COVID-19 Case Prediction CNN and LSTM Neural Network Case Study Comparison

Robert Martinez and Aarush Mathur

ECE884– Neural Networks and Deep Learning

Outlines

- Purpose
- Experimentation
    - Data Preprocessing
    - Benchmark
    - Modified Network 1
    - Modified Network 2
- Results
- Conclusion

Purpose

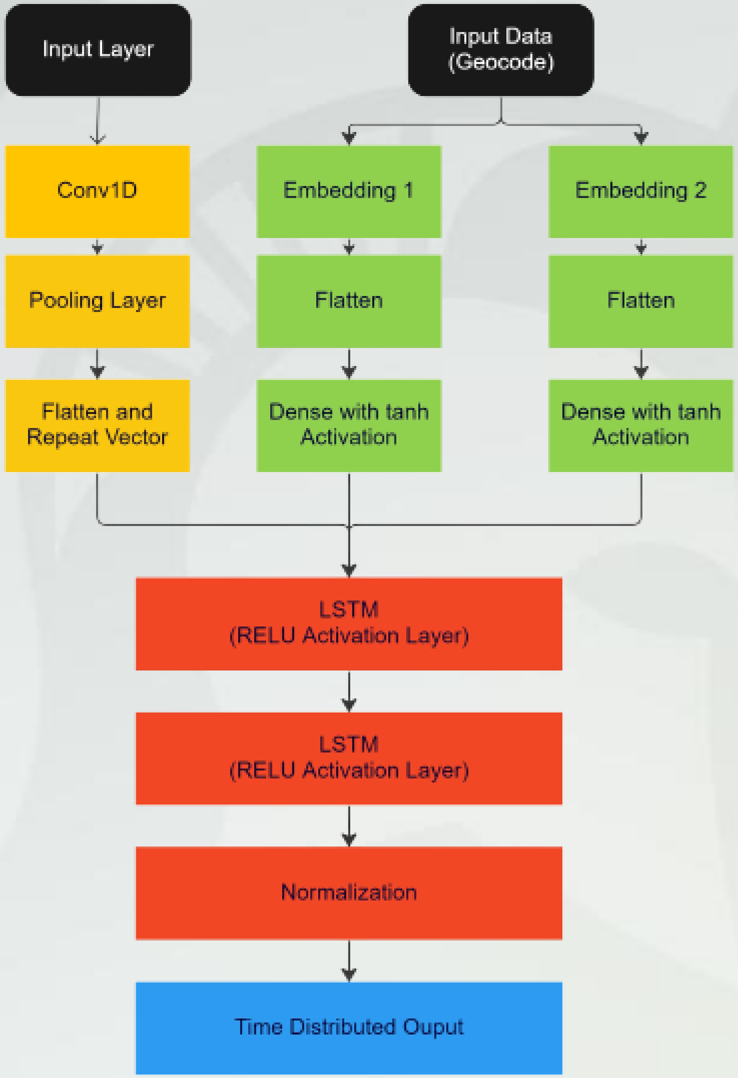
- Create a model in response to the White House’s Office of Science and Technology Policy’s challenge to predict cumulative confirmed COVID 19 Cases in various locations throughout the world [1]

- Data provided from John Hopkins University Center for Systems and Science Engineering on the number of cases and fatalities for a given location [2]

Experiment: Data Preprocessing

- Combining Province/State and Country/Region into single Geocode
- One hot encoding the Geocode
- Normalization of number of cases and fatalities on scale [0,1]
- Log of cases and fatalities

