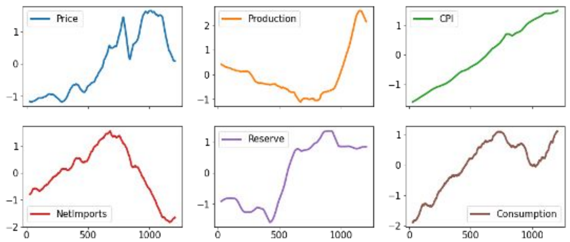
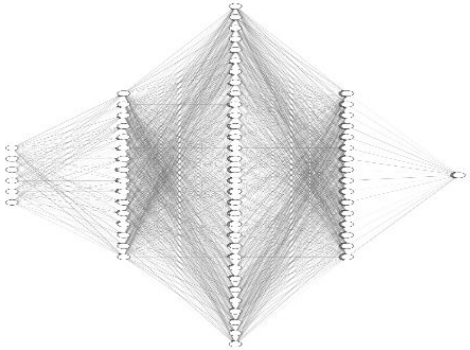
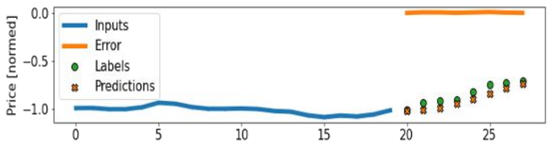
Predicting Gasoline Prices using Neural Networks







Team Oglesby
Dimitrios Bralios, Siva Kumar Valluri, Pallav Ranjan, Pratik Sinha

Towards partial fulfillment of course requirement of
IE 534/ CS 547 Deep Learning
University of Illinois Urbana-Champaign

Feature Selection

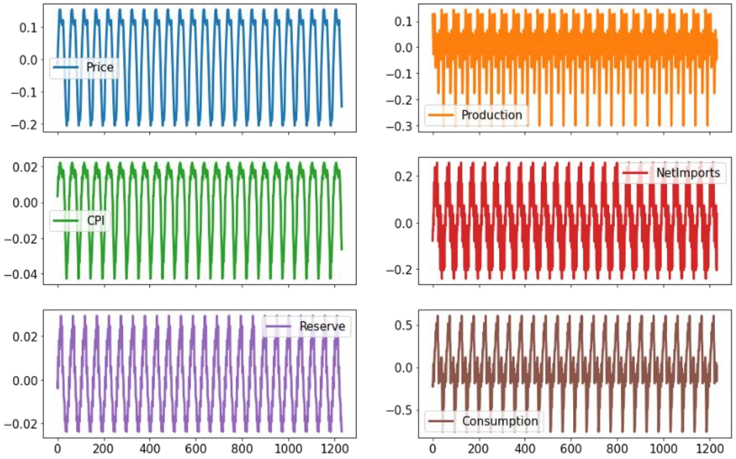
- Gas prices primarily governed by the law of supply and demand[1,2]
- **Supply:**
  - Domestic production
  - Strategic Petroleum Reserve
  - Import from other nations
- **Demand:** reflected by
  - daily domestic consumption
- **Miscellaneous** factors:
  - Inflation (measured as Consumer Price index (CPI))
- Data from US Energy Information Administration

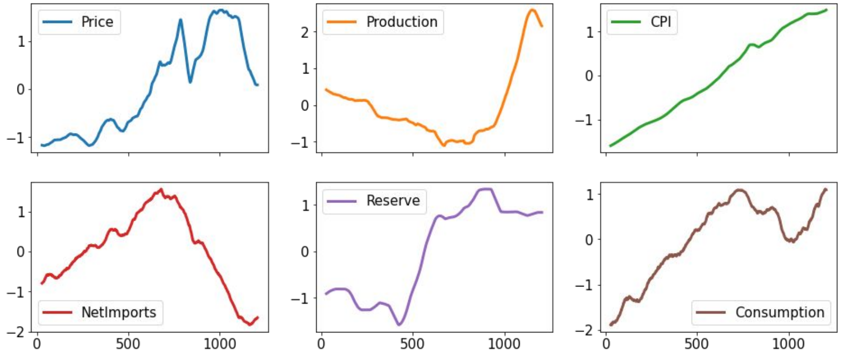
Speaker: Pallav

1. Lahari, M. C., Ravi, D. H., & Bharathi, R. (2018, September). Fuel Price Prediction Using RNN. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1510-1514). IEEE.
2. https://www.investopedia.com/articles/economics/08/gas-prices.asp

