*Software and Data Setup:*

Research was conducted in Python using the Google Colab Pro environment, with Keras library and TensorFlow backend. Colab Pro was chosen as normal Colab had idle timeout issues whilst tuning hyperparameters. I tried to circumvent this by applying JavaScript code in the Colab Console which systematically clicks the connect button. Google appears to have patched this as it did not work, presumably as they want people to buy Pro.

The list of library versions is shown below:

| | | |
|---|---|---|
| ¬ Seaborn 0.11.2 | ¬ Numpy 1.21.6 | ¬ talib 0.4.24 |
| ¬ Keras 2.8.0 | ¬ Matplotlib 3.2.2 | ¬ diebold-mariano-test 0.0.8 |
| ¬ Tensorflow 2.8.0 | ¬ statsmodels 0.9.0 | ¬ sklearn 1.0.2 |
| ¬ krakenex 2.1.0 | ¬ h5py 3.5.0 | ¬ Pandas 1.3.5 |
| ¬ patsy 0.5.2 | ¬ Python 3.7.1 | |

Since the development of the CUDA framework by NVIDIA in 2006, the development of applications that take advantage of the extremely parallel capabilities of the GPU has grown greatly including the area of machine learning. (McNally et al. 2018) benchmarked deep learning models on both a GPU and a CPU and noted that training time on the GPU outperformed the CPU implementation by 67.7%. Due to the benefits of utilising a GPU, my LSTM models were implemented on GPU using Cuda. cuDNN is a wrapper of NVIDIA's cuDNN library, which is an optimized library for CUDA containing various fast GPU implementations. CuDNN is built into Colab which was the Python environment I used to conduct this research. This meant that the process of training models for TanH activation was further expedited. This was particularly important for my section hyperparameter tuning where cells can take a considerable amount of time to execute. As cuDNN only provides support for TanH activation function and not ReLU, only half of the models I ran showed further improved speeds.

McNally, S., Roche, J. and Caton, S., 2018, March. Predicting the price of bitcoin using machine learning. In *2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP)* (pp. 339-343). IEEE.

Links:

| | |
|---|---|
| https://www.python.org/ | https://numpy.org/ |
| https://mrjbq7.github.io/ta-lib/ | https://seaborn.pydata.org/ |
| https://www.statsmodels.org/stable/index.html | https://matplotlib.org/ |
| https://pypi.org/project/diebold-mariano-test/ | https://www.tensorflow.org/ |
| https://scikit-learn.org/stable/ | https://pypi.org/project/krakenex/ |
| https://developer.nvidia.com/cudnn | https://pandas.pydata.org/ |
| https://patsy.readthedocs.io/en/latest/ | https://keras.io/ |
| https://developer.nvidia.com/cuda-toolkit | https://www.h5py.org/ |