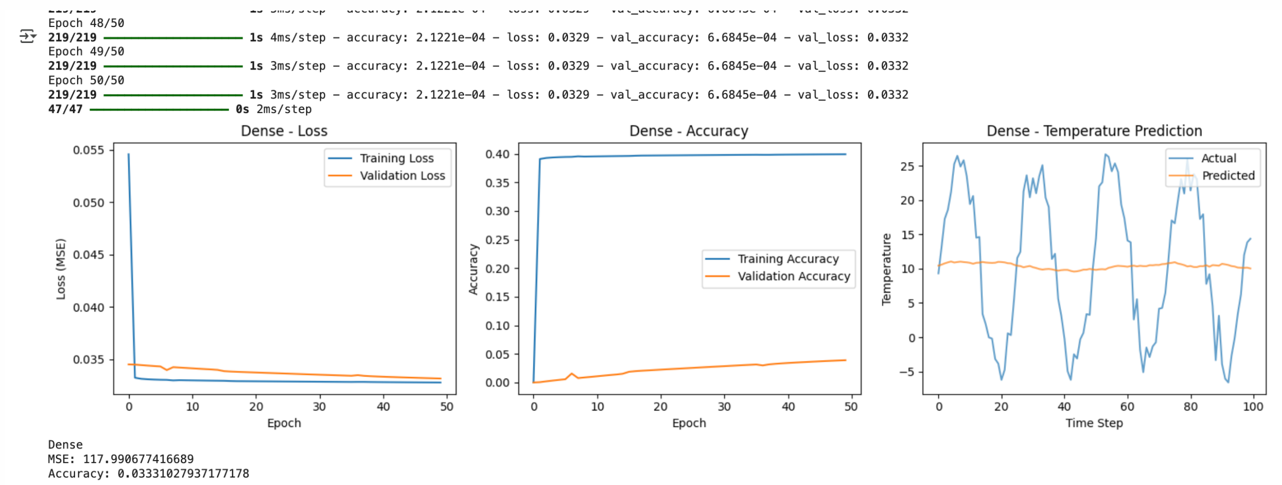
Recurrent Neural Network Report

Yuhuan Huang, 2025

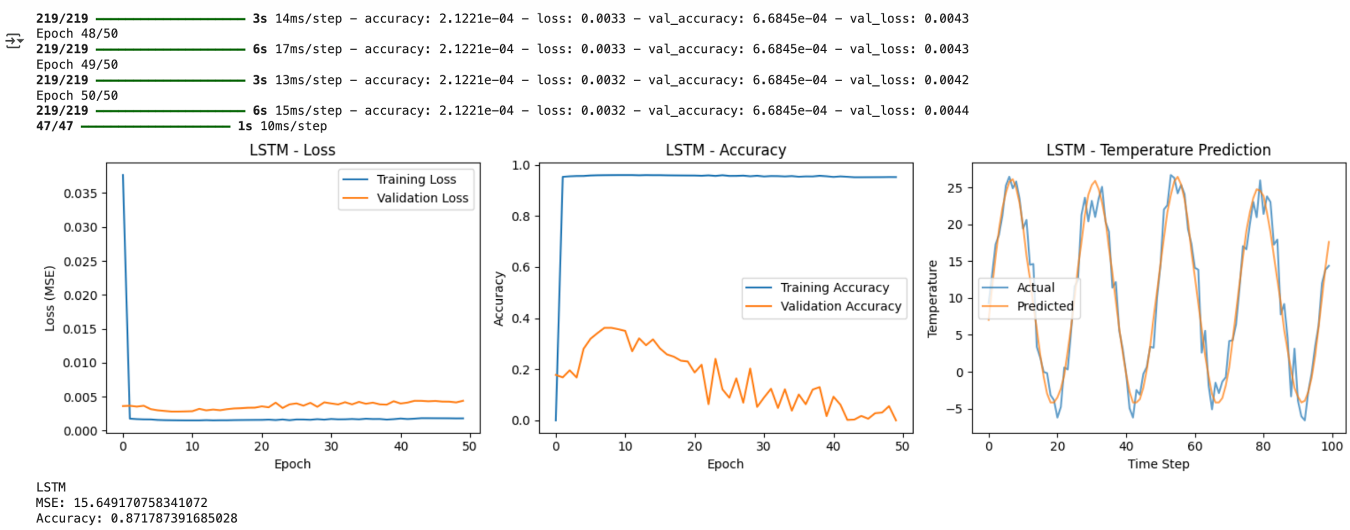
Part1: LSTM on synthetic datasets

Console output:

