Project Lab

Forecasting of Sugar Production Yield

TEAM

20181128 Byeongyeon So

20191349 Jiwon Kim

20201181 Jihwan Oh

Contents

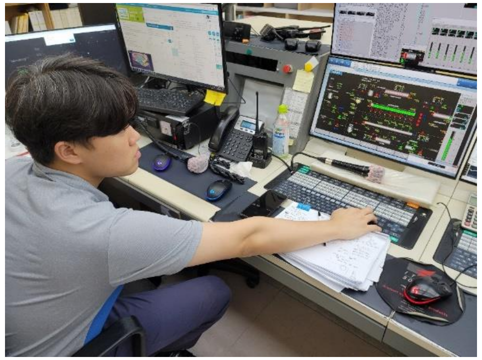
1

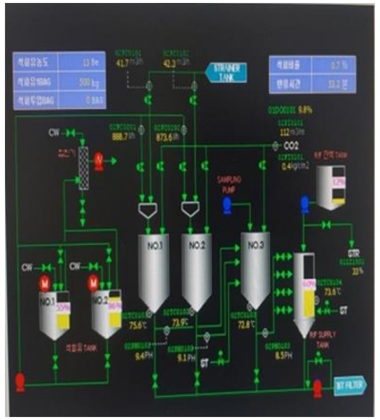
Introduction

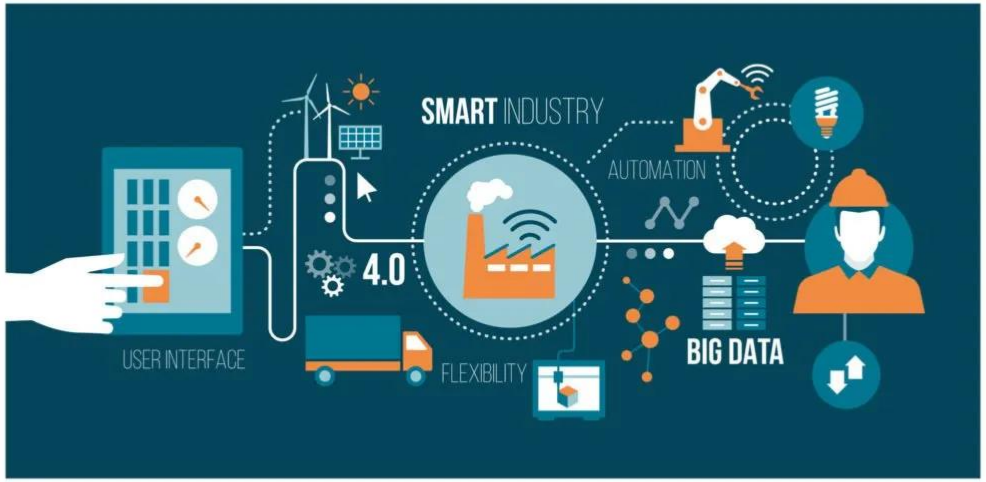
01 Motivation

02 Project Objective

Importance of Operating Sugar Production Stably





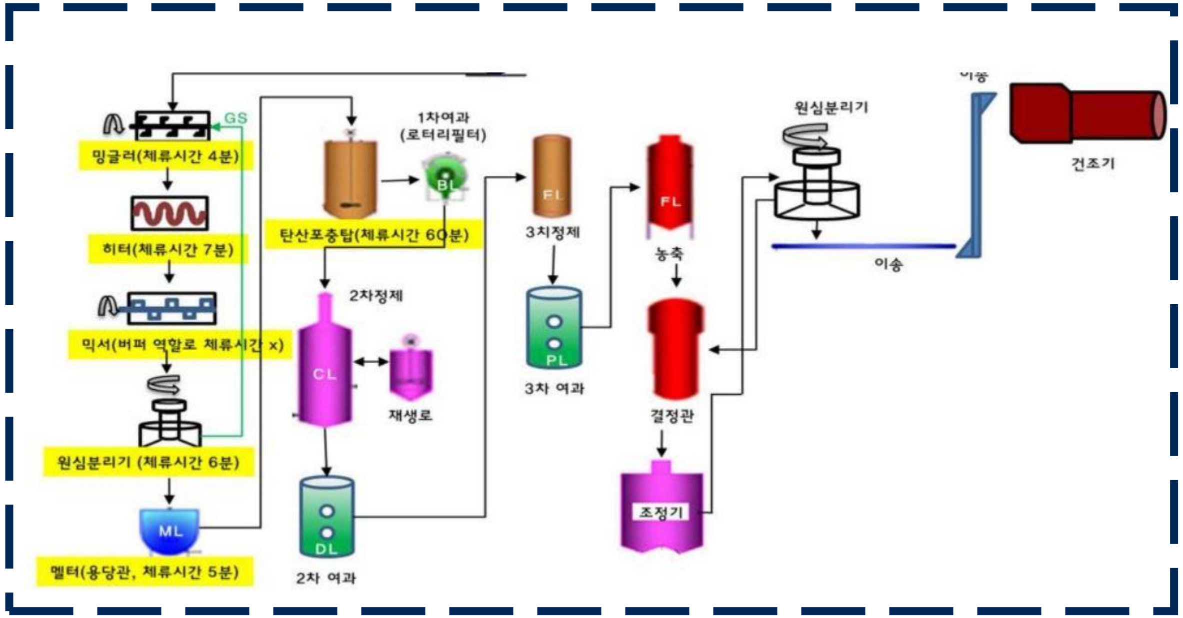


" Manufacturing process control assistance service "

Multi-Stage Time-series Forecasting

Feed (Sugar Cane)

Sugar Manufacturing Process

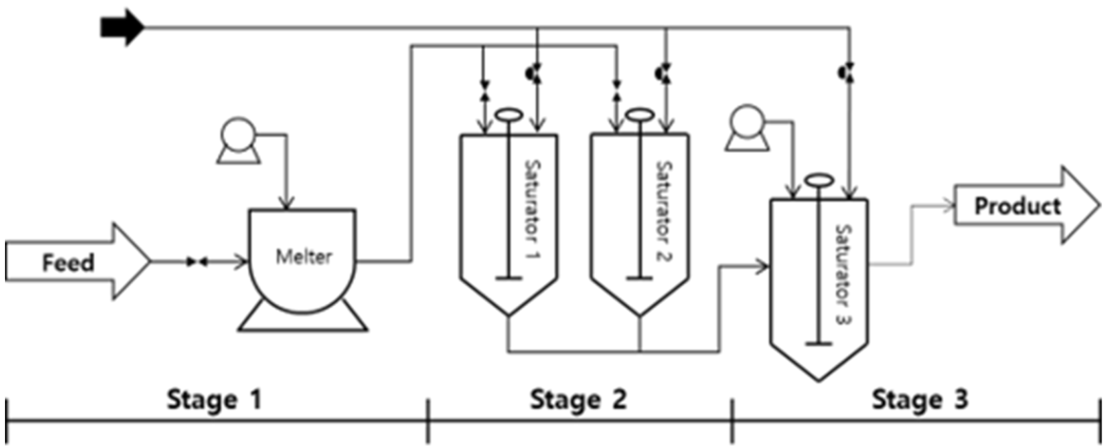


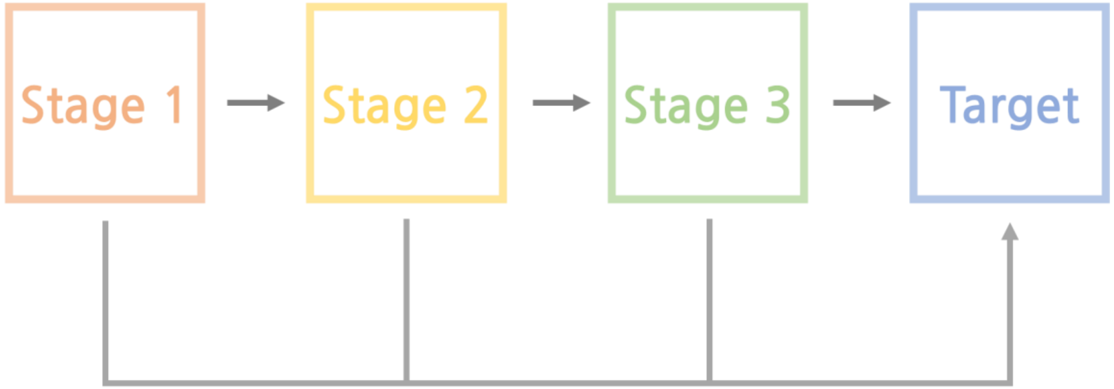
Product

Multi-Stage Time-series Forecasting

- We need to consider Multi-Stage process.
- RULE: **Previous stage can affect to the next stage, but the reverse cannot happen.**
  - In this order, variables in Stage 1 can affect Stage 2, Stage 2 can affect Stage 3, and finally Stage 3 can affect Target.

$$\underline{\boldsymbol{Final\ Product = stage_3(stage_2(stage_1(x_1), x_2), x_3)}}$$





