

Comparison of LSTM, BiLSTM, GRU, BiGRU, and CNN Models

Classification Analysis of Sentiment on Tweets Regarding President Putin's
Participation in Indonesia's G20 with the BiGRU Method

1. LSTM

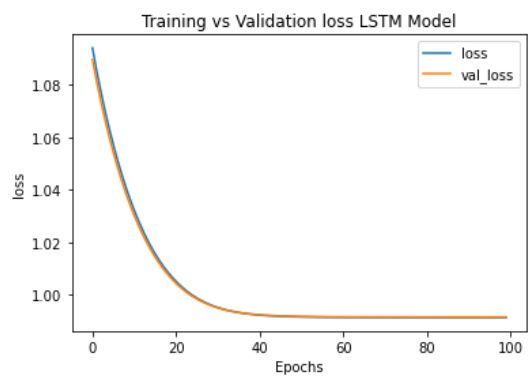
a. LSTM Model

```
def model_lstm(num_words, embedding_dim, maxlen):  
  
    tf.random.set_seed(123)  
    lstm_dim = 32  
    dense_dim = 6  
  
    modelLSTM = tf.keras.Sequential([  
        tf.keras.layers.Embedding(num_words, embedding_dim,  
input_length=maxlen),  
        tf.keras.layers.LSTM(lstm_dim),  
        tf.keras.layers.Dense(dense_dim, activation='relu'),  
        tf.keras.layers.Dense(3, activation='softmax')  
    ])  
  
    modelLSTM.compile(loss='sparse_categorical_crossentropy',  
                      optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),  
                      metrics=['accuracy'])  
  
    return modelLSTM  
  
modelLSTM = model_lstm(NUM_WORDS, EMBEDDING_DIM, MAXLEN)  
modelLSTM.summary()
```

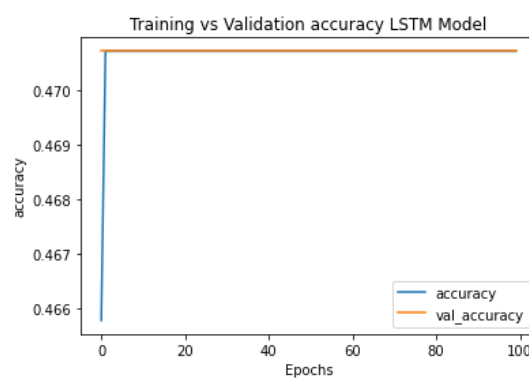
b. Model.fit

```
historyLSTM = modelLSTM.fit(train_padded_seq,  
                            train_label_seq,  
                            epochs=100,  
                            validation_data=(val_padded_seq, val_label_seq))
```

c. Results



Training and Validation Loss for LSTM Model



Training and Validation Accuracy for LSTM Model

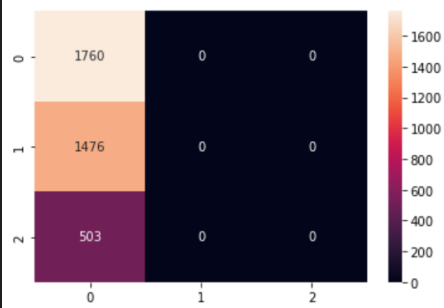
Best Validation Accuracy : 0.4707
On Epoch : 1

Confusin Matrix :

```
[[1760  0  0]
 [1476  0  0]
 [ 503  0  0]]
```

	precision	recall	f1-score	support
0	0.47	1.00	0.64	1760
1	0.00	0.00	0.00	1476
2	0.00	0.00	0.00	503
accuracy			0.47	3739
macro avg	0.16	0.33	0.21	3739
weighted avg	0.22	0.47	0.30	3739