1. Dense:

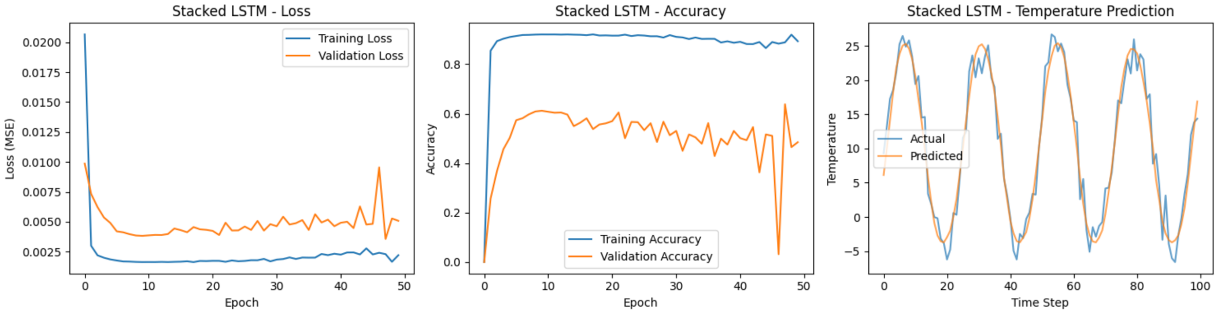


2. LSTM:



3. Stacked-LSTM (two-layer LSTM):

```
Epoch 47/50
219/219 ━━━━━━━━━━━━━━━━━━━━ 7s 30ms/step - accuracy: 2.1221e-04 - loss: 0.0046 - val_accuracy: 6.6845e-04 - val_loss: 0.0096
Epoch 48/50
219/219 ━━━━━━━━━━━━━━━━━━━━ 7s 26ms/step - accuracy: 2.1221e-04 - loss: 0.0055 - val_accuracy: 6.6845e-04 - val_loss: 0.0036
Epoch 49/50
219/219 ━━━━━━━━━━━━━━━━━━━━ 8s 32ms/step - accuracy: 2.1221e-04 - loss: 0.0026 - val_accuracy: 6.6845e-04 - val_loss: 0.0053
Epoch 50/50
219/219 ━━━━━━━━━━━━━━━━━━━━ 7s 27ms/step - accuracy: 2.1221e-04 - loss: 0.0044 - val_accuracy: 6.6845e-04 - val_loss: 0.0051
47/47 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step
```




```
Stacked LSTM
MSE: 18.079572991579212
Accuracy: 0.851875268902937
```

```
Stacked LSTM
MSE: 18.079572991579212
Accuracy: 0.851875268902937
Dense MSE: 117.990677416689
Dense Accuracy: 0.03331027937177178
LSTM MSE: 15.649170758341072
LSTM Accuracy: 0.871787391685028
Stacked LSTM MSE: 18.079572991579212
Stacked LSTM Accuracy: 0.851875268902937
**********************************************************************
File "__main__", line 3, in __main__
Failed example:
    np.isclose(results['Dense']['mse'], 117.33981323767615, atol=1e-02)
Expected:
    np.True_
Got:
    np.False_
**********************************************************************
1 items had failures:
   1 of   6 in __main__
***Test Failed*** 1 failures.
TestResults(failed=1, attempted=6)
```

I am sorry that the MSE test failed, and I don't know how to fix it. From all the above models, I can see the performance: LSTM > Stacked LSTM > Dense

Reflection:

1. The Dense neural network performed poorly compared to both LSTM models. Why might a Dense network struggle with time series prediction, even though it has access to the same 24-hour window of temperature data?

   Ans: Dense is not a time-series model, which means that it would treat the 24-hour data as each independent points, but not time series, so it would lose information given by the periodicity of a time series.

2. The Stacked LSTM (two LSTM layers) performed slightly worse than the single LSTM, despite being a more complex model. What are some potential reasons for this, and what does it tell us about choosing model architectures?

   Ans: Two-layer LSTM may make the model too complex, less efficient, or cause overfitting problem. So when we are choosing models, it is not always better to choose the more complex model.

3. Looking at the temperature predictions plot, you'll notice that all models (even the poor-performing Dense network) capture some basic pattern in the data. What might each model be learning about temperature patterns, and how does this relate to their architectures?

Ans: The LSTM and Stacked LSTM can capture the periodicity of the temperature data, while the Dense model can only capture the general trend (look like "average") of the temperature data. This is because Dense data is not a time-series model, but LSTM models are time-series models.

