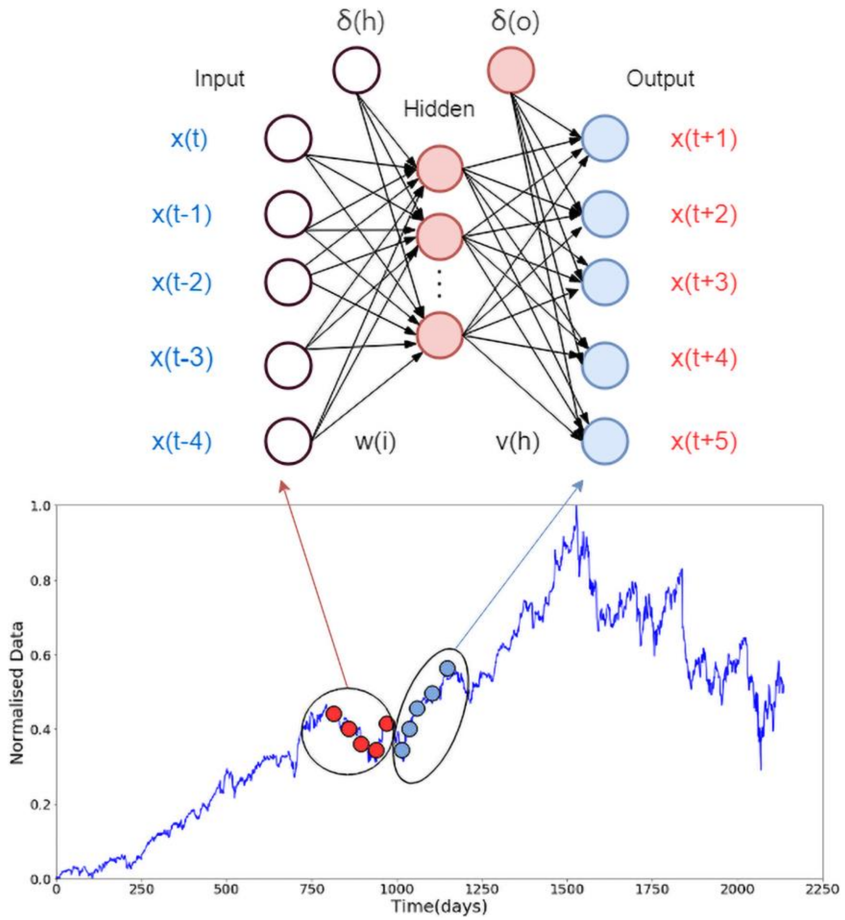
Stock Market Prediction using Ensemble of Graph Theory, Machine Learning and Deep Learning Models

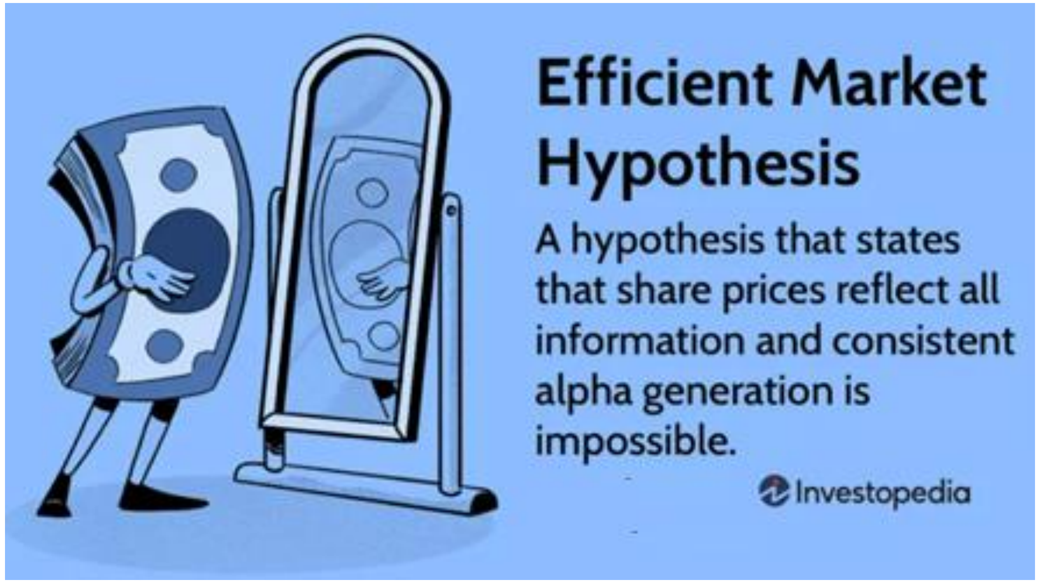
(2020, ACM: Pratik Patil, Ching-She, Katerina Potika, Marjan Orang)



Present my Implementation

Sayed Ahmed

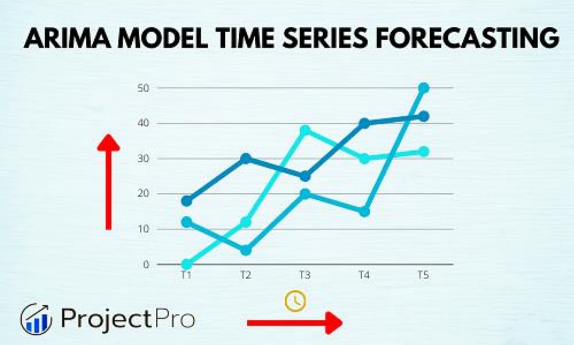
The Problem

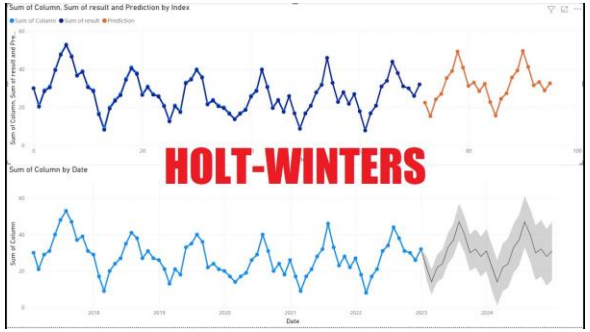


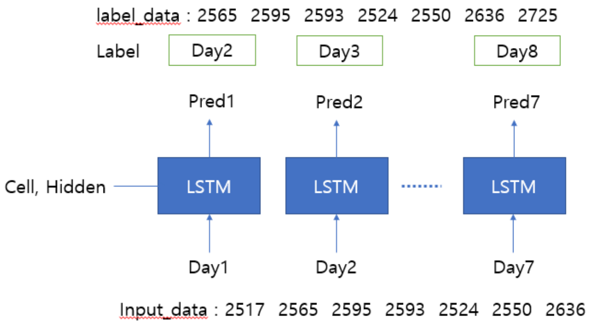
– "Opponents of EMH believe that it is possible to beat the market and that stocks can deviate from their fair market values. " [2]

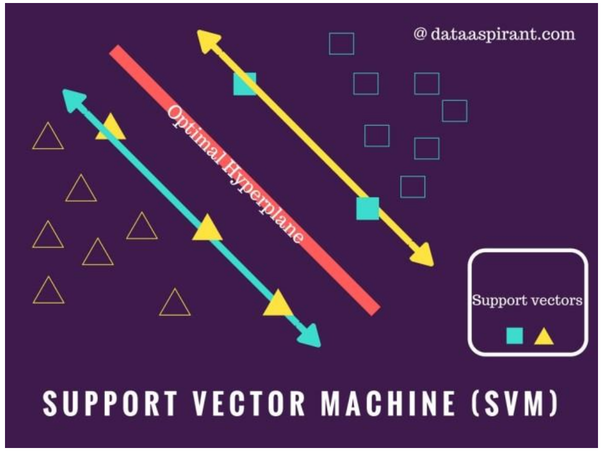
- **We predict stocks using**
  - Usually: Proper selection of
    - Stocks and Time Intervals
      - Statistics
      - Machine Learning
- **This paper goes beyond and Uses**
  - Spatio-Temporal Relationship
    - among stocks
  - Graph theory