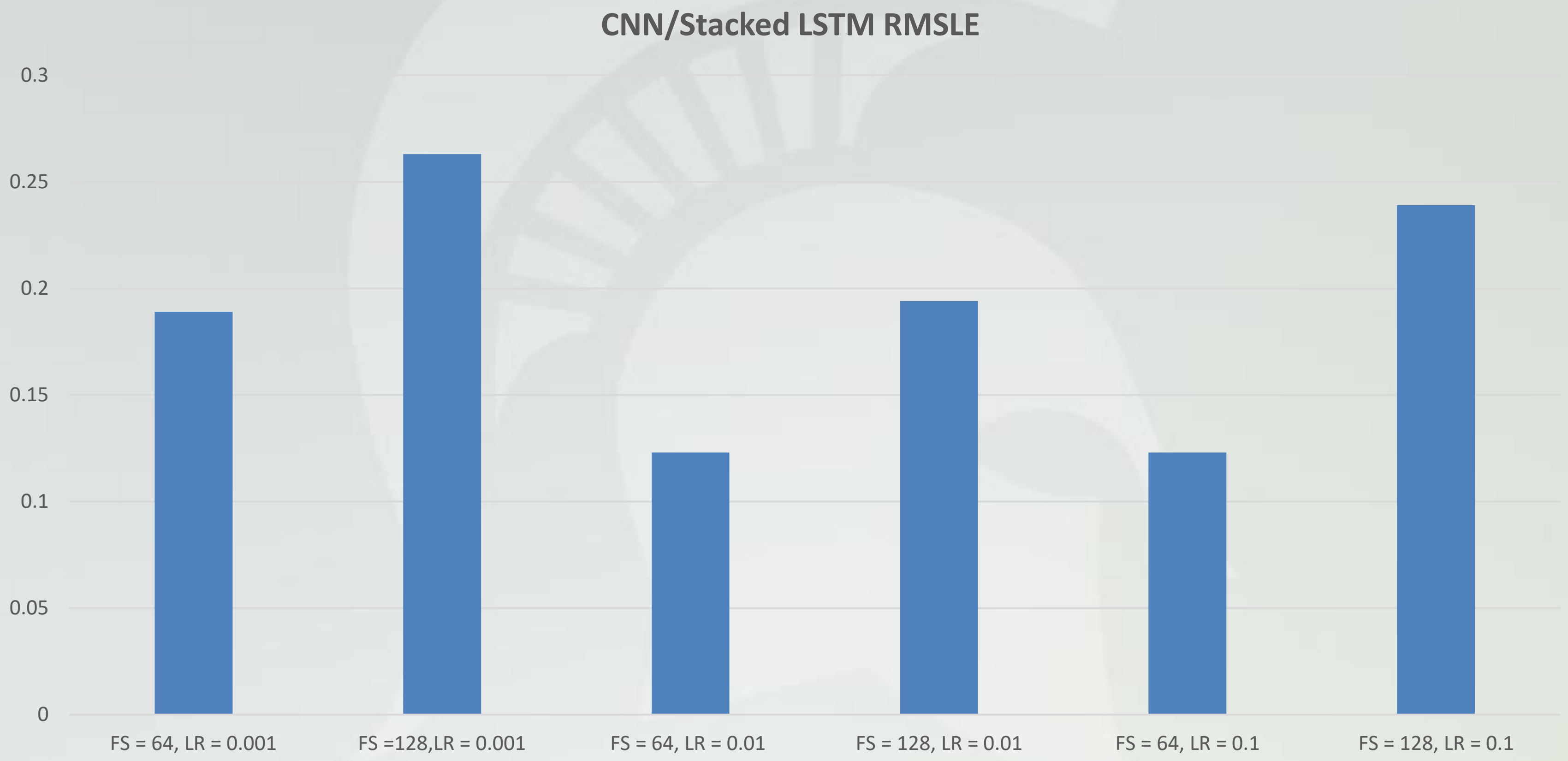
CNN/Stacked LSTM Model Architecture



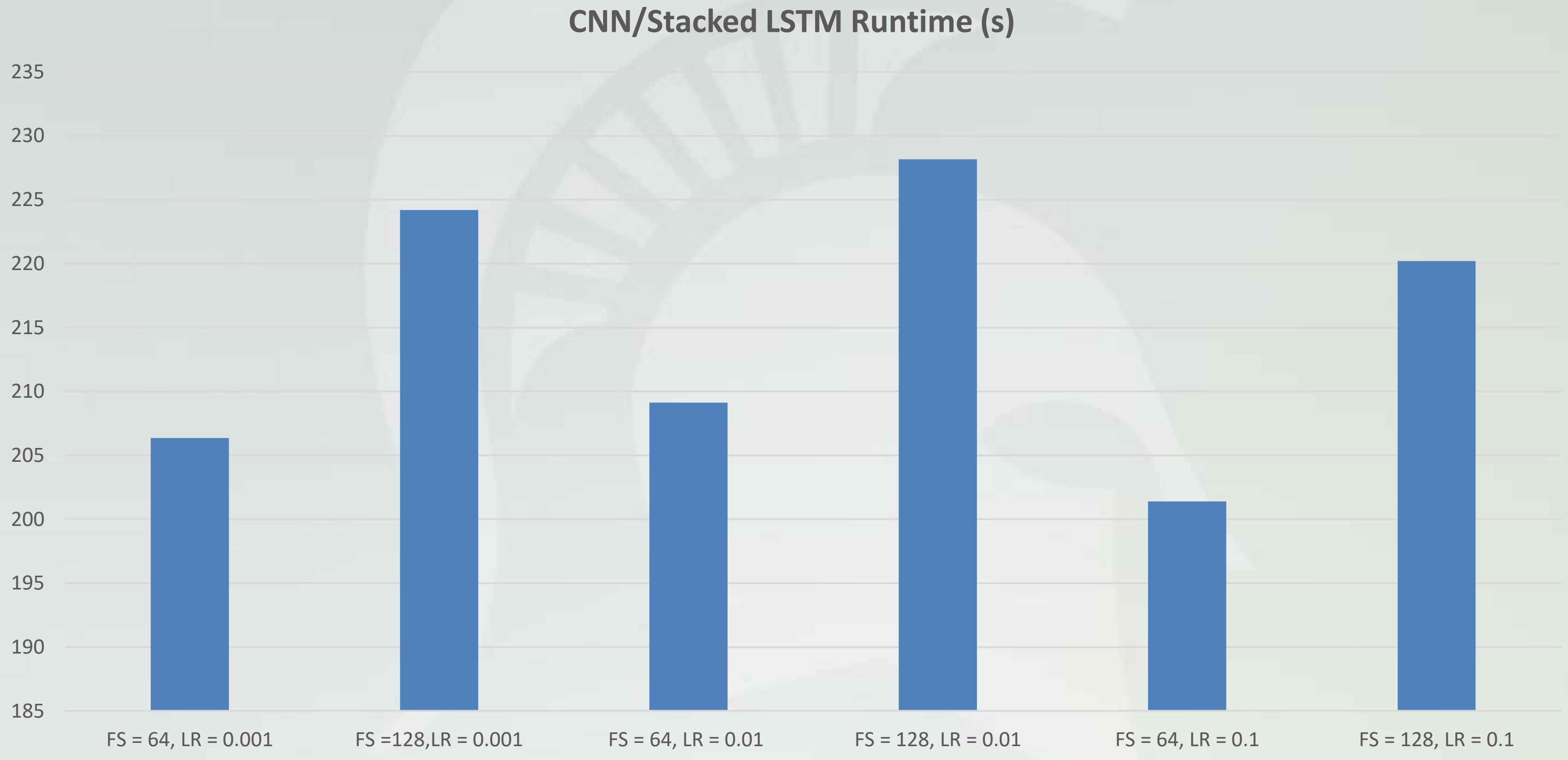
Model Case Study Comparison: Benchmark

- General Case Architecture: CNN/Stacked LSTM
- Optimizer: Adams Optimizer proved best for time series dataset, kept constant
- CNN Filter Size: To see results of smaller convolutional filters vs larger [2]
- Learning Rate: Varied rate to see performance

