**DEPARTMENT OF COMPUTING AND INFORMATION SYSTEMS**
**SCHOOL OF ENGINEERING AND TECHNOLOGY**

**CSC3034 COMPUTATIONAL INTELLIGENCE**

**DEADLINE:**
**17 December 2024**

| No | Student ID | Student Name |
|---|---|---|
| 1 | 21075213 | Gan Hao Zheng |
| 2 | 21074638 | Stephen Tee Wei Xin |
| 3 | 21018429 | Yeong Meng Li |

## INSTRUCTIONS

1. Form a group of 3 to 4 members. The same group from Assignment 1 may be maintained, but not compulsory; inform me if there are any changes in your group so I can assign the submission correctly
2. Construct a neural network on a real-world dataset.
3. Discuss one type of hybrid intelligent system that can be applied as modifications to improve this neural network with the chosen dataset.
4. Submit your code and report by 17 December 2024 23:59.

# Table of Contents

# 1.0 Introduction

The introduction will cover the overview of the project, as well as the dataset selection process that has undergone as a preliminary. Additionally, the segment will delve into deeper detail regarding the dataset's sources, preprocessing methodology, as well as general features and statistics.

## 1.1 Overview and Objectives

This project aims to explores the application of neural networks and hybrid intelligent systems on a real-world dataset. We plan to implement a news prediction system using neural networks to identify fake or true news after training on the chosen dataset.

## 1.2 Dataset Selection

The chosen dataset is selected on Kaggle. This dataset is called fake-and-real-news-dataset which can be assessed through this link. https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset/data

## 1.3 Dataset Description

The dataset has been separated into 2 files for true news and fake news. The fake news is in the Fake.csv file which contains 23502 fake news articles while the true news is in the True.csv file which has 21417 true news articles. The dataset consists of 4 columns which are:

1. Title: title of news article
2. Text: body text of news article
3. Subject: categories of news article such as politics news, world news and US news
4. Date: publish date of news article

## 1.4 Data Preprocessing

In the original dataset, there was not any data preprocessing carried out by the author. However, several key preprocessing steps were applied by us in the code to train the neural network. The preprocessing steps taken by us include:

1. Combining Text: The title and text columns in the dataset are combined using the "clean_text" feature.
2. Data Balancing: A sample size of 5000 entries is taken from both the True.csv and Fake.csv to avoid bias and ensure the neural network learns from equal amount of fake and true news data.
3. Tokenisation: A tokeniser is used to tokenise the data into numerical sequences. The words are converted into integers based on their frequency in the dataset.
4. Padding Sequences: To standardise the input size for the neural network, the tokenised sequences are padded to a fixed length of 200 words.

## 2.0 Neural Network Application and Design

This segment aims to communicate integral information regarding the neural network that has been constructed, utilized, and evaluated within the project scope. Within abstraction, the segment aims to justify the use of neural networks for this particular dataset, the architecture and specifications of the neural network constructed, as well as the overall methodology concerning the training and processing.

## 2.1 Justifying Neural Network Application

The main purpose of applying a neural network in this project is to classify news articles into True News or Fake News based on their textual content. After training the model, sample text can be input for the model to predict whether it is fake news or true news. This project utilises deep learning's ability to extract meaningful pattern from unstructured texts to predict misinformation based on the input data. Since neural network excels in capturing underlying patterns in data, it is suitable to be applied on this dataset as the dataset consists of textual data which is unstructured. Fake news usually includes traits such as informal sentence structure, exaggerated content and lack of credible sources. Besides, the detection of fake news also requires the understanding of context. These characteristics of the dataset can easily be identified by neural network models like Long Short-Term Memory (LSTM). Moreover, the chosen dataset is balanced and contains similar amounts of fake and true news while an equal amount of both types of news is used to train the model to ensure there is no bias. As a result, the application of neural network is aligned with the dataset characteristics which justify the use of it in this project.

