IBM DEEP LERNING AND REIFORCEMENT LEARNING
FINAL PROJECT

Fake news detection

REALISED BY :LEBGA HANANE

JULY 2024

CONTENT

1. Introduction

In the era of digital information, the proliferation of fake news has become a significant concern. Fake news refers to false or misleading information presented as news, often with the intent to deceive. The rapid spread of such misinformation can have serious consequences, including influencing public opinion, undermining trust in legitimate news sources, and causing social unrest.

Given the scale and speed at which fake news can spread, manual detection is impractical. Therefore, there is a pressing need for automated systems that can accurately identify and filter out fake news. This project aims to address this challenge using deep learning sequence models, which are well-suited for natural language processing tasks due to their ability to capture the sequential and contextual information inherent in text data.

2. Main objective

The main objective of this project is to develop and evaluate deep learning models for the task of fake news detection. Specifically, we aim to:

1. **Leverage Deep Learning Techniques:** Utilize advanced deep learning architectures such as LSTM, BiLSTM, Stacked LSTM, GRU, BiGRU, and ConvLSTM to build models capable of detecting fake news.
2. **Compare Model Performance**: Evaluate and compare the performance of various deep learning models in terms of accuracy, precision, recall, and F1-score.
3. **Communicate Findings**: Provide a comprehensive report detailing the data preprocessing steps, model training and evaluation, key findings, and insights. The report will also include visualizations to aid in understanding the model performance and behavior.

3. Data Exploration and Preprocessing

Data Description

- The dataset used for this project was gathered from Kaggle (https://www.kaggle.com/datasets/jruvika/fake-news-detection). It consists of 5MB of data with three features: URLs, Body, and Headline. The dataset is balanced to ensure fair evaluation of the models.Features:
- **URLs:** The web address of the news articles.
- **Body:** The main content of the news articles.
- **Headline:** The headline of the news articles.

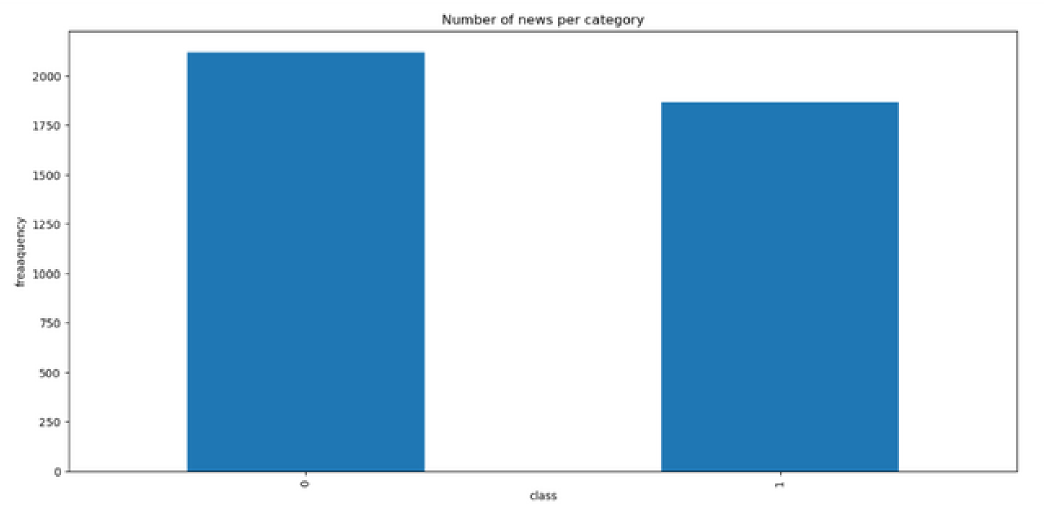

Figure: Plot that show the Frequency of Label class in the original dataset

Data Preprocessing:

- Data Cleaning: Removal of missing values and stopwords.
- Balanced data : extract only the first 1868 instance from each class
- Text Preprocessing: Combining the Body and Headline features to form a single text feature. Text normalization, including lowercasing, removal of URLs, special characters, stop words, and numbers.
- Tokenization: Splitting the text into individual words or tokens.
- Padding and Truncating: Ensuring that all text sequences have the same length for input into the neural network models.

