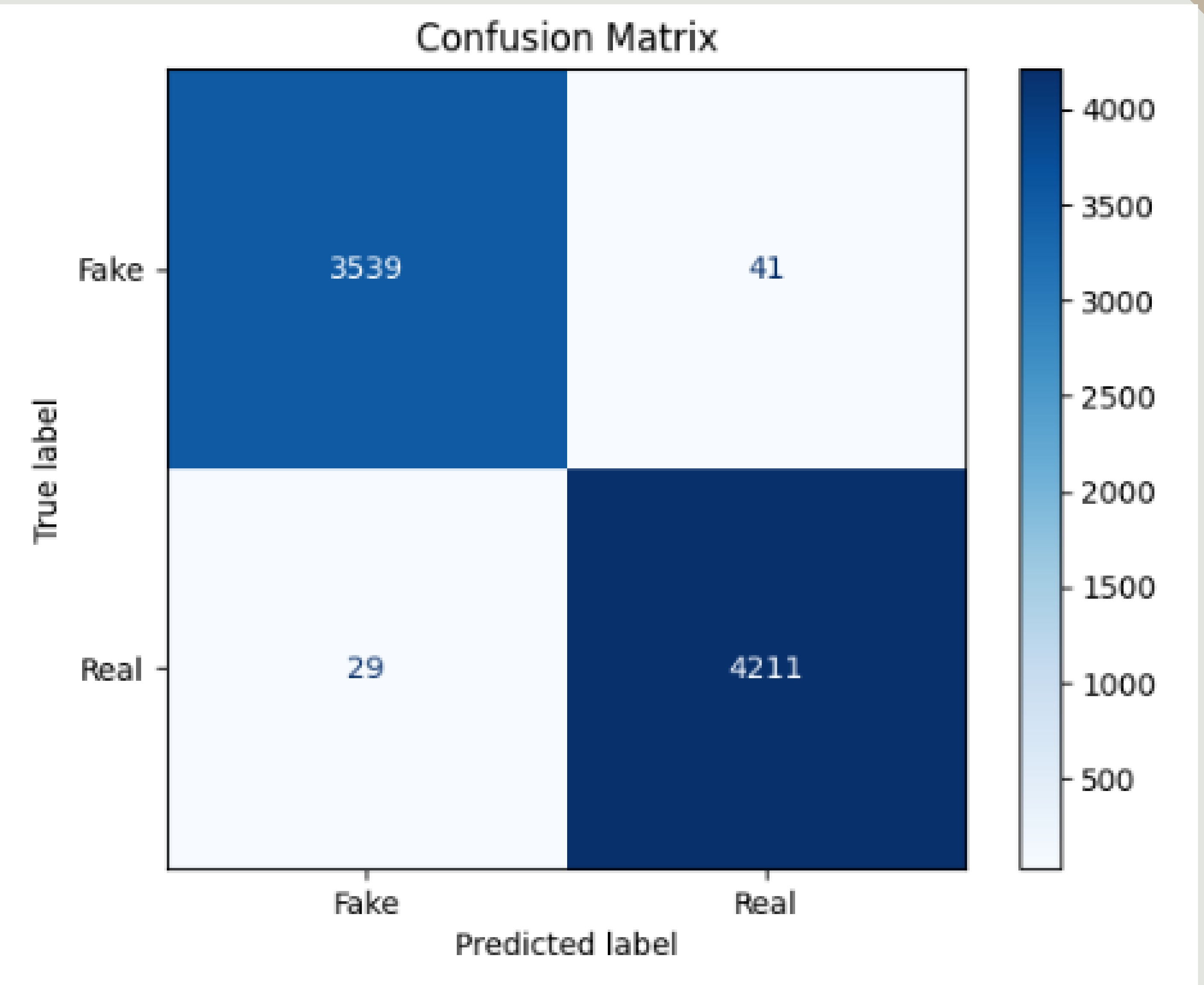
RESULT

TRANSFORMER

Evaluation Metrics:
Accuracy: 0.9910
