



# SMPE: Stock Market Prediction on Edge

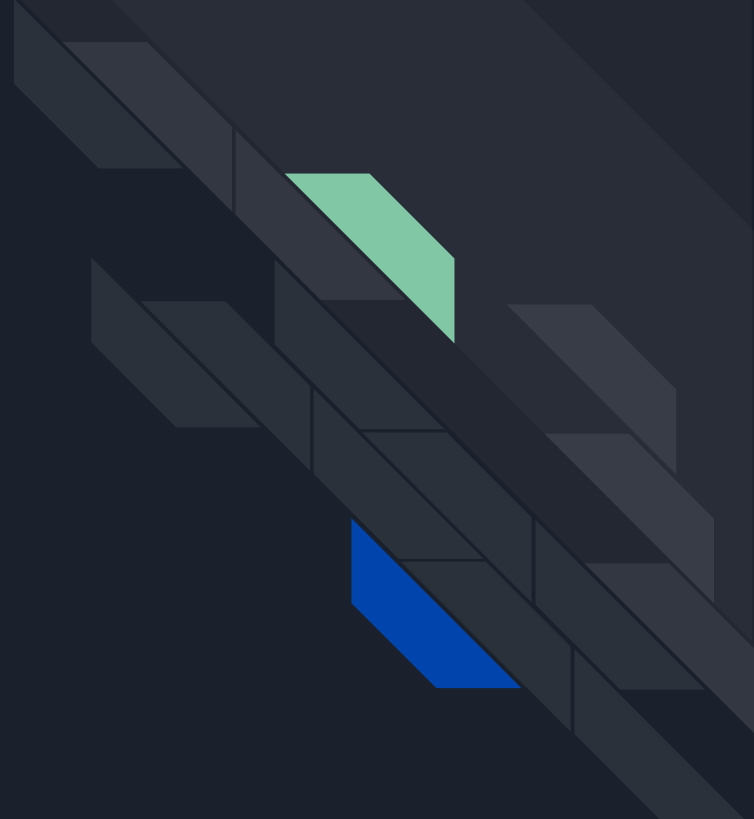
Bryan Chan

Zi Yi Chen

Qi Zhao

# Agenda

- ❖ Introduction & Overview
- ❖ Architecture & Design
- ❖ Evaluation
- ❖ Conclusions & Future Work



# Introduction & Overview



- What is stock market prediction
- Traditional methods and challenges
- Our motivation

# What is stock market prediction?



- Predict the future value of a stock or other financial instruments traded
- Many factors can affect the stock market
- A successful prediction of a stock's future price could yield significant profit



# Traditional methods, challenges and our motivation



- Ideas on using deep neural net models
- Require heavy computation, so prediction is done on the cloud
- Cloud imposes high latency to mobile user
- Day traders travel more frequently nowadays and cannot get up-to-date predictions in time due to latency issue
- We attempt to reduce the latency by offloading the computation on the edge instead of the cloud



# Our Goals



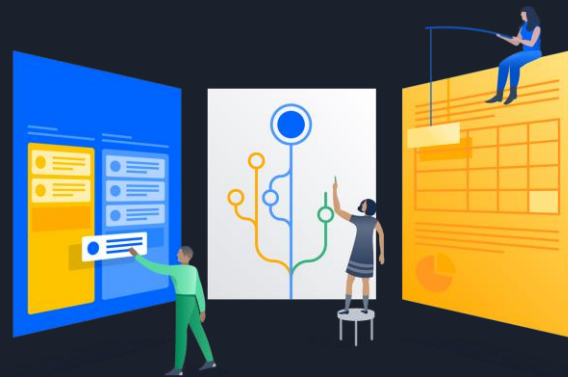
- Reduce latency
- Reduce bandwidth
- Reduce energy consumption