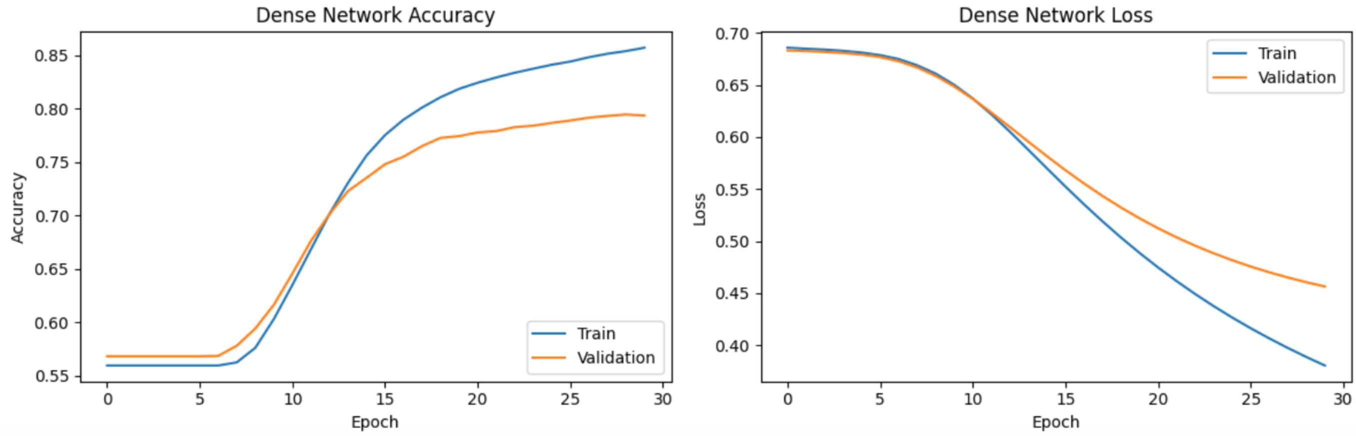
Part2: NLP Architecture Comparisons

1. **The Sarcasm dataset:**

```
Sarcasm dataset configurations:
Learning rates: {'Dense Network': 0.0001, 'LSTM Network': 1e-05, 'Bidirectional LSTM Network': 1e-05, 'Double Bidirectional
Embedding dimensions: {'Dense Network': 16, 'LSTM Network': 64, 'Bidirectional LSTM Network': 64, 'Double Bidirectional LST
Epochs: {'Dense Network': 30, 'LSTM Network': 20, 'Bidirectional LSTM Network': 15, 'Double Bidirectional LSTM Network': 15

Running comparison on Sarcasm dataset with optimized hyperparameters:
  Dense Network: embedding_dim=16, epochs=30, lr=0.0001
  LSTM Network: embedding_dim=64, epochs=20, lr=1e-05
  Bidirectional LSTM Network: embedding_dim=64, epochs=15, lr=1e-05
  Double Bidirectional LSTM Network: embedding_dim=64, epochs=15, lr=1e-05

Initial configuration:
Learning rates: {'Dense Network': 0.0001, 'LSTM Network': 1e-05, 'Bidirectional LSTM Network': 1e-05, 'Double Bidirectional
Embedding dimensions: {'Dense Network': 16, 'LSTM Network': 64, 'Bidirectional LSTM Network': 64, 'Double Bidirectional LST
Epochs config: {'Dense Network': 30, 'LSTM Network': 20, 'Bidirectional LSTM Network': 15, 'Double Bidirectional LSTM Netwo

Final configuration after defaults:
Learning rates: {'Dense Network': 0.0001, 'LSTM Network': 1e-05, 'Bidirectional LSTM Network': 1e-05, 'Double Bidirectional
Embedding dimensions: {'Dense Network': 16, 'LSTM Network': 64, 'Bidirectional LSTM Network': 64, 'Double Bidirectional LST
Epochs config: {'Dense Network': 30, 'LSTM Network': 20, 'Bidirectional LSTM Network': 15, 'Double Bidirectional LSTM Netwo

==================================================
Running Experiment on sarcasm Dataset
==================================================
Loading Sarcasm dataset...
Preprocessing data...
Training set size: 21367
Validation set size: 5342
```

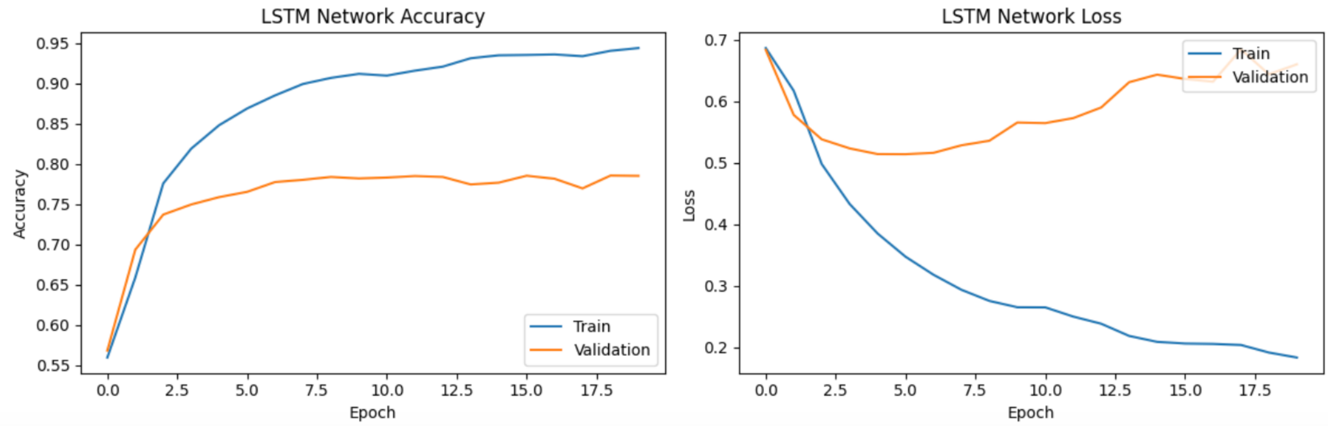
Dense:

```
Epoch 28/30
668/668 ━━━━━━━━━━━━━━━━━━━━ 4s 4ms/step - accuracy: 0.8537 - loss: 0.3998 - val_accuracy: 0.7931 - val_loss: 0.4650
Epoch 29/30
668/668 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.8566 - loss: 0.3909 - val_accuracy: 0.7945 - val_loss: 0.4605
Epoch 30/30
668/668 ━━━━━━━━━━━━━━━━━━━━ 5s 5ms/step - accuracy: 0.8600 - loss: 0.3824 - val_accuracy: 0.7935 - val_loss: 0.4564

Dense Network Validation Accuracy: 0.7935
Training time: 122.17 seconds
```



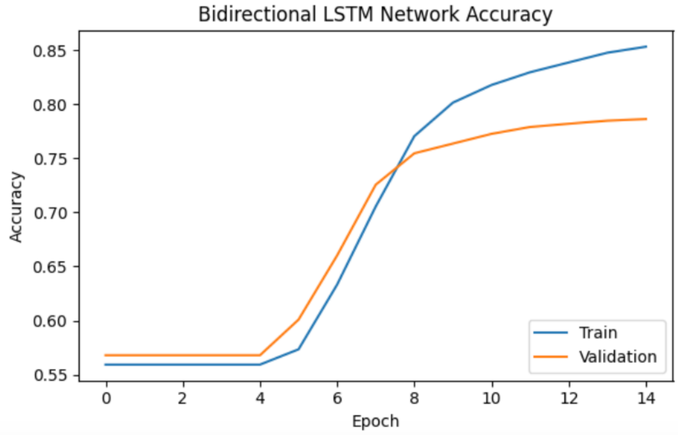
LSTM:

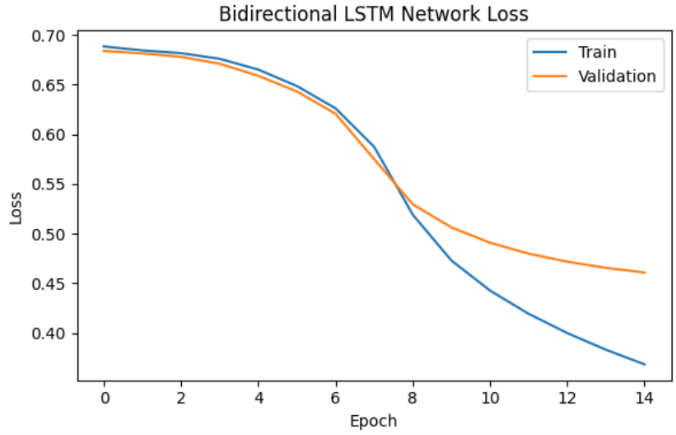
```
Epoch 18/20
668/668 ━━━━━━━━━━━━━━━━━━━━ 138s 163ms/step - accuracy: 0.9287 - loss: 0.2124 - val_accuracy: 0.7696 - val_loss: 0.6844
Epoch 19/20
668/668 ━━━━━━━━━━━━━━━━━━━━ 106s 158ms/step - accuracy: 0.9341 - loss: 0.2013 - val_accuracy: 0.7855 - val_loss: 0.6439
Epoch 20/20
668/668 ━━━━━━━━━━━━━━━━━━━━ 145s 162ms/step - accuracy: 0.9402 - loss: 0.1884 - val_accuracy: 0.7851 - val_loss: 0.6601

LSTM Network Validation Accuracy: 0.7851
Training time: 2420.86 seconds
```



