National College of Ireland

Configuration Manual for Multilingual Fake News Detection Using Hybrid Model for Enhanced Computational Efficiency and Performance in Low-Resource Languages

MSc Research Project
Data Analytics

Srinadh Pippalla
Student ID: 23317728

School of Computing
National College of Ireland

Supervisor: Hamilton Niculescu

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Srinadh Pippalla

Student ID: 23317728

Programme: Data Analytics **Year:** 2025

Module: Research Project

Lecturer: Hamilton Niculescu

Submission Due Date: 11/08/2025

Project Title: Multilingual Fake News Detection Using Hybrid Model for Enhanced Computational Efficiency and Performance in Low-Resource Languages

Word Count: 686 **Page Count:** 8

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Srinadh Pippalla

Date: 11/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual for Multilingual Fake News Detection Using Hybrid Model for Enhanced Computational Efficiency and Performance in Low-Resource Languages

Srinadh Pippalla
23317728

1. Introduction

This manual provides step-by-step guidance for setting up the environment, dependencies, and configurations needed to replicate and run the research project titled **"Multilingual Fake News Detection Using Hybrid Model for Enhanced Computational Efficiency and Performance in Low-Resource Languages."**

This study presents a multilingual fake news detection system utilizing classical machine learning, deep learning (GRU and CNN), and a hybrid model integrating **DeBERTa** with **BiLSTM**. The goal is to enhance both performance and computational efficiency, particularly for **low-resource languages** such as Swahili, Hindi, and Vietnamese.

The dataset used includes multilingual fake and real news articles across seven languages. The study also includes data cleaning, text preprocessing, language filtering, visualization, and comprehensive model evaluation.

2. System Hardware Requirements

Before beginning the setup, please ensure the following system specifications:

- Operating System: Ubuntu 20.04/22.04, macOS, or Windows 10/11 (Linux-based preferred)
- Processor: Intel Core i5/i7 or AMD Ryzen 5 or above
- RAM: 16 GB minimum (32 GB recommended for transformer-based models)
- GPU: NVIDIA GPU with CUDA support (Recommended for DeBERTa+LSTM and faster training)
- Storage: Minimum 10 GB of free disk space for data, libraries, and model checkpoints

3. Software Requirements:

The following software packages and tools must be installed:

- Python Version: Python 3.7 or higher
- Development Tools:
  - Jupyter Notebook, VS Code, or PyCharm (for running and editing scripts)
  - Anaconda (optional, for environment management)

Required Python Libraries:

Install these libraries via pip or conda:

```
pip install langdetect wordcloud transformers keras tensorflow scikit-learn matplotlib
```

- pandas, numpy – For data manipulation and numerical operations
- matplotlib, seaborn – For data visualization and plotting
- langdetect – For detecting language of each text entry
- nltk – For tokenization and stopwords removal (requires downloading resources)
- wordcloud – For generating visual word clouds
- scikit-learn – For classical machine learning models and metrics
- tensorflow, keras – For building deep learning and hybrid models
- transformers – For loading DeBERTa pretrained model and tokenizer
- tqdm – For progress bars during embedding extraction

Download NLTK Resources:

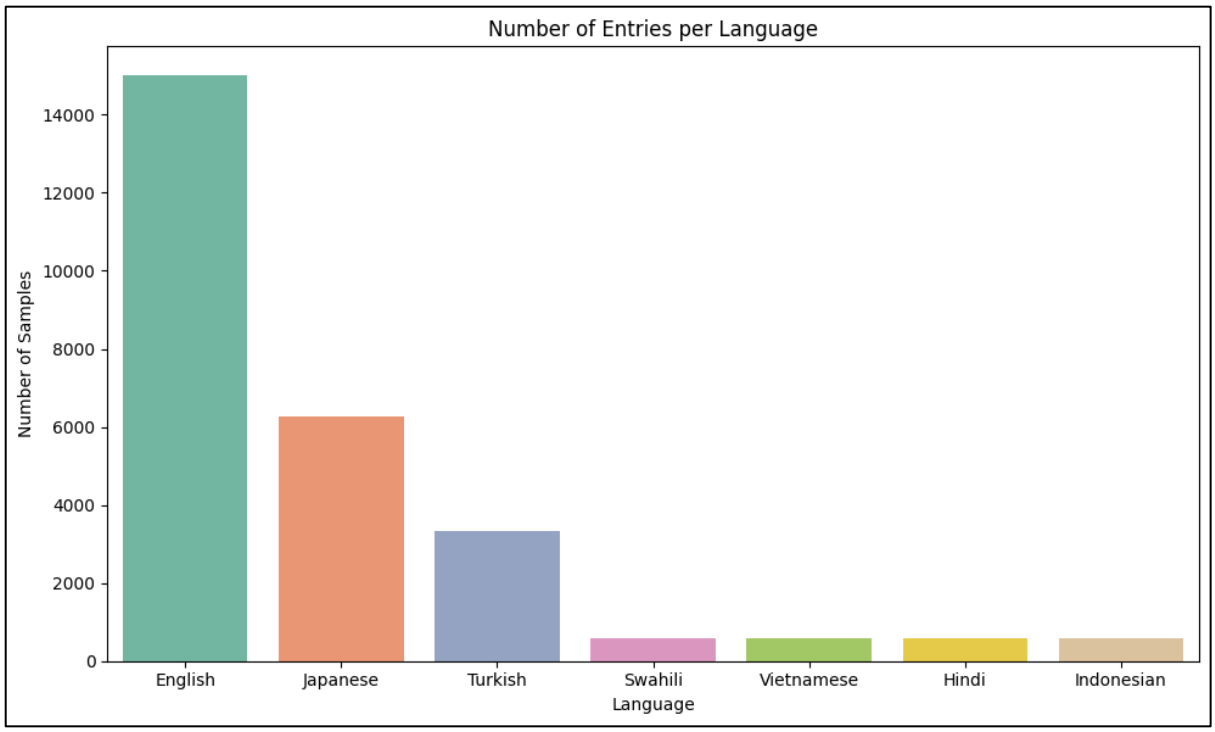
```
import nltk
nltk.download('punkt')
nltk.download('stopwords')
```

4. Dataset Details

Dataset Source:
The dataset used in this research is a cleaned and preprocessed multilingual fake news dataset that includes labeled news texts in seven languages which is gathered from the kaggle:

- English (en)
- Hindi (hi)
- Indonesian (id)
- Japanese (ja)
- Swahili (sw)
- Turkish (tr)
- Vietnamese (vi)

Dataset: https://www.kaggle.com/datasets/begonil/multilingual-fake-news-detection