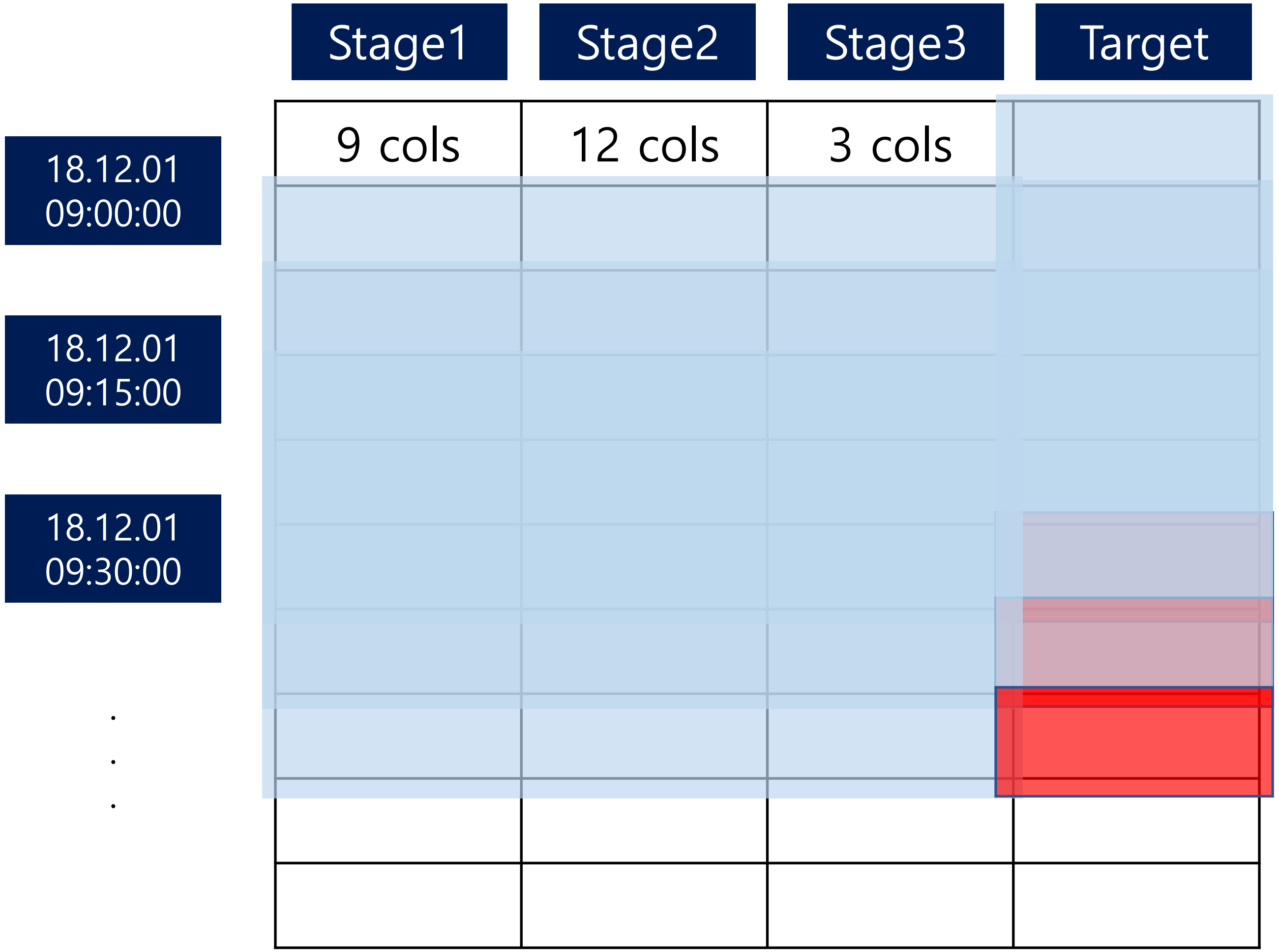
It is important to create a new model that fits these characteristics.

Data for Multi-stage Time-series Forecasting

1. Feed Data
- Shape: (1278, 7)
- Experimental value
- Measured once a day

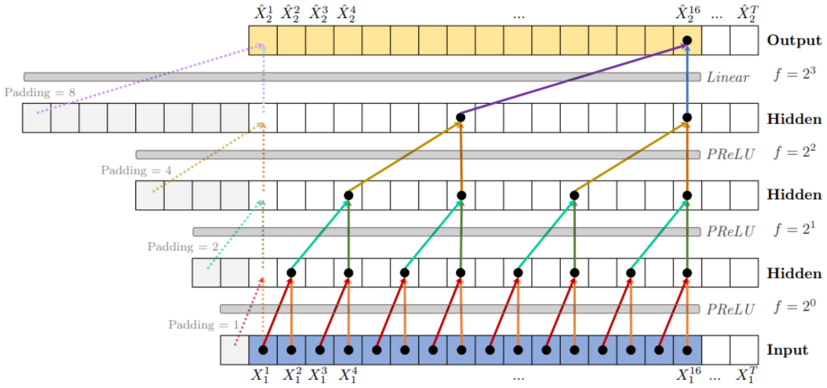
2. Process Operation Data
- Shape: (122640, 17)
- Sensor data, such as process operational status and control value
- Collected every 15 minutes

3. Equipment Data
- Shape: (1278, 3)
- Condition of certain equipment in the process related to the yield of sugar production
- Measured once a day



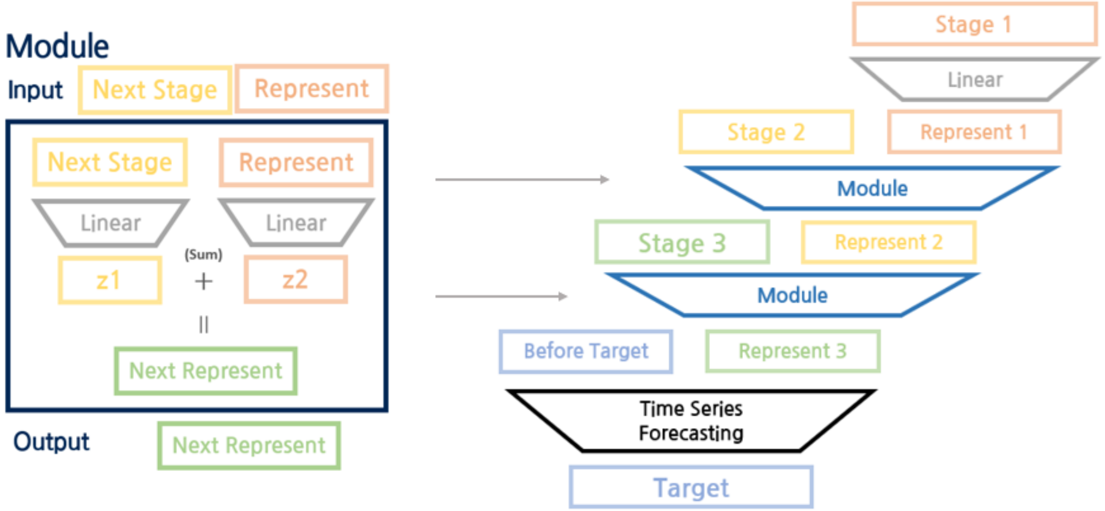
Overview

- **Application of existing models w/o considering Multi-stage process** (Chapter2)
  - Time-Series prediction and forecasting models
  - From Simple to Latest Advanced Models
    - LinearRegression, RandomForest
    - LSTM, AD-DSTCNs, LSTF-Linear, Informer



AD-DSTCNs

- **Development of our own models** (Chapter3)
  - Consider Multi-stage Process
    - Linear
    - Multi-stage AutoEncoder
    - Multi-stage Linear



