Fake And Real News

Fake And Real News

Project Overview

Fake News Classification

In today's digital era, misinformation spreads rapidly across various platforms, making it crucial to develop automated methods for identifying fake news.

This project aims to build a machine learning and deep learning-based classifier that can distinguish between real and fake news articles using Natural Language Processing (NLP) techniques.

Class Distribution

- Real News: 19,999 articles
- Fake News: 19,943 articles

The dataset is well-balanced, meaning both categories have a nearly equal number of samples, reducing bias in model training.

Dataset Summary

The dataset consists of 39,942 news articles, labeled as either real (1) or fake (0).

Each article includes a title, full text, subject category, and publication date.

The goal is to develop a model that can accurately classify news articles based on their textual content.

Fake News Classification

In today's digital era, misinformation spreads rapidly across various platforms, making it crucial to develop automated methods for identifying fake news.

This project aims to build a machine learning and deep learning-based classifier that can distinguish between real and fake news articles using Natural Language Processing (NLP) techniques.

Dataset Summary

The dataset consists of 39,942 news articles, labeled as either real (1) or fake (0).

Each article includes a title, full text, subject category, and publication date.

The goal is to develop a model that can accurately classify news articles based on their textual content.

Class Distribution

- Real News: 19,999 articles
- Fake News: 19,943 articles

The dataset is well-balanced, meaning both categories have a nearly equal number of samples, reducing bias in model training.

Fake And Real News

OUR IMPLEMENTATION

Project Pipline

- load Dataset
- Drop Duplicate Rows
- Check for Null Values
- Identify Rare Words
- Preprocess Text Data
- Split Data
- Apply TF-IDF
- Train and Evaluate the models

Project Pipline

- load Dataset
- Drop Duplicate Rows
- Check for Null Values
- Identify Rare Words
- Preprocess Text Data
- Split Data
- Apply TF-IDF
- Train and Evaluate the models

Fake And Real News

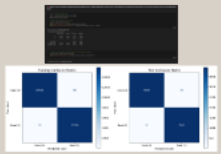
Exploratory Data Analysis

Modeling Techniques

ML Models:

ML Models

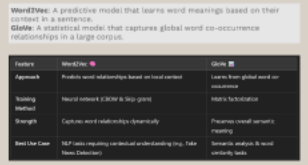
LR Model



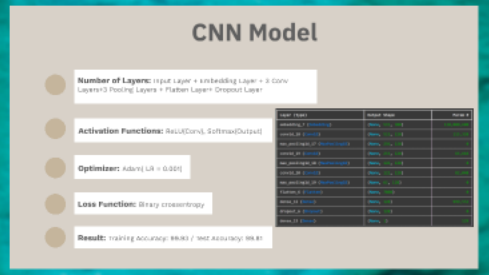
SVM Model

DL Models:

GloVe vs. Word2Vec

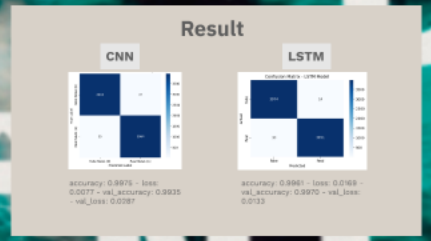


CNN Model

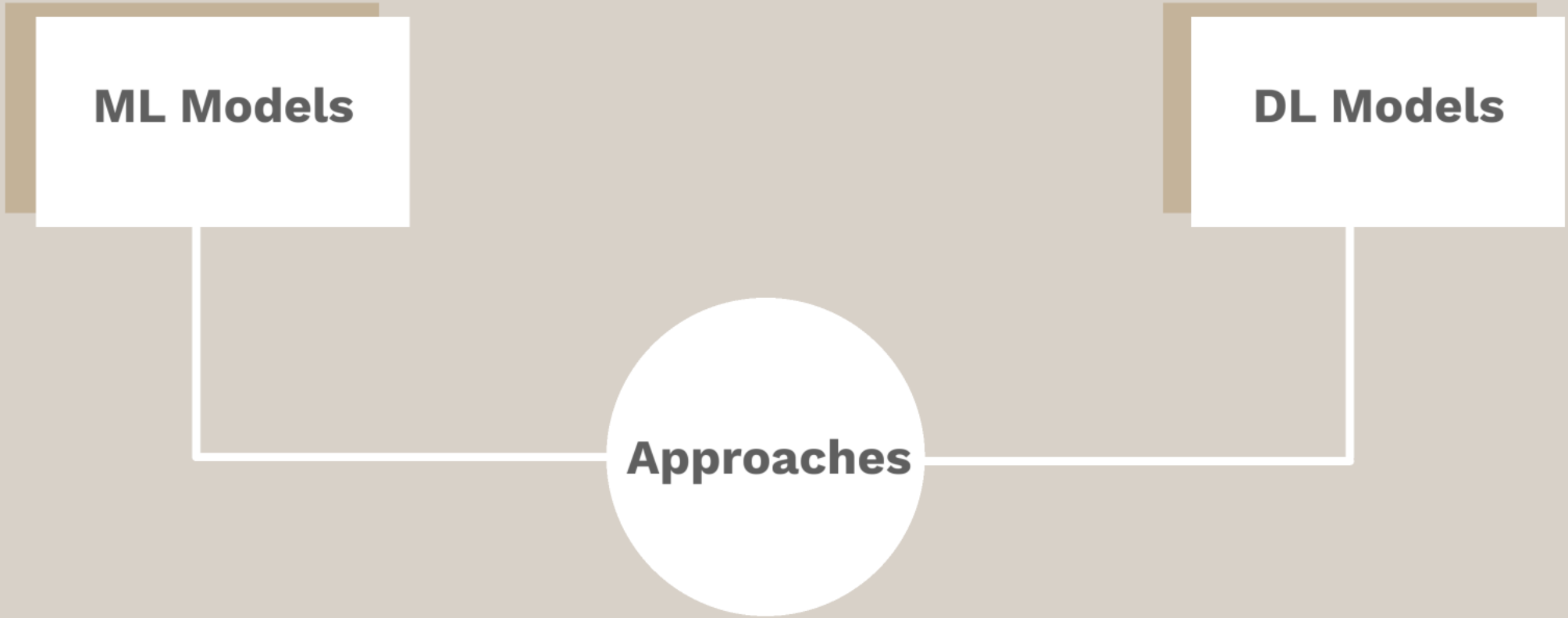


LSTM Model

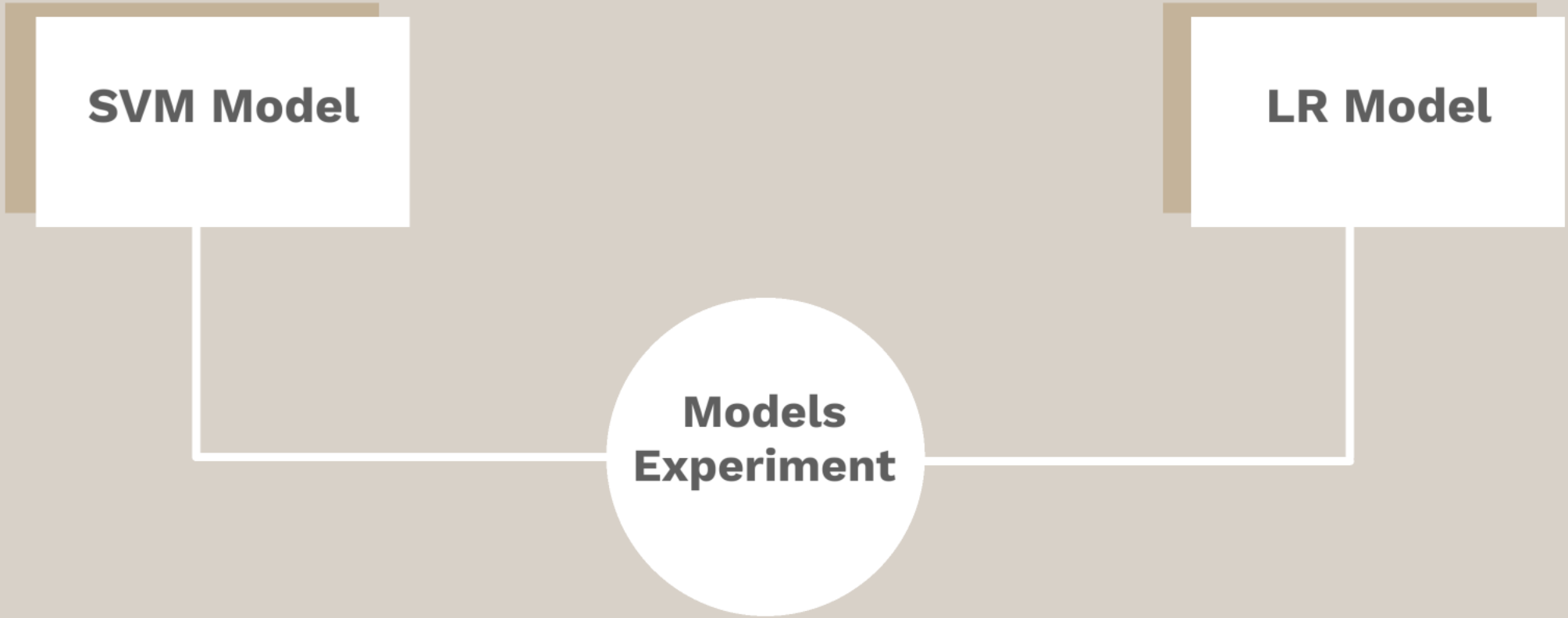
Result



Modeling Techniques



ML Models:



ML Models

📌 Step 4: Split Data into Training and Test Sets

Split the dataset into training (80%) and test (20%) sets before feature extraction to prevent data leakage.

```
# Define features (raw processed text) and labels
X_title = df['title_processed']
X_text = df['text_processed']
y = df['label']

# Split the data
X_title_train, X_title_test, X_text_train, X_text_test, y_train, y_test = train_test_split(
    X_title, X_text, y, test_size=0.2, random_state=42
)

print("Training set size:", len(X_title_train))
print("Test set size:", len(X_title_test))
```

```
Training set size: 31792
Test set size: 7949
```

#📌 Step 5: Apply TF-IDF Feature Extraction Fit the TF-IDF vectorizer on the training data only, then transform both training and test sets to create feature matrices for `title` and `text`.

```
# Initialize TF-IDF vectorizers
tfidf_title = TfidfVectorizer(max_features=5000, min_df=5, ngram_range=(1, 2))
tfidf_text = TfidfVectorizer(max_features=10000, min_df=5, ngram_range=(1, 2))

# Fit and transform on training data
title_tfidf_train = tfidf_title.fit_transform(X_title_train)
text_tfidf_train = tfidf_text.fit_transform(X_text_train)

# Transform test data (do NOT fit on test data)
title_tfidf_test = tfidf_title.transform(X_title_test)
text_tfidf_test = tfidf_text.transform(X_text_test)

# Combine features for training and test sets
X_train = hstack([title_tfidf_train, text_tfidf_train])
X_test = hstack([title_tfidf_test, text_tfidf_test])

print("Training feature matrix shape:", X_train.shape)
print("Test feature matrix shape:", X_test.shape)
```

Python

```
Training feature matrix shape: (31792, 15000)
Test feature matrix shape: (7949, 15000)
```

LR Model

 # 📌 Step 6: Train and Evaluate Logistic Regression Model Train a Logistic Regression model on the TF-IDF features and evaluate its performance on the test set using accuracy and classification report.

```python
# Train Logistic Regression
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

# Predict and evaluate on test data
y_pred = model.predict(X_test)
print("Test Accuracy:", accuracy_score(y_test, y_pred))
print("Test Classification Report:\n", classification_report(y_test, y_pred))
```