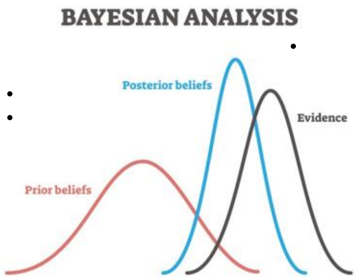
Related Work





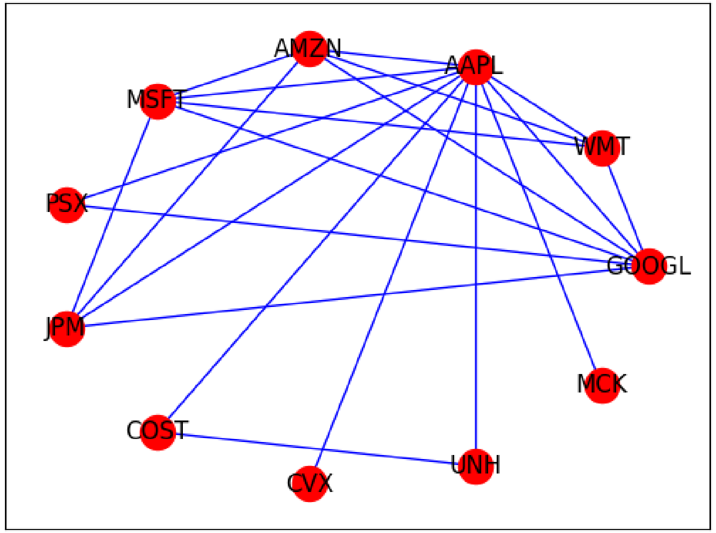


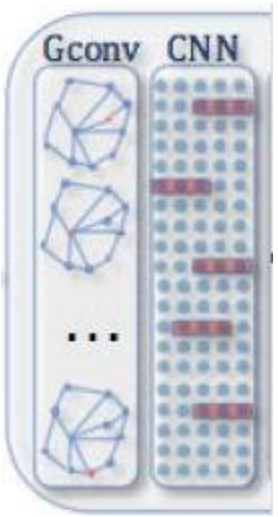




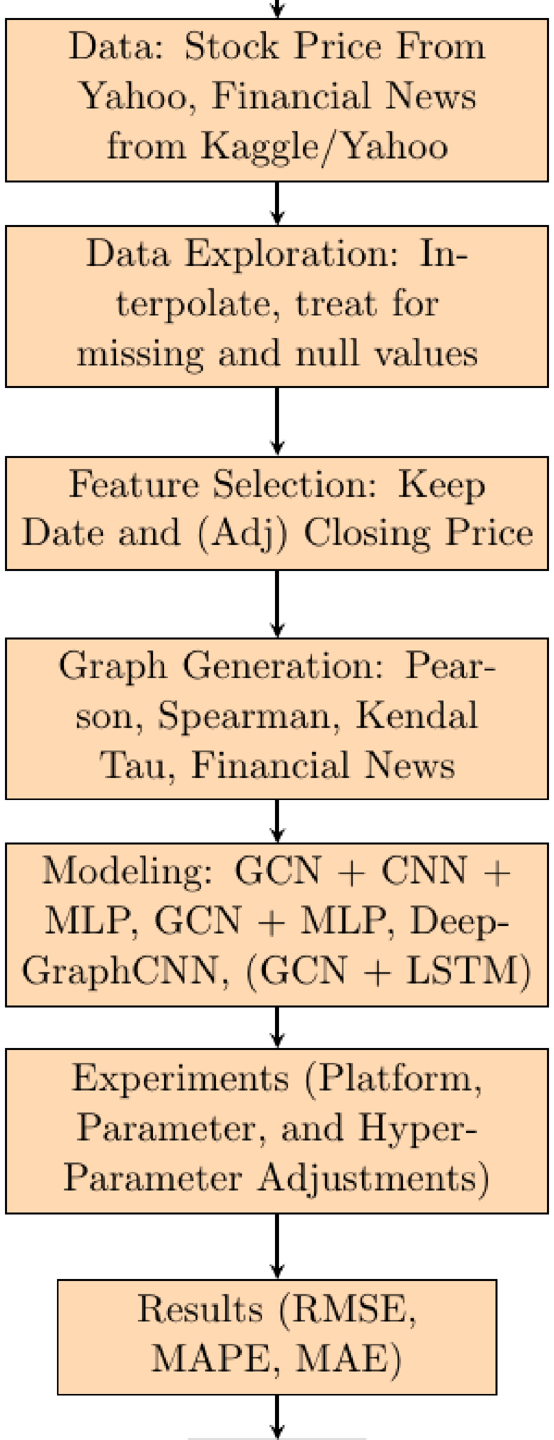
[1]

Graph Based





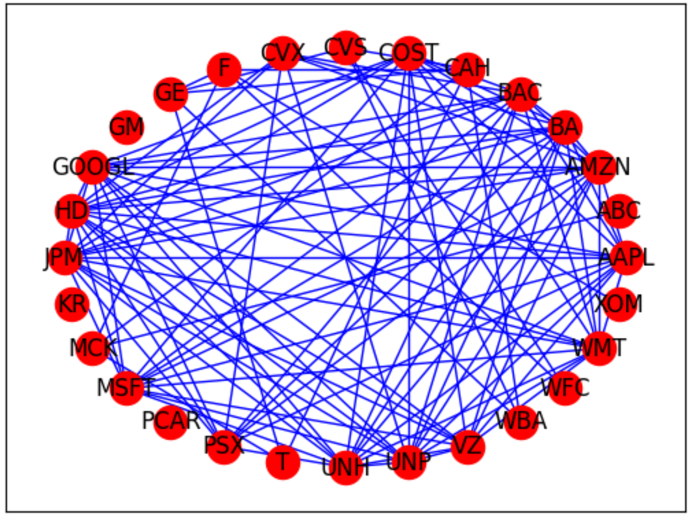
Methodology



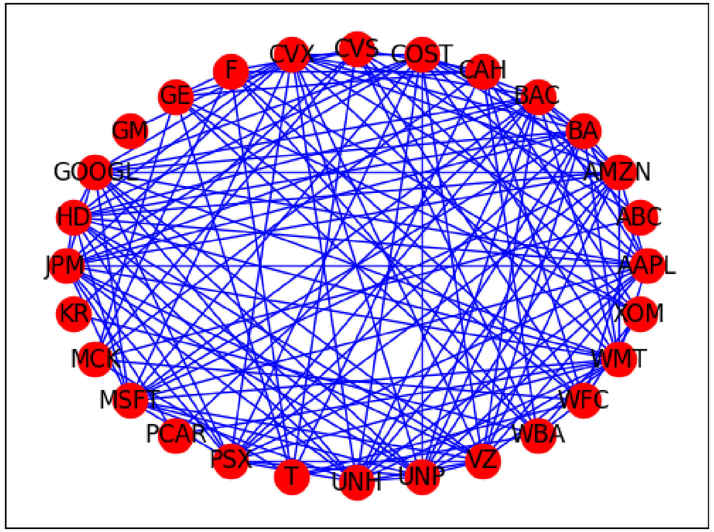
Dataset

- **Final**
  - 30 (28) Stocks from
    - Fortune 500 companies
  - Stock Price Data
    - 1-day interval (2017-01 to 2019-12)
  - News dataset:
    - Yahoo News Articles
    - Couple Thousands (> 15000)
- **Another Dataset/Started with**
  - From Kaggle
  - With Nasdaq, NYSE data