Bidirectional LSTM

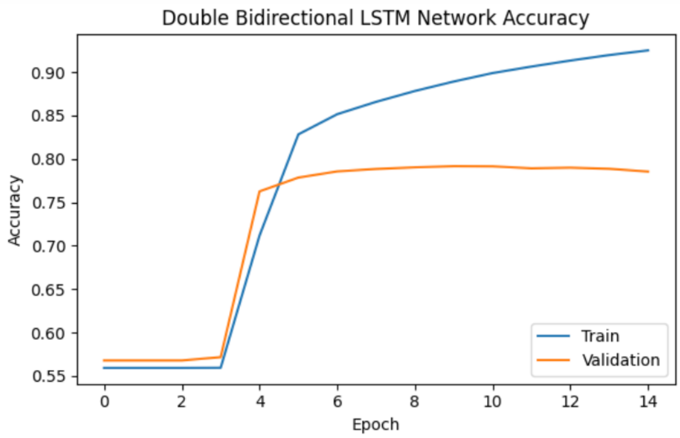
```
Epoch 13/15
668/668 ━━━━━━━━━━━━━━━━━━━━ 83s 114ms/step - accuracy: 0.8380 - loss: 0.4059 - val_accuracy: 0.7819 - val_loss: 0.4717
Epoch 14/15
668/668 ━━━━━━━━━━━━━━━━━━━━ 76s 114ms/step - accuracy: 0.8471 - loss: 0.3885 - val_accuracy: 0.7847 - val_loss: 0.4655
Epoch 15/15
668/668 ━━━━━━━━━━━━━━━━━━━━ 76s 114ms/step - accuracy: 0.8518 - loss: 0.3732 - val_accuracy: 0.7862 - val_loss: 0.4608

Bidirectional LSTM Network Validation Accuracy: 0.7862
Training time: 1172.97 seconds
```

