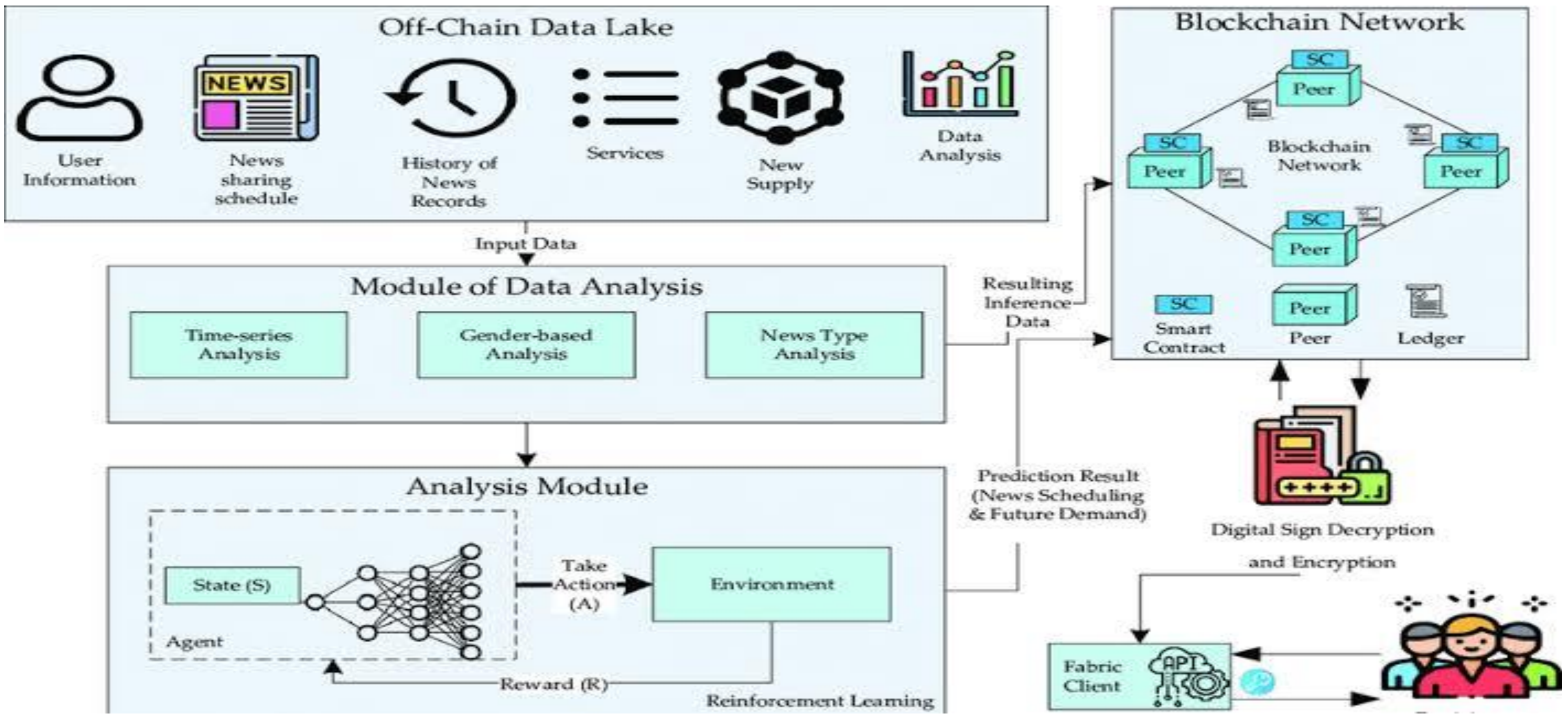
FAKE NEWS DETECTION USING NLP



Detecting fake news using Natural Language Processing (NLP) involves analyzing text data to identify misleading or false information. Here’s a simplified process:

1. **Data Collection**: Gather a dataset of news articles, both real and fake, for training and testing.

2. **Preprocessing**: Clean and preprocess the text data, including tasks like lowercasing, tokenization, and removing stop words and punctuation.

3. **Feature Extraction**: Convert the text data into numerical features that NLP models can understand. Techniques like TF-IDF or word embeddings (e.g., Word2Vec, GloVe) are common choices.

4. **Evaluation**: Assess the model’s performance using metrics like accuracy, precision, recall, and F1-score. Cross-validation and testing on different datasets are important.