0.47071409225463867
Accuracy is: 0.4707140946777213



2. BiLSTM

a. BiLSTM Model

```
def model_BiLSTM(num_words, embedding_dim, maxlen):

    tf.random.set_seed(123)
    lstm_dim = 32
    dense_dim = 6

    modelBiLSTM = tf.keras.Sequential([
        tf.keras.layers.Embedding(num_words, embedding_dim,
input_length=maxlen),
        tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(lstm_dim)),
        tf.keras.layers.Dense(dense_dim, activation='relu'),
        tf.keras.layers.Dense(3, activation='softmax')
    ])

    modelBiLSTM.compile(loss='sparse_categorical_crossentropy',
                        optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),
                        metrics=['accuracy'])

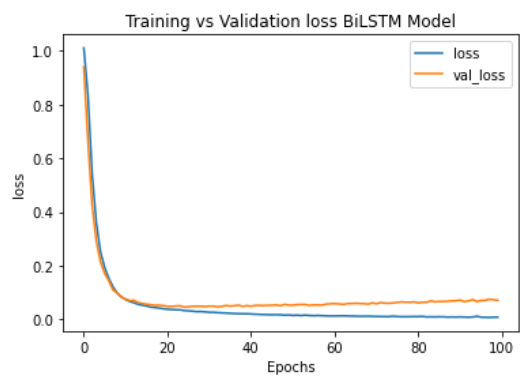
    return modelBiLSTM

modelBiLSTM = model_BiLSTM(NUM_WORDS, EMBEDDING_DIM, MAXLEN)
modelBiLSTM.summary()
```

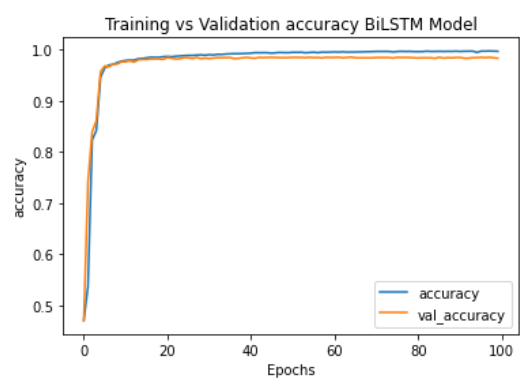
b. ModelCheckpoint & Model.fit

```
mc = ModelCheckpoint('classifier-putinG20-BiLSTM_01.h5',
monitor='val_accuracy', mode='max', verbose=1, save_best_only=True)
historyBiLSTM = modelBiLSTM.fit(train_padded_seq,
                                train_label_seq,
                                epochs=100,
                                callbacks=[mc],
                                validation_data=(val_padded_seq,
val_label_seq))
```

c. Result



Training and Validation Loss for BiLSTM Model



Training and Validation Accuracy for BiLSTM Model

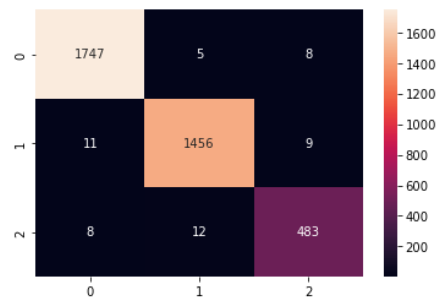
Best Validation Accuracy : 0.98583
On Epoch : 65

Confusin Matrix :

```
[[1747  5  8]
 [ 11 1456  9]
 [  8 12 483]]
```

	precision	recall	f1-score	support
0	0.99	0.99	0.99	1760
1	0.99	0.99	0.99	1476
2	0.97	0.96	0.96	503
accuracy			0.99	3739
macro avg	0.98	0.98	0.98	3739
weighted avg	0.99	0.99	0.99	3739