[Final Model] Multi-stage Linear

Application of Existing Models

01 Machine Learning Models

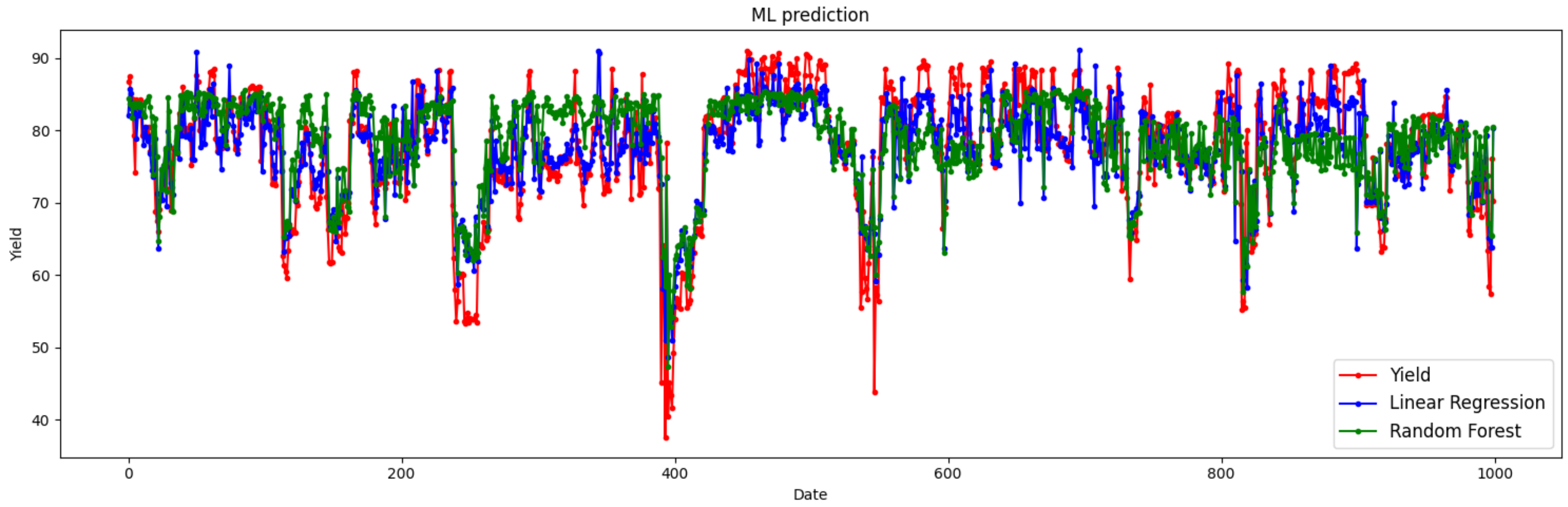
02 Deep Learning Models

2

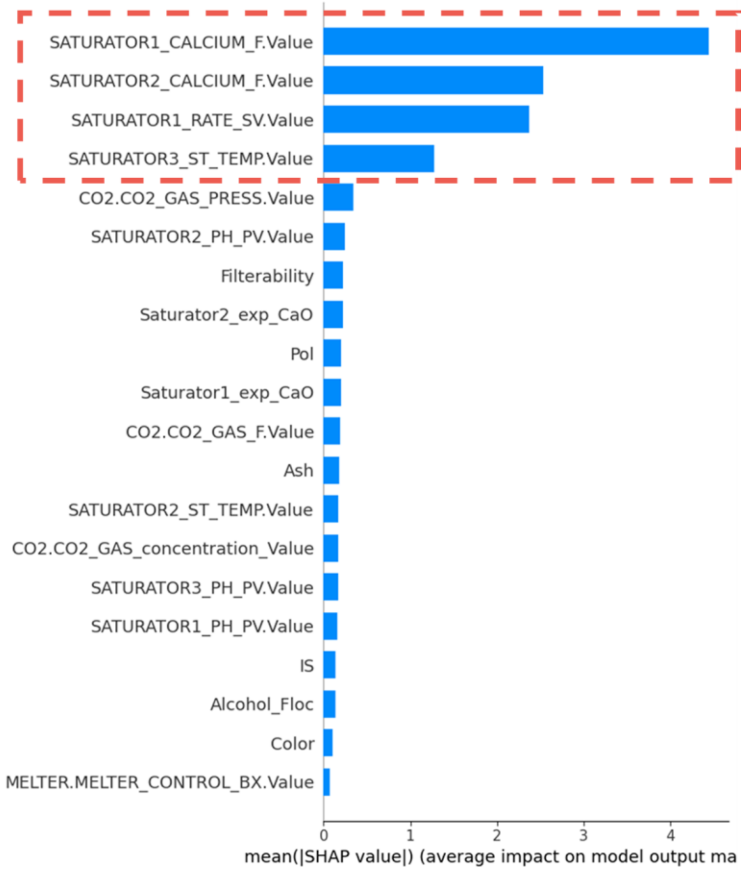
Machine Learning

- First, we applied classic **Linear Regression** and **Random Forest**.
  - They provide us the model interpretability.

Model	MSE
Linear Regression	23.689
Random Forest	35.9489



SHAP Value

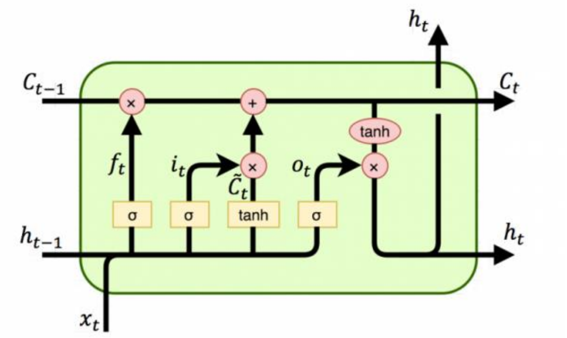


Deep Learning

- Then, we applied on different types of deep learning time series forecasting model.
  - RNN-based, CNN-based, Transformer-based models with whole features.

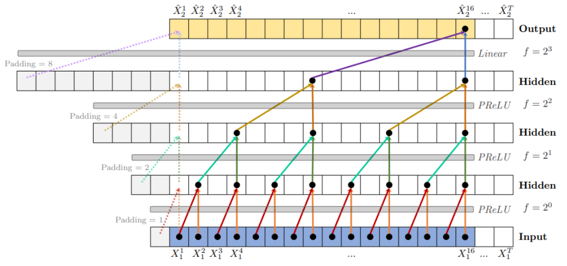
LSTM

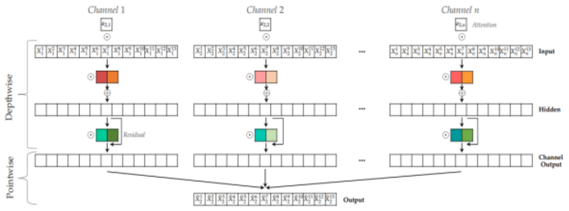
RNN based Model



AD-DSTCNs

TCN based Forecasting





LSTF-Linear

Linear Transform Model

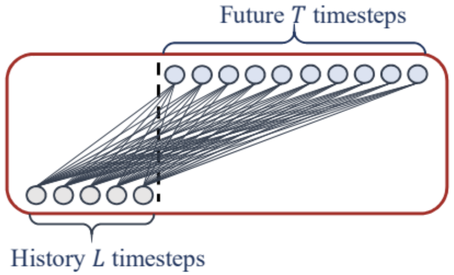
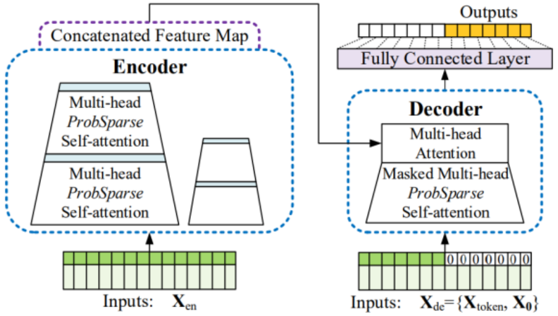


Figure 2. Illustration of the basic linear model.

