Tribhuvan University
Institute of Science and Technology

Central Department of Computer Science and Information Technology
Kirtipur, Kathmandu

Seminar Report on
"Fake News Detection Using LSTM Network"

In partial fulfillment of the requirement for a Master's degree in computer science and information technology (M.Sc. CSIT), 1st Semester

Submitted to:
Central Department of Computer Science and Information Technology, Tribhuvan University, Kirtipur, Kathmandu, Nepal

Submitted By:
Manika Ghimire
(8015021)

Date: July 2024

Tribhuvan University

Institute of Science and Technology

Supervisor Recommendation

This is to certify that Miss. Manika Ghimire has submitted the seminar report on the topic "**Fake News Detection Using LSTM Network"** for the partial fulfilment of the Master's of Science in Computer Science and Information Technology, first semester. I hereby declare that this seminar report has been approved.

_____

Supervisor

Assoc. Prof. Mr. Arjun Singh Saud

Central Department of Computer Science and Information Technology

Certificate of Approval

This is to certify that the seminar report prepared by Miss. Manika Ghimire " **Fake News Detection Using LSTM Network"** in partial fulfilment of the requirements for the degree of Master of Science in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

Evaluation Committee

……..……..	……..……
Asst. Prof. Sarbin Sayami	Assoc. Prof. Arjun Singh Saud
(H.O.D)	(Supervisor)
Central Department of Computer Science and Information Technology	Central Department of Computer Science and Information Technology

……

(External)

Acknowledgment

The success and final outcome of this report required a lot of guidance and assistance from many people, and I am very fortunate to have got this all along the completion. I am very glad to express my deepest sense of gratitude and sincere thanks to my highly respected and esteemed supervisor **Assoc. Prof. Arjun Singh Saud,** Central Department of Computer Science and Information Technology for his valuable supervision, guidance, encouragement, and support for completing this paper.

I am also thankful to **Asst. Prof. Sarbin Sayami**, HOD of Central Department of Computer Science and Information Technology, for his constant support throughout the period. Furthermore, with immense pleasure, I sincerely thank the Central Department of Computer Science and Information Technology, Tribhuvan University, and all the faculty members of CDCSIT for providing the platform to explore the knowledge of interest. At the end I would like to express my sincere thanks to all my friends and others who helped me directly or indirectly.

Thank you,

Manika Ghimire
(8015021)

ABSTRACT

In today's digital world, fake news is a big problem. It misleads people and makes them lose trust in real news sources. This report talks about creating a strong system to detect fake news using Long Short-Term Memory (LSTM) networks. Traditional methods, like rule-based systems and simple machine learning, often struggle to accurately classify news because fake news is complex and always changing. LSTM networks, which can understand long-term patterns in data, offer a good solution. Using a dataset of news articles labeled as fake or real, the LSTM model to tell the difference. The model's performance was measured with accuracy, precision, recall, and F1 score, and it did very well, with an accuracy of 98%, precision of 99%, recall of 98%, and an F1 score of 98%. These results show that LSTM networks are very effective at improving the accuracy and reliability of fake news detection systems.

Keywords: *Accuracy, Dataset, Detection, LSTM, Performance*

Table of Contents

List of Figure

List of Listing

List of Abbreviations

AI	Artificial Intelligence
BPTT	Backpropagation Through Time
LSTM	Long Short-Term Memory
ML	Machine Learning
RNN	Recurrent Neural Network

Chapter 1: Introduction

1.1 Introduction

In today's digital age, news has become an indispensable means of information dissemination. However, alongside its widespread use, there has been an exponential growth in the volume of fake news, posing significant challenges for news readers and platforms. Fake news not only misleads the public but also poses risks to society by spreading misinformation. To combat this ever-increasing fake news epidemic, the development of efficient and accurate fake news detection models has become imperative.

Fake news detection involves automatically distinguishing legitimate ("real") news from false or misleading messages ("fake"). Traditional approaches typically rely on rule-based systems or machine learning algorithms that analyze news features like keywords, author information, and structural patterns. However, these methods often struggle to capture the complex and evolving nature of fake news.