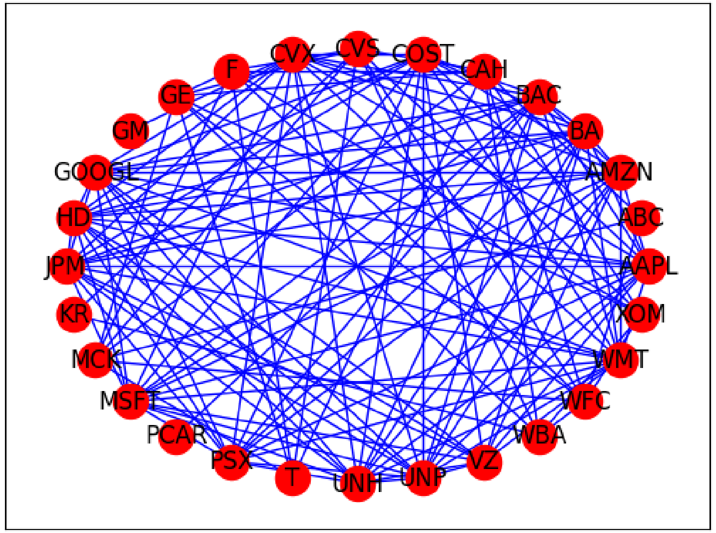
Model Stocks into Graph



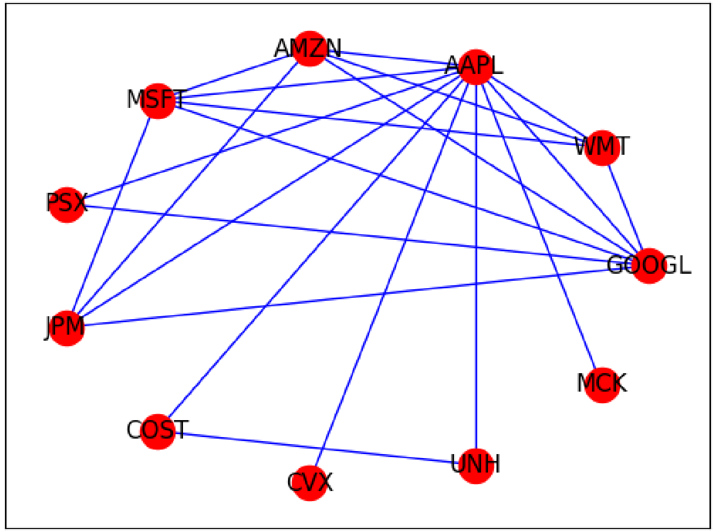
Pearson based



Spearman based



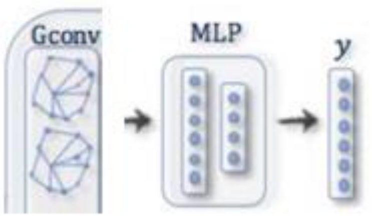
Kendal Tau based



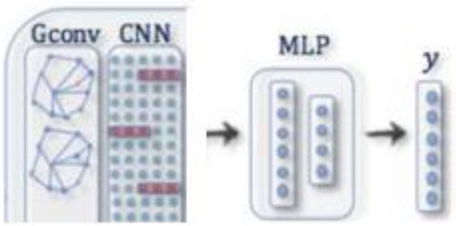
Financial News based

Models and Platforms

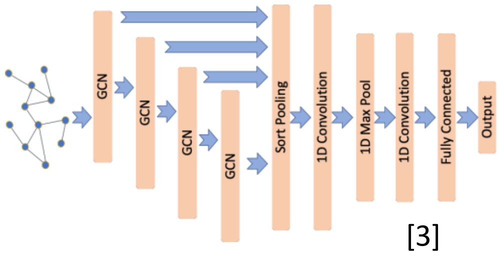
Models



GCN + MLP

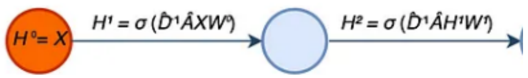


GCN + CNN + MLP



[3]

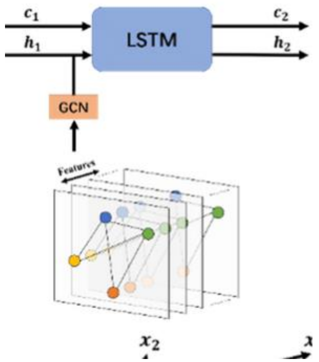
DeepGraphCNN



$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

$$\mathbf{x} *_G \mathbf{g}_\theta \approx \theta(\mathbf{I_n} + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{x}$$

Raw Equations



Platforms

- CPU with 16 GB ram
- Intel Coe i7
- Jupyter Notebook
- Python
- Stellar Graph
- Networkx
- NLTK
- yFinance

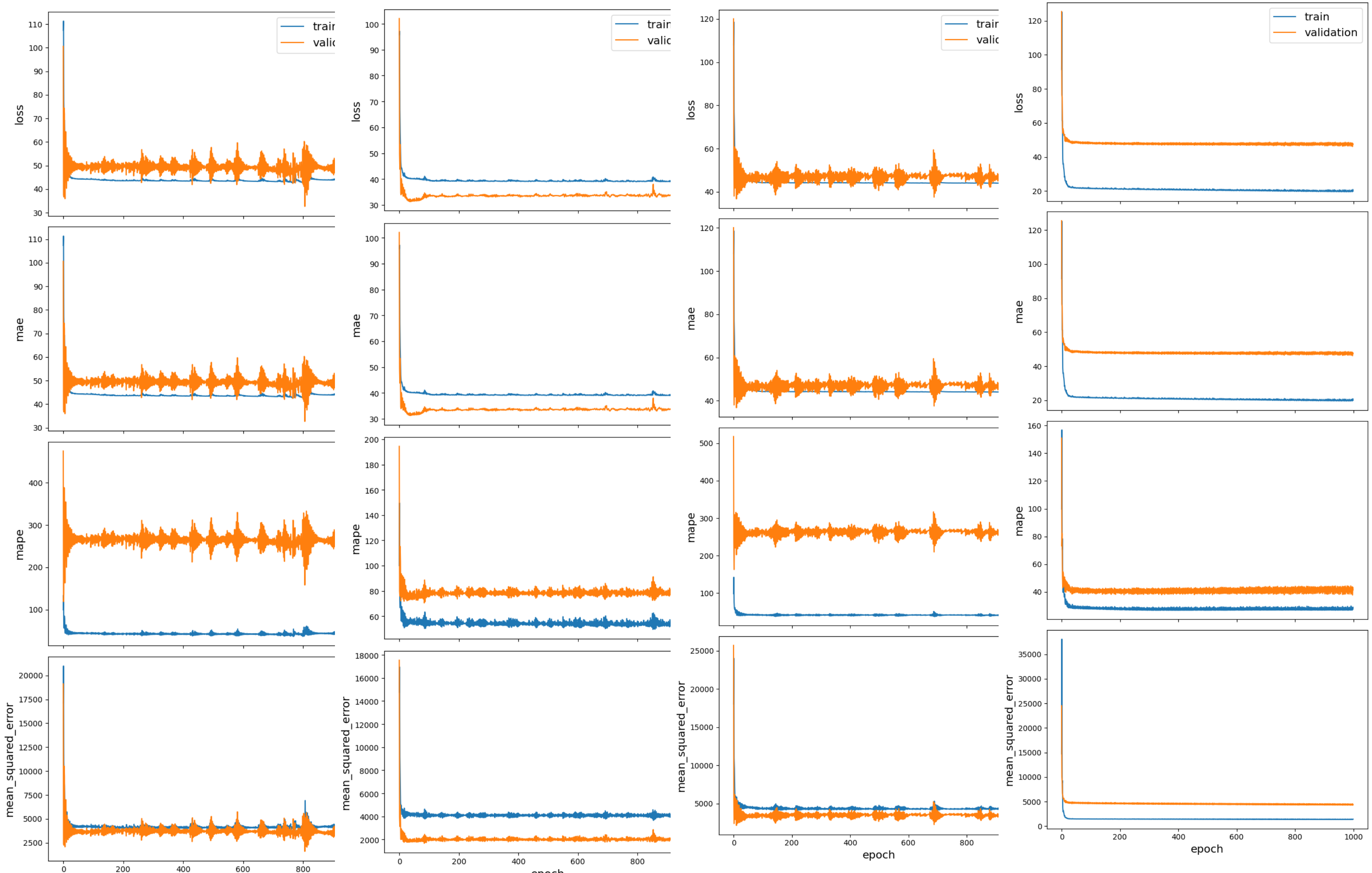
Results: 1st Case: GCN + MLP: Performance (1000 Epochs)