Contributors
Pallav, Dimitrios, Pratik, Siva

Trends in the features



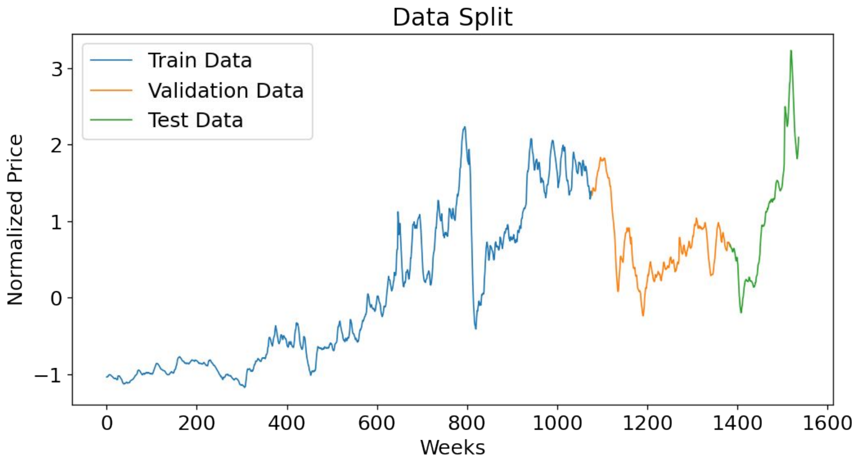


- The data set exhibits seasonality: Annual and Monthly (plot shows annual)
- Trends: Annual seasonality removed data shows trends seen in second image

Contributors
Siva, Pallav, Dimitrios, Pratik

Data Splitting & Normalization

- Before continuing with the experiments we split our data into train, validation and test sets.
  - 20% Validation, 10% Test as shown below

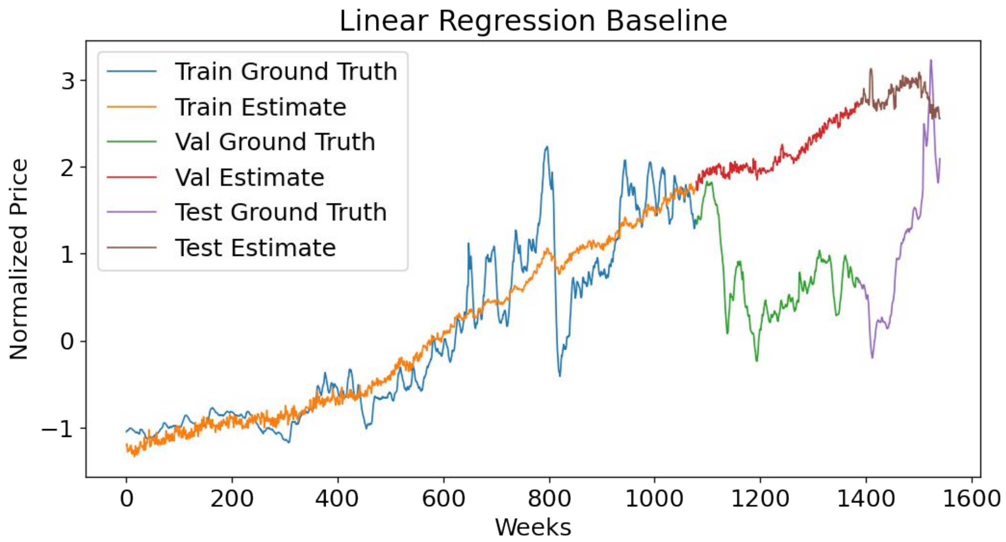


- We select the validation and test sets as the latest data in order to avoid any information leak from them to the train set.
- Then our data are z-score using the mean and standard deviation of the train data

Contributors
Dimitrios, Pallav, Siva, Pratik

Exploring Baseline: Linear Regression Model

- Next week price prediction by linear regression **without** using **historical price data**