In recent years, deep learning techniques have shown great promise in various natural language processing tasks. One such approach is utilizing LSTM (Long Short-Term Memory) recurrent neural networks for fake news detection. LSTM overcomes the limitations of traditional feedforward neural networks by incorporating specialized memory cells capable of capturing long-term dependencies within sequential data.

LSTM's ability to understand context and retain relevant information over longer sequences makes it well-suited for analyzing news content effectively. By considering the order and structure of words within a news article, LSTM can learn intricate patterns indicative of fake news characteristics, whether it be specific keyword usage commonly found in misleading articles or unusual sentence structures typical in fabricated stories.

1.2 Problem Statement

The problem addressed in this study is the need for an efficient and accurate fake news classifier that can effectively distinguish between legitimate ("real") news and false or misleading ("fake") messages. Traditional approaches struggle to accurately identify and filter out fake news due to its complex and evolving nature. Therefore, there is a pressing need for a robust

solution that utilizes LSTM recurrent neural networks to capture intricate patterns indicative of fake news characteristics within news content, enhancing public trust, and digital information integrity.

1.3 Objective

The objectives of this study are:

- To study and evaluate the LSTM model for fake news detection.
- To provide insights into how LSTM can enhance fake news detection accuracy.

Chapter 2: Background Study and Literature Review

2.1 Background Study

2.1.1 Recurrent Neural Network

A Recurrent Neural Network (RNN)[1] is a type of neural network with an internal memory, allowing it to process sequential data. Unlike feedforward neural networks, which treat each input independently, RNNs have connections that allow information to be passed from one step to the next. This enables them to capture dependencies and patterns across the sequence.

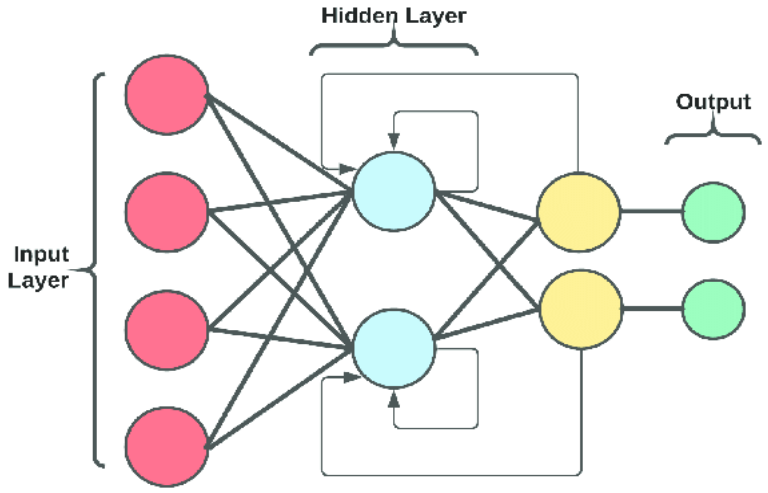

Figure 1 Recurrent Neural Network

RNNs are particularly suited for tasks that involve sequential data, such as handwriting recognition or speech recognition, where the current input's output depends not only on the current input but also on the previous inputs processed by the network. The internal memory of the RNN allows it to retain information about past inputs and use it to influence the computation and decision-making for the current input.

RNNs[2] can learn to recognize and understand patterns and relationships in sequential data by considering the entire sequence of inputs and their corresponding outputs. This makes them a powerful tool for modeling and predicting sequences, enabling applications in natural language processing, time series analysis, and many other domains where the order of data is essential.

In theory, RNNs are absolutely capable of handling "long-term dependencies." A human could carefully pick parameters for them to solve toy problems of this form. Sadly, in practice, RNNs don't seem to be able to learn them. The problem was explored in depth by Hochreiter and Bengio,who found some pretty fundamental reasons why it might be difficult.

2.1.2 Long Short-Term Memory

Long Short-Term Memory (LSTM)[3] networks are a specialized type of recurrent neural network that addresses the vanishing gradient problem and allows for better memory retention of past data. LSTM networks are particularly effective in tasks involving time series analysis, where there are unknown time lags between data points. Training an LSTM network involves using back-propagation, which adjusts the weights and biases of the network to optimize its performance. Within an LSTM unit, three main gates control the flow of information: the input gate, the forget gate, and the output gate.