Loss

MAE

MAPE

MSE

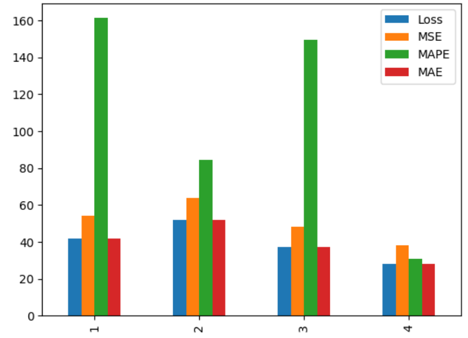


Pearson based

Spearman based

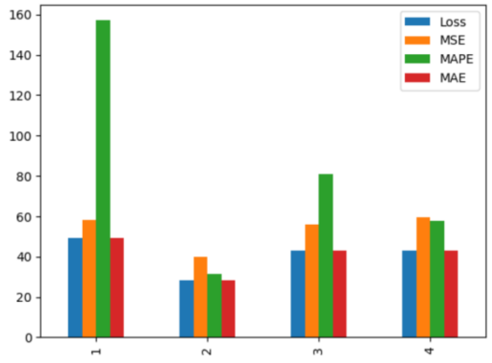
Kendal Tau based

News based



	1	2	3	4
Loss	42.0	51.9	37.4	28.2
MSE	54.1	63.7	48.4	38.3
MAPE	161.1	84.6	149.6	30.7
MAE	42.0	51.9	37.4	28.2

10k Epochs



Loss	49.2	28.3	43.1	43.0
MSE	58.2	39.9	56.1	59.3
MAPE	157.0	31.4	81.0	57.5
MAE	49.2	28.3	43.1	43.0

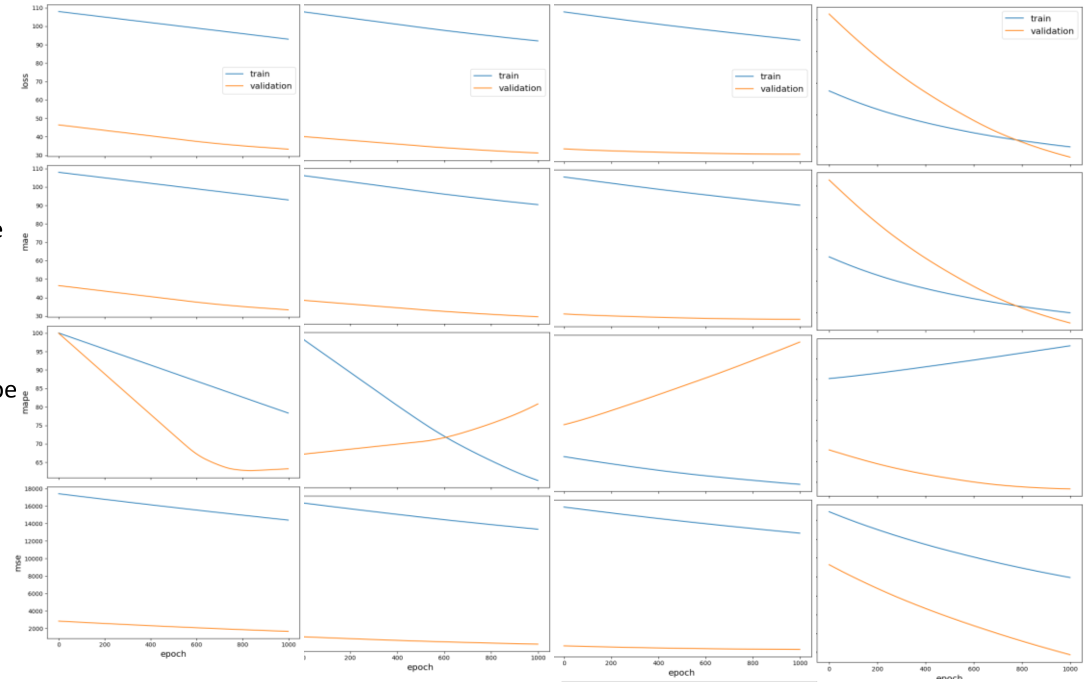
2nd Case: GCN + CNN + MLP (1000 Epochs)

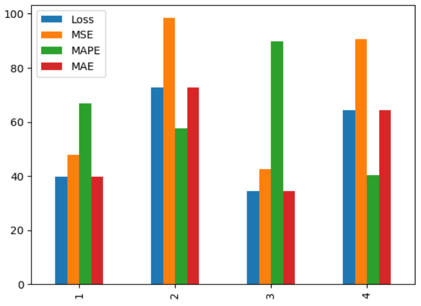
Loss

Mae

Mape

Mse





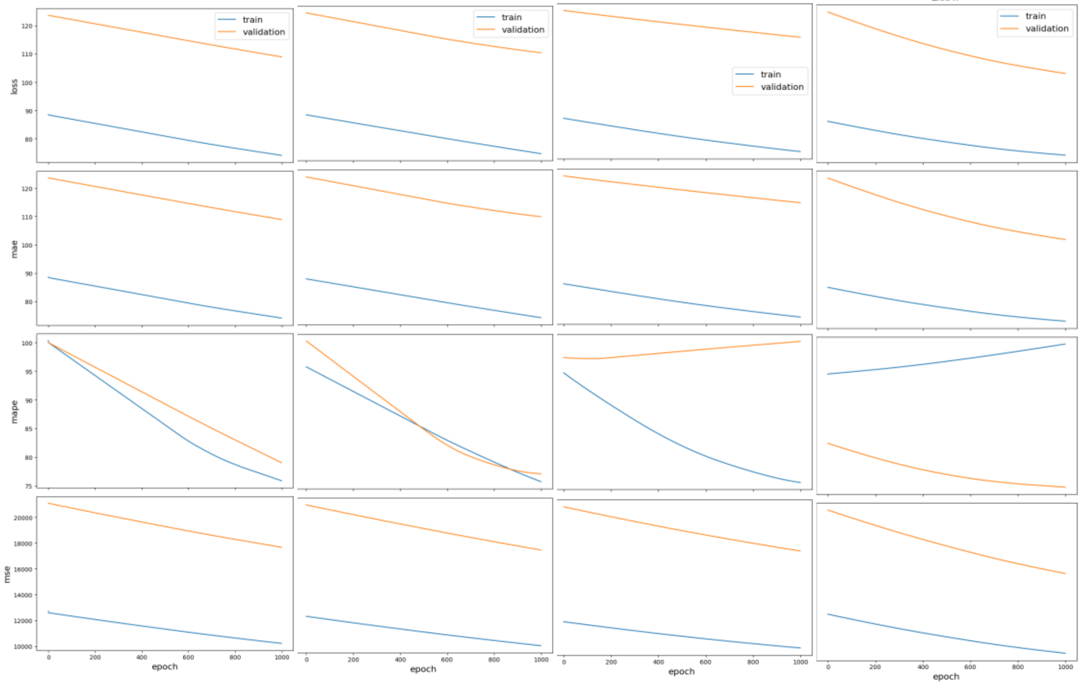
Loss	39.8	72.6	34.6	64.3
MSE	48.0	98.3	42.5	90.6
MAPE	66.7	57.6	89.7	40.4
MAE	39.8	72.6	34.6	64.3

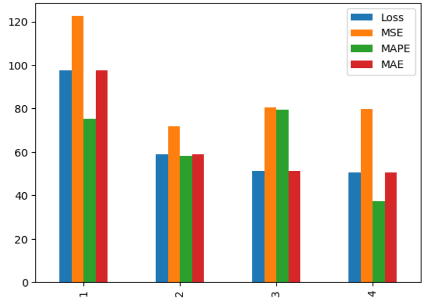
Loss

Mae

Mape

Mse





Loss	97.5	58.8	51.3	50.4
MSE	122.5	71.9	80.7	79.7
MAPE	75.5	58.2	79.4	37.2
MAE	97.5	58.8	51.3	50.4