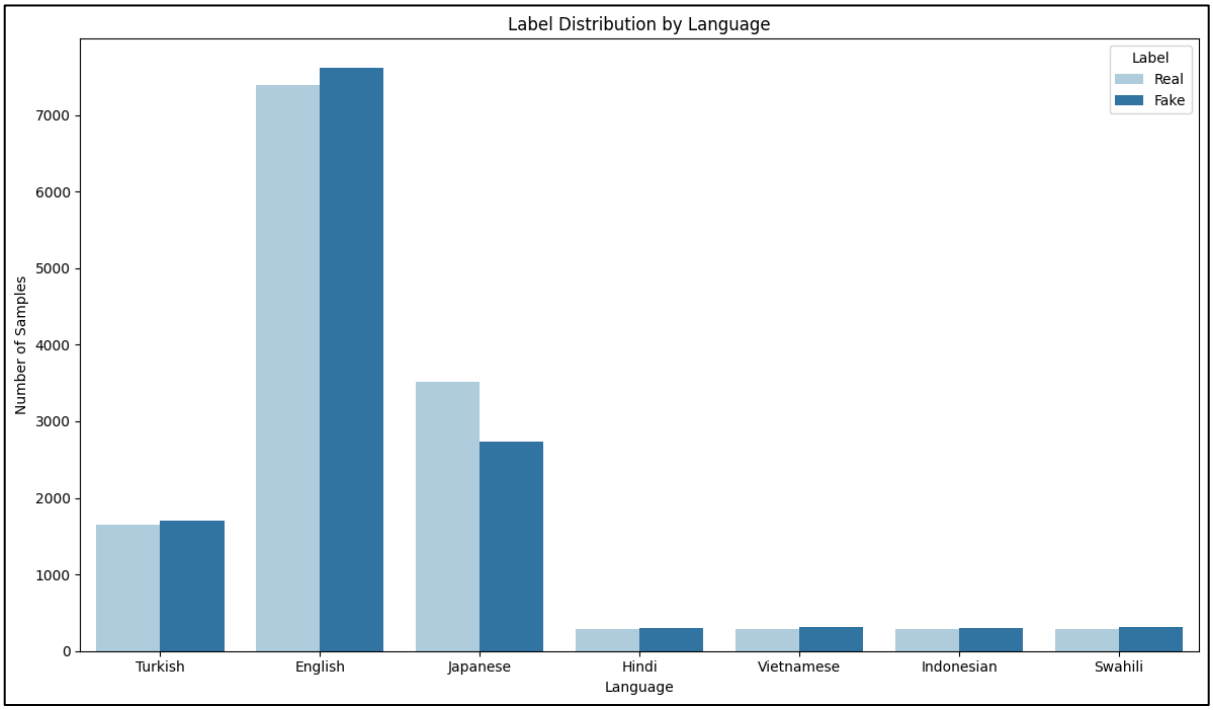


Dataset Format:

- File Name: cleaned_multilingual_fake_news.csv
- Columns Required:
  - text – News content
  - labels – Binary label (0 = real, 1 = fake)
  - Language – Language code for the news entry

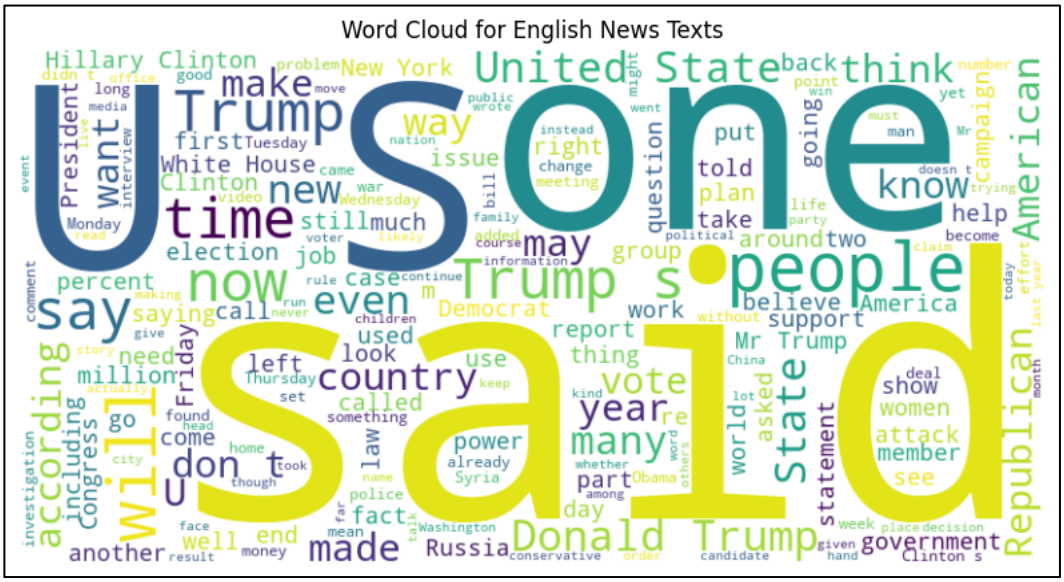
Ensure the CSV file is placed in the root directory of your project before running the code.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26960 entries, 0 to 26959
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Unnamed: 0     26960 non-null  int64 
 1   text           26960 non-null  object
 2   labels         26960 non-null  int64 
 3   Language       26960 non-null  object
 4   text_length    26960 non-null  int64 
 5   Language_Full  26960 non-null  object
dtypes: int64(3), object(3)
memory usage: 1.2+ MB
None
```



5. Model Configuration

This study implements and compares the following models:



A. Classical Machine Learning Models

- Logistic Regression
- Naive Bayes
- Random Forest