5. **Fine-Tuning**: Adjust the model’s hyperparameters, consider ensemble methods, or experiment with different NLP techniques to improve accuracy.

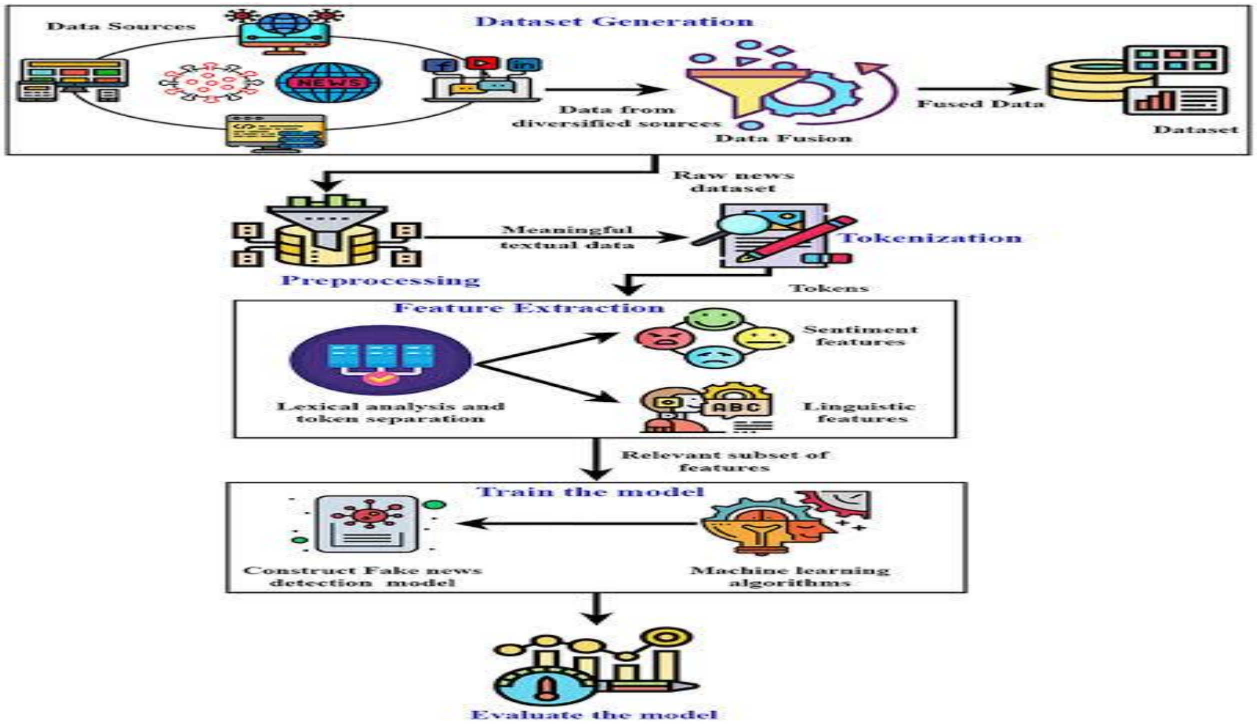
6. **Real-time Monitoring**: Implement the model in a system that can continuously monitor news sources and classify incoming articles.