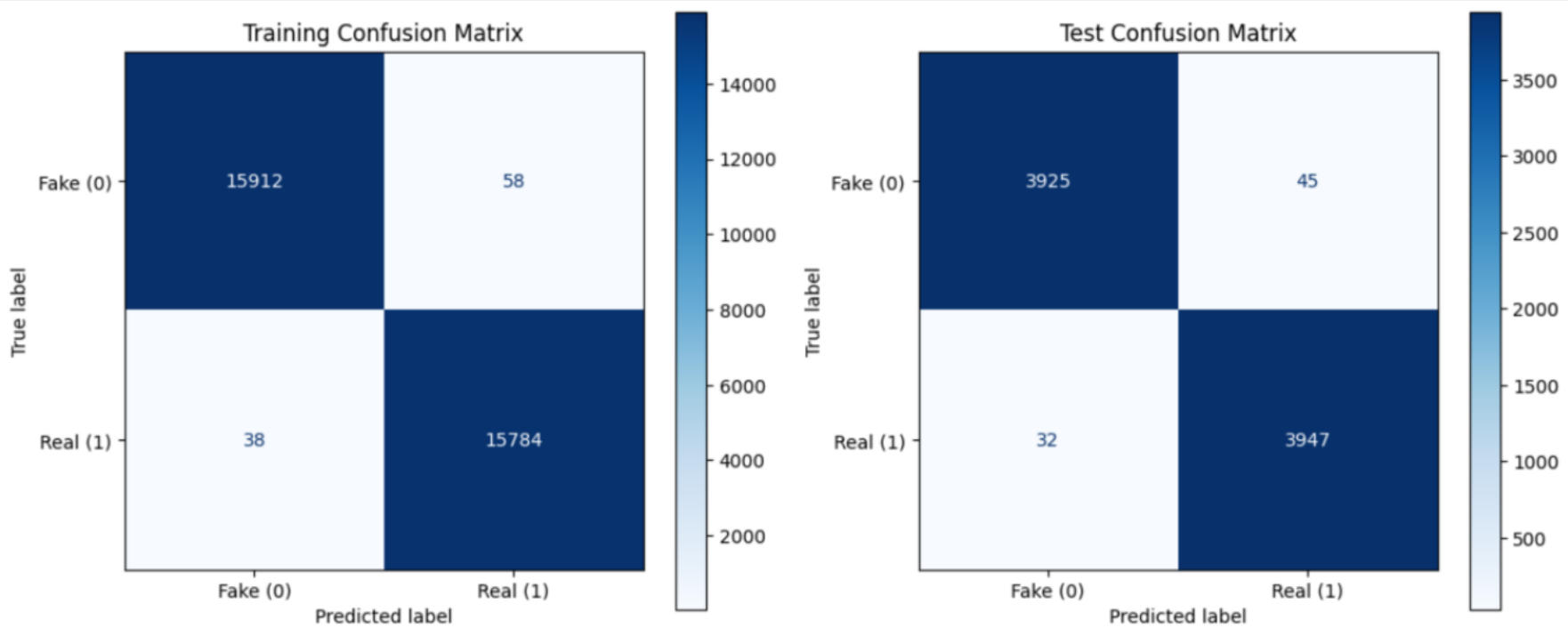
Python

```
Test Accuracy: 0.9903132469493018
Test Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      3970
           1       0.99      0.99      0.99      3979

    accuracy                           0.99      7949
   macro avg       0.99      0.99      0.99      7949
weighted avg       0.99      0.99      0.99      7949
```

```python
# Predict and evaluate on training data (for overfitting check)
y_train_pred = model.predict(X_train)
print("\nTraining Accuracy:", accuracy_score(y_train, y_train_pred))
```

