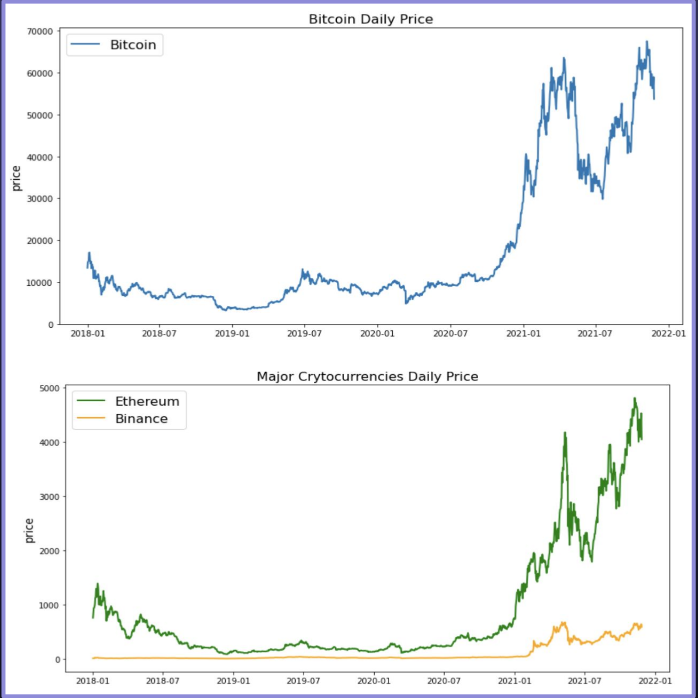
Project Initiative

Symbol	High Price	Max Rise	Max Daily Profit %	Low Price	Max Drop	Max Daily Loss %	Top 3 Months in Volume
ADA	2021-09-02	2021-02-09	33	2020-03-11	2020-03-11	41	2021-02, 2021-05, 2021-01
BNB	2021-05-02	2021-02-18	70	2020-03-11	2020-03-11	44	2021-02, 2020-09, 2021-05
BTC	2021-11-07	2021-02-07	19	2020-03-11	2020-03-11	39	2020-03, 2021-05, 2021-01
ETH	2021-11-07	2021-01-02	26	2020-03-11	2020-03-11	44	2021-01, 2021-05, 2020-03
LTC	2021-05-08	2021-05-23	30	2020-03-11	2020-03-11	38	2021-05, 2020-12, 2021-01

- ❏ Intense amount of data makes impossible for manual high-frequency trading
- ❏ Cross-currency price prediction as coins share similar trends