## 2.2 Neural Network Design

The neural network utilized in the project implements a Recurrent Neural Network (RNN) architecture with the utilization of Bidirectional LSTMs and Dense Layers.

```
# Step 9: Neural Network Model Construction
model = Sequential([
    Embedding(input_dim=vocab_size, output_dim=64, input_length=max_length),
    Bidirectional(LSTM(128, return_sequences=True)),
    Dropout(0.5),
    LSTM(64, return_sequences=False),
    Dense(64, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])
```

*Figure 1: Neural Network Code Block*

### 2.2.1 General Architecture

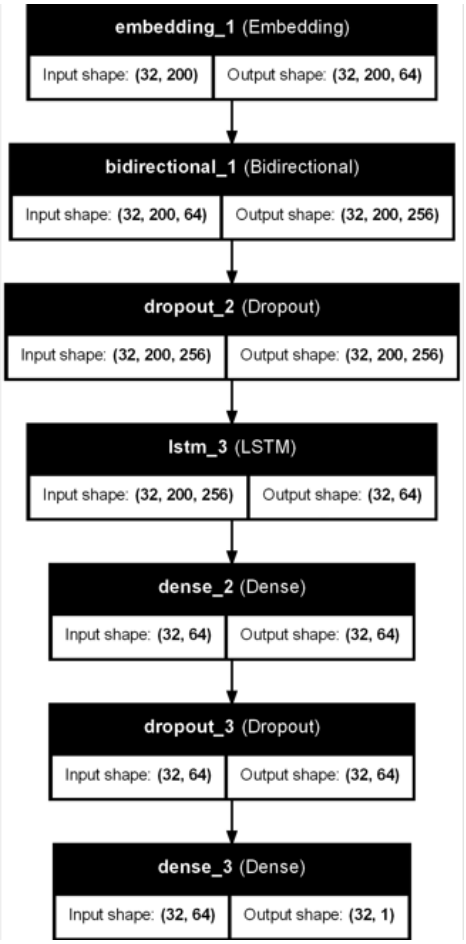


*Figure 2: Model Architecture*

The neural network contains 7 layers of neurons. The layers will be further explained and justified.

1. Embedding Layer

The embedding layer aims to convert the tokenized text input into vectors where semantic relationships can be drawn between different words. This conversion is a crucial process that allows the neural network to process the data in a more meaningful manner.

2. Bidirectional LSTM Layer (128 Neurons)

This layer processes the embedded data to determine what makes fake news fake and what makes true news true, understanding the context clues within the data. Bidirectional LSTM layers allow for sequential data (such as strings of words forming a sentence) to be processed in both forward and backward directions.

3. Dropout Layer

The dropout layer serves as a regulator for the model, allowing for the model to fit the data appropriately, preventing overfitting of data. This allows the model's training and testing accuracy to be similar rather than testing accuracy being significantly higher than training accuracy.

4. LSTM Layer (64 Neurons)

The number of neurons on this layer is reduced to ensure the model's complexity and computational requirements remain appropriate. Besides this, it allows for more features to be determined, improving accuracy.

5. Dense Layer (64 Neurons)
   This layer is responsible for combining features that were extracted by previous LSTM layers to make classification predictions. The number of neurons were chosen to balance complexity and computation expense.
6. Dropout Layer
   A second dropout layer was implemented to further prevent overfitting, allowing for the model to adjust to new data during validation.
7. Output Layer
   The output layer is a single neuron with a sigmoid activation function which allows for a binary classification (appropriate to the context of this dataset). The output from the sigmoid function falls between values 0 and 1, which can be then interpreted as either 0 (fake news) or 1 (real news).

### 2.2.2 Activation Functions

1. Rectified Linear Unit (ReLU)
   Utilized in the dense layer within the model, it is used in hidden layers to reduce the vanishing gradient problem and allows for quicker convergence during the model training phase.
2. Sigmoid
   Utilized in the output layer, it was employed for its suitability for classification tasks such as the one required by this project.

### 2.2.3 Hyperparameter Tuning

The hyperparameters chosen for this model were intentionally chosen, whether through experimentation and appropriation to model complexity and computational capabilities.

1. Embedding Layer Output Dimension
   The value 64 was chosen to ensure enough word embeddings were being used whilst also keeping the model complexity minimal.
2. Number of Neurons in LSTM Layers
   The first LSTM layer utilizes 128 neurons to help the model extract features from the data that could be more complex and underlying, whereas only 64 neurons were utilized in the second layer to ensure the model complexity is kept manageable whilst features are kept.
3. Dropout Rate
   Both dropout layers employ a dropout rate of 0.5, which was chosen from experimenting with the model, this value was chosen as it prevented overfitting whilst maintaining the model's learnability towards data.
4. Learning Rate
   The learning rate was kept at 0.001 in the Adam optimizer to prevent any overfitting. The Adam optimizer was chosen as it adapts the learning rate during the training phase.