3. Data Exploration and Preprocessing

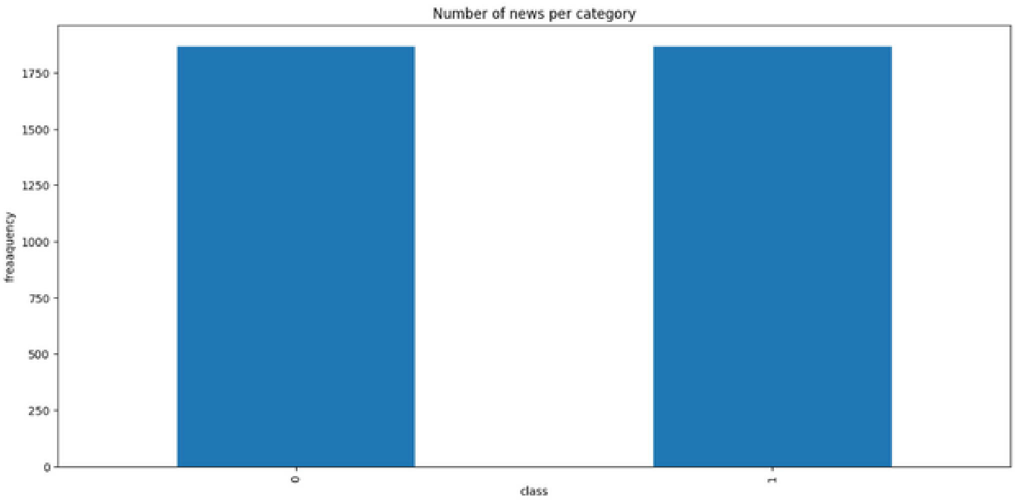

Figure: plot that show the frequency of Label class in the balanced dataset

4. Word Embedding with Word2Vec

Word2Vec Model:

- Trained a Word2Vec model on the preprocessed text data.
- Used Gensim to generate word embeddings with parameters:
  - Vector size: 200
  - Window size: 3
  - Minimum count: 1

5. Model Development

1.LSTM Model

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 1000, 200)         9663800

 lstm_1 (LSTM)               (None, 32)                29824

 dense_1 (Dense)             (None, 1)                 33

=================================================================
Total params: 9693657 (36.98 MB)
Trainable params: 29857 (116.63 KB)
Non-trainable params: 9663800 (36.86 MB)
```

Figure: architecture of LSTM model

2.BiLSTM Model

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 1000, 200)         9663800

 bidirectional (Bidirection  (None, 64)                59648
 al)

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 9723513 (37.09 MB)
Trainable params: 59713 (233.25 KB)
Non-trainable params: 9663800 (36.86 MB)
_________________________________________________________________
```

Figure: architecture of BiLSTM model

3.Stacked LSTM

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 1000, 200)         9663800

 lstm_3 (LSTM)               (None, 1000, 32)          29824

 lstm_4 (LSTM)               (None, 32)                8320

 dense_3 (Dense)             (None, 1)                 33

=================================================================
Total params: 9701977 (37.01 MB)
Trainable params: 38177 (149.13 KB)
Non-trainable params: 9663800 (36.86 MB)
_________________________________________________________________
```

Figure: architecture of Stacked Lstm model

5. Model Development

4.ConvLSTM

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_4 (Embedding)     (None, 1000, 200)         9663800

 conv1d (Conv1D)             (None, 996, 128)          128128

 max_pooling1d (MaxPooling1  (None, 498, 128)          0
 D)

 lstm_5 (LSTM)               (None, 32)                20608

 dense_4 (Dense)             (None, 1)                 33

=================================================================
Total params: 9812569 (37.43 MB)
Trainable params: 148769 (581.13 KB)
Non-trainable params: 9663800 (36.86 MB)
_________________________________________________________________
```