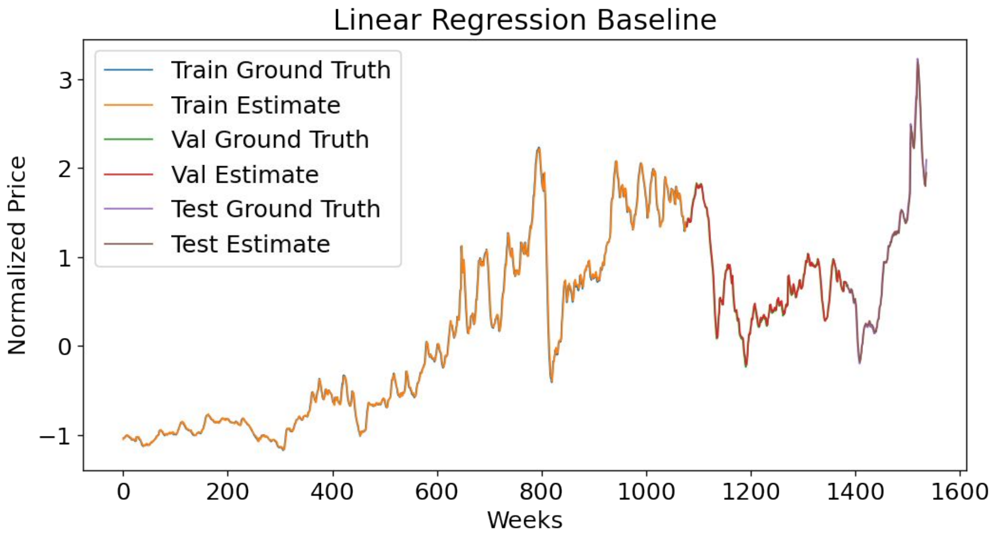


- The limited amount of train data which also have an upward trend, leads to poor performance in the validation and test sets.
- Possible distribution shift over time → Makes prediction harder

Contributors
Dimitrios, Pallav, Siva, Pratik

Exploring Baseline: Linear Regression Model

- Next week price prediction by linear regression **using current price value**

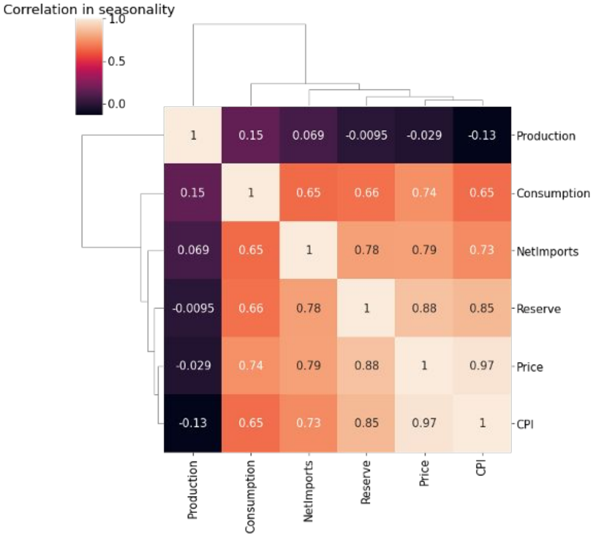


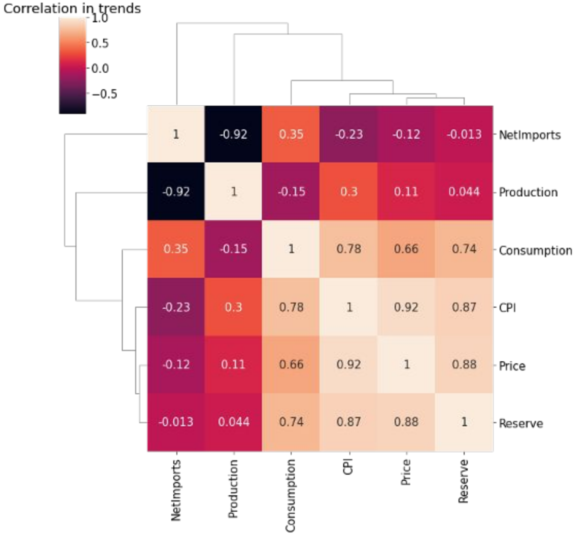
- Drastic improvement of performance → Highlights the need for historical price data

Contributors
Dimitrios, Pallav, Siva, Pratik

Feature Importance: Relevant Features

Pearson Correlation Analysis: CPI > Reserve > Net Imports are the most important features correlated to Price





- The features chosen are clearly correlated as seen in both their seasonality and trends. So, feature importance needs to be verified by other analysis, preferably model-free as well.
- Eg: Net Imports is strongly correlated to Reserve, CPI and Consumption.