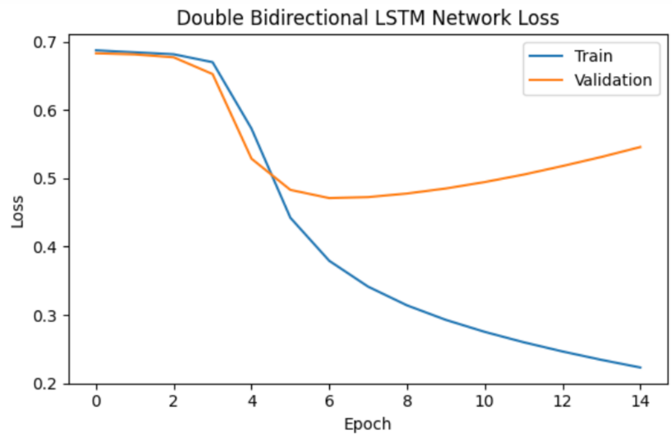


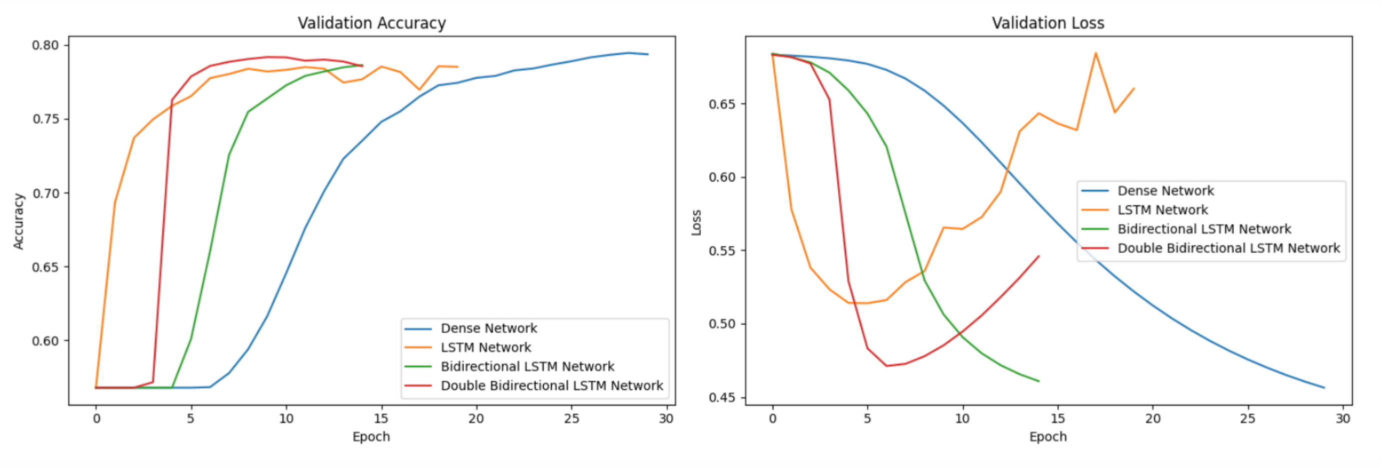


Double Bidirectional LSTM

```
668/668 ━━━━━━━━━━━━━━━━━━━━ 160s 240ms/step - accuracy: 0.8757 - loss: 0.3205 - val_accuracy: 0.7903 - val_loss: 0.4778
Epoch 10/15
668/668 ━━━━━━━━━━━━━━━━━━━━ 161s 241ms/step - accuracy: 0.8863 - loss: 0.2984 - val_accuracy: 0.7917 - val_loss: 0.4853
Epoch 11/15
668/668 ━━━━━━━━━━━━━━━━━━━━ 160s 239ms/step - accuracy: 0.8957 - loss: 0.2802 - val_accuracy: 0.7915 - val_loss: 0.4946
Epoch 12/15
668/668 ━━━━━━━━━━━━━━━━━━━━ 159s 238ms/step - accuracy: 0.9046 - loss: 0.2644 - val_accuracy: 0.7892 - val_loss: 0.5056
Epoch 13/15
668/668 ━━━━━━━━━━━━━━━━━━━━ 160s 240ms/step - accuracy: 0.9112 - loss: 0.2502 - val_accuracy: 0.7900 - val_loss: 0.5181
Epoch 14/15
668/668 ━━━━━━━━━━━━━━━━━━━━ 161s 241ms/step - accuracy: 0.9177 - loss: 0.2372 - val_accuracy: 0.7887 - val_loss: 0.5314
Epoch 15/15
668/668 ━━━━━━━━━━━━━━━━━━━━ 160s 240ms/step - accuracy: 0.9236 - loss: 0.2255 - val_accuracy: 0.7855 - val_loss: 0.5459

Double Bidirectional LSTM Network Validation Accuracy: 0.7855
Training time: 2489.17 seconds
```







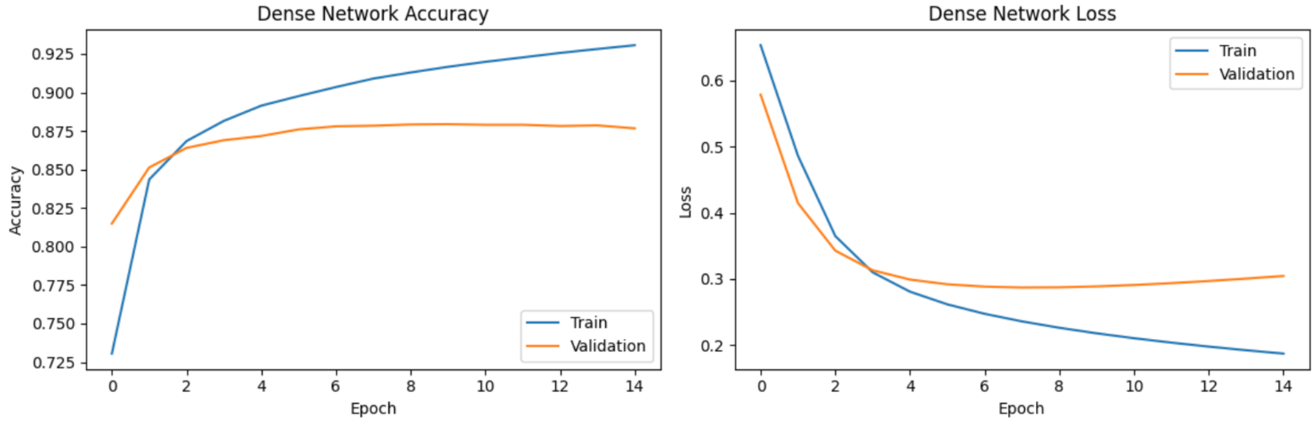

```
Test predictions:

Text: Scientists cure cancer, world peace achieved
  Dense Network: 0.3318 (Negative)
  LSTM Network: 0.0367 (Negative)
WARNING:tensorflow:5 out of the last 50 calls to <function TensorFlowTrainer.make_predict_function.<locals>.one_step_on_dat
  Bidirectional LSTM Network: 0.3236 (Negative)
WARNING:tensorflow:6 out of the last 51 calls to <function TensorFlowTrainer.make_predict_function.<locals>.one_step_on_dat
  Double Bidirectional LSTM Network: 0.3377 (Negative)

Text: Local man wins lottery, plans to spend it all on cat food
  Dense Network: 0.9577 (Positive)
  LSTM Network: 0.9713 (Positive)
  Bidirectional LSTM Network: 0.9267 (Positive)
  Double Bidirectional LSTM Network: 0.9526 (Positive)

Text: Breaking: Water is wet, scientists confirm
  Dense Network: 0.7731 (Positive)
  LSTM Network: 0.8613 (Positive)
  Bidirectional LSTM Network: 0.8451 (Positive)
  Double Bidirectional LSTM Network: 0.9532 (Positive)
```

```
Model Performance Summary:
                                   final_train_acc  final_val_acc  \
Dense Network                               0.8570         0.7935
LSTM Network                                0.9440         0.7851
Bidirectional LSTM Network                  0.8531         0.7862
Double Bidirectional LSTM Network           0.9250         0.7855

                                   best_val_acc  final_train_loss  \
Dense Network                            0.7945            0.3804
LSTM Network                             0.7855            0.1833
Bidirectional LSTM Network               0.7862            0.3683
Double Bidirectional LSTM Network        0.7917            0.2232

                                   final_val_loss  training_time
Dense Network                              0.4564       122.1702
LSTM Network                               0.6601      2420.8608
Bidirectional LSTM Network                 0.4608      1172.9664
Double Bidirectional LSTM Network          0.5459      2489.1669
```