## 2.3 Methodology

### 2.3.1 Data Splitting

The dataset is split into 80% training data and 20% testing data. This split prioritizes the model's access to more data for training. Additionally, 20% of the training data is dedicated to validation which allows for performance monitoring during the training phase as well, preventing the possibility of overfitting.

### 2.3.2 Training Process

The model utilizes the Adam optimizer, specified to a learning rate of 0.001. The training process utilizes "early stopping", which stops the training process if the prediction accuracy for the validation data does not improve after 3 consecutive epochs. The maximum training period is 10 epochs.

### 2.3.3 Stop Conditions

```
# Step 11: Model Training (With Early Stopping)
early_stopping = tf.keras.callbacks.EarlyStopping(
    monitor='val_loss',
    patience=3,
    restore_best_weights=True
)
```

*Figure 3: Early Stopping Callback Code Block*

The model utilizes “early stopping” which is a callback method that reverts the weights of the model if the conditions are appropriate. In this context, the “patience” indicates the number of epochs the callback will wait for before reverting. In this case, 3 was chosen which means the training process will halt and revert before continuing if the prediction accuracy for the validation data does not improve within 3 epochs time. This is done to reduce the possibility of overfitting, which causes the model to be too specific and struggle to adapt to new data in the testing phase

## 2.4 Challenges and Limitations

1. Ambiguous News
   The model may face difficulties in classifying news that are ambiguous (i.e. contains enough features to be either real or fake news). In turn, the model may struggle to classify news with more ambiguity.
2. Hyperparameter Decisions
   The decision process that had undergone to choose the hyperparameters for the model were from experimentation and manual tuning, this does not ensure the most optimal and performative model.
3. Potential Overfitting
   Although the model takes precautions through callback methods, validation splits, and dropout layers, it is possible for overfitting to still occur due to the smaller dataset size causing poor generalization.

## 3.0 Results and Evaluation

This segment aims to discuss the results and findings from the processing done by the constructed neural network model. Additionally, this segment will discuss the evaluation methodology utilized as well as a summary of the system efficacy through metrics and evaluation.