Speaker: Pratik

Contributors
Pratik, Siva, Dimitrios, Pallav

Feature Importance:Additional tests

ANOVA Statistical Test: **CPI > Reserve > Consumption**

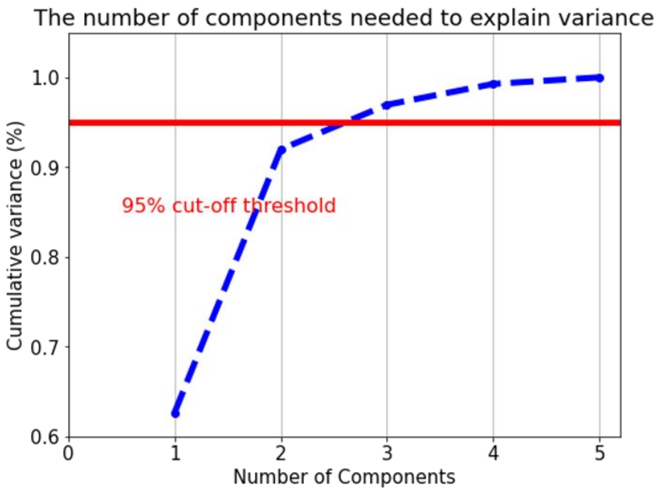
```
The ordered feature list based on importance is: ['CPI', 'Reserve', 'Consumption']
```

Recursive Feature Elimination: Removes codependent features and yields **Production and CPI**

```
The feaures of importance are ['Production', 'CPI']
```

Principal Component Analysis: Clearly shows more than 2 variables are sufficient to get less than 5% variance.

```
Index(['Production', 'CPI', 'NetImports', 'Reserve', 'Consumption'], dtype='object')
```



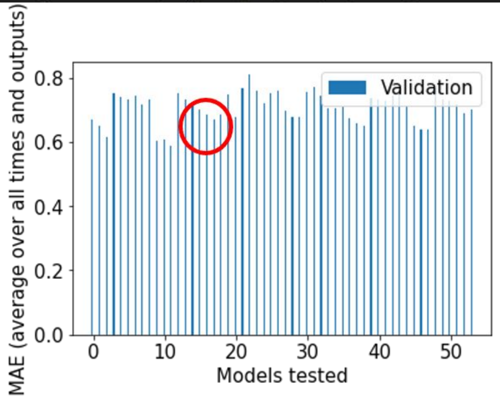
CPI consistently shows up as an important feature while the other features show strong co-dependencies.

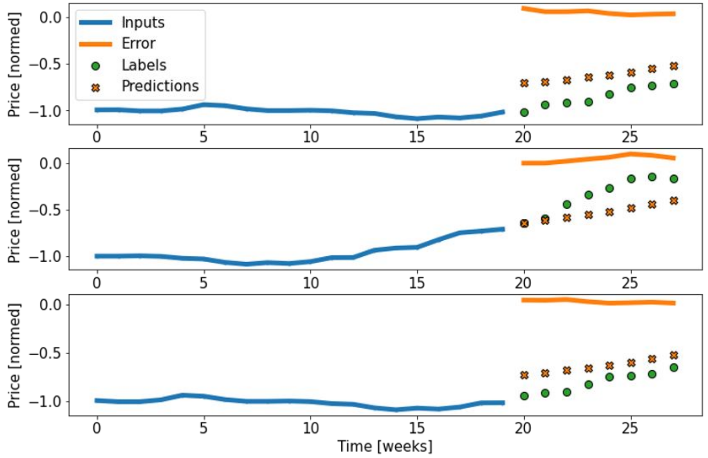
The price dependency is strongly dictated by 2 features of which CPI is likely the strongest feature. Also, Linear regression model improves tremendously when historic price is used as a feature suggesting need for 'memory'

Contributors
Pratik, Siva, Dimitrios, Pallav

Recurrent Neural Networks: Role of Memory

```
model = tf.keras.Sequential([
  tf.keras.layers.LSTM(neuron, activation=activation_def, recurrent_activation=activation_rec, dropout = dropout_rate, return_sequences=False),
  tf.keras.layers.Dense(Output_steps*num_features, activation= activation_dense),
  tf.keras.layers.Reshape([Output_steps, num_features])
```





Optimizing Hyperparameters

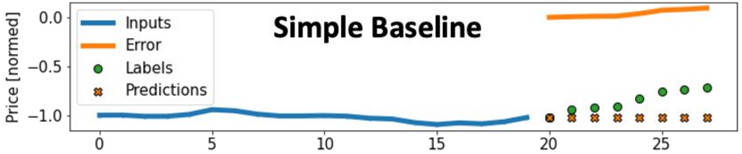
- Input points: [10,**20**,30]
- Activation functions: Feed to LSTM layer (**Tanh**), recurrent feedback (**Sigmoid**), dense layer (**Tanh**)
- Dropout: 0,**0.1**,0.2
- Number of neurons in the LSTM layer: 16,**28**,32,64
- Number of max epochs: 30,50,**100**
- Optimizers: **Adam** and SGD