Model Performance - Benchmarks

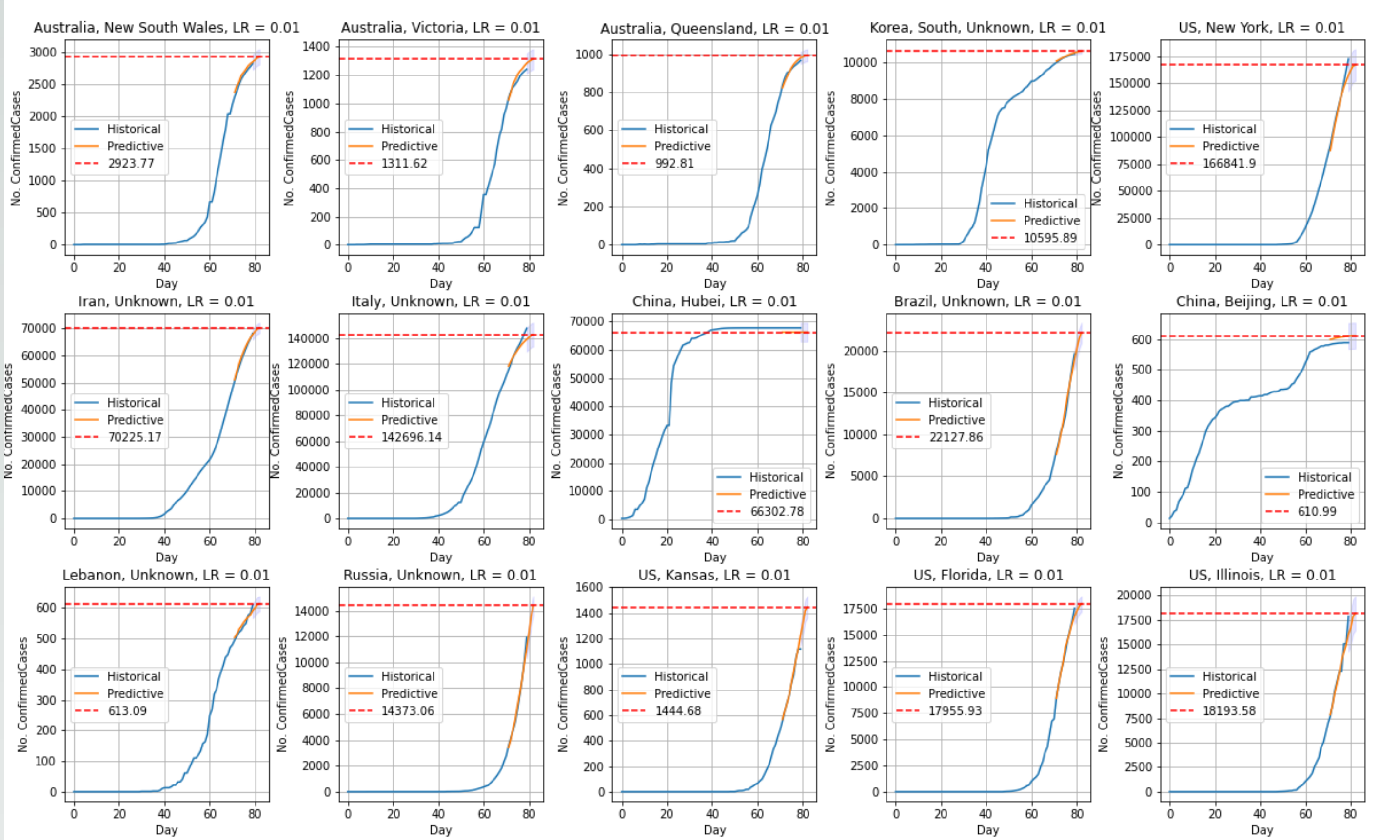


Model Performance - Benchmarks

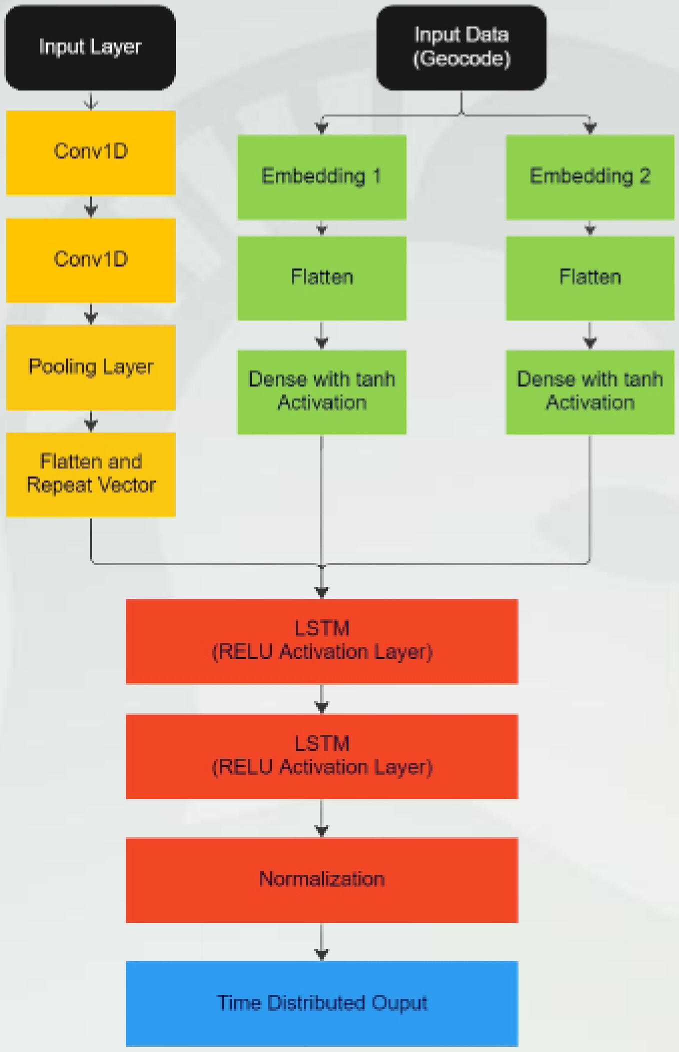


Best Confirmed Case Predictor for Benchmark