Informer

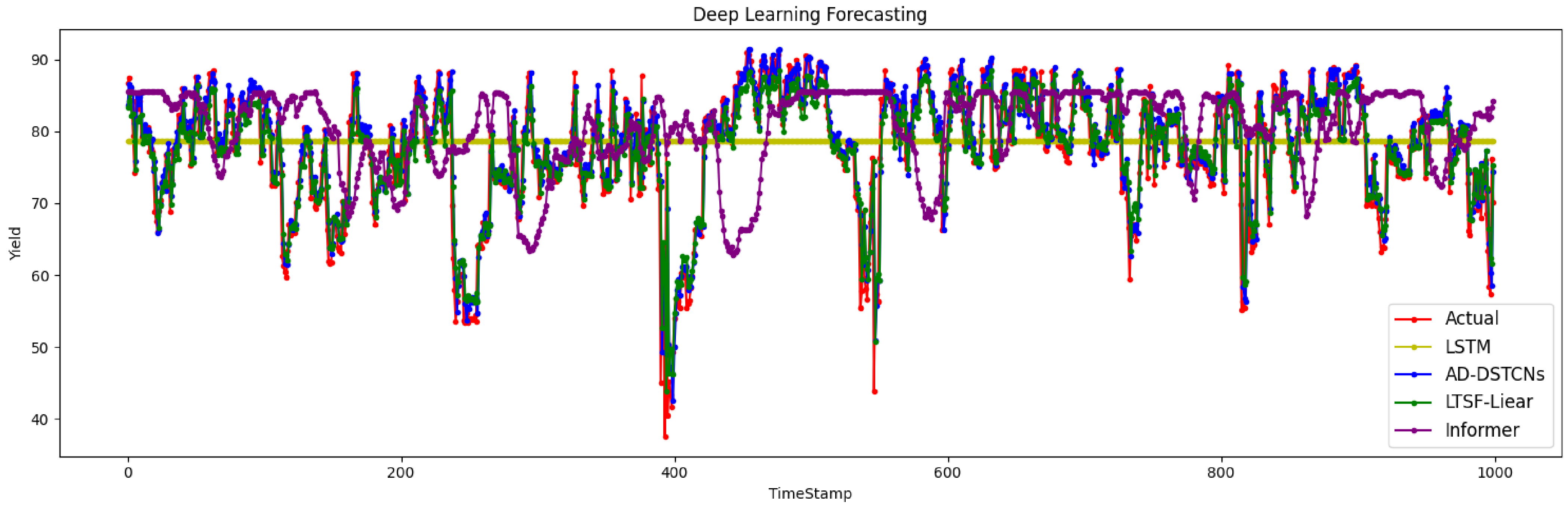
** added after Interim

Transformer based Model



Deep Learning

Model	MSE
LSTM	66.553
AD-DSTCNs	**12.911**
LSTF-Linear	14.665
Informer	39.433



3

Development of Our Own Models

01 Linear

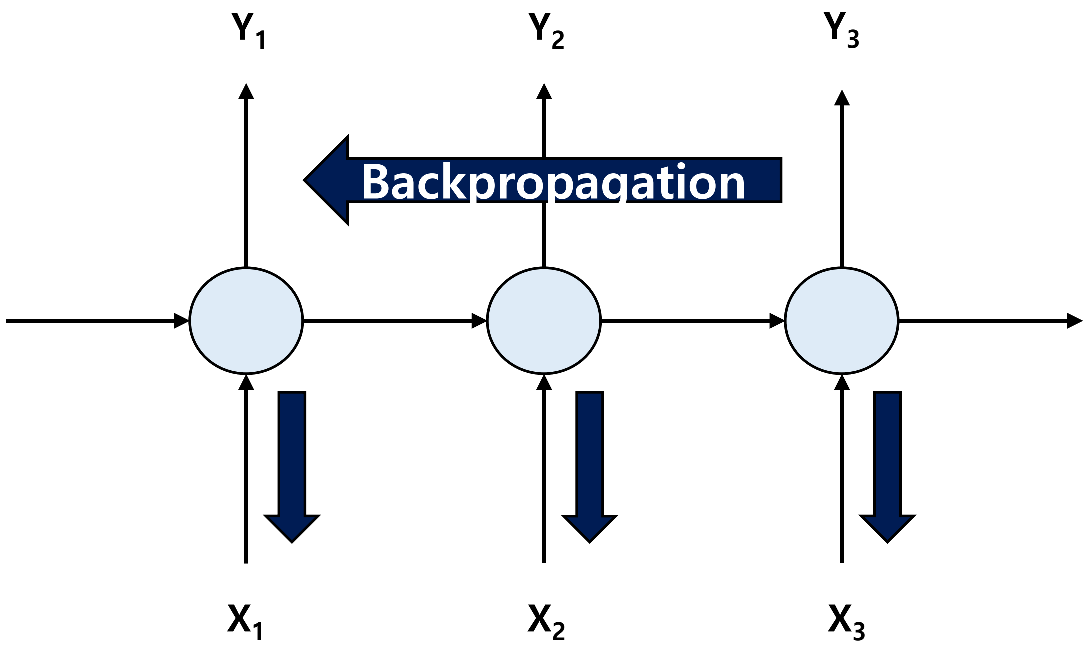
02 Multi-stage AutoEncoder

03 Multi-stage Linear

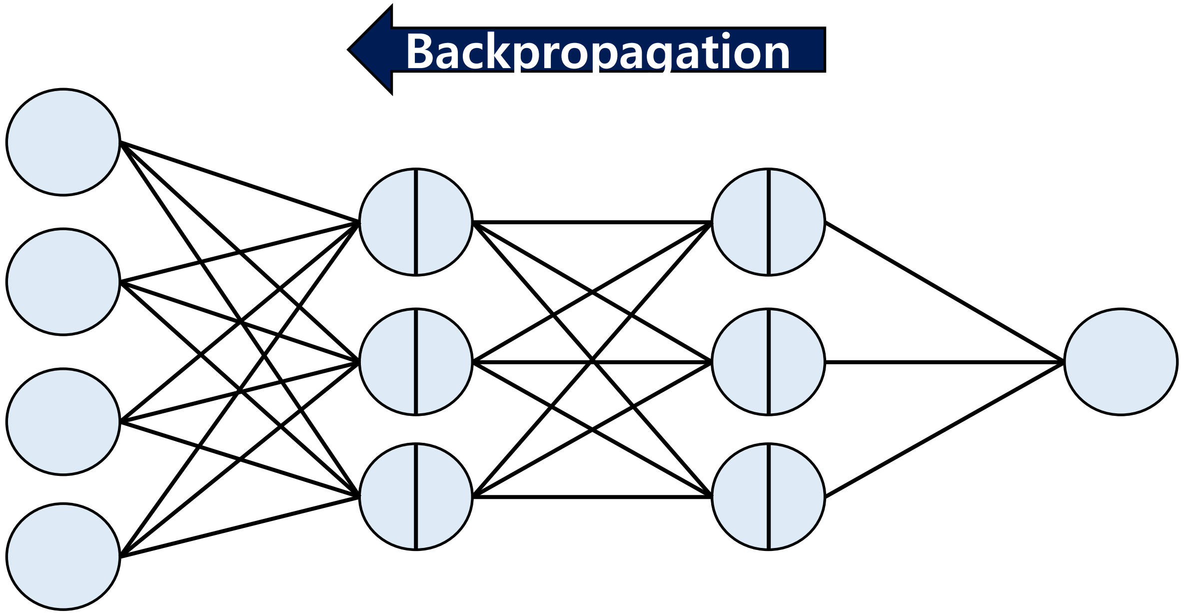
Multi-stage

Why cannot simple RNNs and MLPs reflect Multi-stage process?

- In the process of model training, we update the parameters by the loss through backpropagation.
- At this time, the variables in the next stages affect the variables in the previous stages.



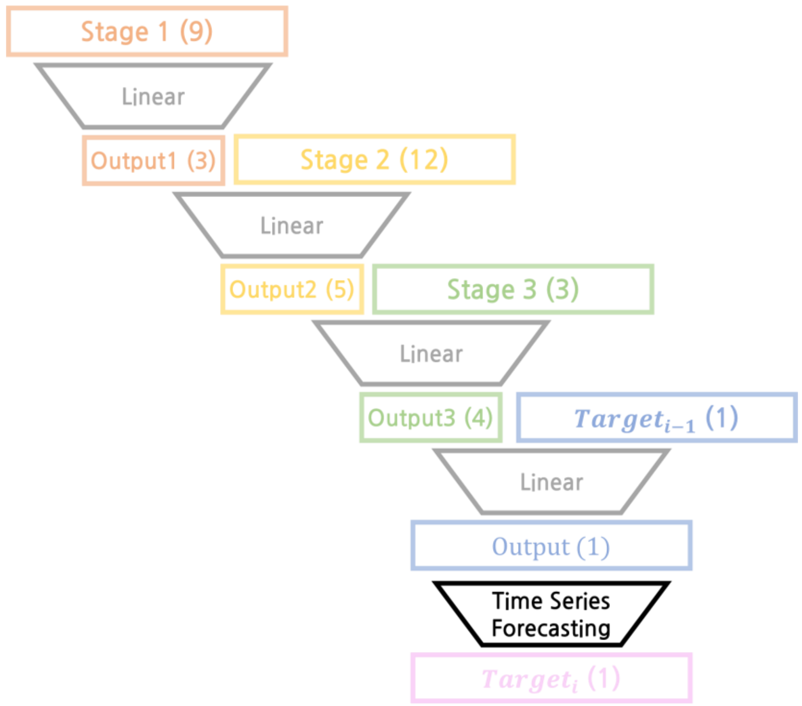
Backpropagation in RNN



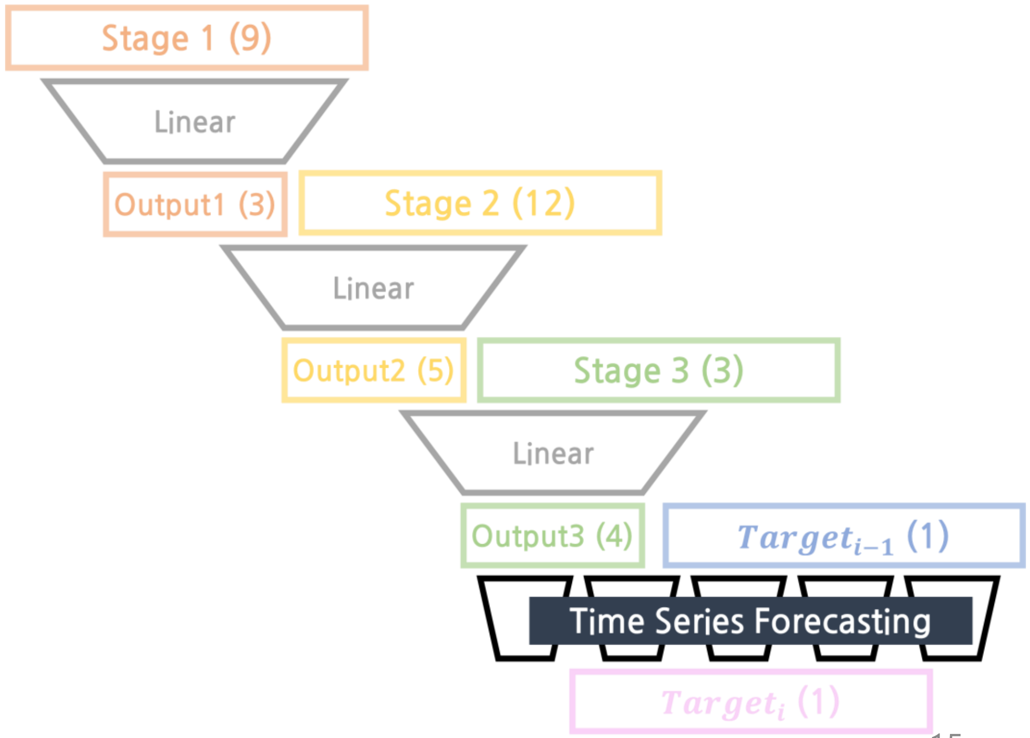
Backpropagation in MLPs

Linear

- Method1
  - Dimensionally reduce each Stage variable and the previous Stage's Representation variable with a linear model, then create the following representation.
  - Finally, use Linear model to predict the target