Python

```
Training Accuracy: 0.9969803724207348
```



SVM Model

```python
from sklearn.svm import SVC

# Train SVM model
svm_model = SVC(kernel='rbf', random_state=42)  # You can adjust kernel (e.g., 'linear', 'rbf', 'poly')
svm_model.fit(X_train, y_train)

# Predict and evaluate on test data
y_pred_svm = svm_model.predict(X_test)
print("Test Accuracy (SVM):", accuracy_score(y_test, y_pred_svm))
print("Test Classification Report (SVM):\n", classification_report(y_test, y_pred_svm))
```

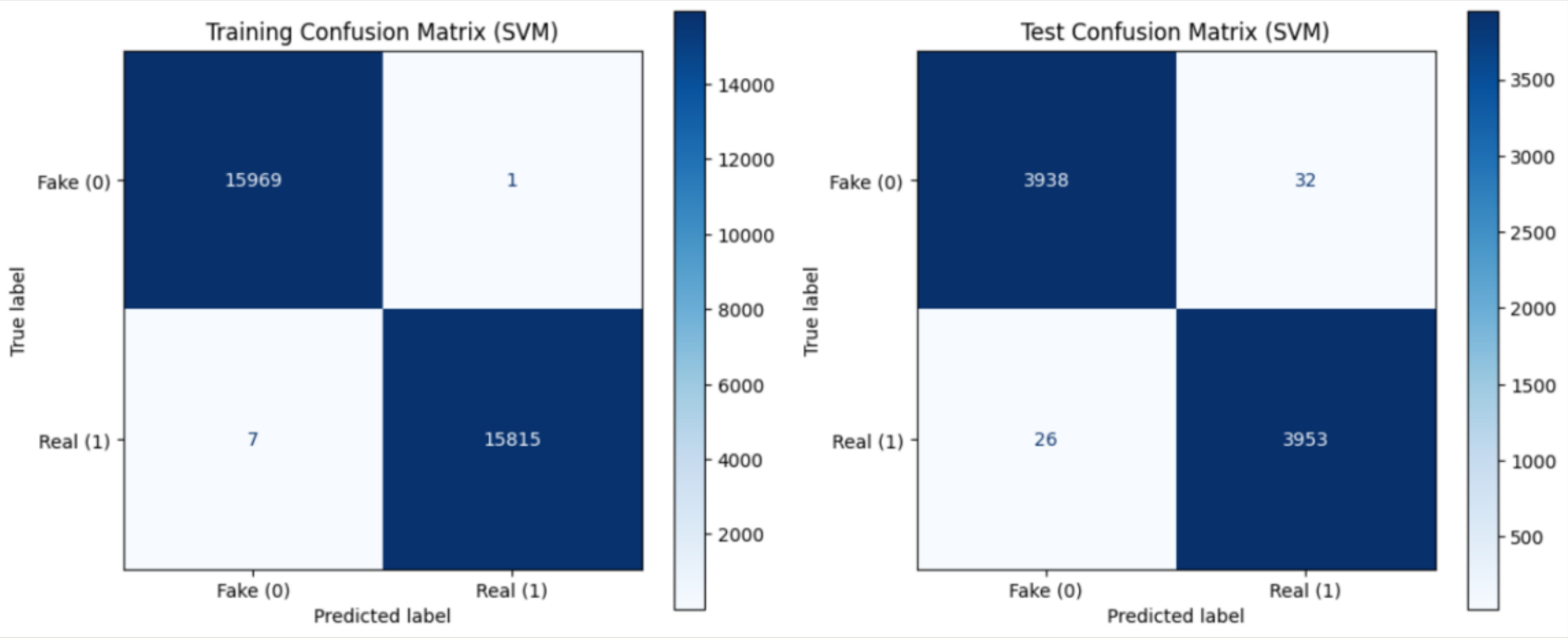
Python

```
Test Accuracy (SVM): 0.9927034847150585
Test Classification Report (SVM):
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      3970
           1       0.99      0.99      0.99      3979

    accuracy                           0.99      7949
   macro avg       0.99      0.99      0.99      7949
weighted avg       0.99      0.99      0.99      7949
```