Pearson **Spearman** **Kendall** **News**

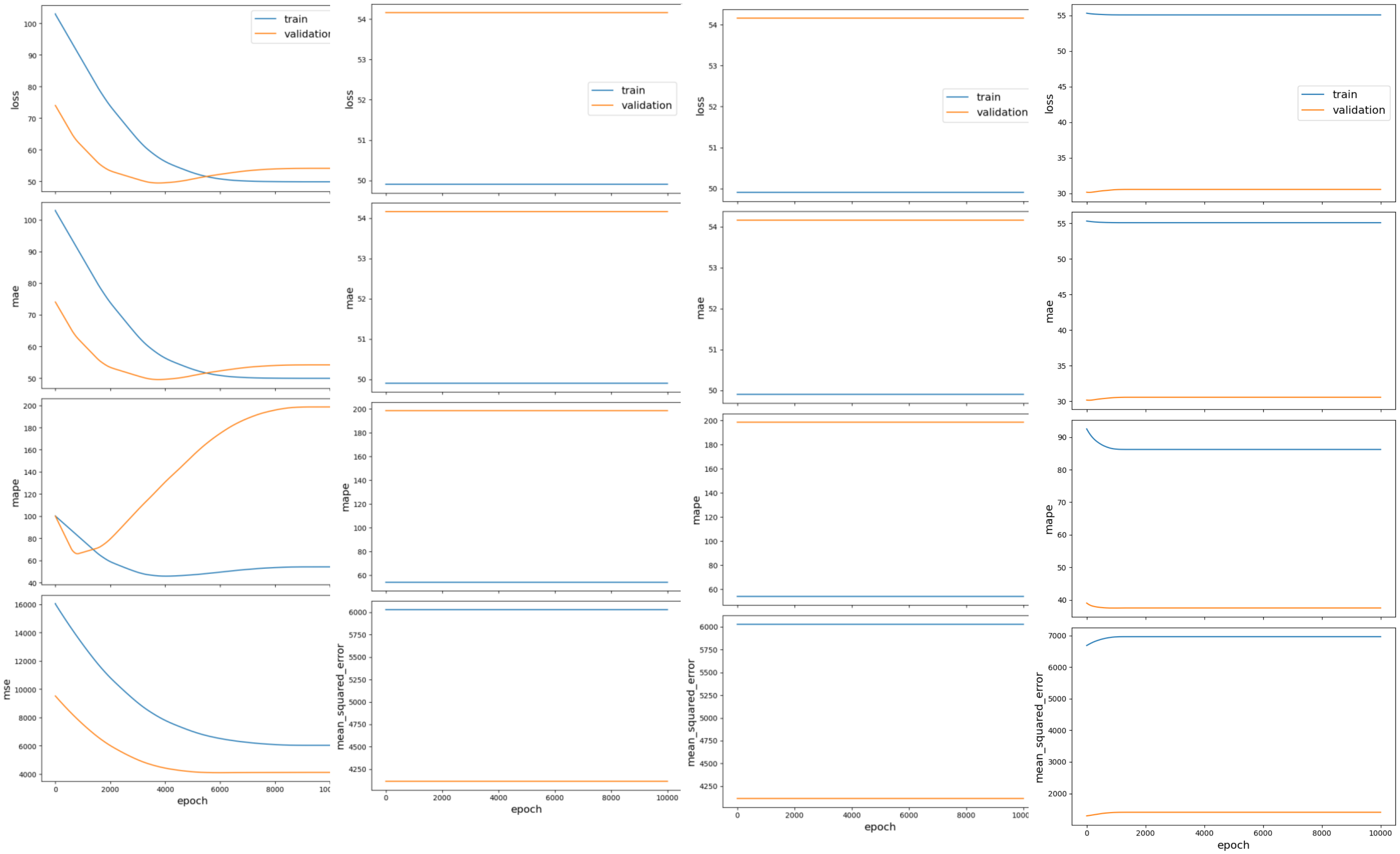
2nd Case: GCN + CNN + MLP (10,000 Epochs)

Loss

MAE

MAPE

MSE

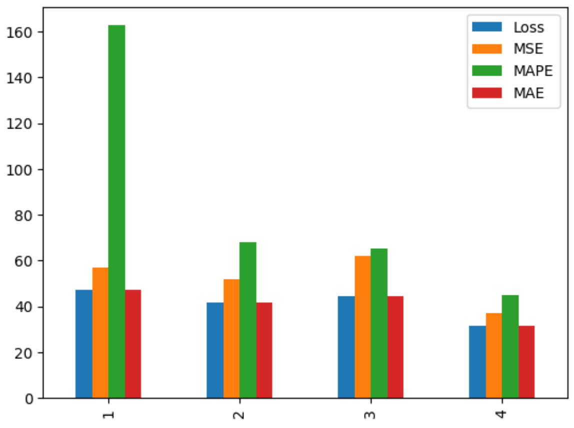


Pearson

Spearman

Kendall

News



Loss	47.4	41.6	44.6	31.5
MSE	57.2	51.9	62.1	37.2
MAPE	162.6	68.0	65.3	45.0
MAE	47.4	41.6	44.6	31.5

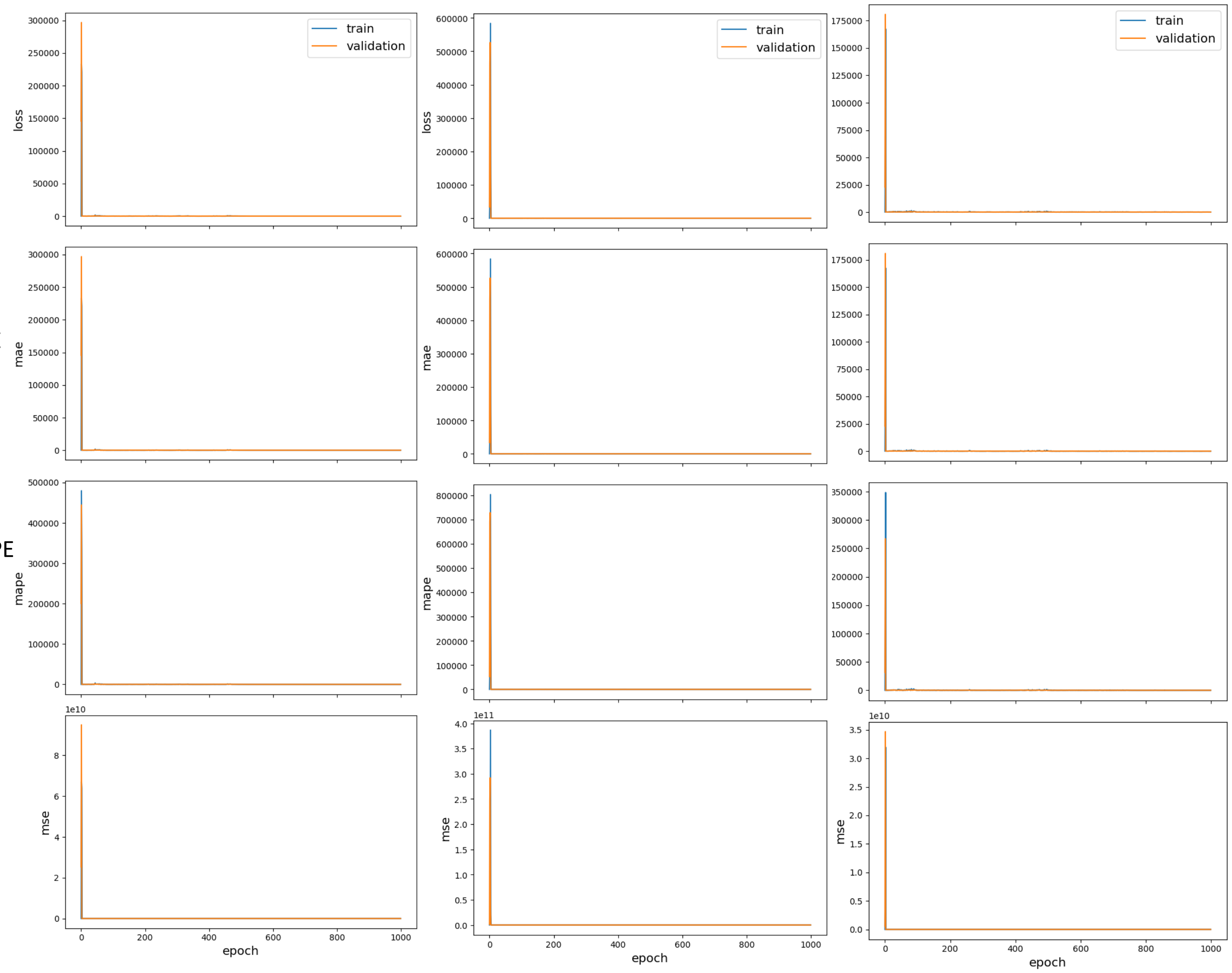
3rd Case: DeepGraphCNN (1000 Epochs)