0.9858250617980957
Accuracy is: 0.9858250869216368



3. GRU

a. GRU Model

```
def model_GRU(num_words, embedding_dim, maxlen):

    tf.random.set_seed(123)
    gru_dim = 32
    dense_dim = 6

    modelGRU = tf.keras.Sequential([
        tf.keras.layers.Embedding(num_words, embedding_dim,
input_length=maxlen),
        tf.keras.layers.GRU(gru_dim),
        tf.keras.layers.Dense(dense_dim, activation='relu'),
        tf.keras.layers.Dense(3, activation='softmax')
    ])

    modelGRU.compile(loss='sparse_categorical_crossentropy',
                      optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),
                      metrics=['accuracy'])

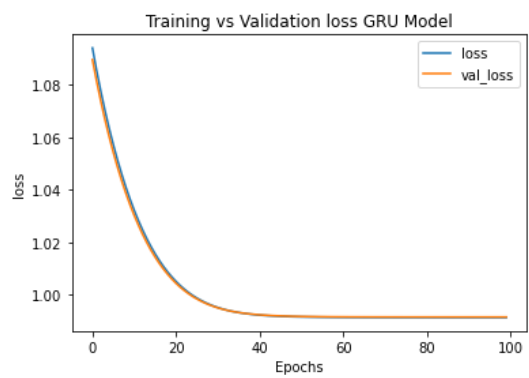
    return modelGRU

modelGRU = model_GRU(NUM_WORDS, EMBEDDING_DIM, MAXLEN)
modelGRU.summary()
```

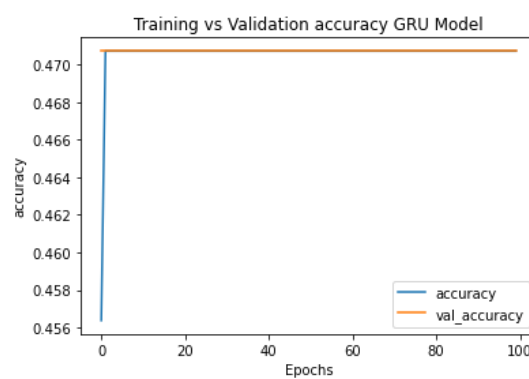
b. Model.fit

```
historyGRU = modelGRU.fit(train_padded_seq,
                           train_label_seq,
                           epochs=100,
                           validation_data=(val_padded_seq, val_label_seq))
```

c. Result



Training and Validation Loss for GRU Model

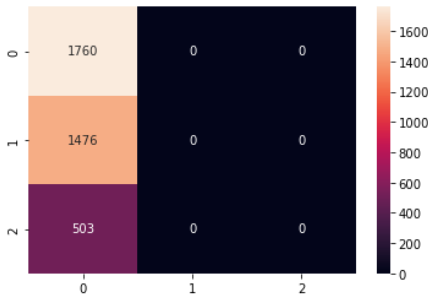


Training and Validation Accuracy for GRU Model

Best Validation Accuracy : 0.4707
On Epoch : 1

Confusin Matrix :

[[1760 0 0] [1476 0 0] [503 0 0]]					
	precision	recall	f1-score	support	
0	0.47	1.00	0.64	1760	
1	0.00	0.00	0.00	1476	
2	0.00	0.00	0.00	503	
accuracy			0.47	3739	
macro avg	0.16	0.33	0.21	3739	
weighted avg	0.22	0.47	0.30	3739	
0.47071409225463867					
Accuracy is: 0.4707140946777213					