## 3.1 Results

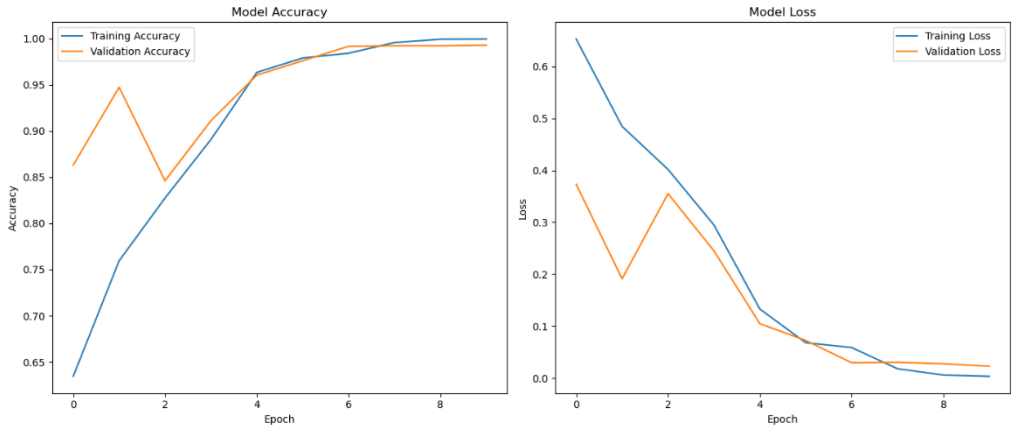


*Figure 4: Model Accuracy and Model Loss*

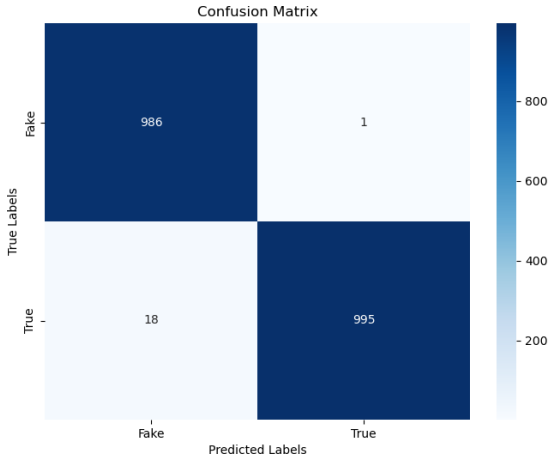


*Figure 6: Classification Report*

| Classification Report: | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.98 | 1.00 | 0.99 | 987 |
| 1 | 1.00 | 0.98 | 0.99 | 1013 |
| accuracy | | | 0.99 | 2000 |
| macro avg | 0.99 | 0.99 | 0.99 | 2000 |
| weighted avg | 0.99 | 0.99 | 0.99 | 2000 |

*Figure 6: Classification Report*

The overall results indicate exceptional model performance on the classification of true and fake news. In figure 4, both training and validation accuracy curves show similar patterns where they both reach > 99% accuracy. Whereas the loss curves illustrate a constant decrease. The final training and validation loss values are extremely low. Figure 6 highlighted that the f1-score for both classes, accuracy, macro average, and weighted average are 99%. Whereas the precision for fake news is 98% and recall is 100%.

## 3.2 Evaluation Metrics and Justification

To evaluate the performance of the model, the following metrics were used:

1) **Accuracy**: The test accuracy achieved was 99%. Accuracy is measured as the ratio of correctly predicted instances to the total number of predictions. While the accuracy metric can provide a simple and straightforward overview of the model's performance, it can be insufficient when it comes to scenarios that involve binary classification where false positives and false negatives have different implications. An example of this would be the model achieving high accuracy, but it might still fail to correctly identify critical examples of fake or true news. Therefore, issues might surface if accuracy is the sole metric that is being relied on.

2) **Precision**: Precision is 98% for fake and 100% for true news prediction. Having high precision ensures that the prediction for fake or true news is highly reliable with minimal false positives.

3) **Recall**: Recall value for fake news is 100% while true news is 98%. In this case, having a high recall value indicates that the model minimizes the number of false negatives by making sure that it does not miss any fake news. If a perfect recall is achieved, it shows that the model is able to detect all occurrences of fake and true news without any oversight.

4) **F1-score**: This metric was also used to provide a fair assessment of precision and recall. F1-score combines both precision and recall in a single metric. In this case, the F1-score for both classes is 99%. This score signifies the model's high reliability between precision and recall for both classes. This metric is particularly useful when it comes to imbalanced class distributions where it is important to balance precision and recall ensuring the model operates optimally under changing conditions.

5) **Loss**: Loss was used to assess the model's training and generalization performance. Having low values on both training loss and validation loss indicates that the model has efficiently learned from the data with no server underfitting or overfitting. In this case, the low validation loss suggests that the model was properly optimized.

6) **Confusion Matrix**: This metric is used to measure the performance of the model. Figure 5 shows that there are 986 true negatives, 1 false positive, 18 false negatives, and 995 true positives. It offers a detailed breakdown of the model's predictions, and a proper analysis of error patterns can be done. By analysing the confusion matrix, it can be determined if the model is biased toward over-flagging or under-flagging news content. The figure shows that the model performs well, achieving high precision and recall while classifying the datasets correctly.

## 3.3 System Evaluation and Summary

Overall, the model performs incredibly well with high accuracy and balanced precision, recall, as well as F1-scores for both classes. The results highlight the model's strength to be able to classify true or fake news accurately consistently, which demonstrates its reliability in distinguishing between the two classes. Another notable strength is its ability to minimize false positives where it has only 1 in this case, ensuring that fake news is not wrongly labelled as legitimate. This allows for an improved practical usability in the model. Furthermore, the model's balanced and consistent performance across both classes indicates its robustness by avoiding bias toward either class.

Despite a strong performance overall, the model also has its flaws. Particularly its ability to handle false negatives where true news is labelled as false. This is a serious weakness as misclassifications can ultimately lead to the undermining of credibility when it comes to the source of information and can potentially lead to scepticism towards legitimate news sources. These errors may occur die to overlapping linguistic traits or patterns in true news articles that resemble those seen in fake news.