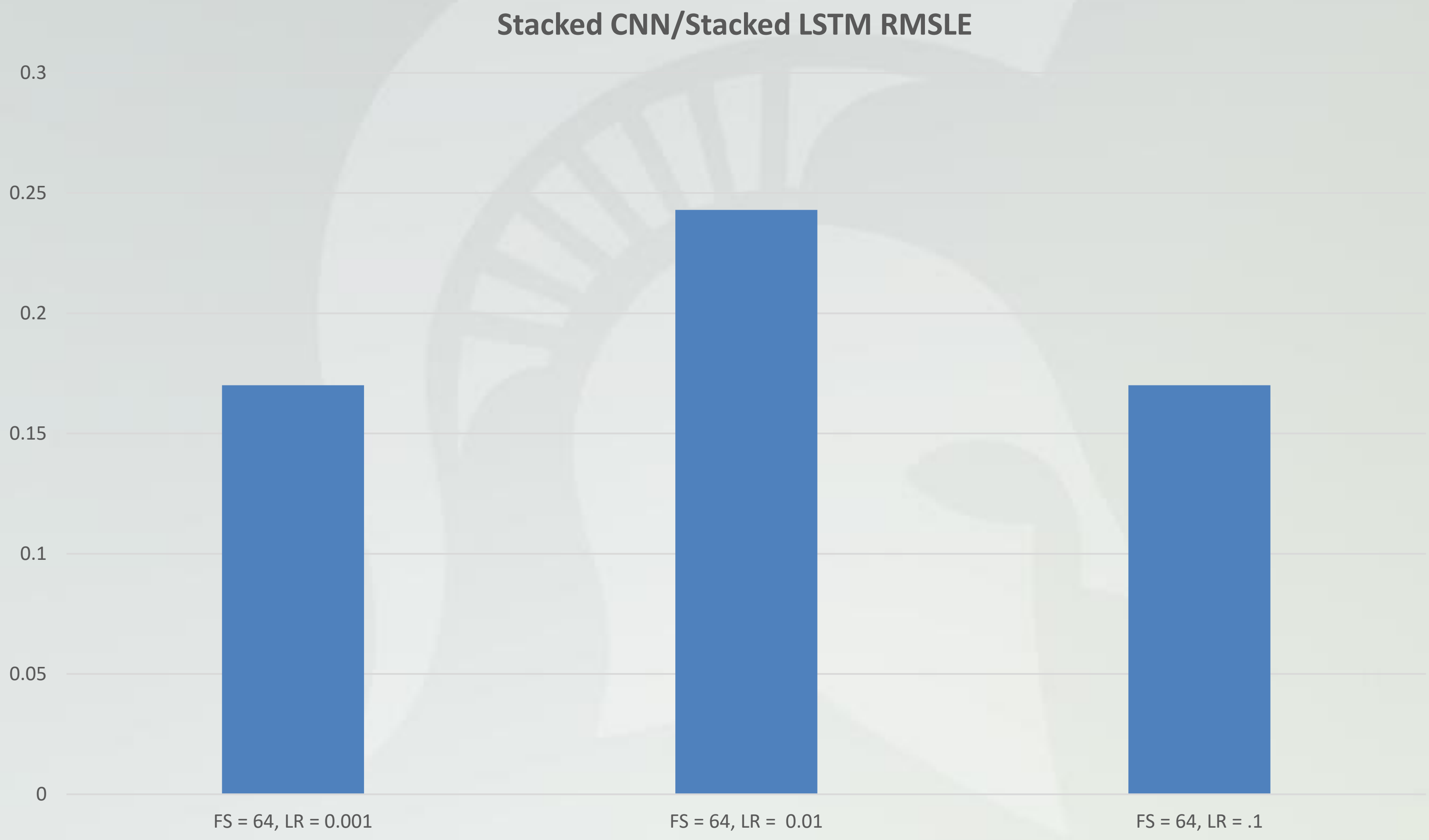
CNN/Stacked LSTM - Filter Size 64, Learning Rate = .01



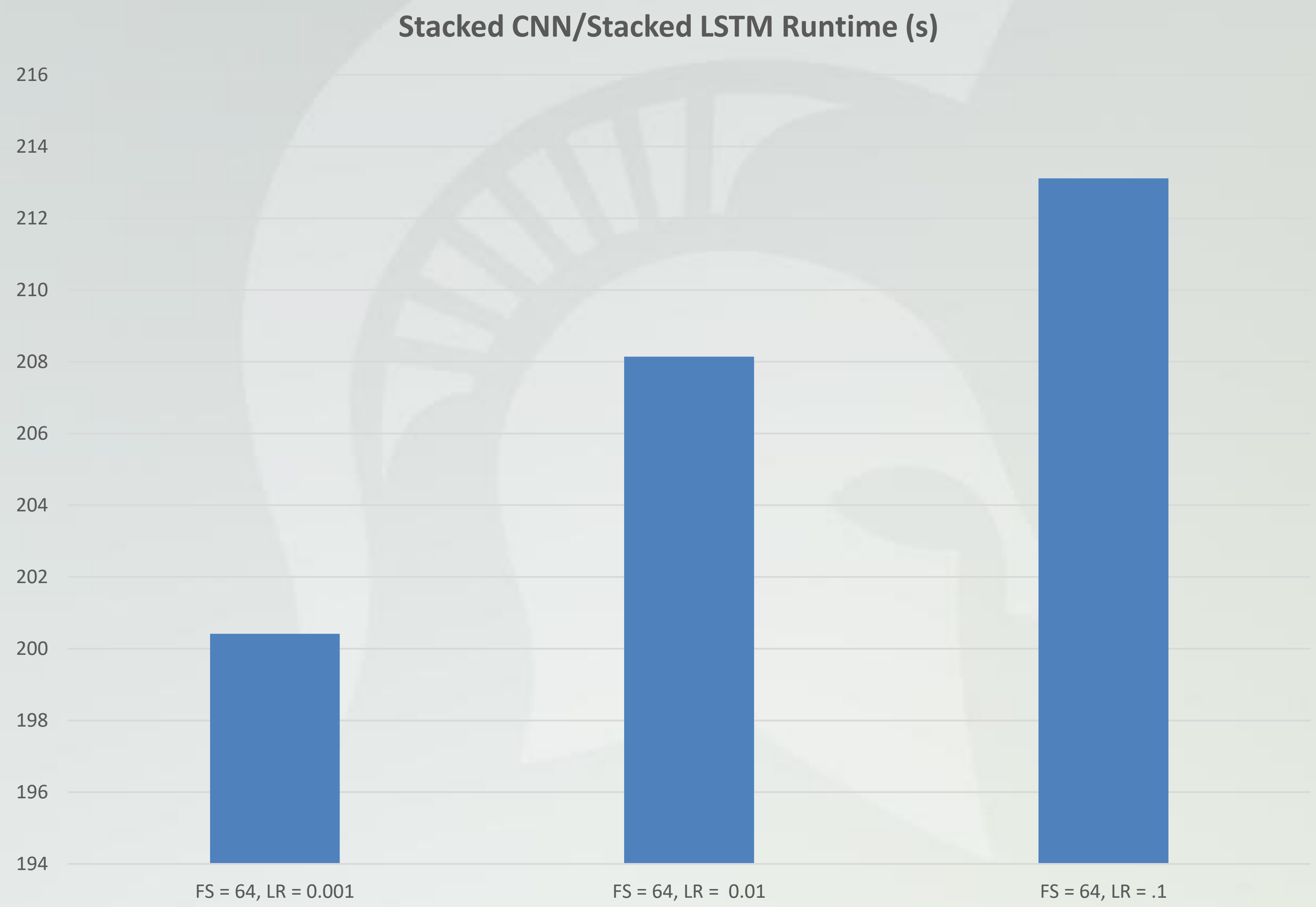
Stacked CNN/Stacked LSTM Model Architecture