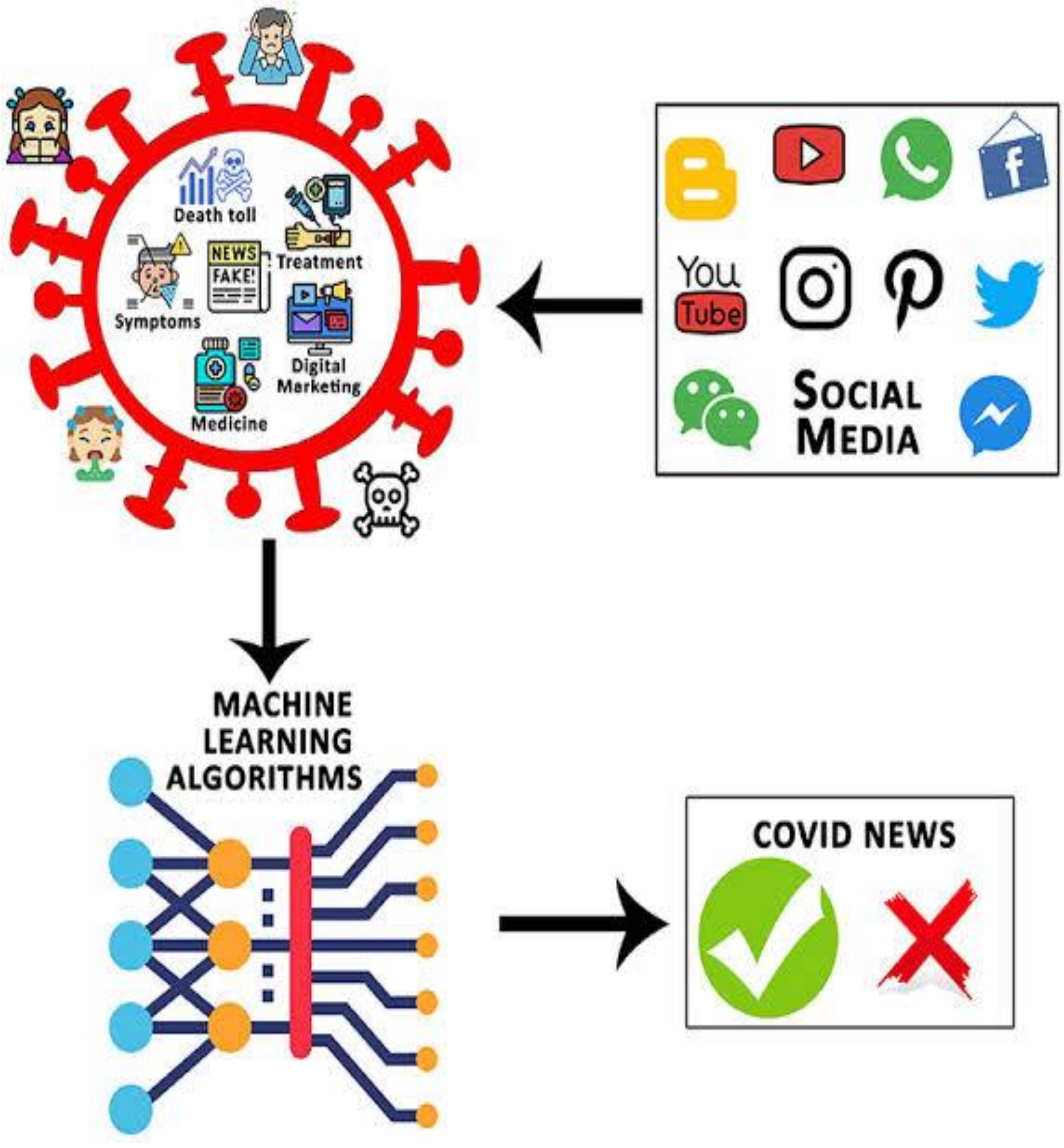
7. **User Interface**: Develop a user-friendly interface for users to fact-check articles by inputting URLs or text.

8. **Combating Bias**: Be aware of bias in data and models, and employ techniques to mitigate it.

9. **Stay Updated**: Fake news evolves, so it’s crucial to keep the model updated with the latest data and trends.

Remember that no model is perfect, and it’s essential to combine NLP techniques with human judgment and other fact-checking methods to effectively combat fake news.





Death toll
Treatment
NEWS
FAKE!
Symptoms
Digital Marketing
Medicine
You Tube
SOCIAL MEDIA
MACHINE LEARNING ALGORITHMS
COVID NEWS

PROGRAM:

```
Import pandas as pd
From sklearn.model_selection import train_test_split
From sklearn.feature_extraction.text import TfidfVectorizer
From sklearn.naive_bayes import MultinomialNB
From sklearn.metrics import accuracy_score, classification_report

# Load and preprocess your dataset
Data = pd.read_csv('fake_news_dataset.csv')
X = data['text']
Y = data['label']

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# TF-IDF vectorization
Tfidf_vectorizer = TfidfVectorizer(max_features=5000)
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)

# Train a classifier (e.g., Multinomial Naïve Bayes)
Clf = MultinomialNB()
Clf.fit(X_train_tfidf, y_train)

# Predict and evaluate
Y_pred = clf.predict(X_test_tfidf)
Accuracy = accuracy_score(y_test, y_pred)
Print("Accuracy:", accuracy)
Print(classification_report(y_test, y_pred))
```