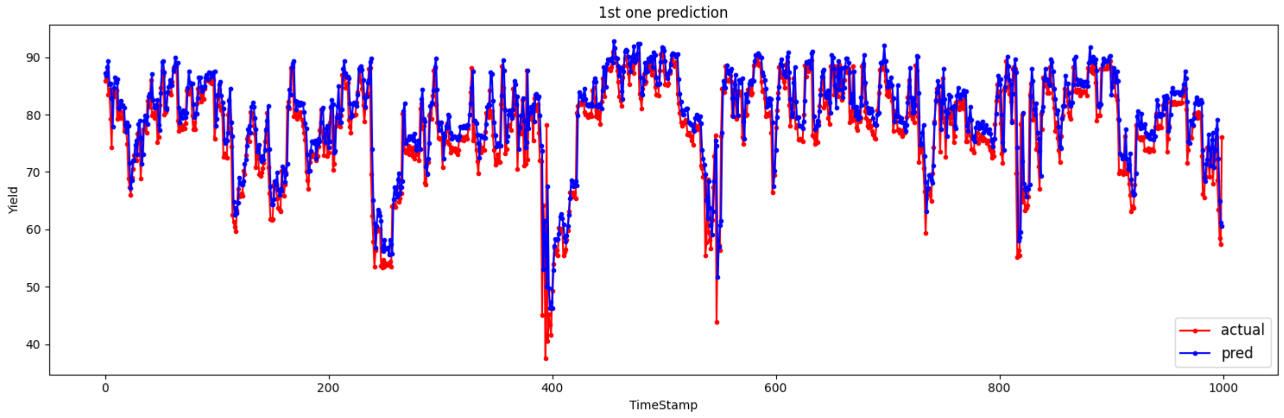


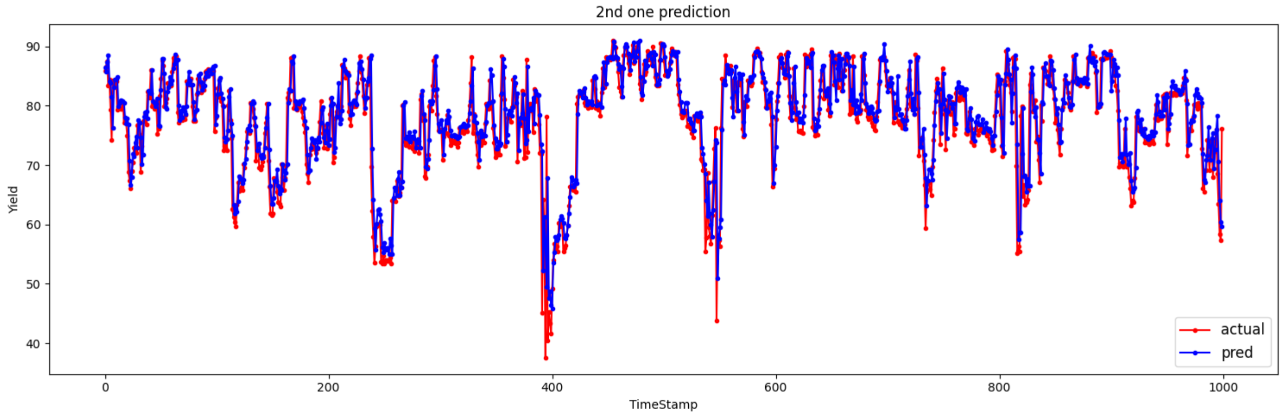
- Method2
  - The method is the same as Method 1, but when predicting the target, Linear is applied to each feature to embed the time information, and then predict the final target.



Linear

Model	MSE
Linear1	17.3874
Linear2	12.3519





- Linear 2 seems to make better predictions than Linear 1 because the model is more complex.
- So, we chose Linear 2 for main model and tried another experiment.

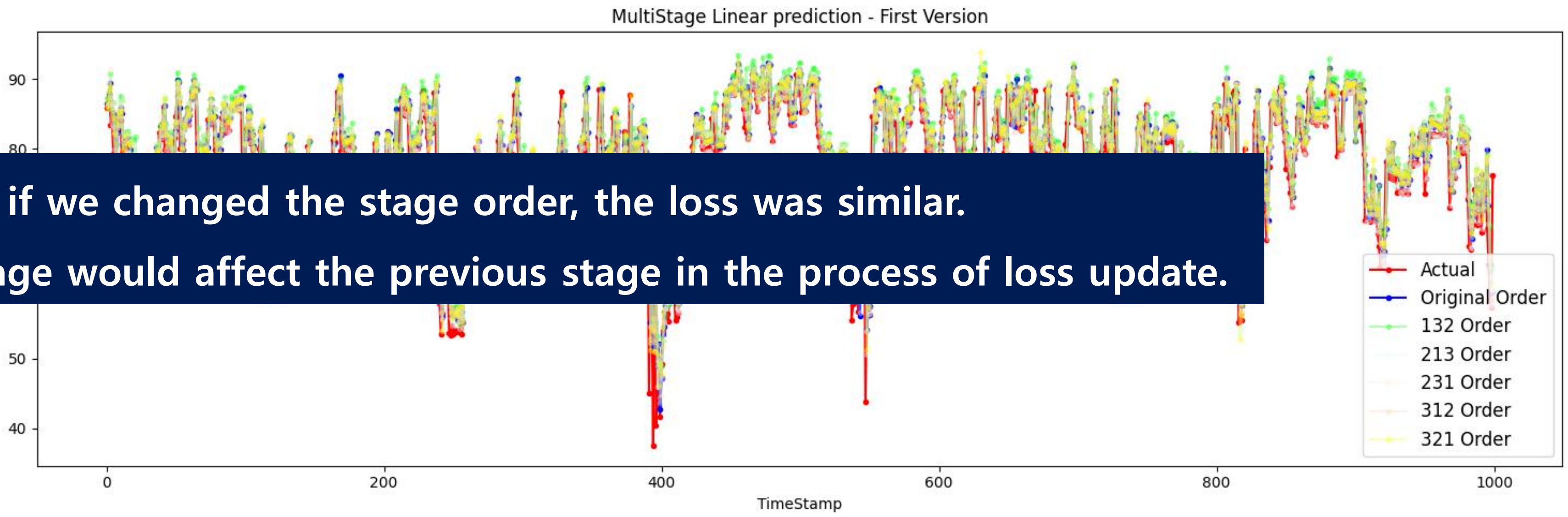
Linear

Order Experiment

- To see if our model actually reflects the process well, we experimented with shuffling the order of the processes
- **Expectation: If the model reflects the ordering well, it will perform best when the data is put in its original order.**
  - Run model 3 times with different random_seed for each order data and record the average score

Result

Model	MSE
1 → 2 → 3	**13.5092**
1 → 3 → 2	
2 → 1 → 3	
2 → 3 → 1	16.3090
3 → 1 → 2	**13.3864**
3 → 2 → 1	**13.1483**



Even if we changed the stage order, the loss was similar.

Maybe the next stage would affect the previous stage in the process of loss update.