Model Performance – Modified Network 1

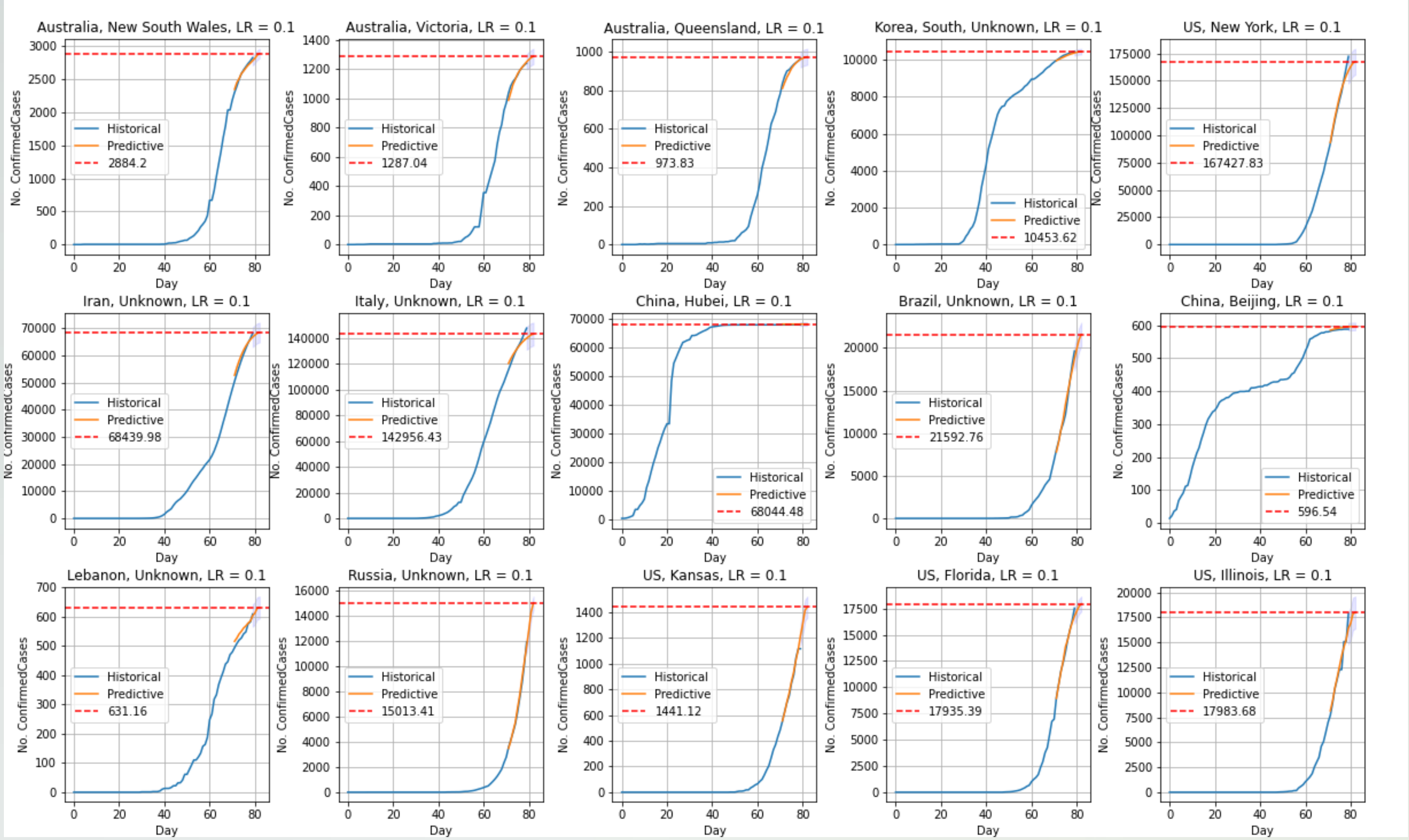


Model Performance – Modified Network 1



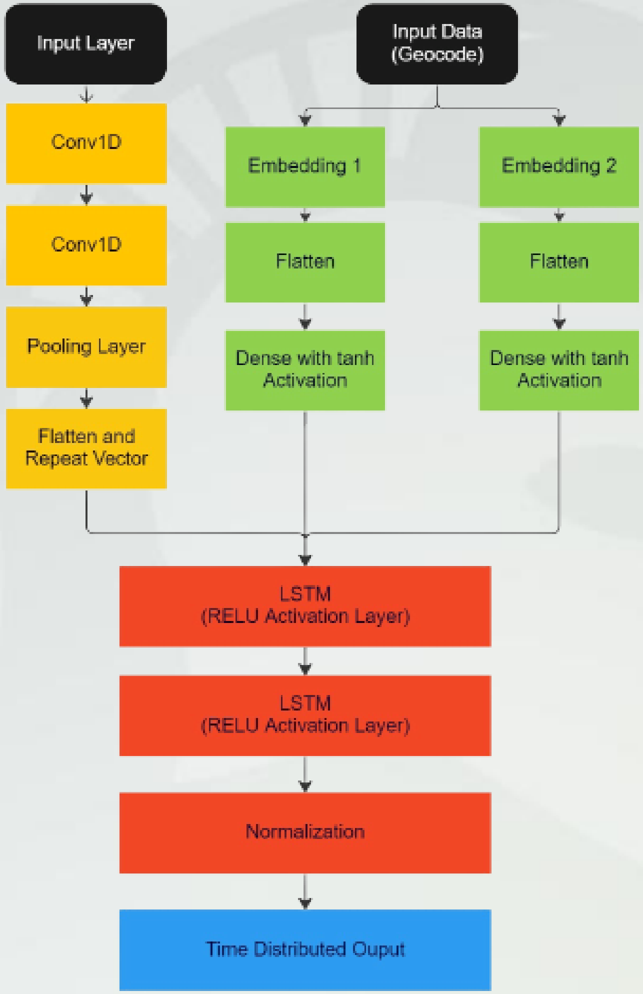
Best Confirmed Case Predictor for Modified Network #1