Loss

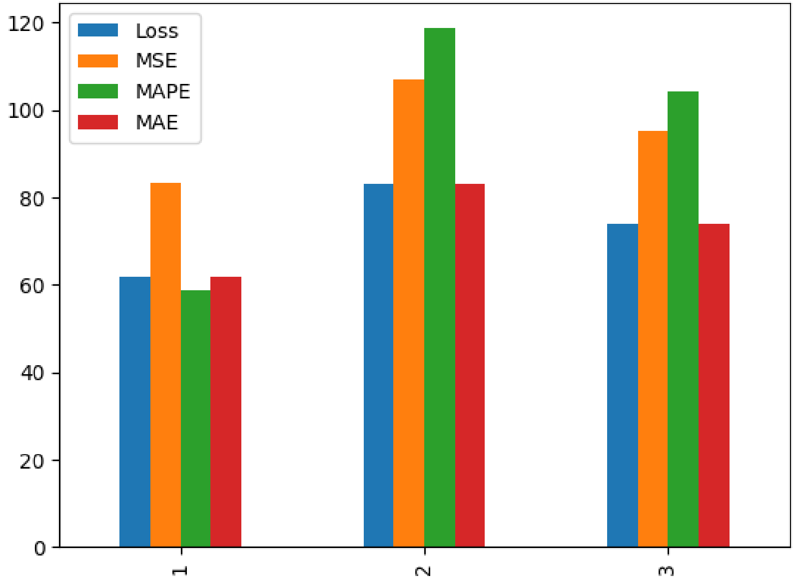
MAE

MAPE

MSE



Pearson Spearman Kendall



Loss	61.6	83.1	74.0
MSE	83.5	107.1	95.0
MAPE	58.7	118.6	104.3
MAE	61.6	83.1	74.0

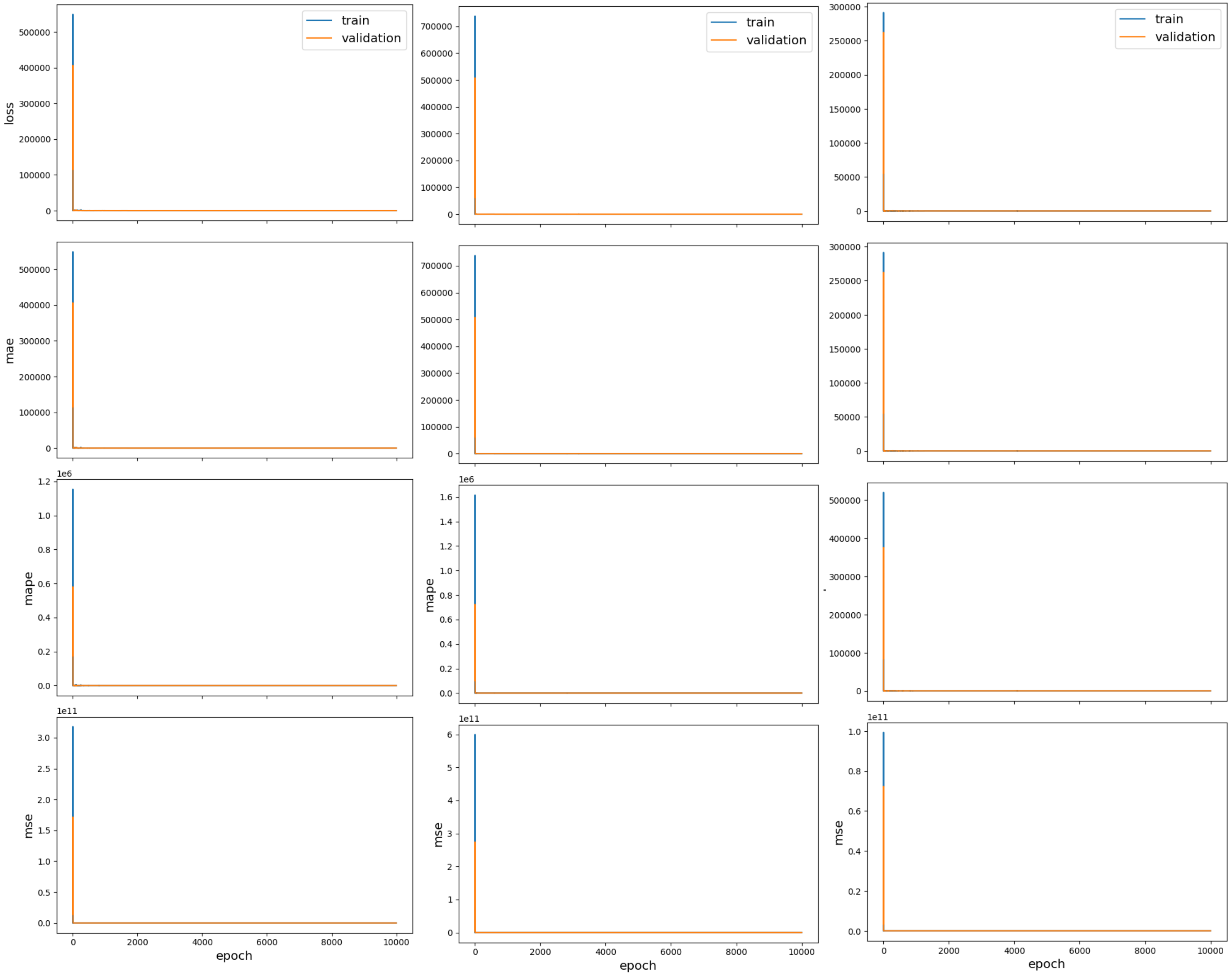
3rd Case: DeepGraphCNN (10, 000 Epochs)

Loss

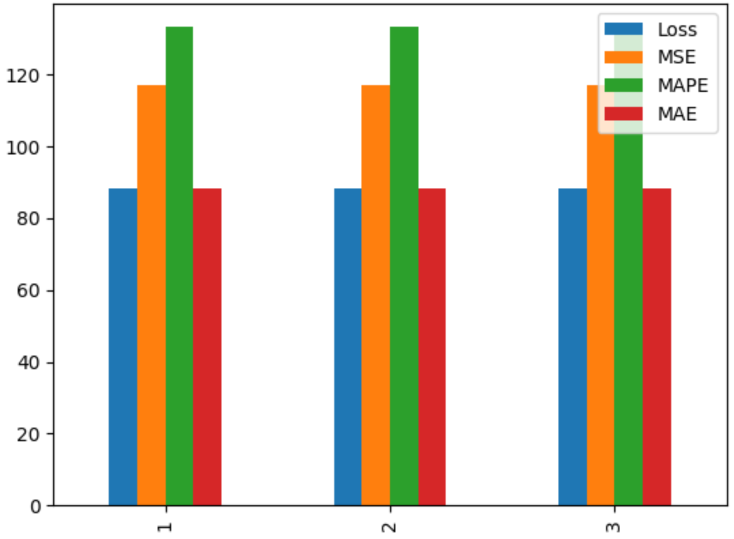
MAE

MAPE

MSE



Pearson Spearman Kendall

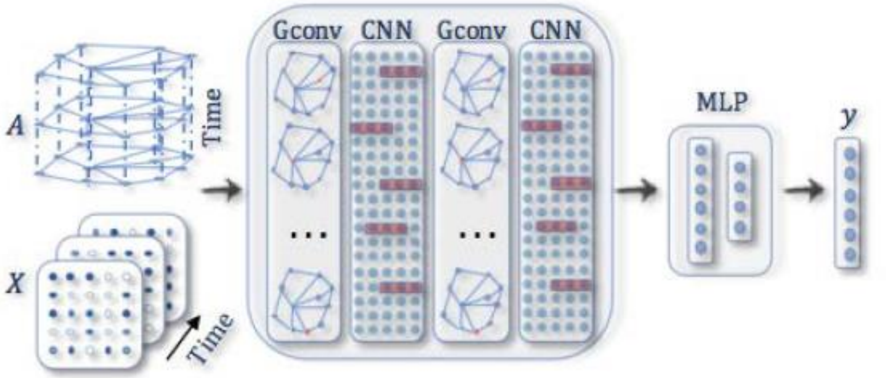


	1	2	3
Loss	88.3	88.4	88.3
MSE	117.1	117.1	117.1
MAPE	133.0	133.1	133.0
MAE	88.3	88.4	88.3