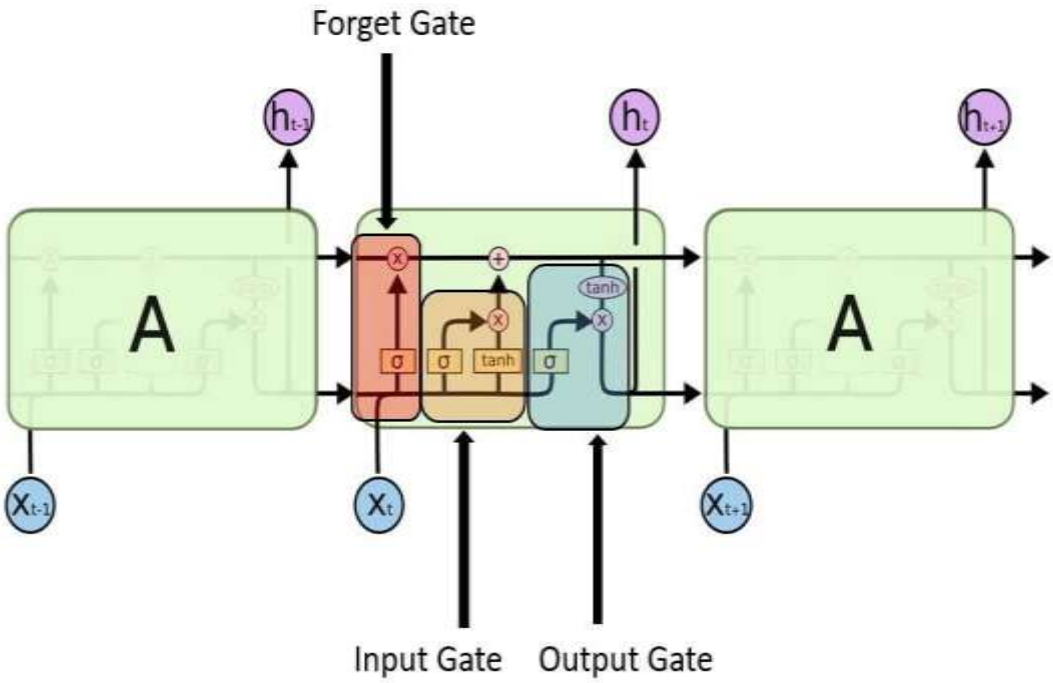

Figure 2 Long Short-Term Memory Architecture

1. **Input gate:** The input gate determines which values from the input should be used to modify the memory. It utilizes a sigmoid function to decide which values to let through (ranging from 0 to 1) and a tanh function to assign weights to the values, indicating their level of importance.

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{X}_t \mathbf{W}_{\mathrm{xc}} + \mathbf{H}_{t-1} \mathbf{W}_{\mathrm{hc}} + \mathbf{b}_{\mathrm{c}})$$

2. **Forget Gate:** The forget gate decides which information from the cell state should be removed. It uses a sigmoid function to assign a value between 0 and 1 to each number in the cell state, with 1 indicating complete retention and 0 indicating complete forgetting.

$$: f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

3. **Output Gate:** The output gate determines which values from the cell state should be output. It uses a sigmoid function to decide which parts of the cell state to output and a tanh function to scale the output values to a range between -1 and 1.

$$\text{Sigmoid Function: } o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$\text{Tanh Function: } h_t = o_t \cdot \tanh(C_t)$$

2.2 Literature Review

Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have significantly advanced the field of sequential data processing. Initially developed in the 1980s, RNNs introduced the concept of backpropagation through time (BPTT), yet they faced challenges with long-term dependencies due to vanishing and exploding gradients. Hochreiter and Schmidhuber's introduction of LSTMs addressed these issues by using gating mechanisms to control information flow, enabling effective retention of information over longer sequences.