4. BiGRU

a. BiGRU Model

```
def model_BiGRU(num_words, embedding_dim, maxlen):

    tf.random.set_seed(123)
    gru_dim = 32
    dense_dim = 6

    modelBiGRU = tf.keras.Sequential([
        tf.keras.layers.Embedding(num_words, embedding_dim,
input_length=maxlen),
        tf.keras.layers.Bidirectional(tf.keras.layers.GRU(gru_dim)),
        tf.keras.layers.Dense(dense_dim, activation='relu'),
        tf.keras.layers.Dense(3, activation='softmax')
    ])

    modelBiGRU.compile(loss='sparse_categorical_crossentropy',
                        optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),
                        metrics=['accuracy'])

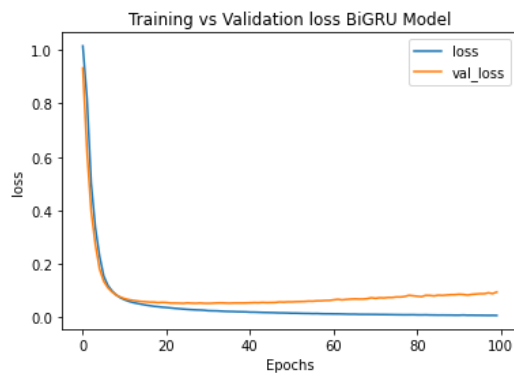
    return modelBiGRU

modelBiGRU = model_BiGRU(NUM_WORDS, EMBEDDING_DIM, MAXLEN)
modelBiGRU.summary()
```

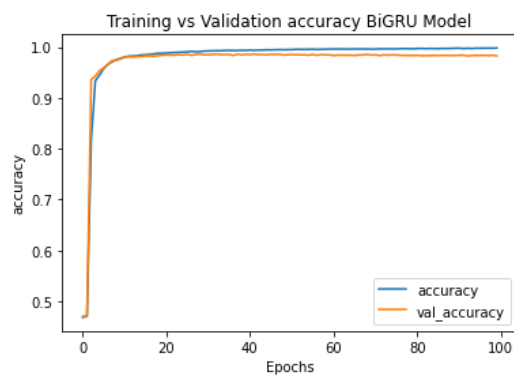
b. ModelCheckpoint & Model.fit

```
mc = ModelCheckpoint('classifier-putinG20-BiGRU_04.h5',
monitor='val_accuracy', mode='max', verbose=1, save_best_only=True)
historyBiGRU = modelBiGRU.fit(train_padded_seq,
                              train_label_seq,
                              epochs=100,
                              callbacks=[mc],
                              validation_data=(val_padded_seq, val_label_seq))
```

c. Result



Training and Validation Loss for BiGRU Model

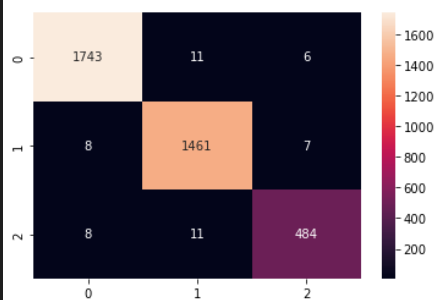


Training and Validation Accuracy for BiGRU Model

Best Validation Accuracy : 0.9864
On Epoch : 29

Confusin Matrix :

[[1743 11 6]					
[8 1461 7]					
[8 11 484]]					
	precision	recall	f1-score	support	
0	0.99	0.99	0.99	1760	
1	0.99	0.99	0.99	1476	
2	0.97	0.96	0.97	503	
accuracy			0.99	3739	
macro avg	0.98	0.98	0.98	3739	
weighted avg	0.99	0.99	0.99	3739	
0.9863600134849548					
Accuracy is: 0.9863599893019523					



5. CNN

a. CNN Model

```
def model_conv(num_words, embedding_dim, maxlen):

    tf.random.set_seed(123)
    filters = 64
    kernel_size = 3
    dense_dim = 6

    modelConv = tf.keras.Sequential([
        tf.keras.layers.Embedding(num_words, embedding_dim,
input_length=maxlen),
        tf.keras.layers.Conv1D(filters, kernel_size, activation='relu'),
        tf.keras.layers.GlobalAveragePooling1D(),
        tf.keras.layers.Dense(dense_dim, activation='relu'),
        tf.keras.layers.Dense(3, activation='sigmoid')
    ])

    modelConv.compile(loss='sparse_categorical_crossentropy',
                      optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),
                      metrics=['accuracy'])

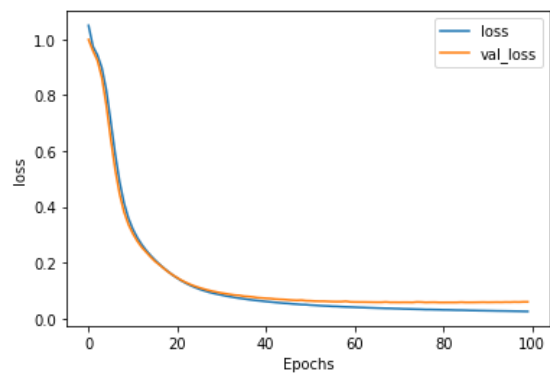
    return modelConv

modelConv = model_conv(NUM_WORDS, EMBEDDING_DIM, MAXLEN)
modelConv.summary()
```