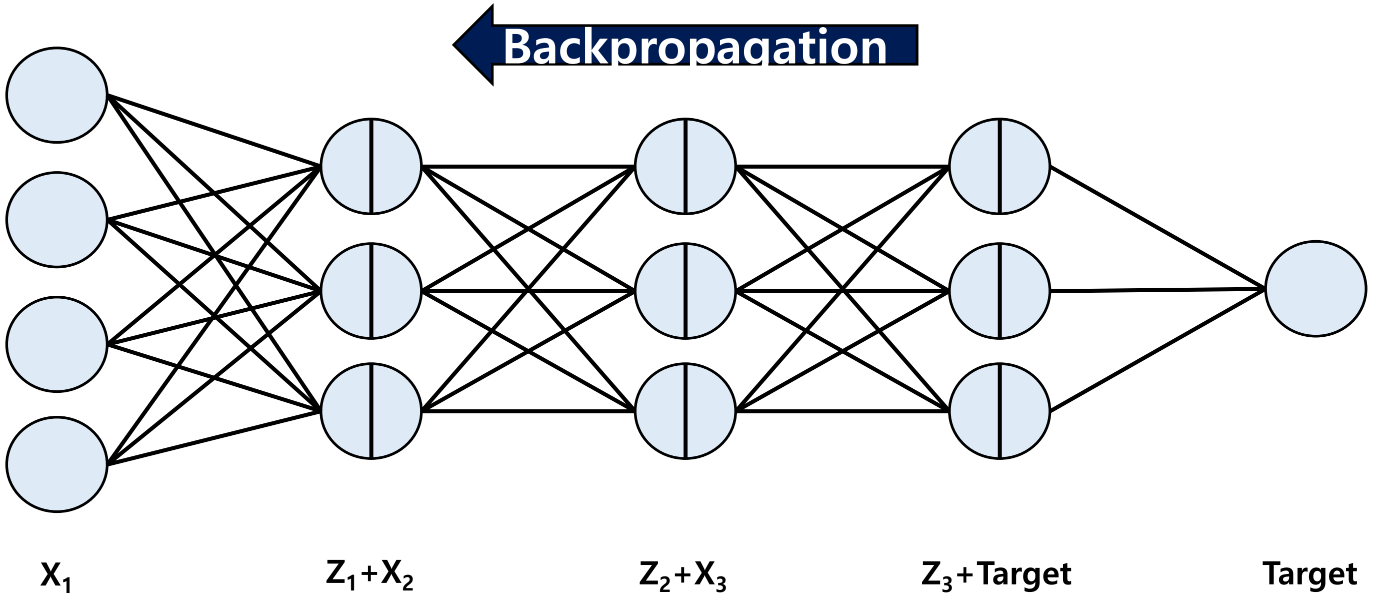
Backpropagation

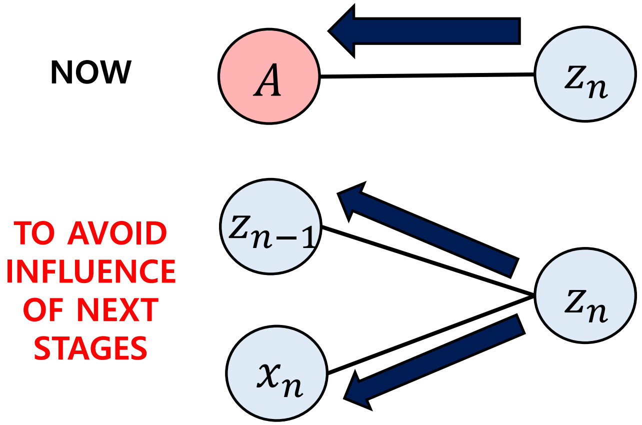
"Is our model truly built so that the future stage does not affect the previous stage?"

- In the process of model training, we update the parameters by the loss through backpropagation.
- However, we found that our linear model violates our RULE.
  - **RULE: Previous stage can affect to the next stage, but the reverse cannot happen.**

$$z_1 = W_1^T x_1,$$

$$z_n = w_n^{injected} \times \underbrace{concat([z_{n-1}, x_n], axis = 1)}_{= \text{ one node: } A}$$



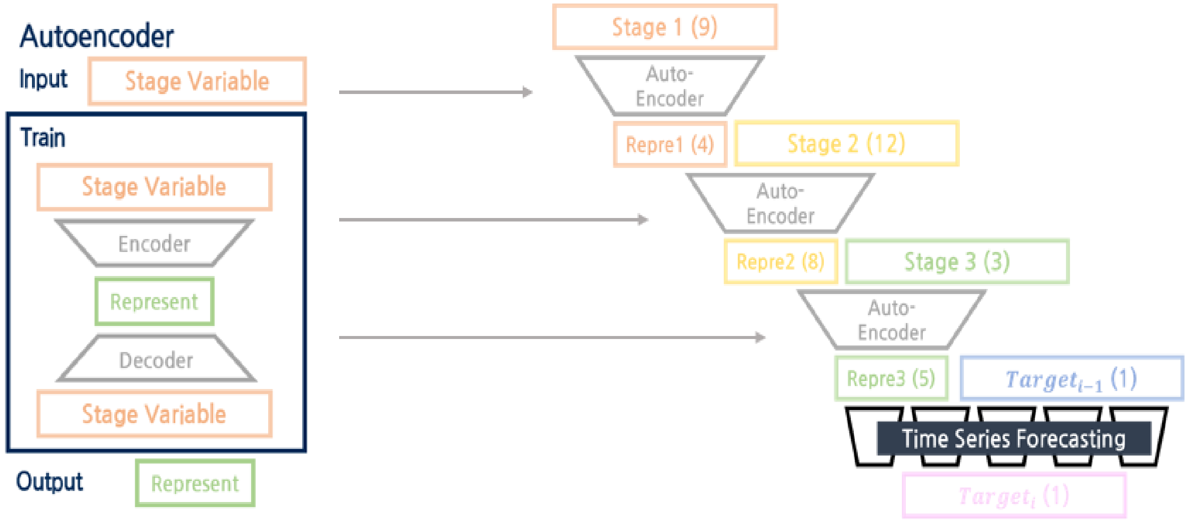


Next Approaches

To further develop the previous Linear model, we also thought of two other versions.

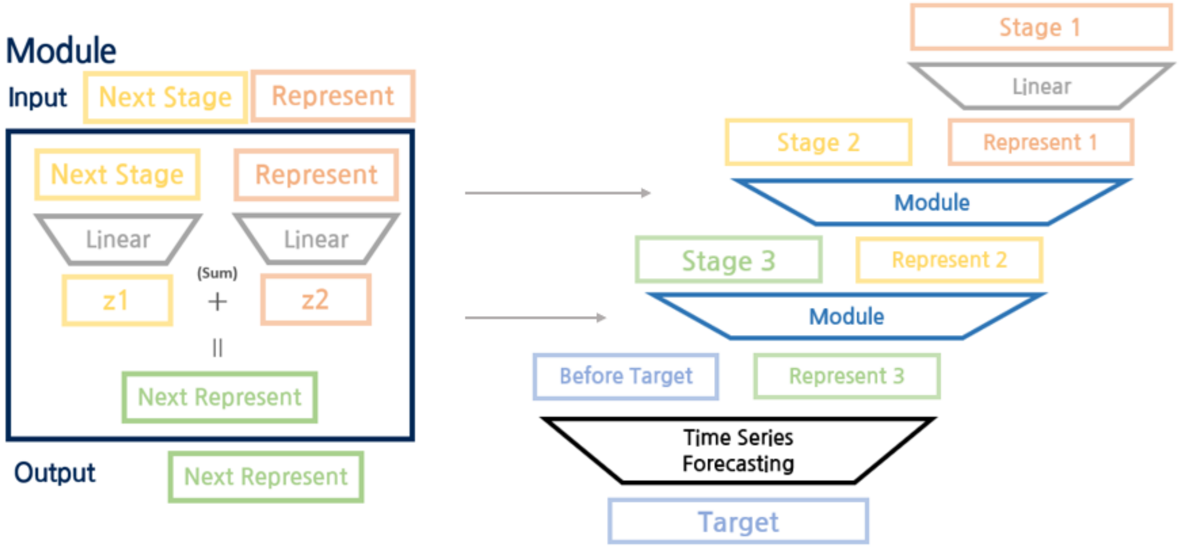
Multi-stage AutoEncoder

- Create a representation variable that does not depend on future dimensions



Multi-stage Linear

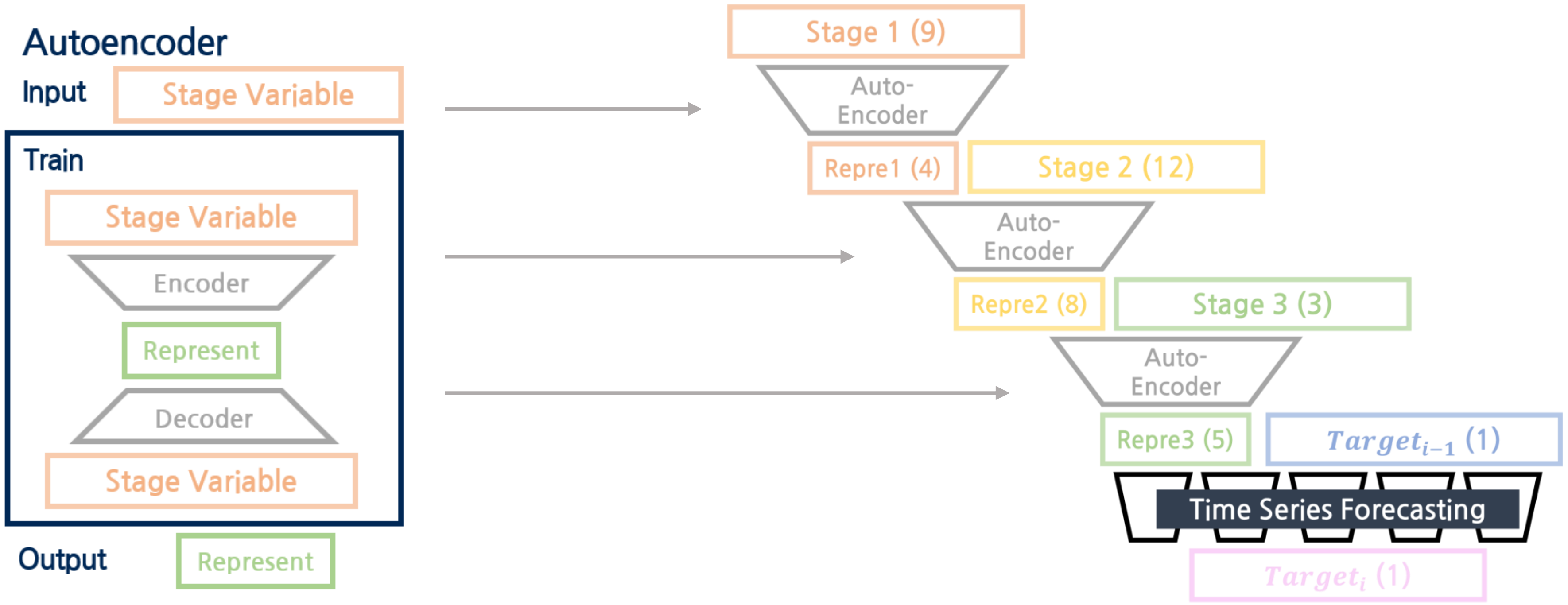
- Create more robust linear model for Multi-stage
  - Advice of our team's professor and teaching assistant