```python
# machine learning models
models = {
    "Logistic Regression": LogisticRegression(max_iter=2),
    "Naive Bayes": MultinomialNB(alpha=100),
    "Random Forest": RandomForestClassifier(n_estimators=2,max_depth=3,random_state=42)
}
```

```python
# evaluate the machine learning models
model_scores = {}
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    model_scores[name] = acc
    print(f"\n{name} Accuracy: {acc:.4f}")
    target_names = ['English', 'Hindi', 'Indonesian', 'Japanese', 'Swahili', 'Turkish', 'Vietnamese']
    print(classification_report(y_test, y_pred, target_names=target_names))
```

Input:

- TF-IDF vectorized cleaned_text
- Labels: Encoded language classes (Language_encoded)

B. Deep Learning Models

- GRU (Gated Recurrent Unit)

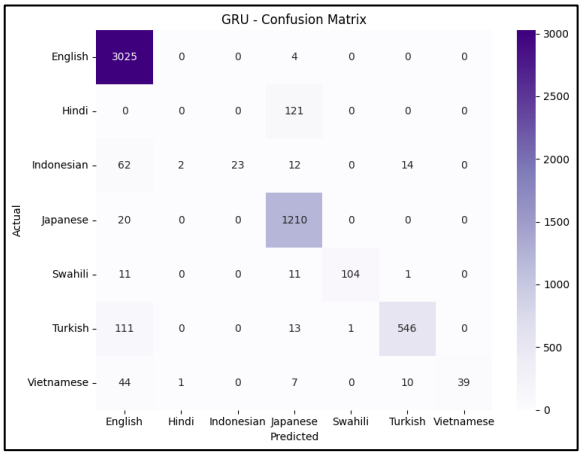
```python
# Define model with GRU
model_gru = Sequential([
    Embedding(input_dim=max_words, output_dim=128, input_length=max_len),
    GRU(64),
    Dense(7, activation='softmax')  # 7 output classes
])

# Compile for multiclass classification
model_gru.compile(
    optimizer='adam',
    loss='sparse_categorical_crossentropy',  # use 'categorical_crossentropy' if one-hot labels
    metrics=['accuracy']
)

# Fit model
model_gru.fit(
    X_train_dl,
    y_train_dl,
    epochs=1,
    batch_size=512,
    validation_split=0.2
)
```

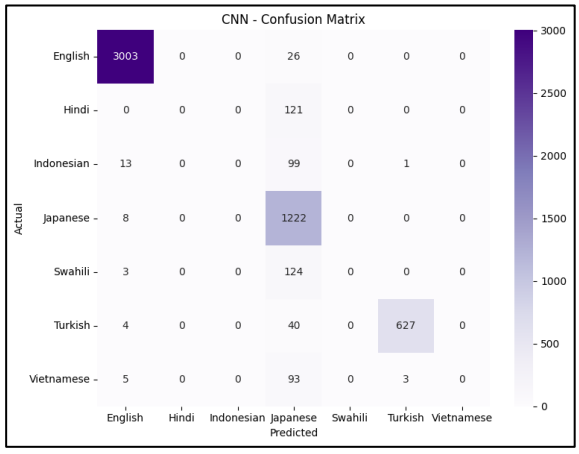
- CNN (Convolutional Neural Network)