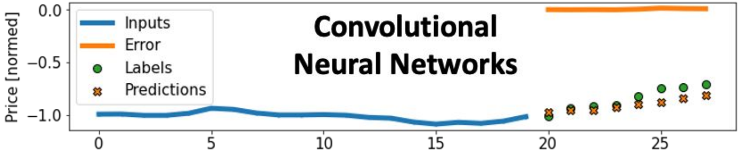
Speaker: Siva

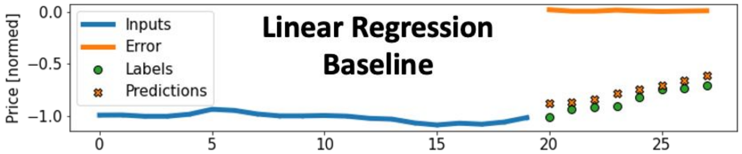
Contributors
Siva, Dimitrios, Pallav, Pratik

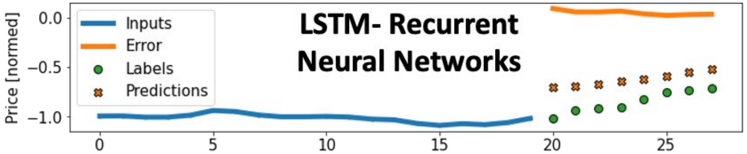
Predicting Multistep Multivariate Time Series

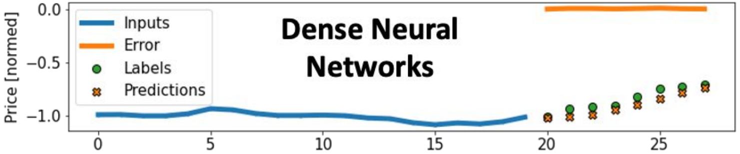
- Objective: Test multiple models, where best model is the one that is able to predict multiple time steps: 8-weeks using 20-week data as input.
- Metric used: Mean Absolute Error (MSE). The model with least complexity and lowest MAE, that beat the baselines are ideal models
- All models are shallow models with 2 layers (1 LSTM/Con1D and 1 dense layer) tested on 30 epochs







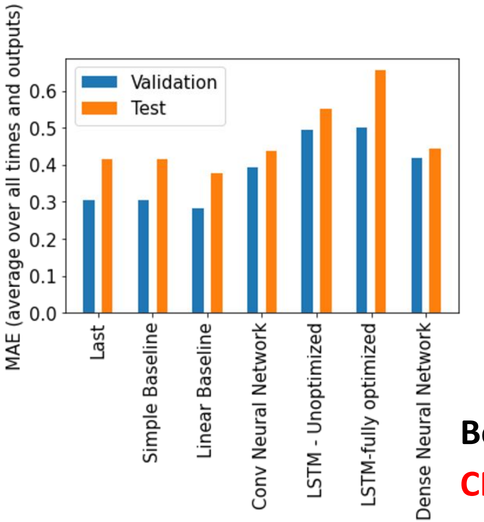




Contributors
Siva, Dimitrios, Pallav, Pratik

Predicting Multistep Multivariate Time Series

- Objective: Test multiple models, where best model is the one that is able to predict multiple time steps: 8-weeks using 20-week data as input.
- Metric used: Mean Absolute Error (MSE). The model with least complexity and lowest MAE, that beat the baselines are ideal models
- All models are shallow models with 2 layers (1 LSTM/Con1D and 1 dense layer) tested on 30 epochs



Shallow Models pursued:

- Convoluted Neural Network (CNN)
- Long/Short-term Memory (LSTM) flavored Recurrent Neural Network (RNN)
- Dense (Fully-connected) Neural Network
- Baselines:
  - Linear, Flat Baselines- Last and Simple Averaged

Best Models:

CNN and Dense NN

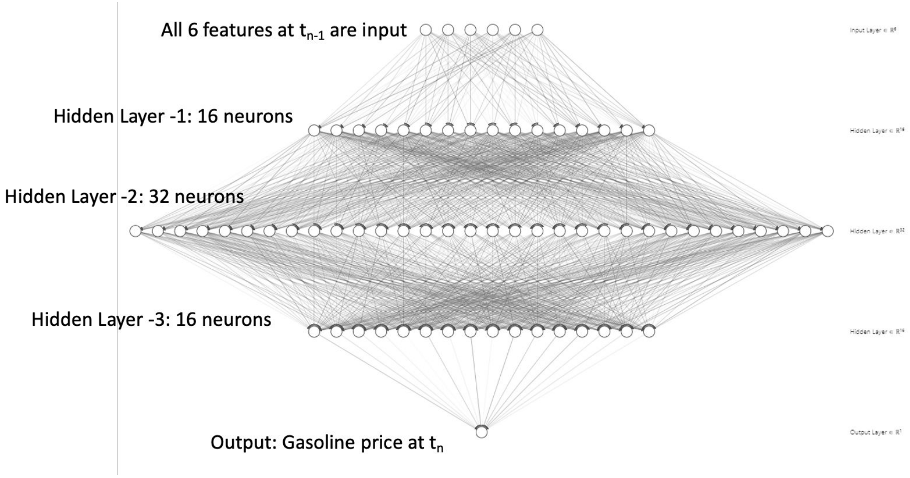
LSTM fails

Contributors
Siva, Dimitrios, Pallav, Pratik

Deeper NNs: DNN, CNN

Based on the preliminary results we develop deeper neural network models

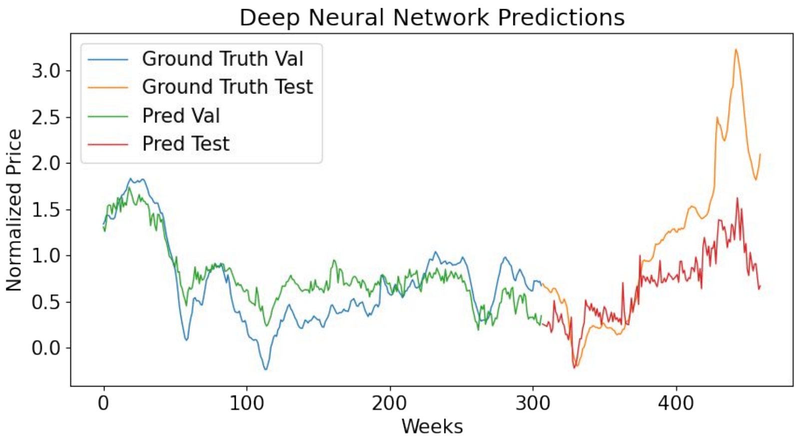
- A deep, four-layer, fully connected (dense) network (shown below)
- A deep, three-layer, convolutional neural network
    - We use **causal** convolution which uses only prior data and not any future data

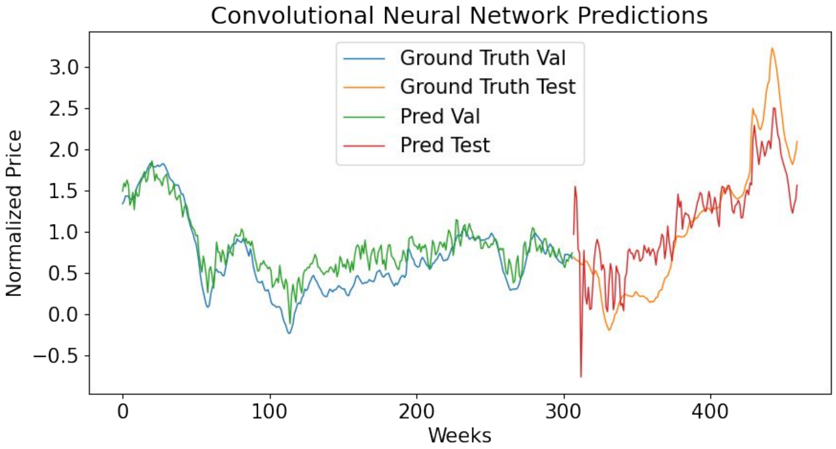


Speaker: Dimitrios

Contributors
Dimitrios, Pallav, Siva, Pratik

Deeper NNs: Results





	MAE CNN	MAE DNN	MSE CNN	MSE DNN
Val	**0.18**	0.23	**0.046**	0.072
Test	**0.35**	0.52	**0.19**	0.48

Contributors
Dimitrios, Pallav, Siva, Pratik

Conclusions

- Linear Regression outperforms simple neural network models
- The Convolutional Neural Network achieves the best results out of the neural network models
- The small number of samples (~1500) render the neural networks ineffective for the task

Contributors
Pallav, Dimitrios, Pratik, Siva