2. The IMDB dataset

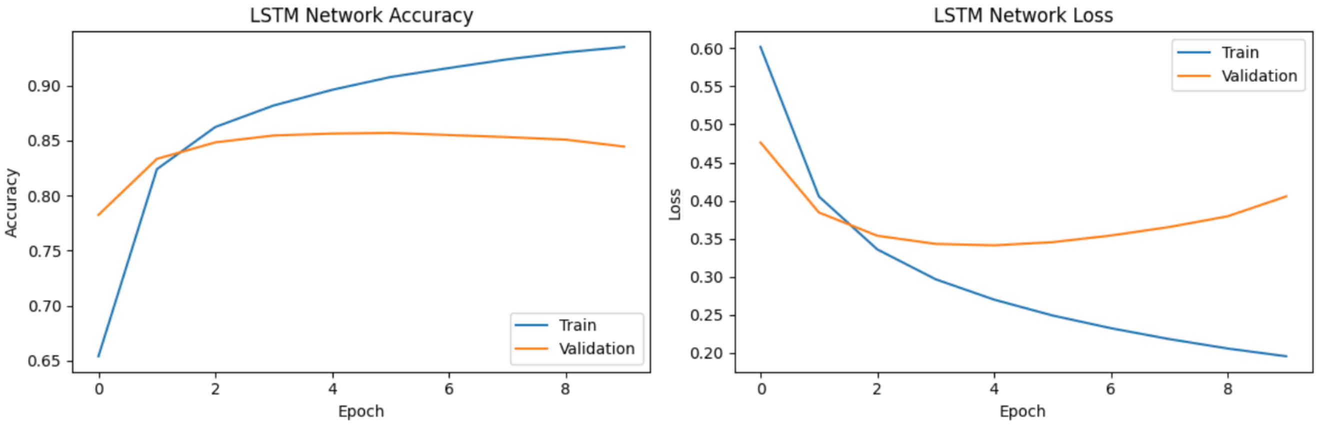
Dense:

```
Epoch 12/15
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 9s 5ms/step - accuracy: 0.9217 - loss: 0.2017 - val_accuracy: 0.8790 - val_loss: 0.2938
Epoch 13/15
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 8s 7ms/step - accuracy: 0.9244 - loss: 0.1956 - val_accuracy: 0.8782 - val_loss: 0.2970
Epoch 14/15
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 8s 7ms/step - accuracy: 0.9275 - loss: 0.1900 - val_accuracy: 0.8786 - val_loss: 0.3006
Epoch 15/15
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 7s 5ms/step - accuracy: 0.9297 - loss: 0.1849 - val_accuracy: 0.8767 - val_loss: 0.3046

Dense Network Validation Accuracy: 0.8767
Training time: 136.06 seconds
```

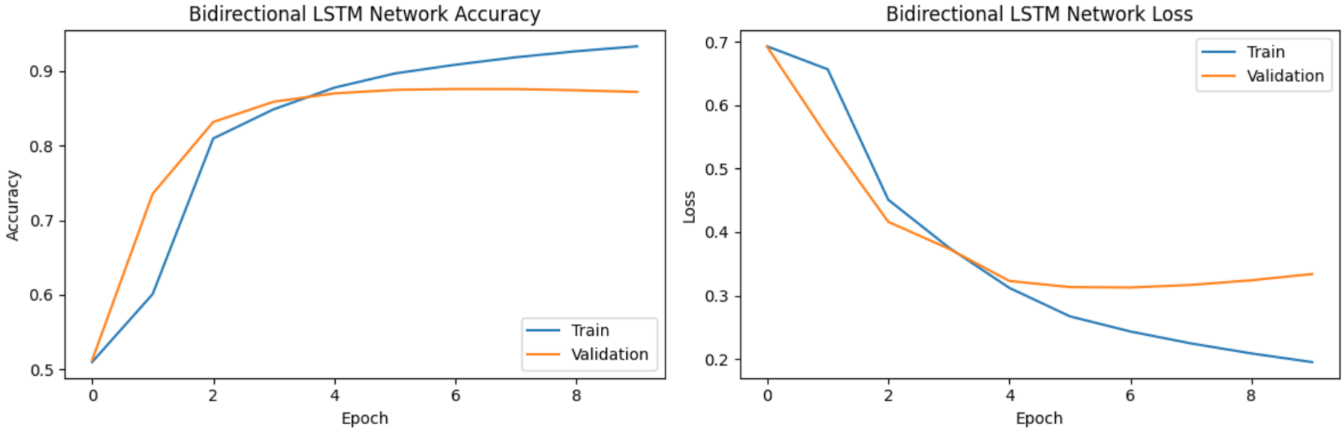


LSTM:

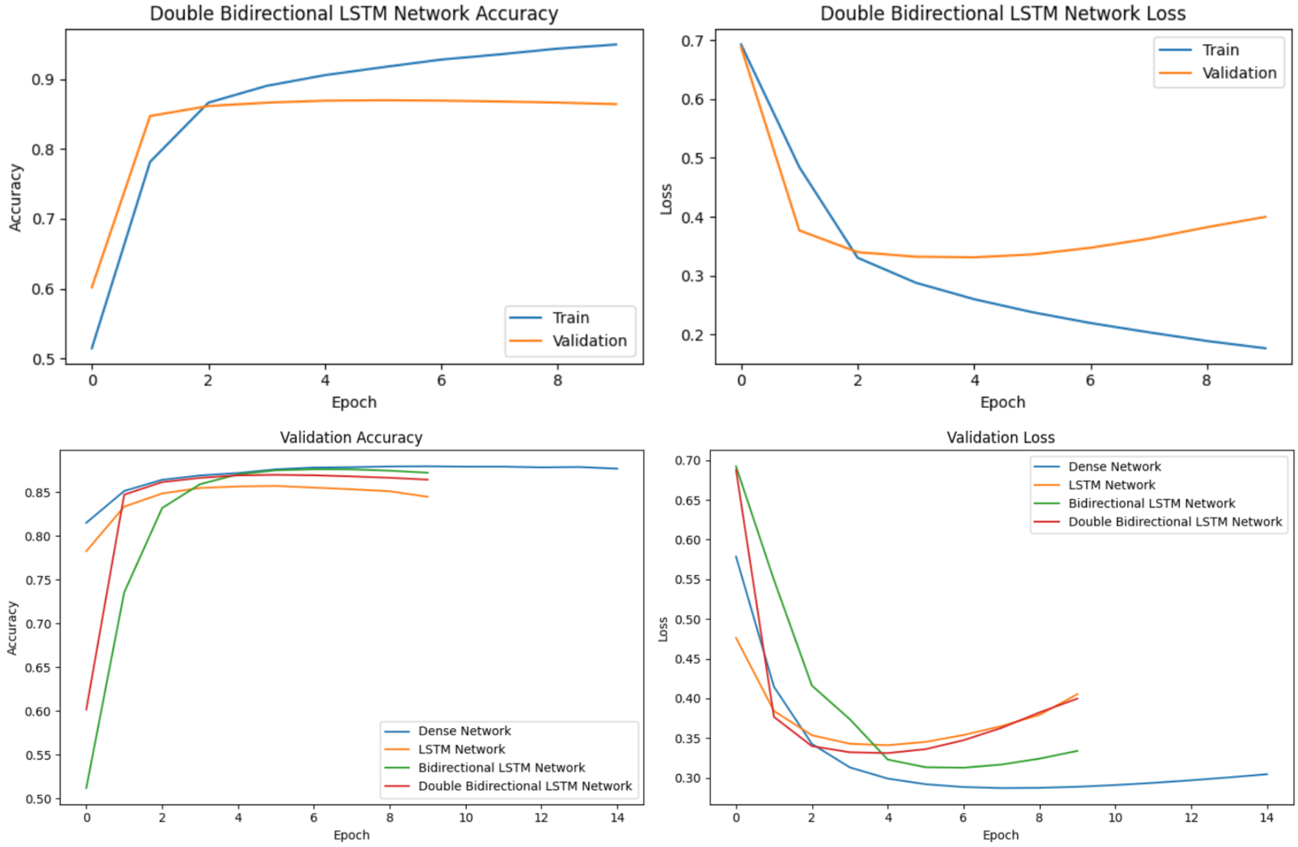
```
Epoch 8/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 250s 166ms/step - accuracy: 0.9242 - loss: 0.2186 - val_accuracy: 0.8532 - val_loss: 0.3652
Epoch 9/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 202s 161ms/step - accuracy: 0.9315 - loss: 0.2056 - val_accuracy: 0.8509 - val_loss: 0.3793
Epoch 10/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 201s 161ms/step - accuracy: 0.9366 - loss: 0.1943 - val_accuracy: 0.8446 - val_loss: 0.4054

LSTM Network Validation Accuracy: 0.8446
Training time: 2080.61 seconds
```