# Scenarios

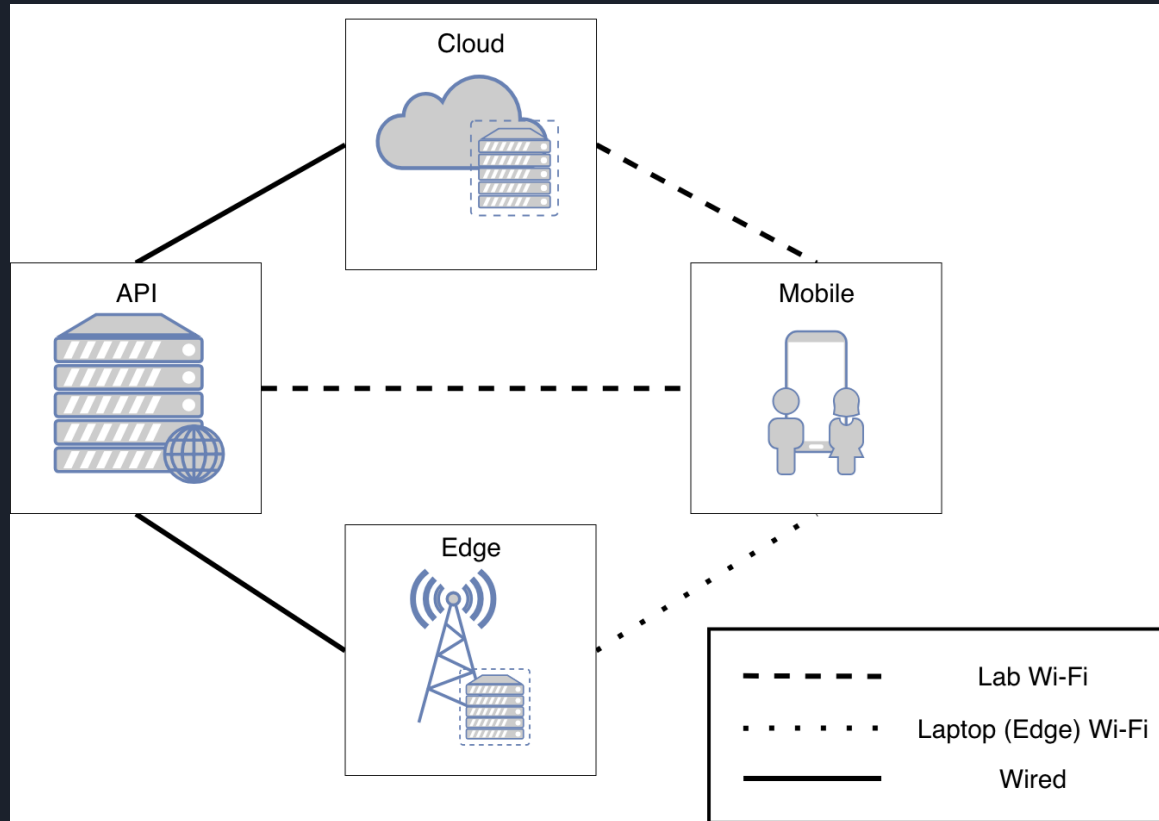
All scenarios use the same LSTM model trained prior to the experiments.

All experiments are done by a custom Android application

- S1 (Cloud): Predictions made on cloud and relayed to app
- S2 (Edge): Predictions made on edge and relayed to app
- S3 (Mobile): Predictions made on app

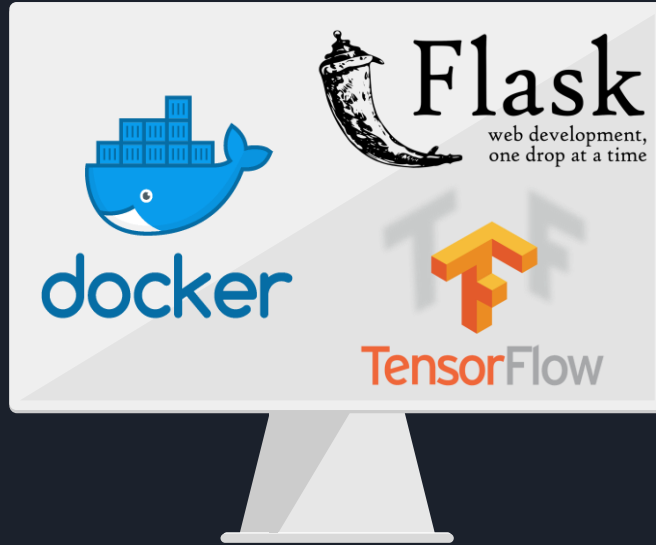


# Network Topology



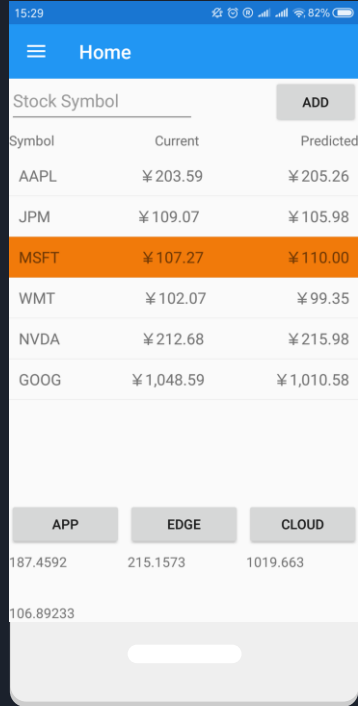


# RESTful Service on Cloud & Edge



- ❑ Dockerized RESTful service built on Python Flask
- ❑ A simple LSTM (only one layer) model built using TensorFlow and stored locally in the container