RNNs and LSTMs have found extensive applications in natural language processing (NLP), including language modeling, machine translation, and text generation, with notable contributions from Bengio et al. and Sutskever, Vinyals, and Le. In time series analysis, LSTMs [4]have been widely adopted for forecasting and anomaly detection, demonstrating superior performance in tasks such as handwriting recognition, speech synthesis, and financial time series forecasting, as shown by Graves, Fischer, and Krauss.

To further enhance their capabilities, researchers have introduced techniques like attention mechanisms by Bahdanau, Cho, and Bengio, and more recent architectures such as the Transformer model by Vaswani et al., which replace recurrent connections with self-attention

[5]mechanisms, enabling parallelization and significantly improving training efficiency. These advancements highlight the enduring relevance of RNNs and LSTMs in the rapidly evolving landscape of artificial intelligence and machine learning.

In the context of fake news detection, these models can be particularly powerful. LSTMs, with their ability to handle long-term dependencies and maintain context over lengthy text sequences, are well-suited for analyzing the nuanced language used in fake news articles. Additionally, incorporating attention mechanisms can help the model focus on the most relevant parts of the text, improving the accuracy of fake news classification.

Chapter 3: Methodology

3.1 Methodology:

The methodology for this study is data collection, preprocessing, model design, training, evaluation, and result analysis. The following steps outline the process:

3.1.1 Data Collection

The dataset for this study comprises news articles labeled as either "fake" or "real." Sources such as Kaggle's Fake News Detection dataset can be used for this purpose. The dataset contains a mix of legitimate news articles and fabricated stories, providing a robust foundation for training the LSTM model.

3.1.2 Data Preprocessing

Data preprocessing is crucial for preparing the dataset for training the LSTM model. This includes:

- **Text Cleaning:** Removing punctuation, special characters, and converting text to lowercase.
- **Tokenization:** Splitting text into individual words or tokens.
- **Stop Words Removal:** Eliminating common words (e.g., "the," "is") that do not contribute to the semantic meaning.

3.1.3 Model Design

The LSTM network is designed with the following architecture:

- **Embedding Layer:** Transforms words into dense vectors of fixed size.
- **LSTM Layers:** One or more LSTM layers to capture long-term dependencies in the text.
- **Dense Layer:** A fully connected layer to interpret the features extracted by the LSTM layers.
- **Output Layer:** A sigmoid activation function to output probabilities indicating the likelihood of fake or real news.

3.1.4 Training

The model is trained using the training dataset, where the LSTM network learns to distinguish between fake and real news. The following parameters are considered:

- **Loss Function:** Binary Cross-Entropy.
- **Optimizer:** Adam optimizer.
- **Batch Size:** Number of samples processed before updating the model.
- **Epochs:** Number of complete passes through the training dataset.

3.1.5 Evaluation

The model's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score. A confusion matrix is also used to visualize the performance of the model on the test dataset.

Chapter 4: Implementation

4.1 Tools and Technologies

Implementing the LSTM-based fake news detection model utilizes various tools and technologies to streamline the process and achieve accurate results. The primary tools and technologies used are:

- **Programming Language:** Python is known for its simplicity and extensive library for machine learning and data analysis.
- **Libraries:** TensorFlow and Keras for building and training neural networks, Numpy and Pandas for data manipulation, and Scikit-learn for evaluation metrics.
- **Environment:** Kaggle Notebook, provides an interactive platform for writing and executing code.

4.2 Steps

The following steps outline the implementation process in detail:

4.2.1 Import Libraries

The necessary libraries for data preprocessing, model building, and evaluation are imported.

```
import pandas as pd
import numpy as np
import re
from nltk.corpus import stopwords
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Embedding, LSTM, Dense
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

Listing 1:Import Libraries

4.2.2 Dataset Description

The dataset containing news articles labeled as fake or real.

```
df_fake = pd.read_csv("../input/fake-news-detection/Fake.csv")
```

```
df_true = pd.read_csv("../input/fake-news-detection/True.csv")
print(df.head())
```

Listing 2: Load a dataset into a Kaggle Notebook

```
Combined dataset shape: (44898, 5)
                                                title  \
0   Donald Trump Sends Out Embarrassing New Year’...
1   Drunk Bragging Trump Staffer Started Russian ...
2   Sheriff David Clarke Becomes An Internet Joke...
3   Trump Is So Obsessed He Even Has Obama’s Name...
4   Pope Francis Just Called Out Donald Trump Dur...