Stacked CNN/Stacked LSTM - Filter Size 64, Learning Rate = .1

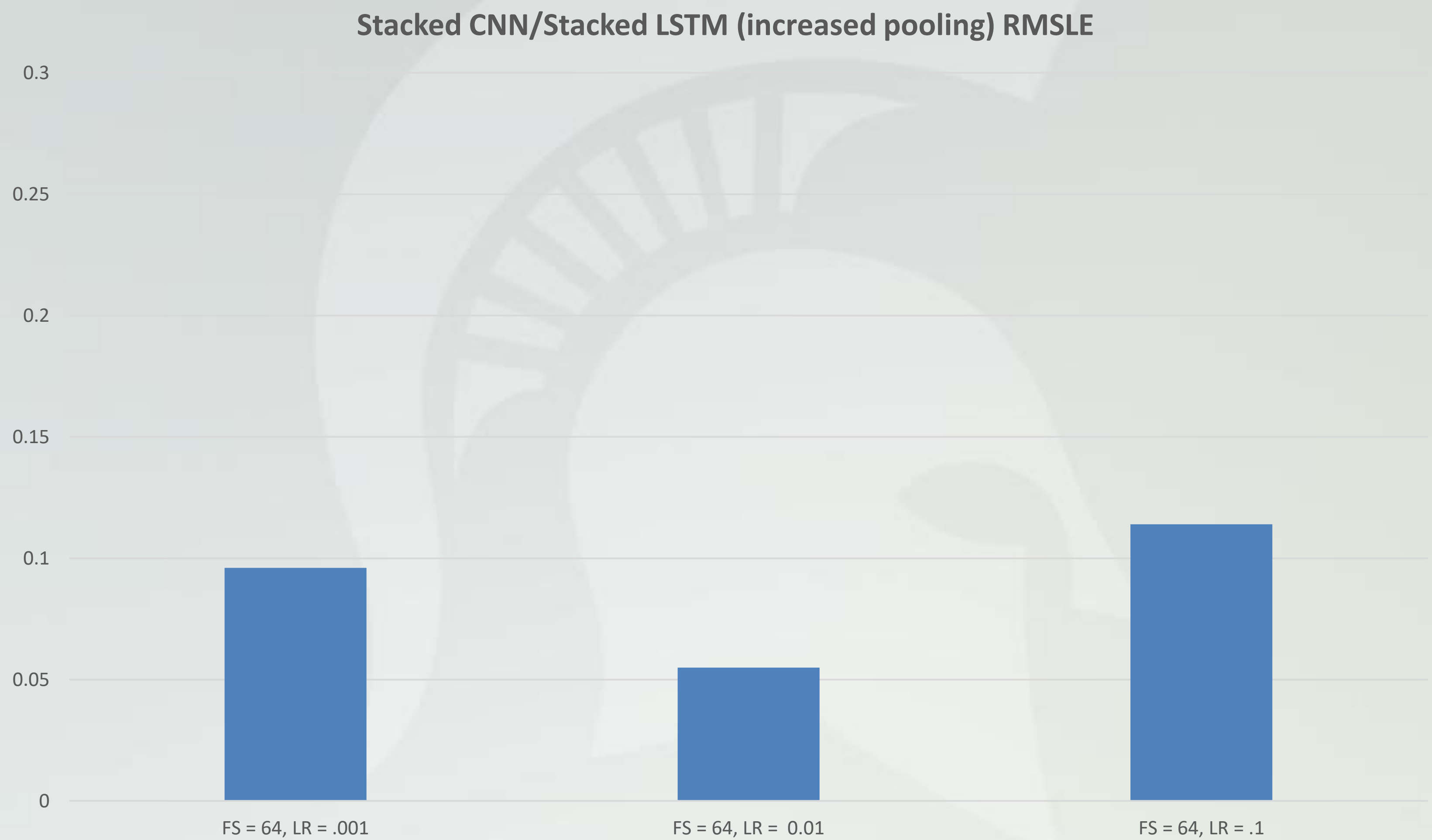


Stacked CNN/Stacked LSTM Model Architecture

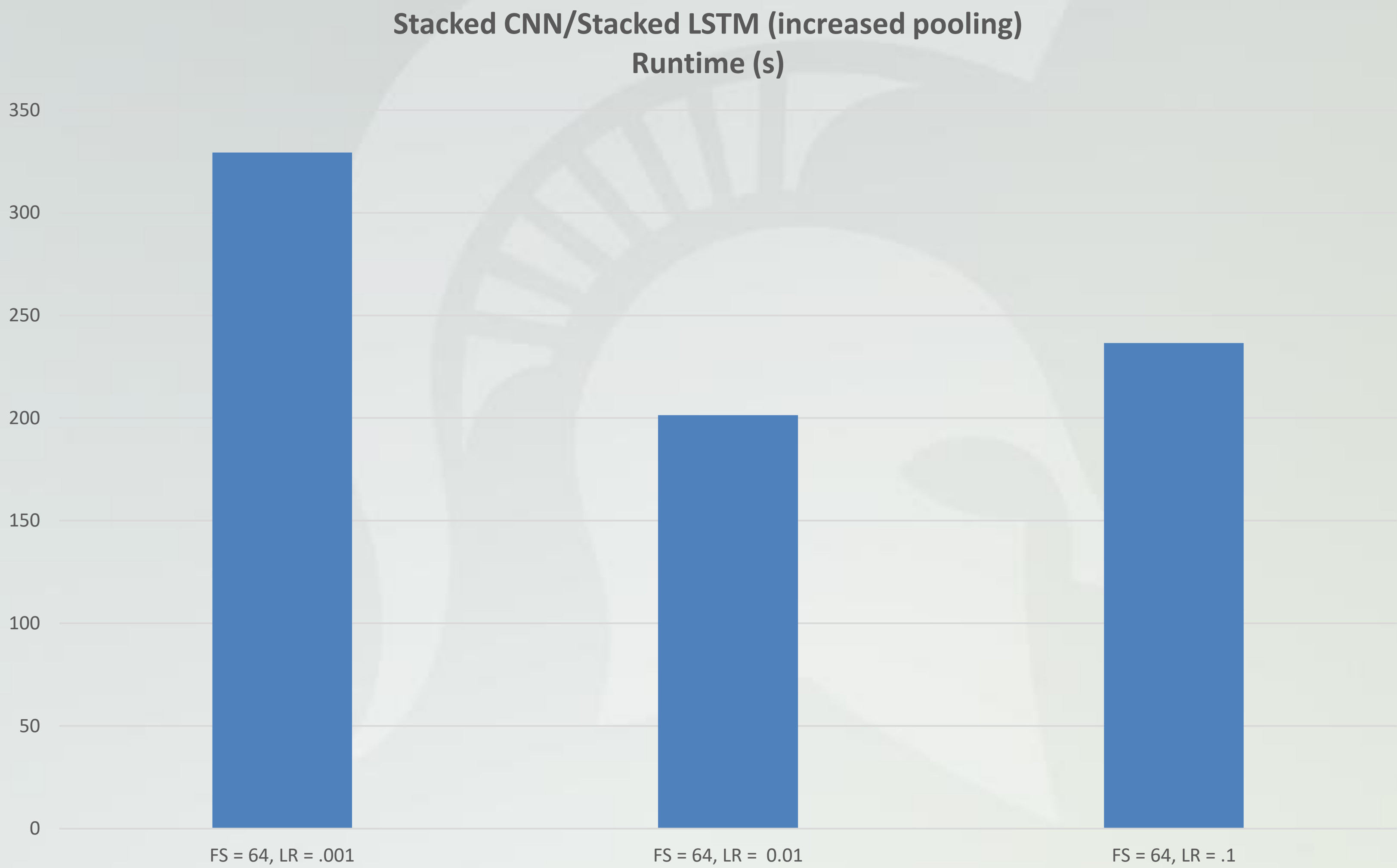
increased pooling