Figure: architecture of convolutional LSTM model

5.GRU

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_6 (Embedding)     (None, 1000, 200)         9663800

 gru_1 (GRU)                 (None, 32)                22464

 dense_6 (Dense)             (None, 1)                 33

=================================================================
Total params: 9686297 (36.95 MB)
Trainable params: 22497 (87.88 KB)
Non-trainable params: 9663800 (36.86 MB)
_________________________________________________________________
```

Figure: architecture of GRU model

6.BiGRU

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_6 (Embedding)     (None, 1000, 200)         9663800

 gru_1 (GRU)                 (None, 32)                22464

 dense_6 (Dense)             (None, 1)                 33

=================================================================
Total params: 9686297 (36.95 MB)
Trainable params: 22497 (87.88 KB)
Non-trainable params: 9663800 (36.86 MB)
_________________________________________________________________
```

Figure: architecture of BiGRU model

6. Model Training and Evaluation

Training:

- Split data into training and testing sets (80-20 split).
- Trained each model on the training set with early stopping to prevent overfitting in architecture such as stacked LSTM, GRU, BiGRU, ConvLSTM

Evaluation:

- Evaluated models on accuracy and loss metrics using “binary_crossentropy”.
- Plotted training and validation performance (accuracy and loss) over epochs.

1.LSTM

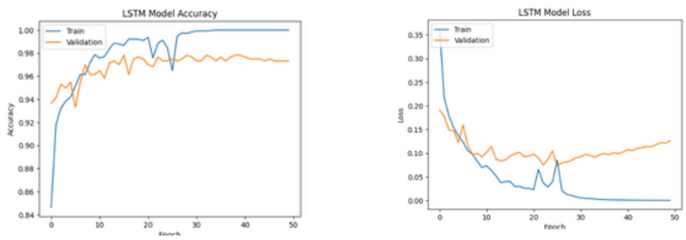

Figure: the training and validation accuracy and loss over epochs using LSTM

2.Bilstm

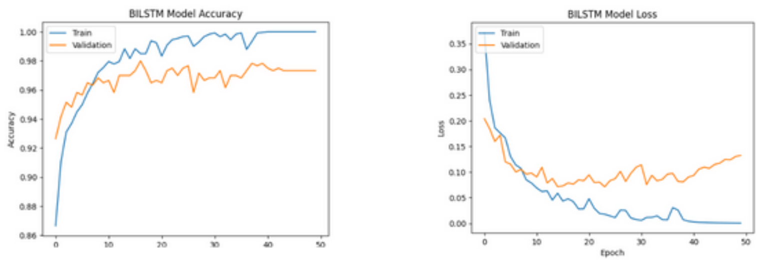

Figure: the training and validation accuracy and loss over epochs using Bilstm

3.Stacked LSTM

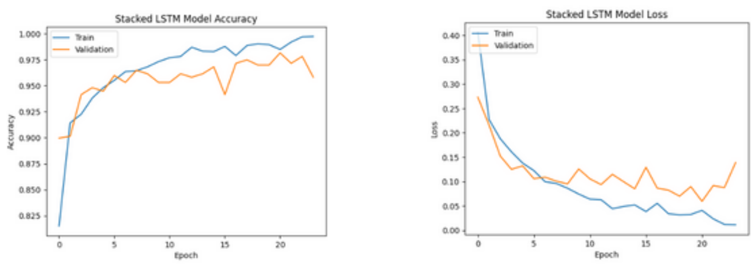

Figure: the training and validation accuracy and loss over epochs using Stacked LSTM