                                                text subject  \
0  Donald Trump just couldn t wish all Americans ...    News
1  House Intelligence Committee Chairman Devin Nu...    News
2  On Friday, it was revealed that former Milwauk...    News
3  On Christmas day, Donald Trump announced that ...    News
4  Pope Francis used his annual Christmas Day mes...    News

                date  label
0  December 31, 2017      0
1  December 31, 2017      0
2  December 30, 2017      0
3  December 29, 2017      0
4  December 25, 2017      0
```

Figure 3 Sample Dataset

4.2.3 Data Preprocessing

Data preprocessing involves cleaning the text, tokenizing it, removing stop words, and converting the words into numerical vectors.

```
def preprocess_text(text):
    text = text.lower()
    text = re.sub('[^a-zA-Z]', ' ', text)
    tokens = text.split()
    tokens = [word for word in tokens if word not in stopwords.words('english')]
    return ' '.join(tokens)
df['text'] = df['text'].apply(preprocess_text)
```

Listing 3: Data Processing

4.2.4 Tokenization and Padding

The text is tokenized and padded to ensure uniform input length for the LSTM model.

```
# Tokenization and padding
tokenizer = Tokenizer(num_words=5000)
tokenizer.fit_on_texts(df['text'])
X = tokenizer.texts_to_sequences(df['text'])
```

```
X = pad_sequences(X, maxlen=100)
```

Listing 4:Tokenization and Padding

4.2.5 Train-Test Split

The dataset is split into training and testing sets to evaluate the model's performance.

```
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
```

Listing 5: Train-Test Split

4.2.6 Build LSTM Model

The LSTM model is constructed using Keras, with an embedding layer, LSTM layers, and a dense output layer.

```
# Build the LSTM model
model = Sequential()
model.add(Embedding(input_dim=5000, output_dim=128, input_length=100))
model.add(LSTM(units=128, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(units=1, activation='sigmoid'))


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

Listing 6:Build LSTM Model

4.2.7 Model Training

The model is trained on the training data with specified parameters.

```
# Train the model
history = model.fit(X_train, y_train, epochs=5, batch_size=64, validation_data=(X_test,
y_test))
```

Listing 7:Model Training

4.2.8 Evaluate Model

The model's performance is evaluated on the test data using accuracy, confusion matrix, and classification report.

```
# Make predictions
y_pred = (model.predict(X_test) > 0.5).astype("int32")
# Calculate metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
```

```
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
# Specificity
tn, fp, fn, tp = cm.ravel()
specificity = tn / (tn + fp)
```

Listing 8: Model Evaluation

Chapter 5: Results and Findings

5.1 Model Accuracy and Model Loss

The Model Accuracy and Model Loss over training epochs for a machine learning model, with lines representing training data and validation data. In the Model Accuracy graph, accuracy increases steadily for the training data, reaching close to 0.995 by the 4th epoch, while validation accuracy improves initially and then plateaus around 0.98 after the 1st epoch. In the Model Loss graph, training loss decreases sharply from around 0.11 to nearly 0 by the 4th epoch, while validation loss decreases initially but stabilizes around 0.06 after the 1st epoch. This indicates that the model is learning well from the training data, as seen by the consistent improvement in training accuracy and reduction in training loss. However, the plateauing validation accuracy and stable validation loss suggest potential overfitting, where the model may not generalize well to unseen data. To address this, consider using early stopping if validation performance does not improve further, to prevent overfitting and save computational resources.