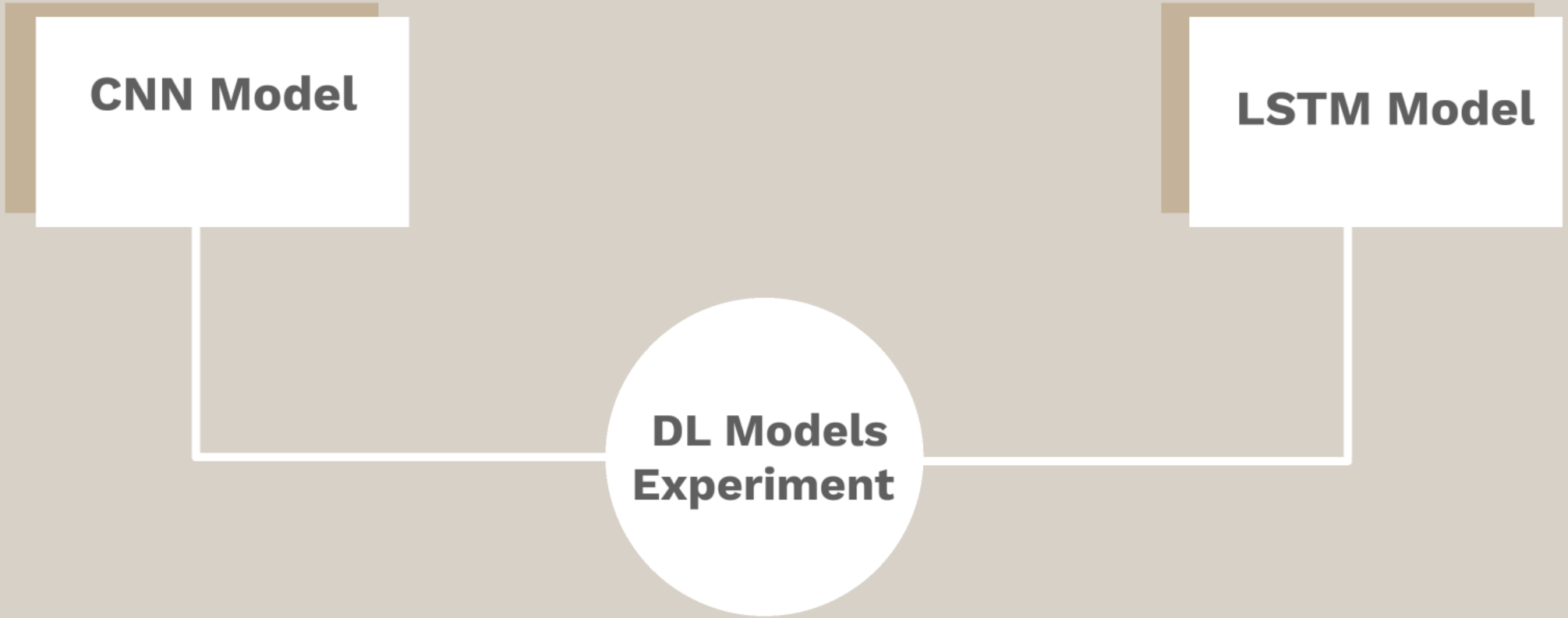
```python
# Predict and evaluate on training data (for overfitting check)
y_train_pred_svm = svm_model.predict(X_train)
print("\nTraining Accuracy (SVM):", accuracy_score(y_train, y_train_pred_svm))
```

Python

```
Training Accuracy (SVM): 0.9997483643683945
```



DL Models:



GloVe vs. Word2Vec

Word2Vec: A predictive model that learns word meanings based on their context in a sentence.
GloVe: A statistical model that captures global word co-occurrence relationships in a large corpus.