Precision: 0.9904
AUC: 0.9992
EER: 0.0094

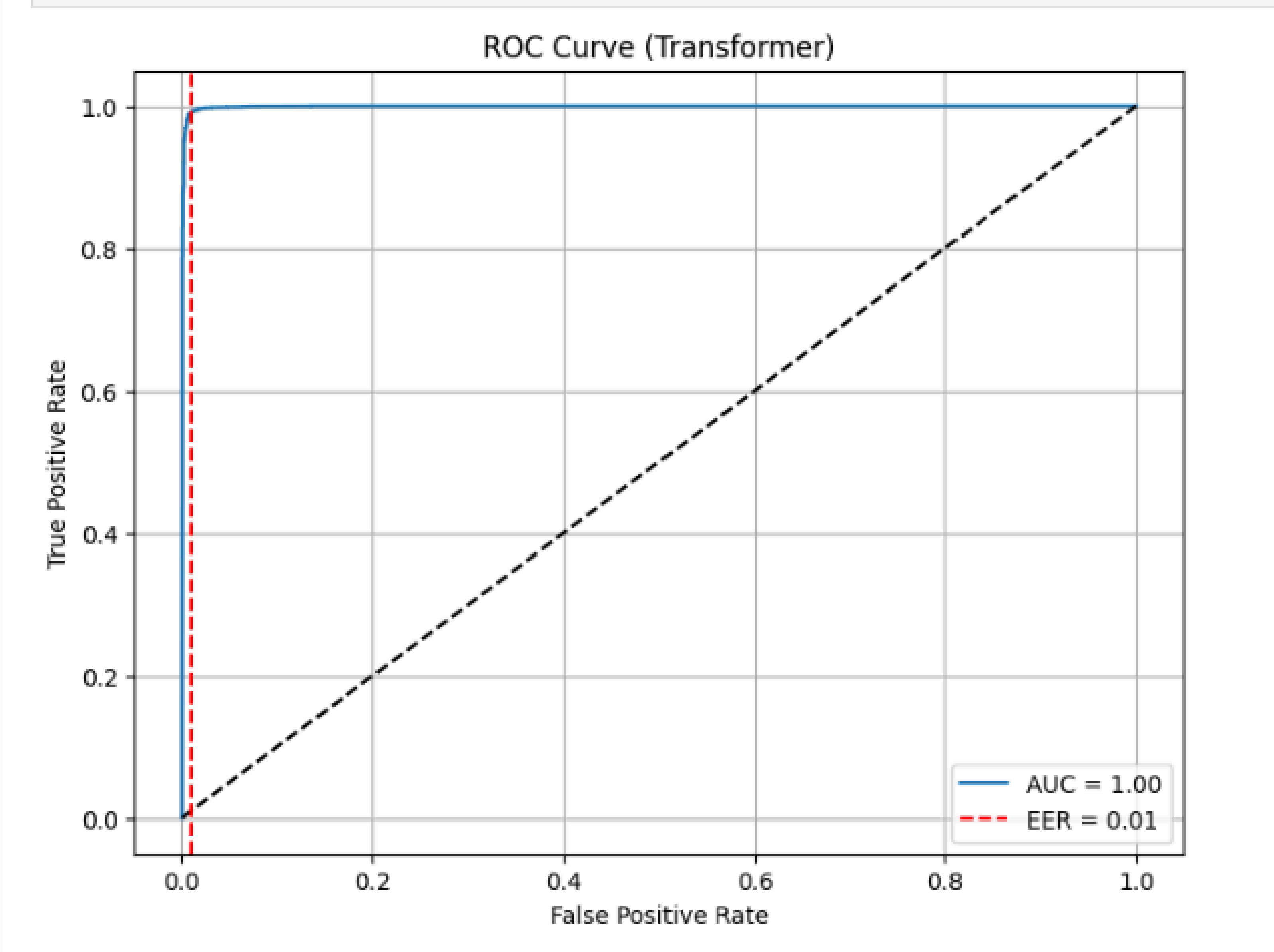
Classification Report:

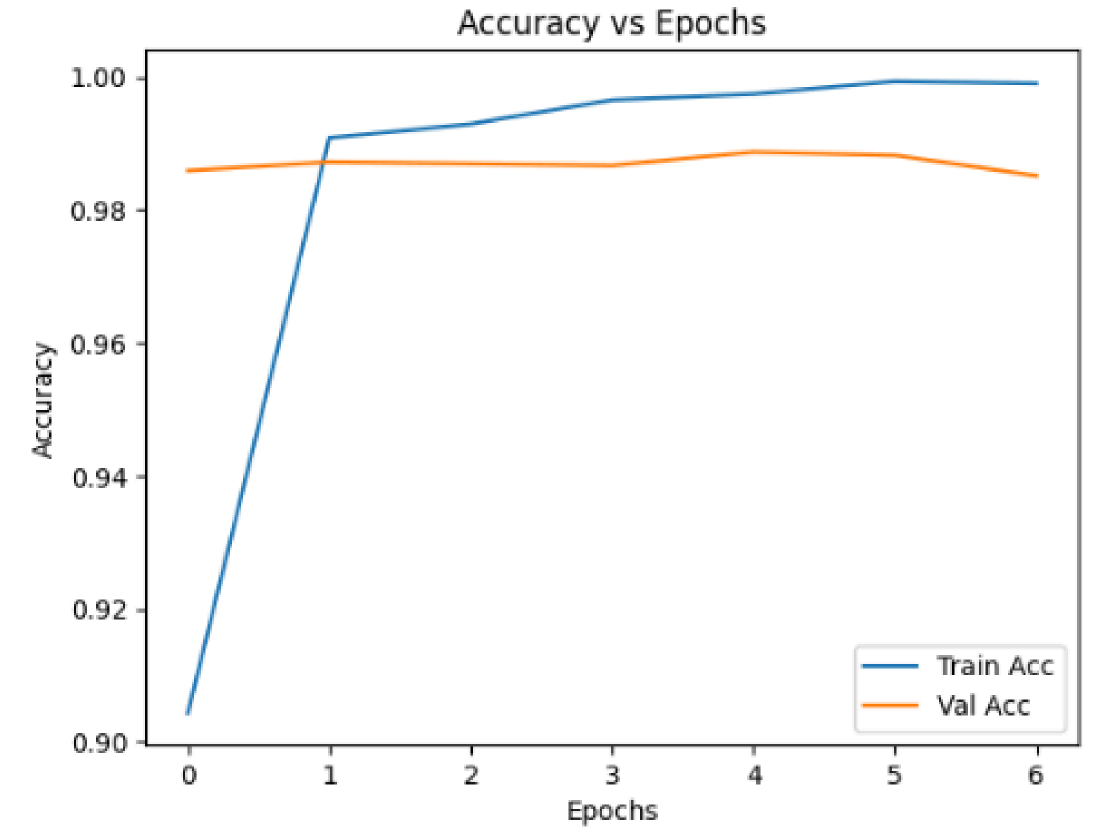
	precision	recall	f1-score	support
Fake	0.99	0.99	0.99	3580
Real	0.99	0.99	0.99	4240
accuracy			0.99	7820
macro avg	0.99	0.99	0.99	7820
weighted avg	0.99	0.99	0.99	7820