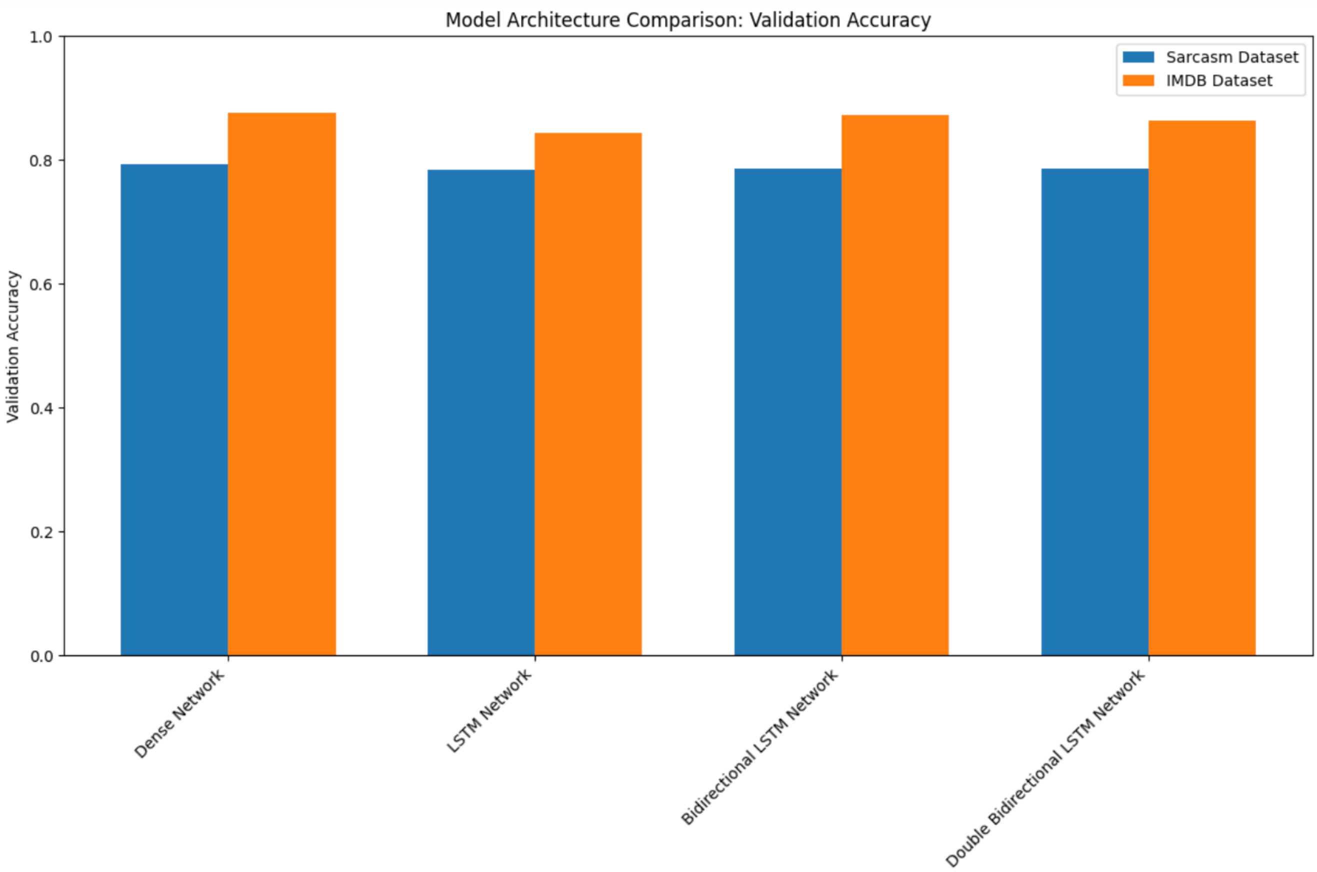
```
Training Bidirectional LSTM Network...
Embedding dimension: 64
Using learning rate: 1e-05 for Bidirectional LSTM Network
Training for 10 epochs
Epoch 1/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 141s 110ms/step - accuracy: 0.5048 - loss: 0.6930 - val_accuracy: 0.5121 - val_loss: 0.6922
Epoch 2/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 140s 108ms/step - accuracy: 0.5401 - loss: 0.6857 - val_accuracy: 0.7355 - val_loss: 0.5492
Epoch 3/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 141s 108ms/step - accuracy: 0.7921 - loss: 0.4917 - val_accuracy: 0.8318 - val_loss: 0.4162
Epoch 4/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 142s 108ms/step - accuracy: 0.8408 - loss: 0.3910 - val_accuracy: 0.8590 - val_loss: 0.3739
Epoch 5/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 140s 106ms/step - accuracy: 0.8686 - loss: 0.3314 - val_accuracy: 0.8702 - val_loss: 0.3231
Epoch 6/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 144s 108ms/step - accuracy: 0.8935 - loss: 0.2745 - val_accuracy: 0.8749 - val_loss: 0.3134
Epoch 7/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 136s 109ms/step - accuracy: 0.9066 - loss: 0.2479 - val_accuracy: 0.8760 - val_loss: 0.3128
Epoch 8/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 138s 106ms/step - accuracy: 0.9170 - loss: 0.2280 - val_accuracy: 0.8759 - val_loss: 0.3168
Epoch 9/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 144s 108ms/step - accuracy: 0.9259 - loss: 0.2116 - val_accuracy: 0.8744 - val_loss: 0.3242
Epoch 10/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 142s 108ms/step - accuracy: 0.9328 - loss: 0.1977 - val_accuracy: 0.8721 - val_loss: 0.3340

Bidirectional LSTM Network Validation Accuracy: 0.8721
Training time: 1416.51 seconds
```



```
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 323s 228ms/step - accuracy: 0.9353 - loss: 0.2017 - val_accuracy: 0.8679 - val_loss: 0.3630
Epoch 9/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 286s 229ms/step - accuracy: 0.9443 - loss: 0.1861 - val_accuracy: 0.8663 - val_loss: 0.3824
Epoch 10/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 285s 228ms/step - accuracy: 0.9501 - loss: 0.1723 - val_accuracy: 0.8642 - val_loss: 0.3998

Double Bidirectional LSTM Network Validation Accuracy: 0.8642
Training time: 3070.10 seconds
```



```
Test predictions:

Text: this film was absolutely terrible the worst acting i have seen
  Dense Network: 0.0741 (Negative)
  LSTM Network: 0.0179 (Negative)
  Bidirectional LSTM Network: 0.0800 (Negative)
  Double Bidirectional LSTM Network: 0.0227 (Negative)

Text: a masterpiece of cinema with outstanding performances and direction
  Dense Network: 0.9119 (Positive)
  LSTM Network: 0.9879 (Positive)
  Bidirectional LSTM Network: 0.9244 (Positive)
  Double Bidirectional LSTM Network: 0.9609 (Positive)

Text: neither good nor bad just a mediocre film that passes the time
  Dense Network: 0.2011 (Negative)
  LSTM Network: 0.0475 (Negative)
  Bidirectional LSTM Network: 0.0930 (Negative)
  Double Bidirectional LSTM Network: 0.0606 (Negative)

 Model Performance Summary:
                                   final_train_acc  final_val_acc  \
 Dense Network                              0.9306         0.8767
 LSTM Network                               0.9352         0.8446
 Bidirectional LSTM Network                 0.9333         0.8721
 Double Bidirectional LSTM Network          0.9495         0.8642

                                   best_val_acc  final_train_loss  \
 Dense Network                           0.8794            0.1872
 LSTM Network                            0.8570            0.1953
 Bidirectional LSTM Network              0.8760            0.1952
 Double Bidirectional LSTM Network       0.8697            0.1769

                                   final_val_loss  training_time
 Dense Network                             0.3046       136.0565
 LSTM Network                              0.4054      2080.6131
 Bidirectional LSTM Network                0.3340      1416.5087
 Double Bidirectional LSTM Network         0.3998      3070.1048
```




```
Comparison complete! Review the results and charts above.
```