6. Model Training and Evaluation

4.Convolutional LSTM

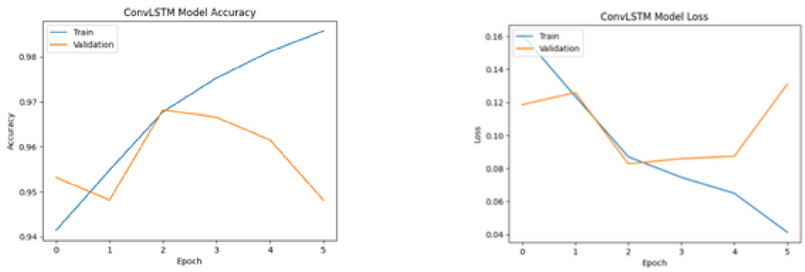

Figure: the training and validation accuracy and loss over epochs using ConvLstm

5.GRU

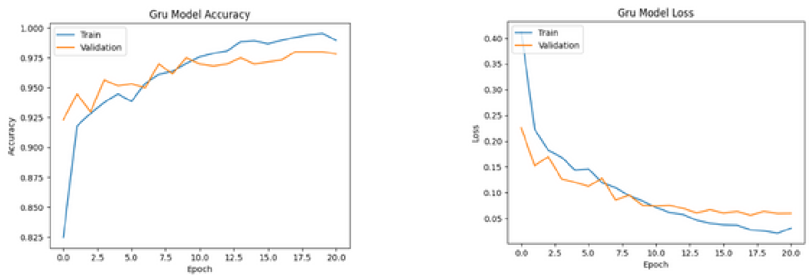

Figure: the training and validation accuracy and loss over epochs using GRU

6.BiGRU

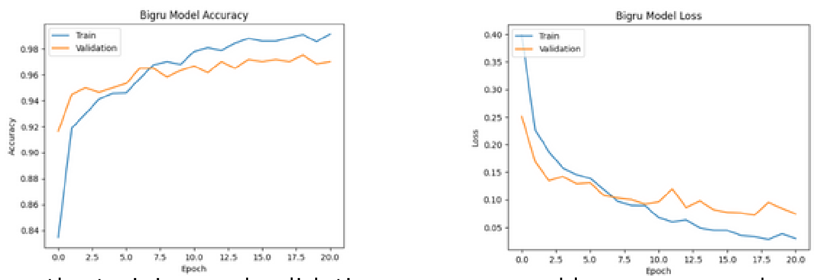

Figure: the training and validation accuracy and loss over epochs using BiGRU

7. Results and Discussion

The models were evaluated in the test set based on accuracy, precision, recall, and F1-score. The results showed that the deep learning models were effective in detecting fake news, with the BiGRU model achieving the highest overall performance.

1. LSTM

```
24/24 [==============================] - 1s 20ms/step
              precision    recall  f1-score   support

        Fake       0.97      0.97      0.97       392
    Not Fake       0.97      0.97      0.97       356

    accuracy                           0.97       748
   macro avg       0.97      0.97      0.97       748
weighted avg       0.97      0.97      0.97       748
```

Figure: Results in the test set

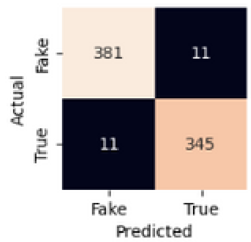

Figure: Confusion matrix

2. BiLSTM

```
24/24 [==============================] - 1s 28ms/step
              precision    recall  f1-score   support

        Fake       0.98      0.95      0.97       392
    Not Fake       0.95      0.97      0.96       356

    accuracy                           0.96       748
   macro avg       0.96      0.96      0.96       748
weighted avg       0.96      0.96      0.96       748
```

Figure: Results in the test set

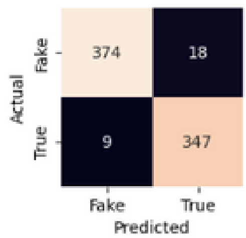

Figure: Confusion matrix

3. Stacked LSTM

```
24/24 [==============================] - 1s 29ms/step
              precision    recall  f1-score   support

        Fake       0.98      0.97      0.98       392
    Not Fake       0.97      0.98      0.97       356

    accuracy                           0.97       748
   macro avg       0.97      0.97      0.97       748
weighted avg       0.97      0.97      0.97       748
```

Figure: Results in the test set

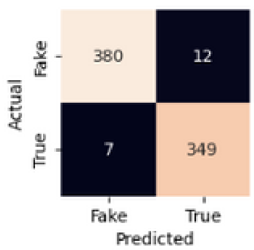

Figure: Confusion matrix

7. Results and Discussion

4. Convolutional LSTM

```
24/24 [==============================] - 1s 12ms/step
              precision    recall  f1-score   support

        Fake       0.98      0.96      0.97       392
    Not Fake       0.96      0.98      0.97       356

    accuracy                           0.97       748
   macro avg       0.97      0.97      0.97       748
weighted avg       0.97      0.97      0.97       748
```