Multi-stage AutoEncoder

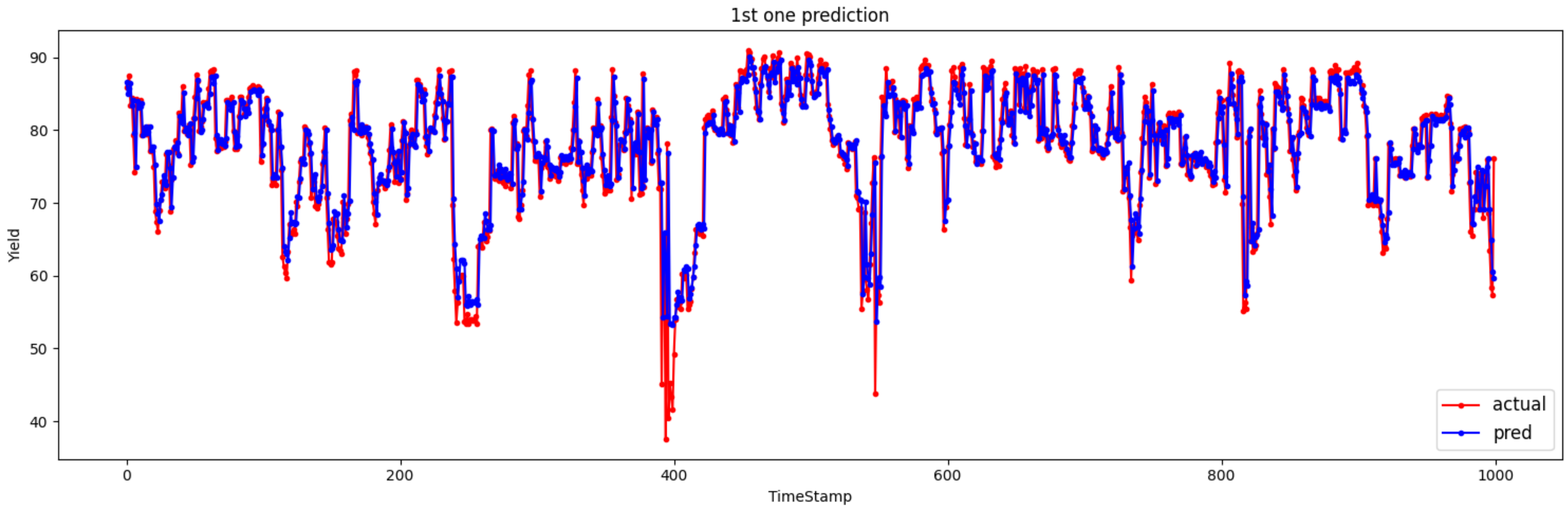
Multi-stage AutoEncoder for Multi-stage Forecasting

- Finding good stage representation variables with Multi-stage AutoEncoder
  - Starting with the second, put the previous representation and its Stage variables into the same Multi-stage AutoEncoder
  - The 1st Representation variable is not affected by the 2nd Stage, and the 2nd Representation is affected by the 1st variable but not the 3rd Stage variable.
- Continue to make representation and predict the next target with the final representation and the previous targets



Multi-stage AutoEncoder

Model	MSE
Multi-stage AutoEncoder	13.4170



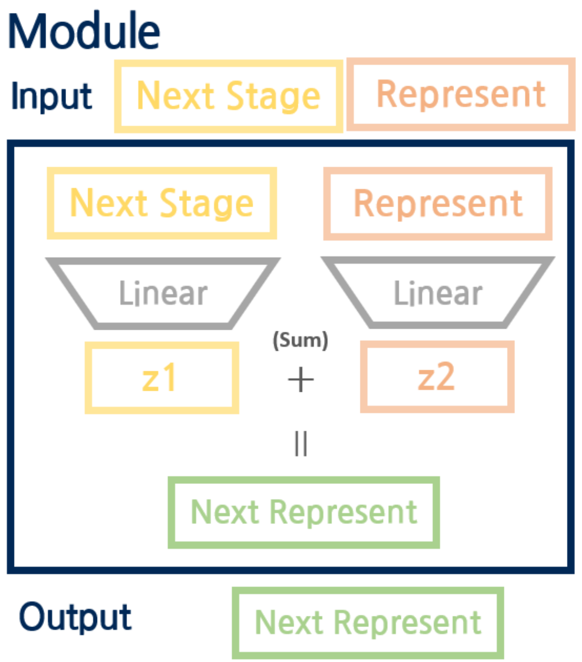
However, there is a big limitation: The model is not an end-to-end model.

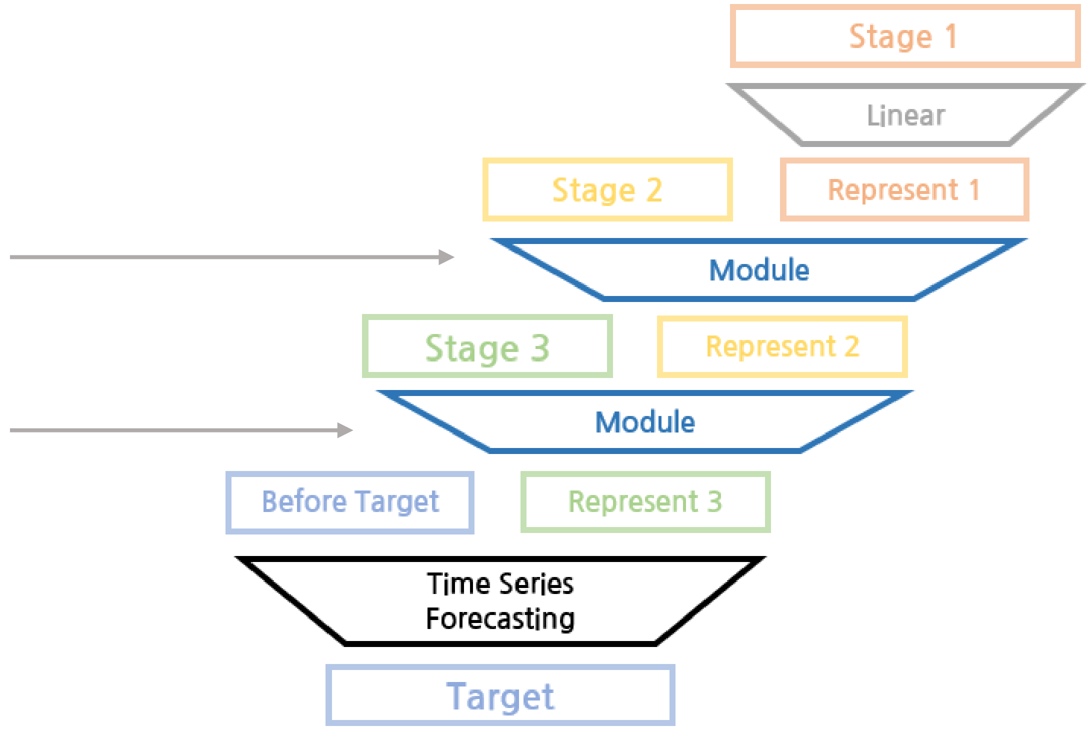
Each of latent variable are not made for better forecasting, just for good represent of each input

Multi-stage Linear

Linear for Multi-stage Forecasting

- We made the following modifications to the original linear model
  - After equating the dimensions of the representation, apply a linear model to each of the previous representation and the next stage variables to create the following representation by summing them together





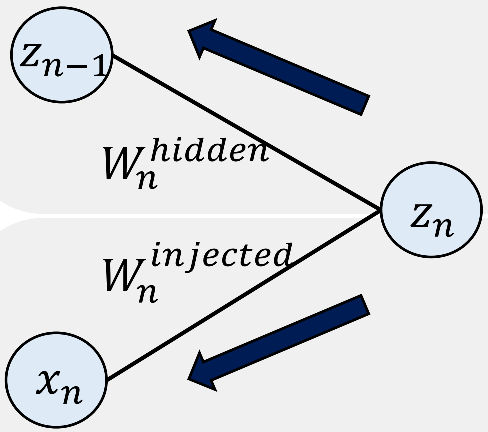
Multi-stage Linear

"Is Multi-stage Linear truly built so that the future stage does not affect the previous stage?"