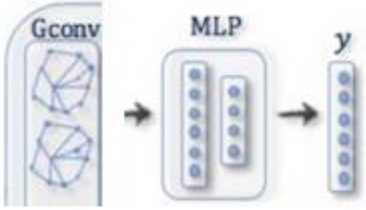
More Comparison: My Work with the Paper

Compare Architectures

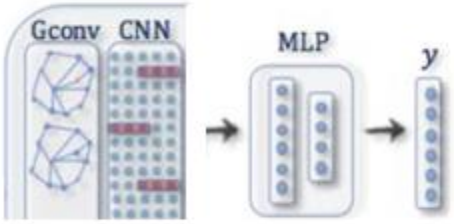
Paper



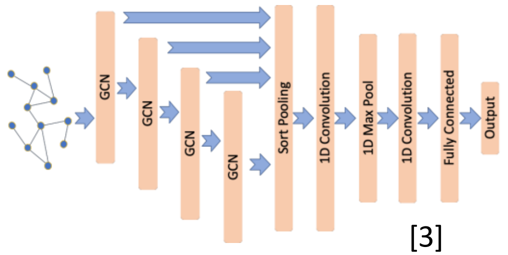
Mine



GCN + MLP

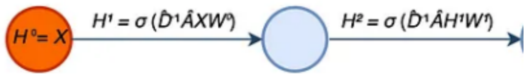


GCN + CNN + MLP



[3]

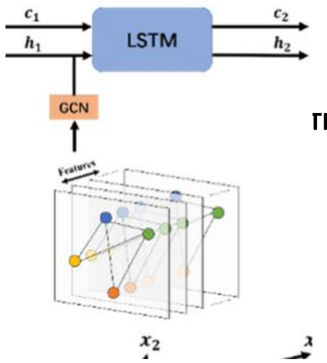
DeepGraphCNN



$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

$$\mathbf{x} *_G \mathbf{g}_\theta \approx \theta(\mathbf{I_n} + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{x}$$

Raw Equations



TM

[1] [3] [4]

Compare Performance Numbers (1000 Epochs)

Paper					Our Approaches											
-> Archi	(GCN + CNN)* + MLP				GCN + MLP				GCN + CNN + MLP				DeepGraph CNN			
	pea	spe	ken	news	pea	spe	ken	news	pea	spe	ken	news	pea	spe	ken	news
RMSE	12	12	10	9	54.1	63.7	**48.4**	**38.3**	122	71.9	80.7	79.7	83	107	95	
MAPE	15	5	4	4	161.1	84.6	149.6	**30.7**	75.5	58.2	**79.4**	37.2	58	118	104	
MAE	8	6	5	5	42.0	51.9	**37.4**	**28.2**	97.5	58.8	51.3	50.4	61	83	74	

Rank:
RMSE: News, Kendall, Pearson
MAPE: News, News, Spearman
MAE: News, Kendall, Pearson

Which Architecture is Better?
GCN + MLP

Compare Performance Numbers (10, 000 Epochs)

Paper					Our Approaches											
-> Archi	(GCN + CNN)* + MLP				GCN + MLP				GCN + CNN + MLP				DeepGraph CNN			
	pea	spe	ken	news	pea	spe	ken	news	pea	spe	ken	news	pea	spe	ken	news
RMSE	12	12	10	9	58.2	39.9	56.1	59.3	57.2	51.9	62.1	37.2	117.1	117.1	117.1	
MAPE	15	5	4	4	157.0	31.4	81.0	57.5	162	68.0	65.3	45.0	133	133.1	133.0	
MAE	8	6	5	5	49.2	28.3	43.1	43.0	47.4	41.6	44.6	31.5	88.3	88.4	88.3	

Rank:
RMSE: News, Spearman, Spearman
MAPE: Spearman, News, News
MAE: Spearman, News, Spearman

Which Architecture is Better?
GCN + MLP although GCN + CNN + MLP – Very close/equivalent

Overall: 1000 Epochs are as good as 10,000 epochs

Compare: Ranking or Performance

- **Paper**
  - Causation: Financial News
  - Kendall tau correlation (nonlinear)
  - Spearman (nonlinear)
  - Pearson (linear)

Ours

Was hard to conclude.

- Causation
- Spearman
- Pearson/Kendall

Challenges, Limitations and Future Work

Challenges

- Exact Dataset
  - Finding or Generating
- Find and implement Exact Architecture
  - Not exact specific details
- Exact Platform to execute
- Adapting data for Deep Network
  - Esp. for different time steps

Limitations of my work

- Implement exact architecture
- Execute and tune
  - to get the same or better performance
- News dataset

- **Future Work**
  - Address the challenges and limitations mentioned
  - Also, resolve Exploding Gradient
    - To reduce training complexity

References

[1] Pratik Patil, Ching-She, Katerina Potika, Marjan Orang [Stock Market Prediction Using Ensemble of Graph Theory, Machine Learning and Deep Learning Models]

[2] Investopedia: Efficient Market Hypothesis (EMH): Definition and Critique. https://investopedia.com/terms/e/efficientmarkethypothesis.asp

[3] **Supervised graph classification with Deep Graph CNN** https://stellargraph.readthedocs.io/en/latest/demos/graph-classification/dgcnn-graph-classification.html

[4] **Graph Convolutional Networks (GCN) & Pooling** https://jonathan-hui.medium.com/graph-convolutional-networks-gcn-pooling-839184205692

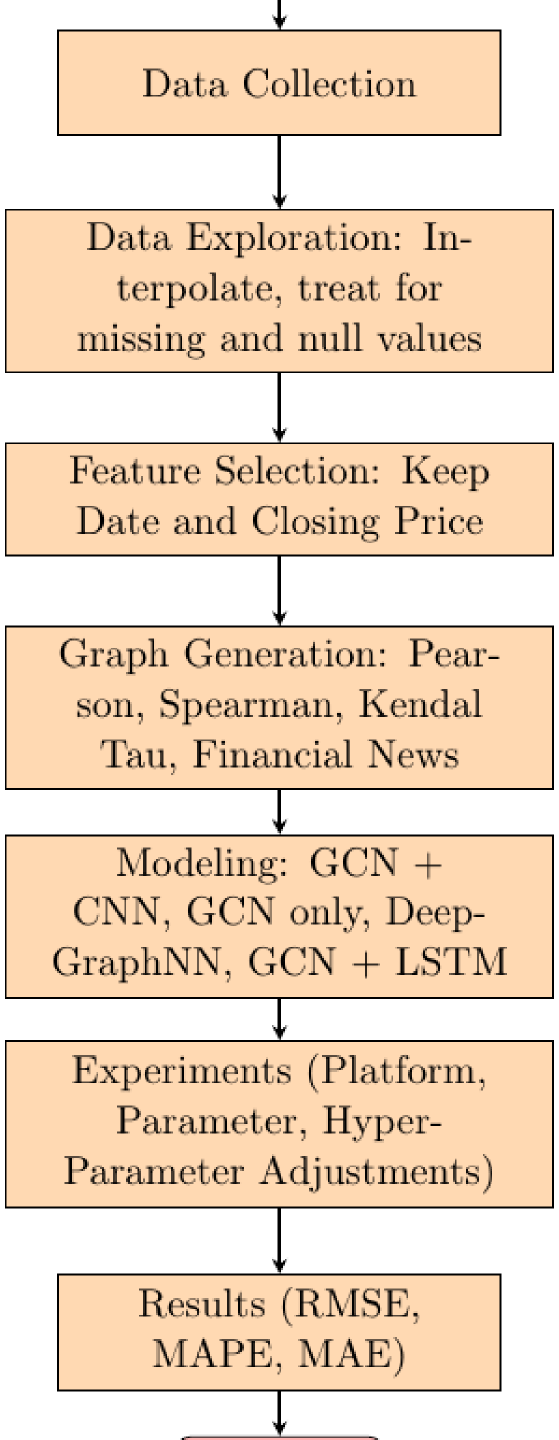
I Hope that you Enjoyed the Presentation

If you have any **Questions**?