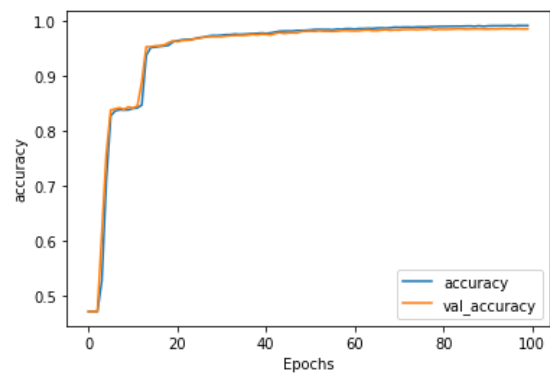
b. ModelCheckpoint & Model.fit

```
mc = ModelCheckpoint('classifier-putinG20-CNN_01.h5', monitor='val_accuracy',
mode='max', verbose=1, save_best_only=True)
historyConv = modelConv.fit(train_padded_seq,
                           train_label_seq,
                           epochs=100,
                           callbacks=[mc],
                           validation_data=(val_padded_seq, val_label_seq))
```

c. Result



Training and Validation Loss for CNN Model

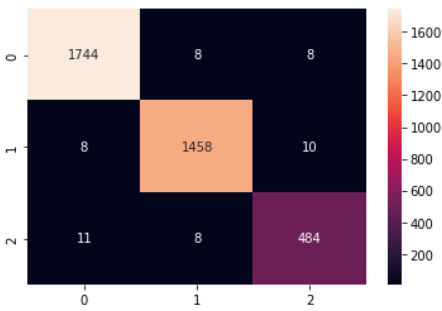


Training and Validation Accuracy for CNN Model

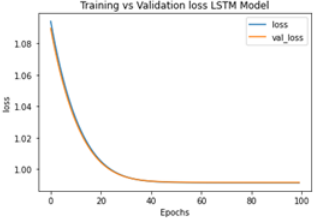
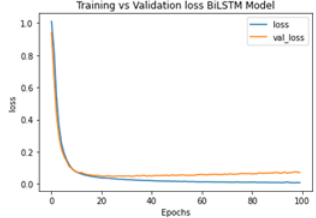

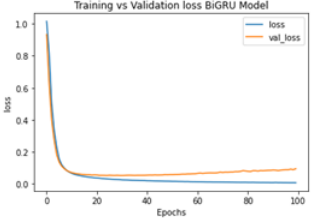
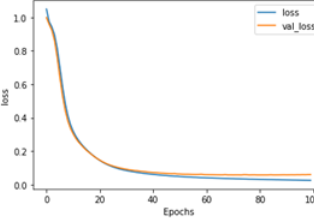
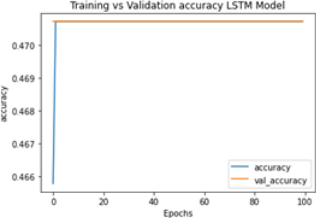
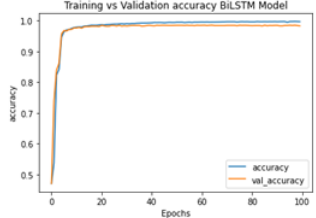
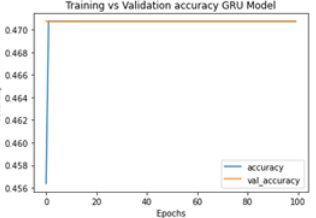
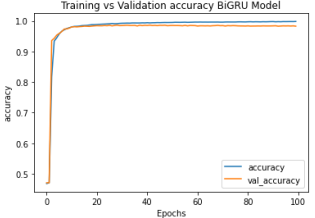
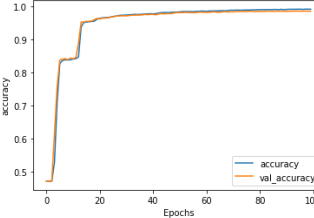
Best Validation Accuracy : 0.98583
On Epoch : 94