Model Performance – Modified Network 2



Model Performance – Modified Network 2



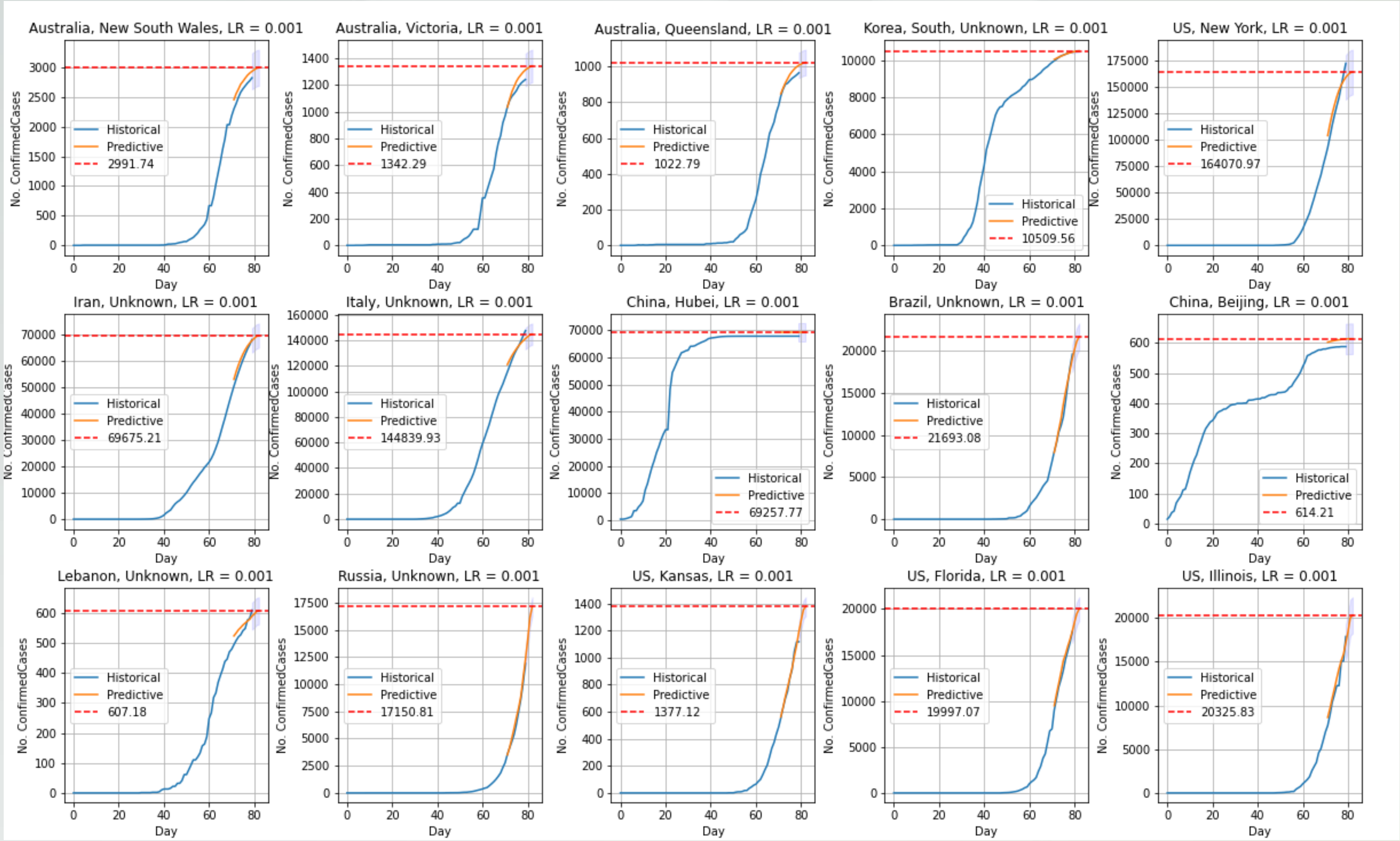
Conclusions

Best RMSLE: 0.055 - Stacked CNN/Stacked LSTM (increased pooling) (FS = 64, LR = .01)