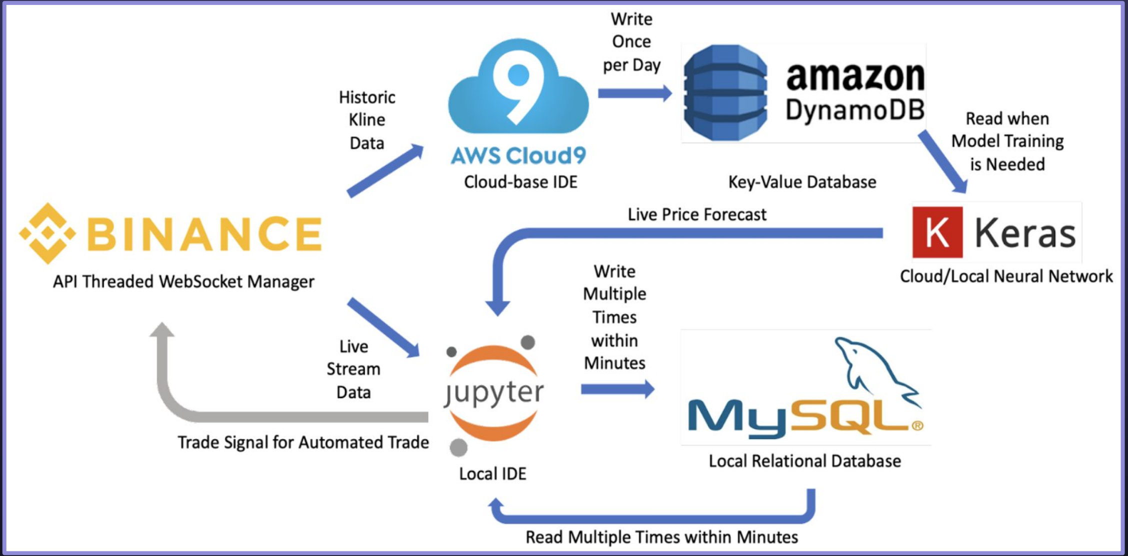
My Role and Responsibility

- ❏ Constructed data warehouse and data streaming
- ❏ Designed recurrent neural network to forecast next-day Bitcoin price

Data Wrangling Process

Historical Kline Data is requested from Binance API and stored in key-value database directly from cloud IDE, to be read into either cloud or local tensorflow shell to train neural network model

Live stream data is requested after connecting to web socket and writen into local relational database, later being used for live price forecast using predefined neural network



Stream Data: Relational Database

Benefit

- ❑ Live stream data is well-structured
- ❑ Fast writing speed
- ❑ Running on local server reduce latency

Limitation

- ❑ Coins information stored in seperate table
- ❑ Multiple joins required to query desired format

```
cnx = create_engine('mysql+pymysql://root:Sun990722@127.0.0.1/Bitcoin')
bm = BinanceSocketManager(client)
ts = bm.multiplex_socket(['btcusdt@ticker', 'ethusdt@ticker','bnbusdt@ticker',
                          'solusdt@ticker', 'adausdt@ticker'])

async with ts as tscm:
    while True:
        msg = await ts.recv()
        frame = createframe(msg)
        if frame.iloc[0,0] == 'BTCUSDT':
            frame.to_sql('BTC', cnx, if_exists = 'append', index = False)
        elif frame.iloc[0,0] == 'ETHUSDT':
            frame.to_sql('ETH', cnx, if_exists = 'append', index = False)
        elif frame.iloc[0,0] == 'BNBUSDT':
            frame.to_sql('BNB', cnx, if_exists = 'append', index = False)
        elif frame.iloc[0,0] == 'SOLUSDT':
            frame.to_sql('SOL', cnx, if_exists = 'append', index = False)
        elif frame.iloc[0,0] == 'ADAUSDT':
            frame.to_sql('ADA', cnx, if_exists = 'append', index = False)
        print(frame)
```

```
  price change percent
0               -0.734
    symbol                    time         open        close         high
0  BNBUSDT 2021-11-30 05:36:13.308  607.80000000  613.80000000  634.20000000

           low traded base asset volume traded quote asset volume  \
0  604.20000000          1116517.84400000         692088530.81090000

   number of trades      best bid best bid quantity      best ask  \
0            901200  613.70000000        49.82400000  613.80000000
```

Real Time
Read & Write

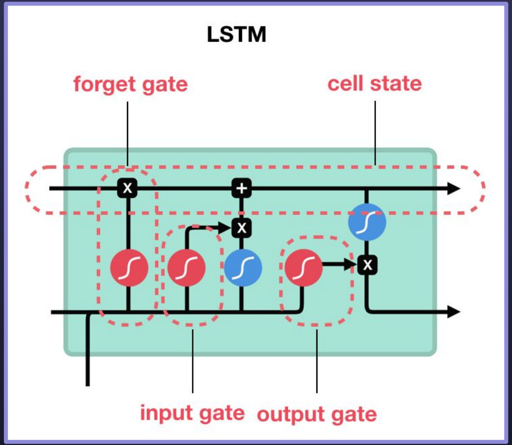
Result Grid   Filter Rows: Export:

symbol	time	open	close
BTCUSDT	2021-11-30 05:40:43	57248.38000000	56728.23000000
BTCUSDT	2021-11-30 05:40:42	57248.37000000	56728.22000000
BTCUSDT	2021-11-30 05:40:41	57251.34000000	56732.35000000
BTCUSDT	2021-11-30 05:40:40	57251.34000000	56734.90000000
BTCUSDT	2021-11-30 05:40:39	57251.34000000	56736.61000000
BTCUSDT	2021-11-30 05:40:38	57251.34000000	56738.28000000
BTCUSDT	2021-11-30 05:40:37	57251.38000000	56740.77000000
BTCUSDT	2021-11-30 05:40:36	57251.38000000	56741.04000000
BTCUSDT	2021-11-30 05:40:35	57251.38000000	56741.03000000
BTCUSDT	2021-11-30 05:40:34	57251.38000000	56742.02000000

Model Construction

Goal

- ❏ Use daily trade data of five cryptocurrencies from 2018 to predict future Bitcoin price
- ❏ Generate **likelihood between 0 to 1** of next-day price increase



Long Short Term Memory

- ❏ Stateful in theory
- ❏ All states are propagated to the next batch
- ❏ Alleviate vanishing gradient problem: train longer sequence

Hyperparameter Tuning: Bayesian Optimization

- ❏ Faster than exhaustive search
- ❏ 'Smarter' than random search — learn from past results to determine next best parameter

Model Summary

Architecture:

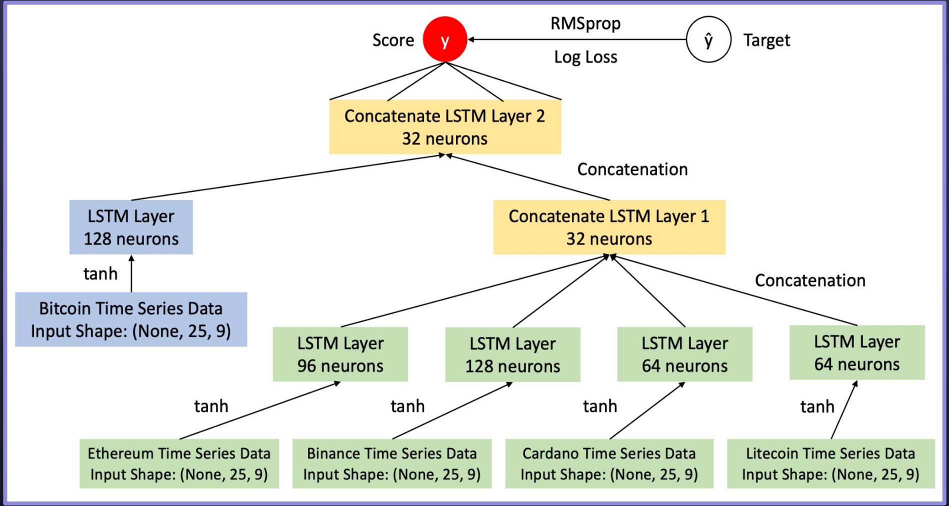
- ❏ Wide neural network
  - ❏ Better memorization than generalization
- ❏ RMSprop is commonly used optimizer in RNN
- ❏ Hyperbolic tangent function produces better prediction compared to relu function

Insight:

- ❏ More neurons in Ethereum and Binance coin hidden layers

Accuracy:

- ❏ 0.54 ROC AUC



Model Performance

One-day Buy Sell Strategy

- ❑ Use LSTM model to predict next-day Bitcoin price
- ❑ Buy if prediction value is above certain threshold and sell at the end of the same day
- ❑ Avoid consecutive buying

Backtesting (0.1% transaction fee)

- ❑ **582%** market return from March 2020 to November 2021 (0.05 threshold)
- ❑ 364% benchmark return if buy and sell without price forecast
- ❑ Model avoids small market crash.