Feature	Word2Vec 🧠	GloVe 📊
Approach	Predicts word relationships based on local context	Learns from global word co-occurrence
Training Method	Neural network (CBOW & Skip-gram)	Matrix factorization
Strength	Captures word relationships dynamically	Preserves overall semantic meaning
Best Use Case	NLP tasks requiring contextual understanding (e.g., Fake News Detection)	Semantic analysis & word similarity tasks

CNN Model

Number of Layers: Input Layer + Embedding Layer + 3 Conv Layers+3 Pooling Layers + Flatten Layer+ Dropout Layer

Activation Functions: ReLU(Conv), Softmax(Output)

Optimizer: Adam(LR = 0.001)

Loss Function: Binary crossentropy

Result: Training Accuracy: 99.93 / Test Accuracy: 99.81

Layer (type)	Output Shape	Param #
embedding_7 (Embedding)	(None, 515, 300)	120,000,600
conv1d_18 (Conv1D)	(None, 513, 128)	115,328
max_pooling1d_17 (MaxPooling1D)	(None, 256, 128)	0
conv1d_19 (Conv1D)	(None, 253, 128)	65,664
max_pooling1d_18 (MaxPooling1D)	(None, 126, 128)	0
conv1d_20 (Conv1D)	(None, 122, 128)	82,048
max_pooling1d_19 (MaxPooling1D)	(None, 61, 128)	0
flatten_6 (Flatten)	(None, 7808)	0
dense_12 (Dense)	(None, 128)	999,552
dropout_6 (Dropout)	(None, 128)	0
dense_13 (Dense)	(None, 1)	129

LSTM Model

Number of Layers: Input Layer + Embedding Layer + 2 LSTM Layers Flatten Layer+ 2 Dropout Layer

Activation Functions: Softmax(Output)

Optimizer: Adam

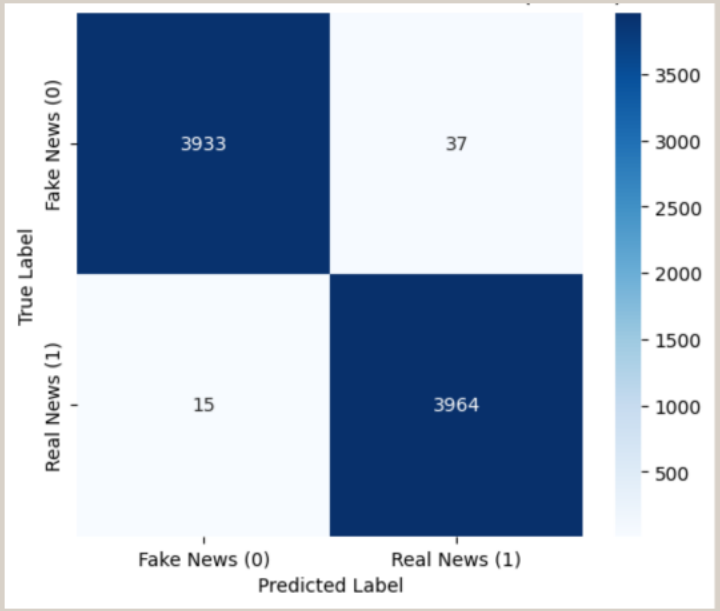
Loss Function: Binary crossentropy

Result: Training Accuracy: 99.61 / Test Accuracy: 99.73

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 517, 100)	15,602,300
lstm (LSTM)	(None, 517, 128)	117,248
dropout (Dropout)	(None, 517, 128)	0
lstm_1 (LSTM)	(None, 64)	49,408
dropout_1 (Dropout)	(None, 64)	0
dense (Dense)	(None, 1)	65