# Android Application



Stock Symbol	Current	Predicted
AAPL	¥203.59	¥205.26
JPM	¥109.07	¥105.98
MSFT	¥107.27	¥110.00
WMT	¥102.07	¥99.35
NVDA	¥212.68	¥215.98
GOOG	¥1,048.59	¥1,010.58

APP	EDGE	CLOUD
187.4592	215.1573	1019.663
106.89233		

## Functionalities:

- ❑ Choice to predict a specified symbol
- ❑ Choice to use one of the scenarios to perform prediction
- ❑ Displays the predictions of historical/hottest symbols
- ❑ Displays different latency factors
- ❑ Use the model trained previously to predict on the phone (TensorFlow Lite)

# Execution Environment

Scenario 1:



- East US
- 1 Core
- 1.5 GB RAM

Scenario 2:



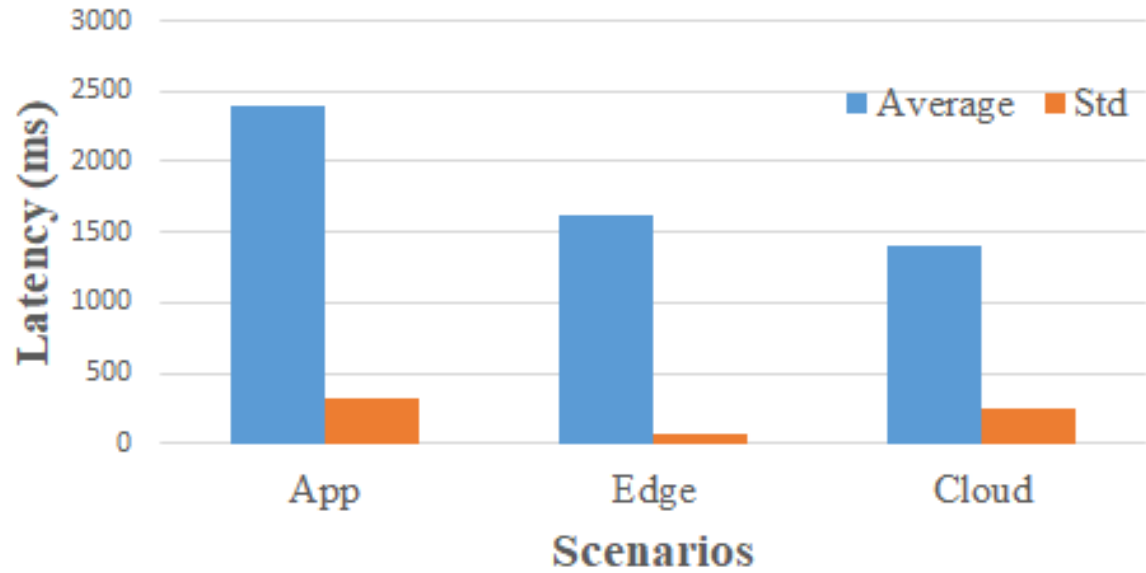
- Intel Core i7
- 802.11n

Scenario 3:



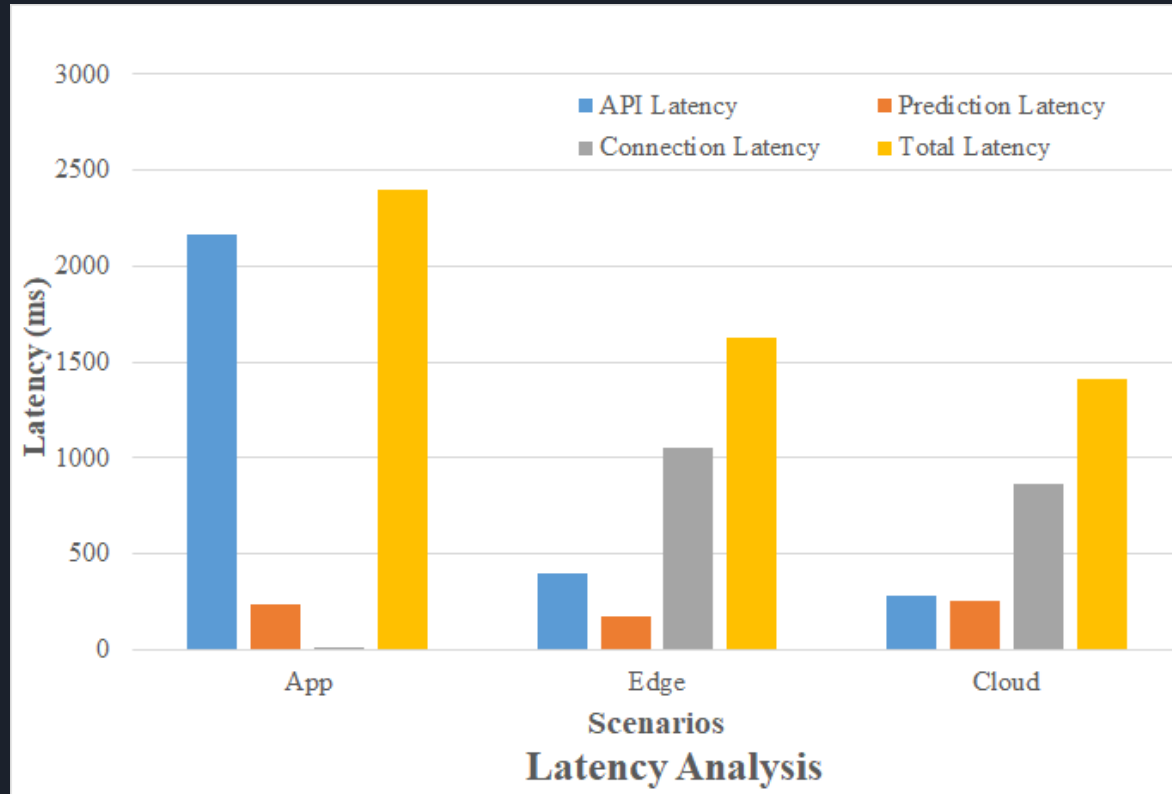
- Snapdragon 625
- 802.11n

# Evaluation

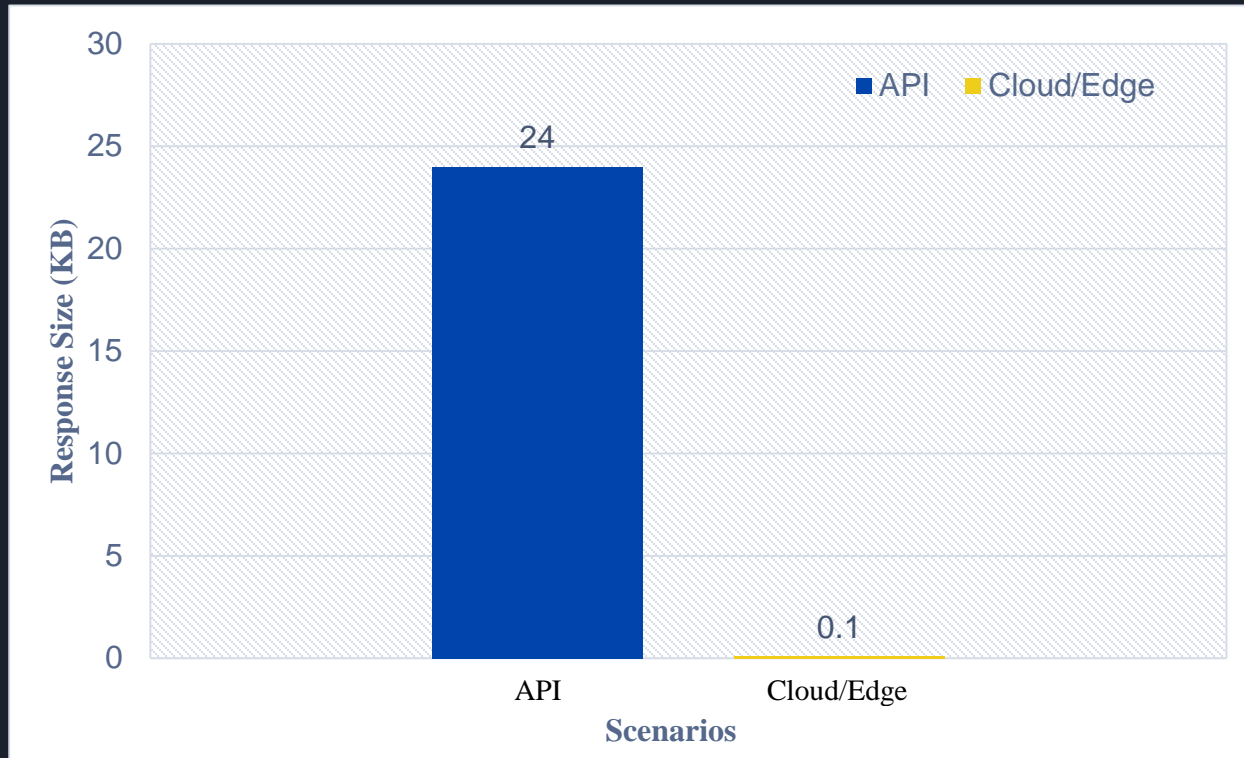


**Average Total Latency and Standard deviation**

# Evaluation Cont.



# Evaluation Cont.



## Future work & Limitation

- Investigate stream data processing where users can get real-time update & prediction with the app open
- Perform prediction with high accuracy (possibly a more complex model)
- Experiments under a controlled environment
- A better API that can provide live-data





**Questions?**