Figure: Results in the test set

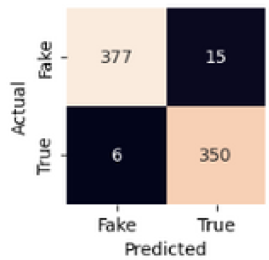

Figure: Confusion matrix

5. GRU

```
24/24 [==============================] - 1s 15ms/step
              precision    recall  f1-score   support

        Fake       0.98      0.96      0.97       392
    Not Fake       0.96      0.98      0.97       356

    accuracy                           0.97       748
   macro avg       0.97      0.97      0.97       748
weighted avg       0.97      0.97      0.97       748
```

Figure: Results in the test set

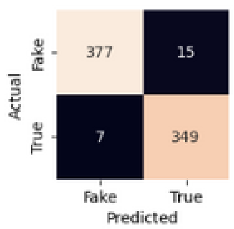

Figure: Confusion matrix

6. BiGRU

```
24/24 [==============================] - 1s 29ms/step
              precision    recall  f1-score   support

        Fake       0.98      0.97      0.98       392
    Not Fake       0.97      0.98      0.97       356

    accuracy                           0.98       748
   macro avg       0.98      0.98      0.98       748
weighted avg       0.98      0.98      0.98       748
```

Figure: Results in the test set

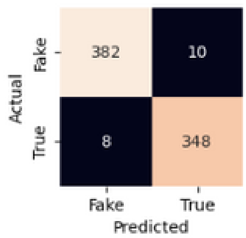

Figure: Confusion matrix

Discussion :

- The Bidirectional Gated Recurrent Unit (BiGRU) model demonstrated exceptional performance in the task of fake news detection. With an accuracy of 98% and F1-scores of 98% for the fake news class and 97% for the non-fake news class, the BiGRU model outperformed other models tested in this project, including LSTM, BiLSTM, Stacked LSTM, GRU, and ConvLSTM.

8. Key Findings and Insights

Key Findings

1. **High Accuracy:**
    - The BiGRU model achieved an accuracy of 98%, indicating that it correctly classified 98% of the news articles in the test set. This high level of accuracy suggests that the model is highly effective at distinguishing between fake and non-fake news.
2. **F1-Score:**
    - The F1-score for the fake news class was 98%, and for the non-fake news class, it was 97%. The F1-score considers both precision and recall, making it a robust measure of the model's performance. High F1-scores for both classes indicate that the model has a balanced performance, with low false positive and false negative rates.
3. **Bidirectional Context:**
    - The BiGRU architecture leverages information from both past and future contexts, enabling it to capture nuanced patterns in the text. This bidirectional approach likely contributed to the model's superior performance compared to unidirectional models like standard GRU and LSTM.
    -

Strengths of the BiGRU Model

1. **Effective Sequence Modeling:**
    - GRU cells are known for their ability to handle long-term dependencies in sequences, which is crucial for text data where the meaning of a word can depend on distant words in the sequence.
2. **Bidirectional Processing:**
    - By processing the input text in both forward and backward directions, the BiGRU model captures a more comprehensive understanding of the text, improving its ability to detect fake news.
3. **Efficient Training:**
    - GRUs are computationally more efficient than LSTMs, as they have fewer parameters and require less memory, allowing for faster training and inference.

Future work :

- **Generalizability:** Future work should involve testing the BiGRU model on more diverse datasets to ensure its generalizability across different domains and sources of news.

- **Model Interpretability:** Enhancing the interpretability of the BiGRU model using techniques such as attention mechanisms, explainable AI methods(SHAP, LIME) which are so suitable for text classication could provide deeper insights into how the model makes its predictions.

9. Conclusion

Conclusion :

This project demonstrates the power of deep learning sequence models in tackling the challenging task of fake news detection. By systematically exploring various models and leveraging the strengths of the BiGRU architecture, we have developed a highly accurate and efficient model that holds significant potential for real-world applications. Continued research and refinement will help in further enhancing the model's capabilities and ensuring its effectiveness in diverse settings.