```python
# Define model CNN Model
model_cnn = Sequential([
    Embedding(input_dim=max_words, output_dim=128, input_length=max_len),
    Conv1D(128, 5, activation='relu'),
    GlobalMaxPooling1D(),
    Dense(7, activation='softmax')  # 7 output classes
])

# Compile for multiclass classification
model_cnn.compile(
    optimizer='adam',
    loss='sparse_categorical_crossentropy',  # use categorical_crossentropy if labels are one-hot encoded
    metrics=['accuracy']
)

# Fit model
model_cnn.fit(
    X_train_dl,
    y_train_dl,
    epochs=1,
    batch_size=512,
    validation_split=0.2
)
```

Configuration:

- Tokenizer: Keras Tokenizer with 10,000 max words
- Sequence Length: 250
- Embedding Dim: 128
- Batch Size: 512





C. Hybrid Model – DeBERTa + BiLSTM

- Uses microsoft/deberta-base from HuggingFace Transformers
- DeBERTa is frozen to reduce training time
- Embeddings are passed to a Bidirectional LSTM classifier

```python
# Load tokenizer and frozen DeBERTa
tokenizer = DebertaTokenizerFast.from_pretrained("microsoft/deberta-base")
deberta = TFDebertaModel.from_pretrained("microsoft/deberta-base")
deberta.trainable = False  # freeze it

MAX_LEN = 128
```

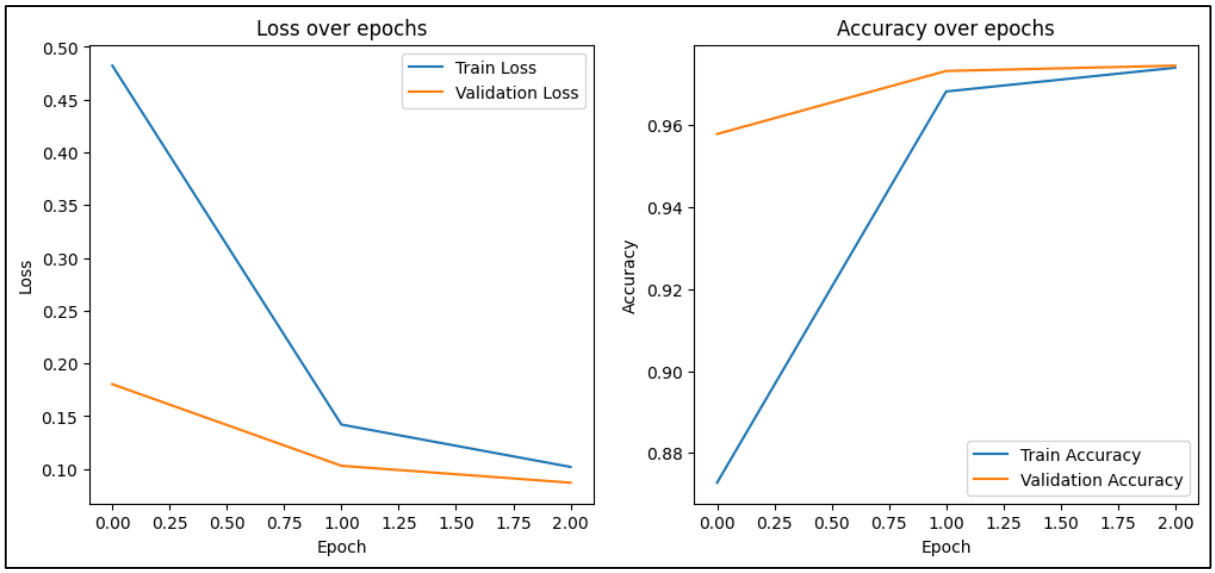
Configuration:

- Max Token Length: 128

- LSTM Units: 128
- Dense Layer Units: 128
- Dropout: 0.3
- Epochs: 3
- Batch Size: 32
- Loss Function: categorical_crossentropy
- Optimizer: Adam (2e-5 learning rate)