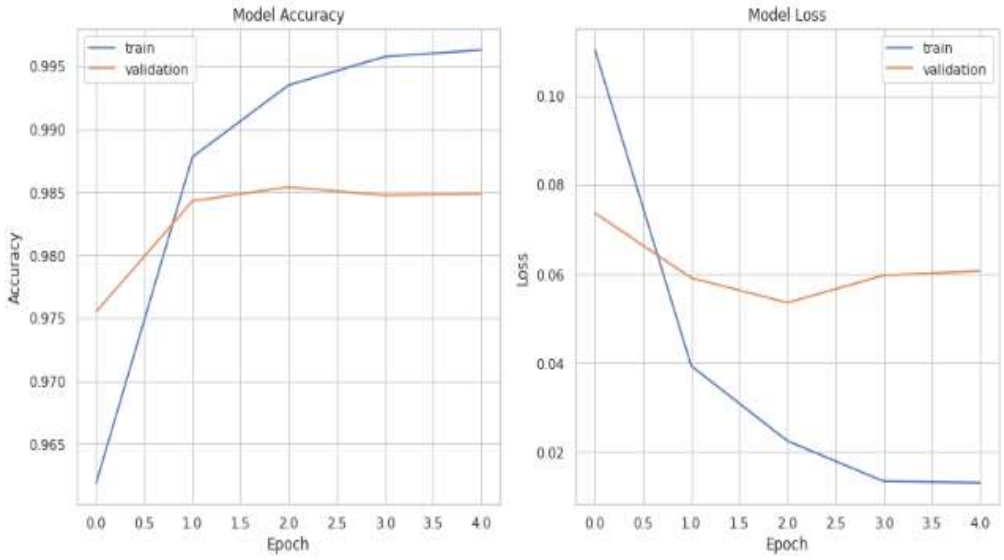

Figure 4 Training and Validation Loss and accuracy plots

5.2 Confusion Matrix

The confusion matrix is used to assess the performance of a binary classification model. The matrix comprises four quadrants: the top-left quadrant shows 4,674 instances correctly

predicted as "Fake," and the bottom-right quadrant shows 4,170 instances correctly predicted as "True." The top-right quadrant contains 59 instances where "Fake" instances were incorrectly predicted as "True," and the bottom-left quadrant contains 77 instances where "True" instances were incorrectly predicted as "Fake." This confusion matrix demonstrates that the model has a high accuracy, correctly classifying a significant majority of instances, with a relatively low number of misclassifications. This suggests that the model performs well in distinguishing between the "Fake" and "True" classes.