Reflections:

1. Model Complexity vs. Performance
   The Dense Network outperformed more complex architectures on both datasets. What might explain this counterintuitive result, and what implications does this have for the principle that "more complex models yield better results"?

   Ans: Although Dense is "simplier" than the more complexed LSTM model, possibly the dataset of the IMDB doesn't have time-series characteristics, and using Dense would be enough and more efficient. So it is not true that complex models always yield better result. There's a trade-off between complexity and efficiency.

2. Training-Validation Gap Analysis
   The LSTM Network on the sarcasm dataset showed a large gap between training accuracy and validation accuracy. What does this pattern suggest, and what techniques could you implement to address this issue?

   Ans: It shows the over-fitting problem. To mitigate the over-fitting problem, we can add the Dropout layer, or we can set the early stopping.

3. Task Difficulty Comparison
   All models performed better on the IMDB dataset than on the sarcasm dataset. What characteristics of sarcasm might make it inherently more difficult to detect compared to sentiment analysis?

   Ans: I think a possible reason is that the reviews of IMDB have stronger or more intense sentiments, while the sentences on the sarcasm dataset have more ambigue emotions. So it makes the sarcasm dataset harder to detect.

4. Computational Efficiency Analysis
   The training times varied dramatically between architectures. Considering the performance results, analyze the trade-offs between model complexity, computational resources, and accuracy. When might you choose a simpler model despite having resources for more complex ones?

   Ans: A more complex model may requires larger computational resources (and a lot of time to train the models), like the LSTMs in this example, and when the dataset itself is large and complex, a more sophiscated model may bring higher accuracy. However, sometimes when the data itself is simple, and a simple model can provides good accuracy. In this case, we don't need to use a very complex model, even if we have enough resources.

5. Architecture Enhancement Proposal
   Based on these results, propose a modified architecture or approach that might improve performance on the sarcasm detection task while maintaining reasonable com-

putational efficiency. Justify your design choices.

Ans: Perhaps I would include the Attention model, which is introduced in the following lectures. This design may be better than having multiple layers of LSTM.

Part3: Dense vs LSTM on longer docuements

Sorry that I haven't done this part yet!