I will be happy to **answer**

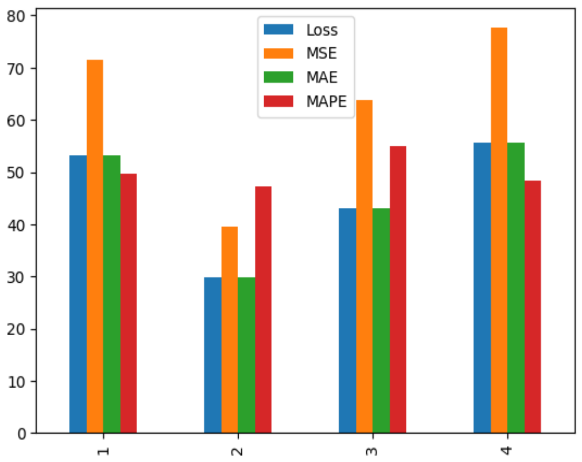
Methodology

- Data Collection
- Data Exploration
  - find date range where data exist for all stocks under consideration
    - Remove out of date data
  - Still looked for NAN/NA
    - Replaced those with surrounding values (interpolate) (8 or 9)
    - Even NAN, dropped them (1 row)
    - 4301, 4292
  - Verified all stocks has equal number of values
  - Any Plotting/Unitary/Bi-Variate analysis
    - Scaling/normalization – does not seem that I need (will see)
- Feature selection/engineering
  - Kept: Date, Ticker, Adjusted Close
    - For accuracy try the other close
- Graph Generation
- Model (Deep Learning Models)
  - GCN + CNN based stock price prediction
  - GCN only
  - DeepgraphNN
  - GCN + LSTM
- Experiments
- Results

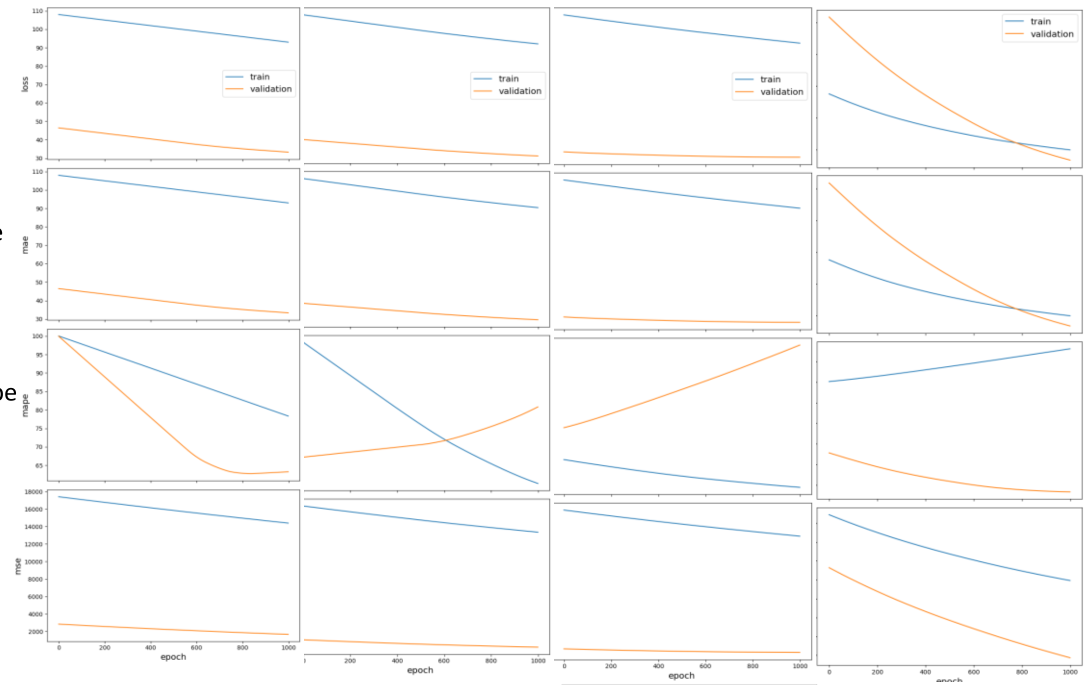


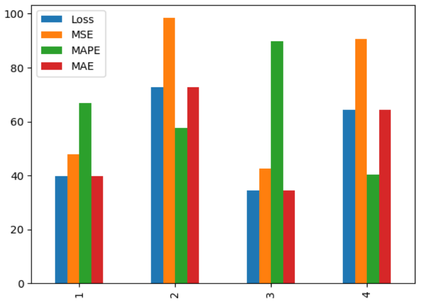
2nd Case: GCN + CNN + MLP (1000 Epochs)

15k Epochs

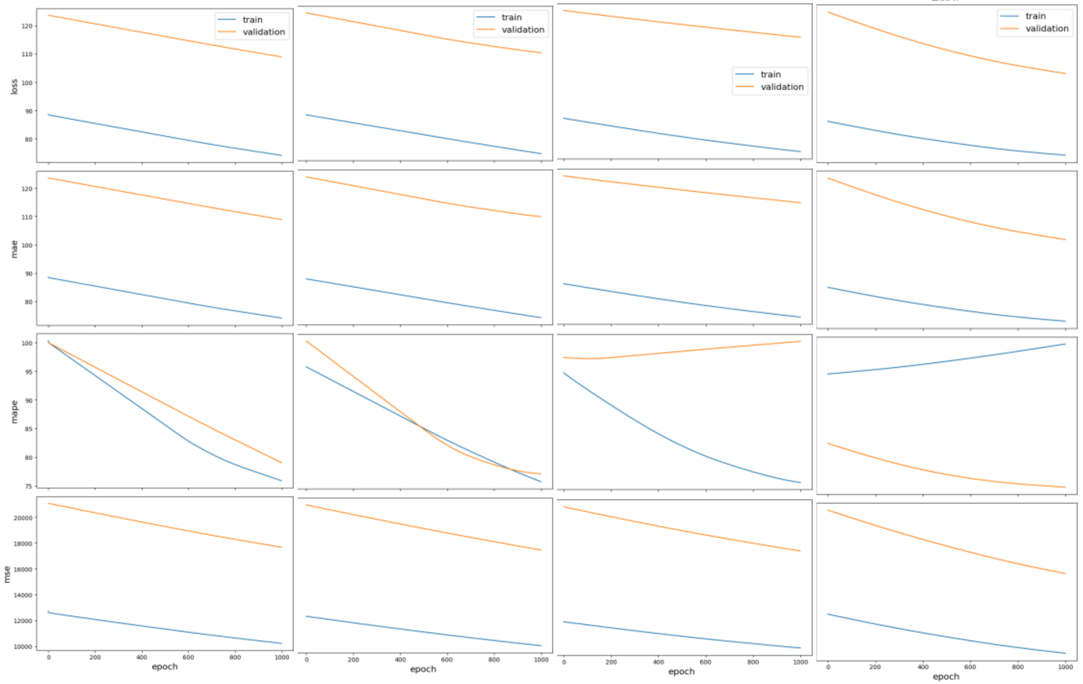


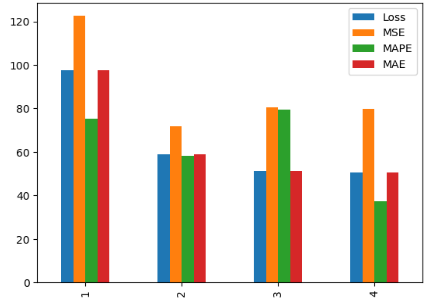
	Loss	MSE	MAE	MAPE
1	53.140034	71.558536	53.140034	49.651329
2	29.864273	39.612246	29.864273	47.168633
3	43.128960	63.828448	43.128960	54.905254
4	55.650585	77.589062	55.650585	48.469971





Loss	39.8	72.6	34.6	64.3
MSE	48.0	98.3	42.5	90.6
MAPE	66.7	57.6	89.7	40.4
MAE	39.8	72.6	34.6	64.3





Loss	97.5	58.8	51.3	50.4
MSE	122.5	71.9	80.7	79.7
MAPE	75.5	58.2	79.4	37.2
MAE	97.5	58.8	51.3	50.4

Pearson **Spearman** **Kendall** **News**