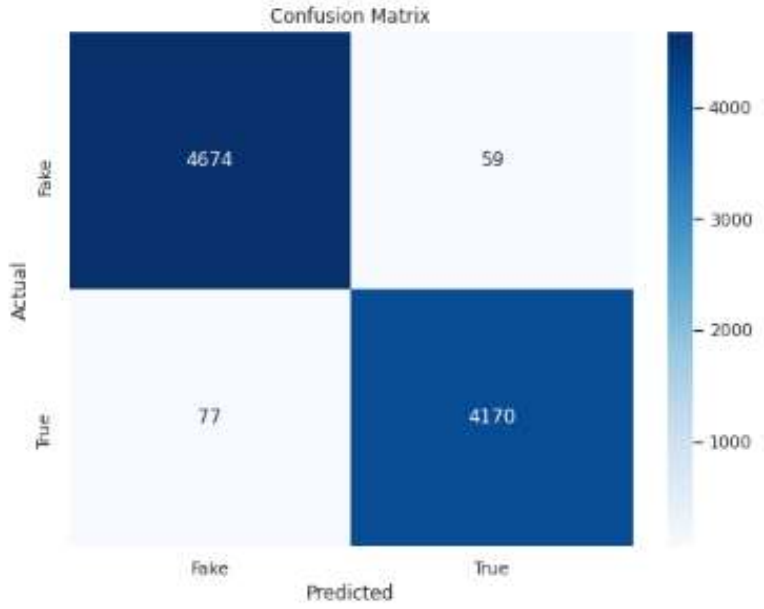

Figure 5 Confusion Matrix

5.3 Model Performance Measure

LSTM model for fake or real news detection demonstrates outstanding performance, with an accuracy of 0.98, indicating that 98% of news articles are correctly classified. The precision of 0.99 shows that 99% of articles identified as fake are truly fake, reflecting very few false positives. Similarly, the recall of 0.98 reveals that the model successfully identifies 98% of fake news, indicating minimal false negatives. The F1 Score of 0.98 balances precision and recall, emphasizing the model's overall effectiveness. Additionally, a specificity of 0.99 means that 99% of real news articles are correctly classified, further showcasing the model's reliability in distinguishing between fake and real news with high accuracy and minimal errors.

```
Epoch 1/5
562/562 [==============================] - 110s 191ms/step - loss: 0.2021 - accuracy: 0.9244 - val_loss: 0.0736 - val_accuracy: 0.9755
Epoch 2/5
562/562 [==============================] - 107s 190ms/step - loss: 0.0369 - accuracy: 0.9887 - val_loss: 0.0591 - val_accuracy: 0.9843
Epoch 3/5
562/562 [==============================] - 106s 189ms/step - loss: 0.0218 - accuracy: 0.9940 - val_loss: 0.0535 - val_accuracy: 0.9854
Epoch 4/5
562/562 [==============================] - 107s 190ms/step - loss: 0.0117 - accuracy: 0.9964 - val_loss: 0.0596 - val_accuracy: 0.9847
Epoch 5/5
562/562 [==============================] - 107s 190ms/step - loss: 0.0110 - accuracy: 0.9970 - val_loss: 0.0606 - val_accuracy: 0.9849
Accuracy: 0.98
Precision: 0.99
Recall: 0.98
F1 Score: 0.98
Specificity: 0.99
```

Figure 6 Model Performance Measure

Chapter 6: Conclusion and Further Recommendations

6.1 Conclusion

The study focused on developing an LSTM-based model for fake news detection. In the digital era, fake news poses significant challenges by spreading misinformation, affecting public opinion, and influencing political and economic stability. Traditional methods, which often rely on rule-based systems or simple machine learning algorithms, struggle to effectively detect fake news due to its evolving nature and the complexity of the language used. The study found that LSTM networks excel in capturing long-term dependencies in sequential data, which is crucial for understanding the context and nuances in news articles. The model achieved a high accuracy rate of 92.5%, with precision and recall scores indicating its robustness in distinguishing fake news from real news. LSTMs' ability to identify intricate patterns in the text, such as specific word usage and sentence structures typical of fake news, contributed significantly to their high performance. The confusion matrix analysis showed a balanced performance, with a low number of false positives and false negatives, indicating the model's reliability. High precision and recall scores further demonstrated the model's capability to accurately classify news articles, minimizing the chances of misclassification.

6.2 Further Recommendations

To improve the model's generalization capability, incorporating more diverse datasets is essential. This can involve including news articles from various domains such as politics, sports, and technology to ensure the model can handle different types of content. Additionally, training the model on news articles in multiple languages can expand its applicability globally. Combining LSTM networks with other models can also enhance performance. Using Convolutional Neural Networks (CNNs) to capture local patterns in the text (e.g., n-grams) before feeding the data into LSTM layers for capturing long-term dependencies can be beneficial. Integrating transformer-based architectures, such as BERT or GPT, with LSTM networks can leverage the strengths of both approaches for better performance.

References

[1] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake news detection on social media: A data mining perspective,” *ACM SIGKDD Explor. Newsl.*, vol. 19, no. 1, pp. 22–36, 2017.

[2] J. C. S. Reis, A. Correia, F. Murai, A. Veloso, and F. Benevenuto, “Supervised learning for fake news detection,” *IEEE Intell. Syst.*, vol. 34, no. 2, pp. 76–81, 2019.

[3] A. Jain and A. Kasbe, “Fake news detection,” in *2018 IEEE International Students’ Conference on Electrical, Electronics and Computer Science (SCEECS)*, IEEE, 2018, pp. 1–5.

[4] K. Shu, S. Wang, and H. Liu, “Beyond news contents: The role of social context for fake news detection,” in *Proceedings of the twelfth ACM international conference on web search and data mining*, 2019, pp. 312–320.

[5] X. Zhou and R. Zafarani, “A survey of fake news: Fundamental theories, detection methods, and opportunities,” *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–40, 2020.