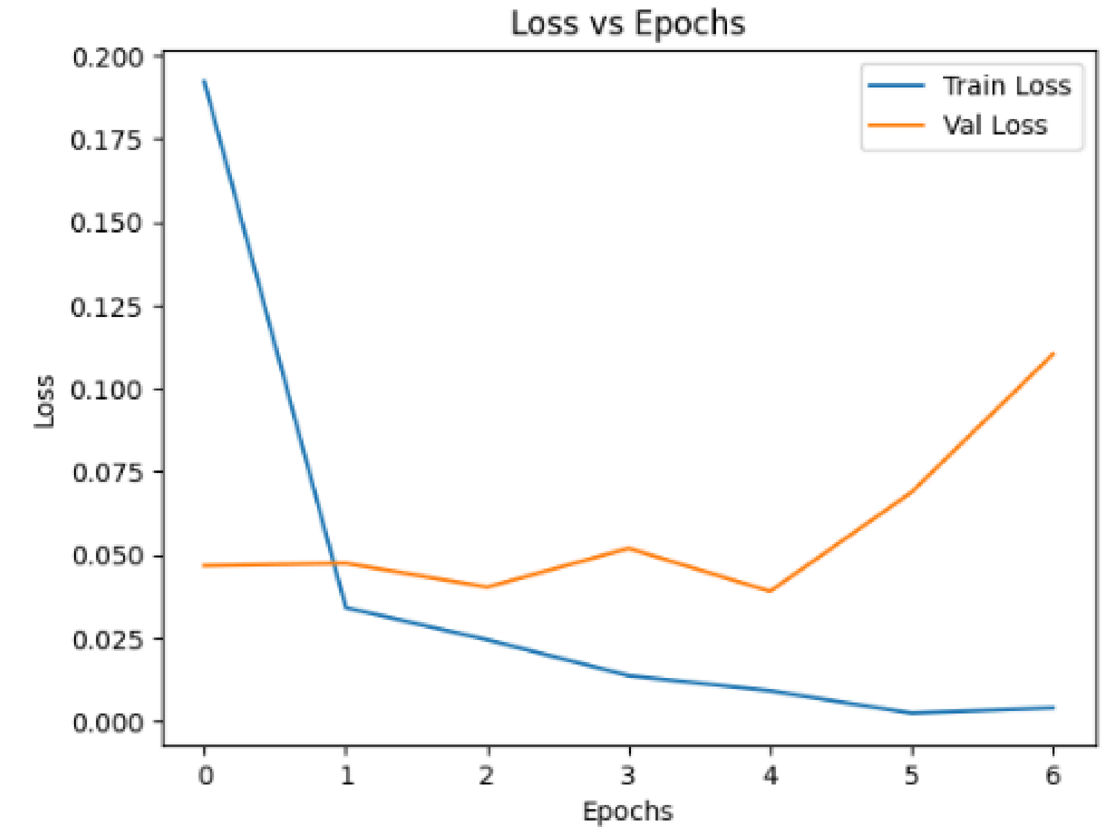


RESULT

TRANSFORMER







HYPERPARAMETER TUNING

Model	Batch SIze	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 (token len = 907)	Adam	0.0094	0.9992
	64 (token len = 500)	Adam	0.0108	0.9987
	32	RMSProp	0.0125	0.9993

SUMMARY

Metric	Logistic Regression	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 (avg)	0.98 (avg)	0.99 (avg)	0.99 (avg)
Fake Class (F1)	0.98	0.98	0.99	0.99
Real Class (F1)	0.99	0.98	0.99	0.99
Confusion Matrix	TP: 3494 FP: 86 FN: 36 TN: 4204	TP: 3539 FP: 41 FN : 92 TN: 4148	TP: 3550 FP: 30 FN: 61 TN: 4179	TP : 3539 FP : 49 FN : 21 TN : 4211

- Bi-GRU and Transformer are the top performers, both with F1-score of 0.99.
- Transformer has the lowest EER (0.0094) and fewest false negatives, making it highly reliable.
- Bi-GRU leads in precision (0.9929) and AUC (0.9994), showing strong class separation.
- Logistic Regression captures surface-level patterns but lacks semantic understanding, leading to good performance on specific datasets that may not generalize well.
- Bi-LSTM shows higher errors, especially in detecting real news (FN = 92).
- Overall, Transformer and Bi-GRU are the most effective, with Transformer better at minimizing errors and Bi-GRU better at precision.

CONCLUSION

- Transformer and Bi-GRU deliver the best overall performance across all metrics.
- Transformer excels in minimizing errors, with the lowest EER and false negatives, making it highly reliable for real-world detection.
- Bi-GRU achieves the highest precision and AUC, indicating strong confidence and class separation.
- Logistic Regression performs well by learning surface-level patterns, not semantic meaning — which explains its good metrics, but makes it highly dataset-specific and less generalizable
- Bi-LSTM is competitive but has higher misclassification rates, especially for real news.
- Recommendation: Choose Transformer for balanced reliability or Bi-GRU for slightly better precision depending on the application context.

Thank You