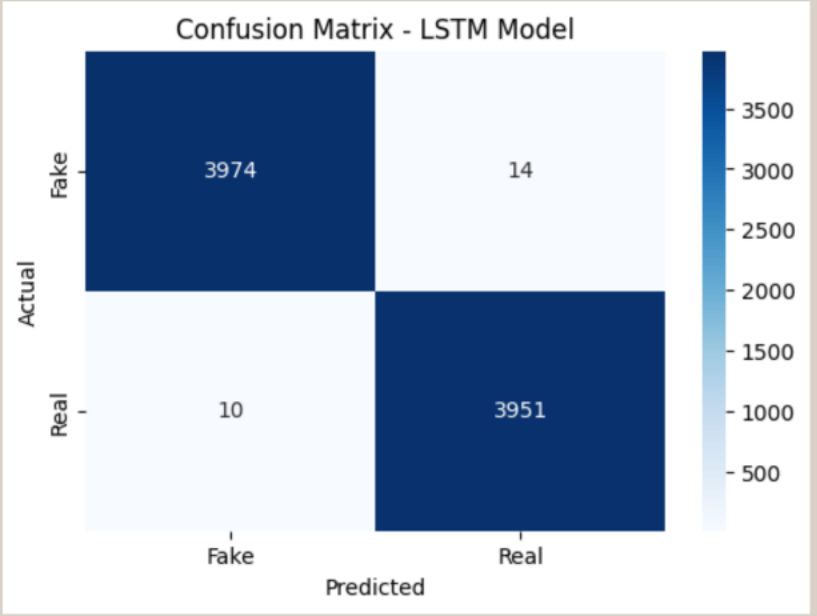
Result

CNN



accuracy: 0.9975 - loss: 0.0077 - val_accuracy: 0.9935 - val_loss: 0.0287

LSTM



accuracy: 0.9961 - loss: 0.0169 - val_accuracy: 0.9970 - val_loss: 0.0133

Conclusion and Insights

Streamlit Devlopment

Our " Fake News Detector " project, allowing users to easily check the credibility of news articles in real time.

Give it a Try!

Scan the QR to Try Our Application

Conclusion

In summary, our "Fake and Real News" project uses machine learning to help people tell the difference between real and fake news. By providing quick checks of news articles, we raise awareness about misinformation.

This approach encourages critical thinking and helps people become smarter consumers of news, empowering them to make informed choices.

AuraNeura Group

Team Members :

- Mayar Fawaz althebati
- Shatha Saleh Alqubaisi
- Raghad Khaled Almutairi
- Wareef Yousef Alqurashi

Malala Yousafzai shot, critically injured

Nation prays for Malala

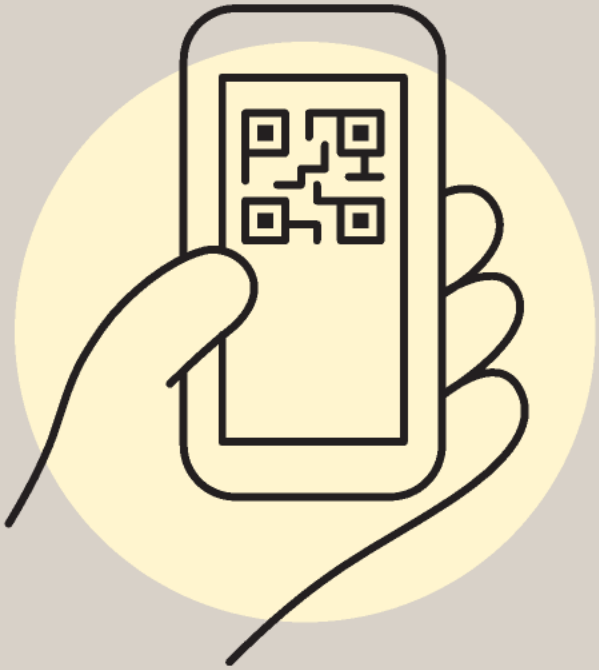
claims responsibility; president, PM, Nawaz, Imran, others condemn attack

sincere talks with

Streamlit Devlopment

Our " Fake News Detector " project, allowing users to easily check the credibility of news articles in real time.

Give it a Try!



Scan the QR to Try Our Application

Conclusion

In summary, our "Fake and Real News" project uses machine learning to help people tell the difference between real and fake news. By providing quick checks of news articles, we raise awareness about misinformation.

This approach encourages critical thinking and helps people become smarter consumers of news, empowering them to make informed choices.

AuraNeura Group

Team Members :

- Mayar Fawaz althebati
- Shatha Saleh Alqubaisi
- Raghad Khaled Almutairi
- Wareef Yousef Alqurashi

Fake And Real News