Confusin Matrix :

[[1744 8 8]					
[8 1458 10]					
[11 8 484]]					
	precision	recall	f1-score	support	
0	0.99	0.99	0.99	1760	
1	0.99	0.99	0.99	1476	
2	0.96	0.96	0.96	503	
accuracy			0.99	3739	
macro avg	0.98	0.98	0.98	3739	
weighted avg	0.99	0.99	0.99	3739	
0.9858250617980957					
Accuracy is: 0.9858250869216368					



SUMMARY

Model (Top Layer)	LSTM	BiLSTM	GRU	BiGRU	CNN
Optimizer	Adam	Adam	Adam	Adam	Adam
Learning Rate	1e-4	1e-4	1e-4	1e-4	1e-4
Best Validation Accuracy On Epoch	0.4707	0.98583	0.4707	0.9864	0.98583
	1	65	1	29	94
Graph Train & Val Loss					
Graph Train & Val Accuracy					
Confusion Matrix	<pre>[[1760 0 0] [1476 0 0] [503 0 0]] precision recall f1-score support 0 0.47 1.00 0.64 1760 1 0.00 0.00 0.00 1476 2 0.00 0.00 0.00 503 accuracy 0.47 3739 macro avg 0.16 0.33 0.21 3739 weighted avg 0.22 0.47 0.30 3739 0.47071409225463867 Accuracy is: 0.4707140946777213</pre>	<pre>[[1747 5 8] [11 1456 9] [8 12 483]] precision recall f1-score support 0 0.99 0.99 0.99 1760 1 0.99 0.99 0.99 1476 2 0.97 0.96 0.96 503 accuracy 0.99 3739 macro avg 0.98 0.98 0.98 3739 weighted avg 0.99 0.99 0.99 3739 0.9858250617980957 Accuracy is: 0.9858250869216368</pre>	<pre>[[1760 0 0] [1476 0 0] [503 0 0]] precision recall f1-score support 0 0.47 1.00 0.64 1760 1 0.00 0.00 0.00 1476 2 0.00 0.00 0.00 503 accuracy 0.47 3739 macro avg 0.16 0.33 0.21 3739 weighted avg 0.22 0.47 0.30 3739 0.47071409225463867 Accuracy is: 0.4707140946777213</pre>	<pre>[[1743 11 6] [8 1461 7] [8 11 484]] precision recall f1-score support 0 0.99 0.99 0.99 1760 1 0.99 0.99 0.99 1476 2 0.97 0.96 0.97 503 accuracy 0.99 3739 macro avg 0.98 0.98 0.98 3739 weighted avg 0.99 0.99 0.99 3739 0.9863600134849548 Accuracy is: 0.9863599893019523</pre>	<pre>[[1744 8 8] [8 1458 10] [11 8 484]] precision recall f1-score support 0 0.99 0.99 0.99 1760 1 0.99 0.99 0.99 1476 2 0.96 0.96 0.96 503 accuracy 0.99 3739 macro avg 0.98 0.98 0.98 3739 weighted avg 0.99 0.99 0.99 3739 0.9858250617980957 Accuracy is: 0.9858250869216368</pre>