```python
# Define LSTM Classifier Model
lstm_model = tf.keras.Sequential([
    tf.keras.layers.Input(shape=(MAX_LEN, 768)),
    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(128, return_sequences=False)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Dense(num_classes, activation='softmax')
])

lstm_model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=2e-5),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

```python
# Train the Model
history = lstm_model.fit(
    X_train_seq, y_train_cat,
    validation_data=(X_val_seq, y_val_cat),
    batch_size=32,
    epochs=3
)
```

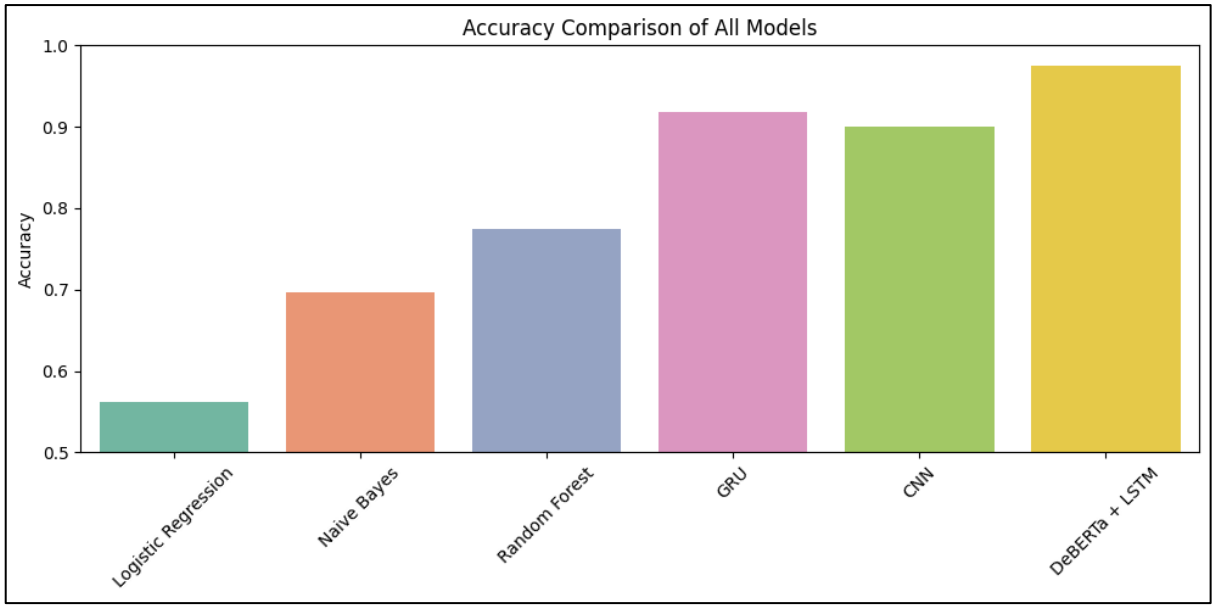


6. Model Evaluation Metrics

Each model is evaluated using:

- **Accuracy**
- **Confusion Matrix**
- **Classification Report (Precision, Recall, F1-score)**

A final bar chart is generated comparing all model accuracies.



7. Conclusion

This configuration manual provides a comprehensive guideline to set up and run the multilingual fake news detection models presented in the research. By following the steps above, users can replicate the experiments, compare model performances, and extend the study to additional languages or datasets. For best results, use a GPU-enabled system for training the DeBERTa + LSTM model.

References

Python: https://www.python.org
Transformers Library: https://huggingface.co/transformers
Dataset Source: https://www.kaggle.com/datasets/begonil/multilingual-fake-news-detection
DeBERTa Base Model: https://huggingface.co/microsoft/deberta-base
TensorFlow: https://www.tensorflow.org/
NLTK: https://www.nltk.org/