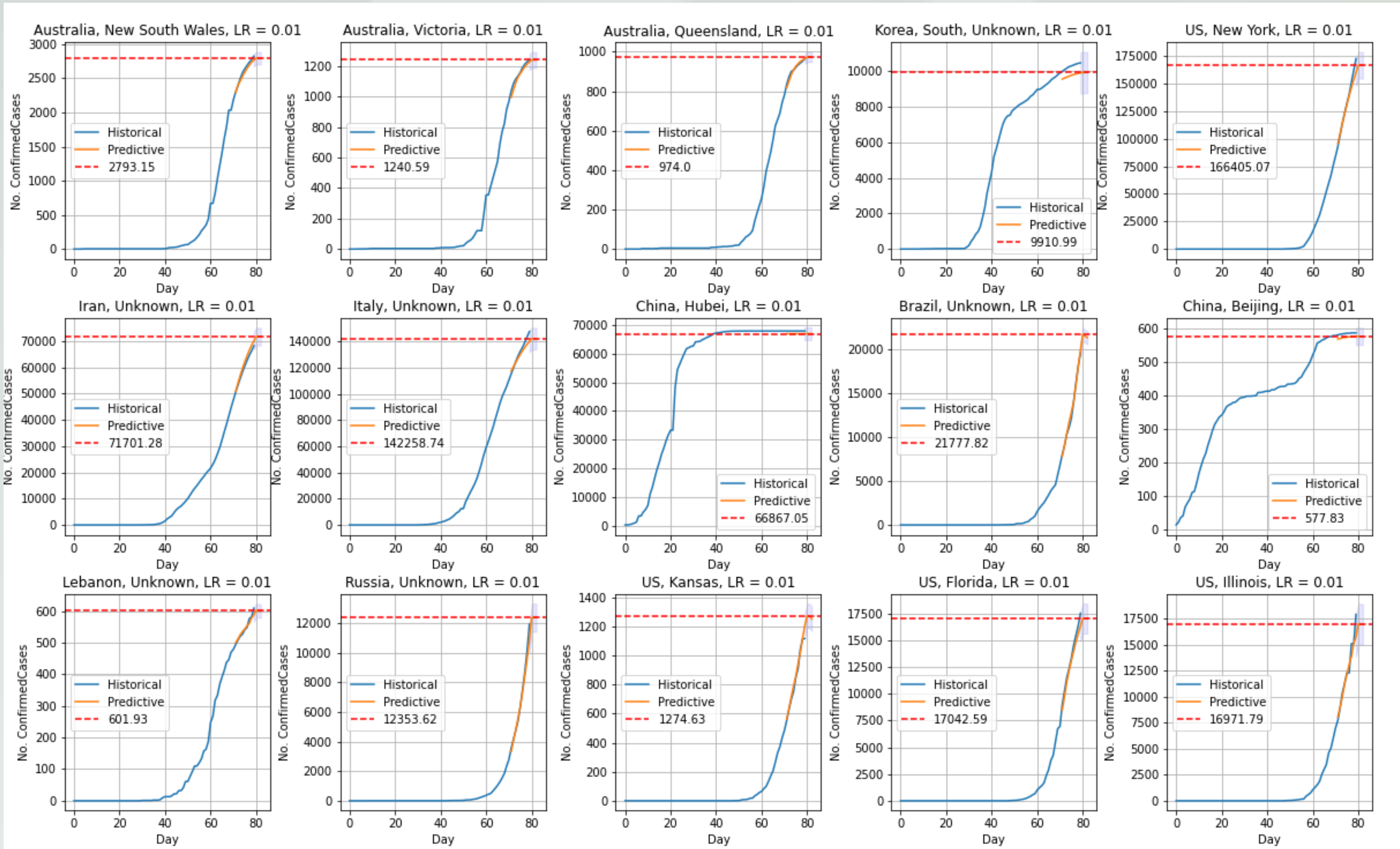
Best runtime – 200.41 (s) – Stacked CNN/Stacked LSTM (FS =64, LR = .001)

Best RMSLE and Runtime: .055 and 201.42 (s) - Stacked CNN/Stacked LSTM (increased pooling) (FS = 64, LR = .01)

Worst Case – CNN/Stacked LSTM, FS = 128, LR = .001
Run time = 224.2 (s), RMSLE = .263



Best Case - Stacked CNN (increased Pooling)/Stacked LSTM FS = 64, LR = .01 – Run time = 201.38 (s), RMSLE = .055



Lessons Learned and Future Work

Filter Size: Varied between 64 and 128. 64 delivered better performance on average

Pooling and CNN Layer: The best accuracy of testing was when using CNN structure with increased convolutional layers and increased pooling layers

Future Implementation: more complex datasets and other architectures can be explored and compared like GRU RNNs for example

References

- [1] “Covid19 Global Forecasting (Week 4),” Kaggle. [Online]. Available: https://www.kaggle.com/competitions/covid19-global-forecasting-week-4.
- [2] “CSSEGISandData/COVID-19,” *GitHub*. https://github.com/CSSEGISandData/COVID-19/blob/master/README.md
- [3] “(PDF) The Impact of Filter Size and Number of Filters on Classification Accuracy in CNN,” www.researchgate.net. https://www.researchgate.net/publication/342999107_The_Impact_of_Filter_Size_and_Number_of_Filters_on_Classification_Accuracy_in_CNN
- [4] Yohancheong, “Forecasting covid-19 infections (CNN LSTM),” Kaggle, 23-May-2020. [Online]. Available: https://www.kaggle.com/code/yohancheong/forecasting-covid-19-infections-cnn-lstm.