# 4.0 Proposed Hybrid Intelligent System: Adaptive Neuro-Fuzzy System (ANFIS)

In order to address the limitations specified in Section 2.4 Challenges and Limitations, the project also proposes a neuro-fuzzy system that utilizes neural networks and fuzzy logic to classify news to binary categories of true or false.

### 4.1 Justification

The purpose of this proposed hybrid system is to handle the limitations of the current model directly.

1. Adapting to Ambiguity
   The current model may face difficulties in classifying news that are more ambiguous due to their composition of both "true" and "false" characteristics. To address this, the addition of fuzzy logic allows for imprecise data to be processed more accurately by assigning membership degrees to features, rather than a binary "true" or "false". This allows the model to comprehend and classify news that potentially lie between the current black and white classifications.
2. Addressing Overfitting
   ANFIS incorporates fuzzy rules into its neural network, which allows for more accurate predictions. This allows for the model to base its classification based on rules to guide itself, rather than on memorization of patterns in the training data.

### 4.2 Potential Methodology

The hybrid solution could be implemented in a sequential manner that utilizes the strengths of both fuzzy layers and LSTM layers.

Phase 1: LSTM based Neural Network Model (Embedding – BiLSTM – Dropout – LSTM)

The input data is embedded in the embedding layer and is passed to the first LSTM layer where the first set of features are learned. The dropout layer is kept for regularization purposes. This process is then repeated at a smaller scale to ensure features are sufficiently extracted from the data. The abstracted features that have been extracted will be passed directly into the input layer of the neuro-fuzzy system. Consider the scenario where the LSTM layer may extract a correlation between the sentiment and the reliability of the news.

Phase 2: Neuro-Fuzzy Model (Input – Fuzzification – Fuzzy Rules – Normalization – Defuzzification)

The input layer receives the abstract features from the LSTM layers. These features are then passed through a fuzzification layer. In this layer, the "sentiment" feature can be fuzzified with different degrees of membership to certain categories. In this case, consider the categories "negative", "neutral", and "positive" sentiment.

After the feature is fuzzified, it passes through a fuzzy rule layer. These rules will be determined by a domain expert or from previous data exploration. For example, a rule could state "If sentiment = positive, source reliability = high, then news = true". These rules allow for the different features extracted by the LSTM layers (sentiment, source reliability, etc.) to pass through predefined rules to ensure more accurate classification.

After this, the fuzzy outputs are then normalized in the output function layer or normalization layer, where they are standardized to a value between 0 and 1 based on the weights of features and rules from the last layer, which prepares them for classification.

The normalized output is then defuzzied in the defuzzification layer. This is where a fuzzed instance can be classified. For instance, an output that has a value of 0.67 for "Real" and 0.33 for "Fake". It will be classified as 1, indicating true.

Phase 3: Evaluation

Similar metrics will be used to determine the accuracy of the hybrid model.

## 4.3 Application Examples

In the current dataset, the goal is to classify news articles to categories of either true or false. This section aims to give an example of how that works. The LSTM based model that we currently have could be adapted to extract features of a particular news article; Sentiment (Positive, Neutral, Negative), Keyword Count ("Experts", "Warn", "Breaking"), and even Context ("Alleged", "Claims"). The combination of these features is then passed through a set of neuro-fuzzy layers. Fuzzy rules such as: "If sentiment = positive, keyword count = high, then article is real". This allows for more nuanced decision making, rather than unsupervised trend learning. Based on these combinations, the model will assign a value that falls between 0 and 1. If the value is closer to 1, it will be determined as true and vice versa.

### 4.4 Potential Improvements

The hybrid model faces an issue of rigidity due to the fuzzy rules being determined before any dataset is passed through. Additionally, this hinders the LSTM's ability to uncover new patterns. This will present an issue in scenarios requiring more flexibility, especially if the rules are not specific or reliable enough.

In order to improve on this, a reinforcement learning approach could be taken for rules, where the rules can account for new patterns found by the LSTM layers to ensure updated and reliable predictions.

## 5.0 Conclusion

In conclusion, the project manages to deliver a neural network model that adequately performs classification tasks on the chosen dataset. It performs adequately and is evaluated through a comprehensive collection of metrics. It possesses some natural limitations due to a consideration of feasibility and a limitation of computational capacity. However, the project proposes a potential improvement through a hybrid model that addresses some of the limitations of the current model.