- <u>Multi-stage Linear limits the next stage not to affect previous stages.</u>

$$z_1 = \sigma(W_1^T x_1),$$
$$z_n = \sigma\left(W_n^{hidden} z_{n-1} + W_n^{injected} x_n\right),$$
$$y_n = MLP(z_n)$$

TO AVOID INFLUENCE OF NEXT STAGES



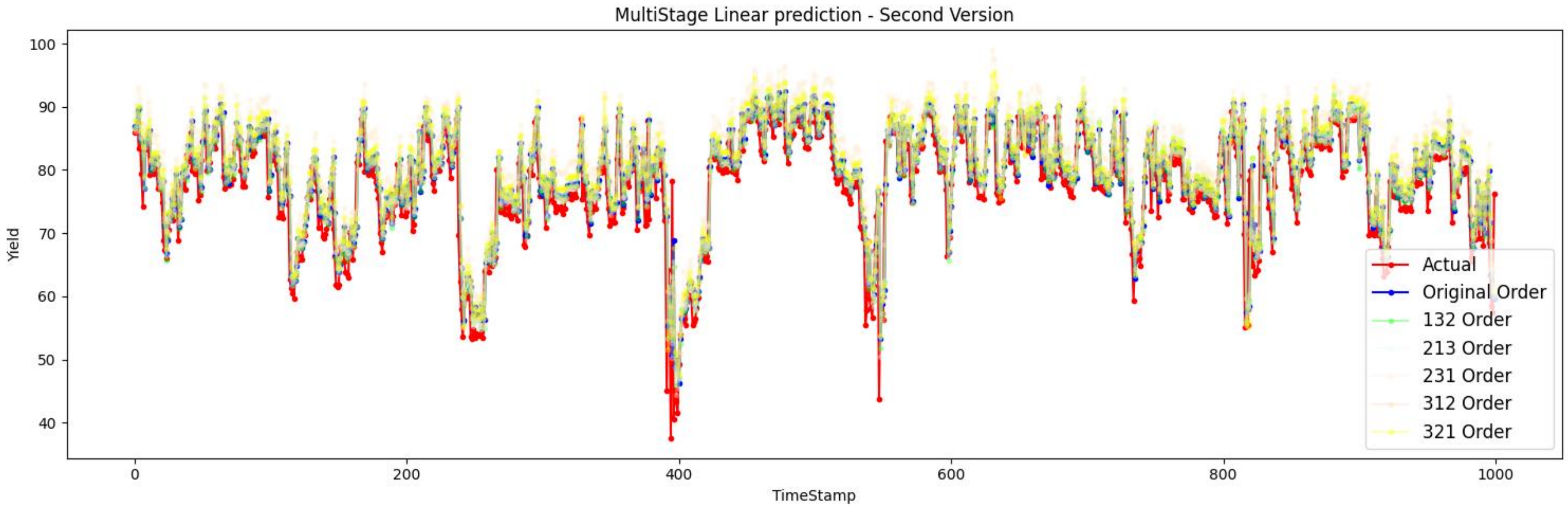
$$\frac{\partial \mathcal{L}}{\partial W_n^{hidden}} = \frac{\partial z_n}{\partial W_n^{hidden}} \cdot \frac{\partial \mathcal{L}}{\partial z_n}$$
$$= z_{n-1} \cdot \frac{\partial \mathcal{L}}{\partial z_n}$$

$$\frac{\partial \mathcal{L}}{\partial W_n^{injected}} = \frac{\partial z_n}{\partial W_n^{injected}} \cdot \frac{\partial \mathcal{L}}{\partial z_n}$$
$$= x_n \cdot \frac{\partial \mathcal{L}}{\partial z_n}$$

Multi-stage Linear

Order Experiment

Model	MSE
1 → 2 → 3	**12.9894**
1 → 3 → 2	14.0530
2 → 1 → 3	13.7356
2 → 3 → 1	15.3857
3 → 1 → 2	13.1917
3 → 2 → 1	14.9665



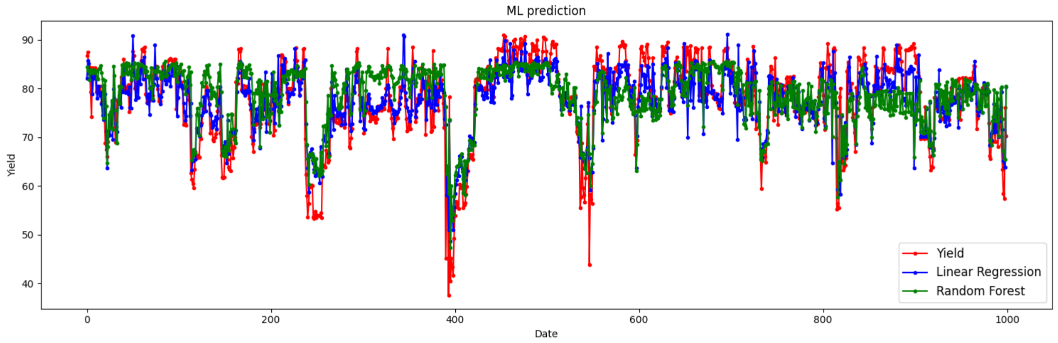
- In this model, we saw that the best performance was achieved when we reflected the original manufacturing order.
- **We can envision this Multi-stage model working in the real world.**

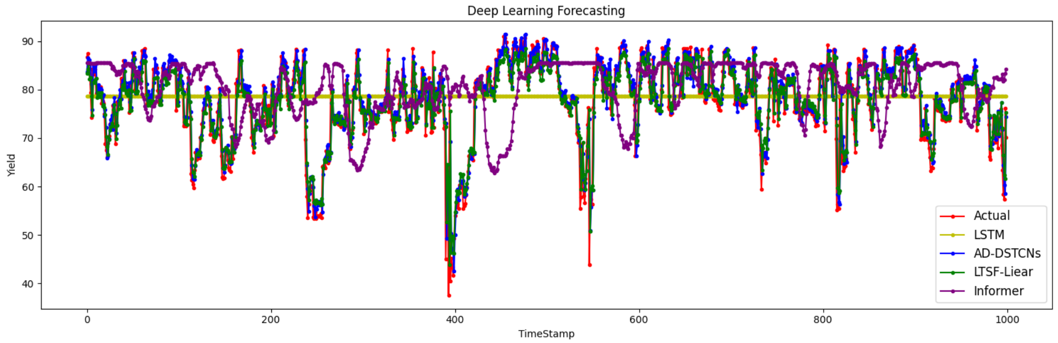
Conclusion

4

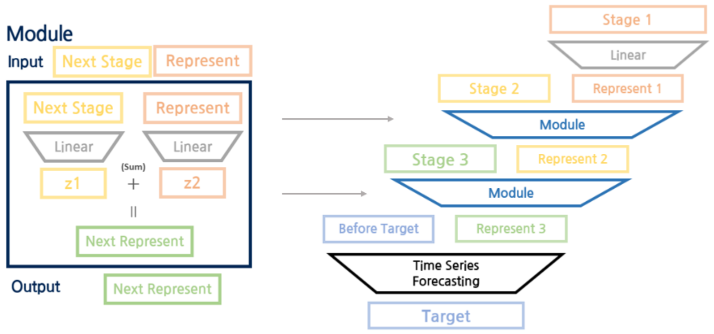
Overview

- **Application of existing models w/o considering Multi-stage process** (Chapter2)
  - Time-series prediction and forecasting models
  - From simple to latest advanced models
    - LinearRegression, RandomForest
    - LSTM, AD-DSTCNs, LSTF-Linear, Informer

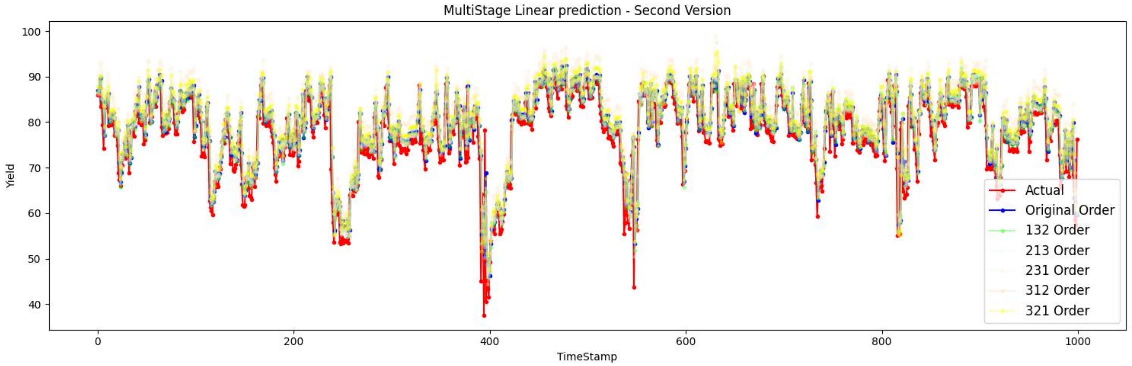




- **Development of our own models** (Chapter3)
  - Consider Multi-stage Process
    - Linear
    - Multi-stage AutoEncoder
    - Multi-stage Linear



[Final Model] Multi-stage Linear



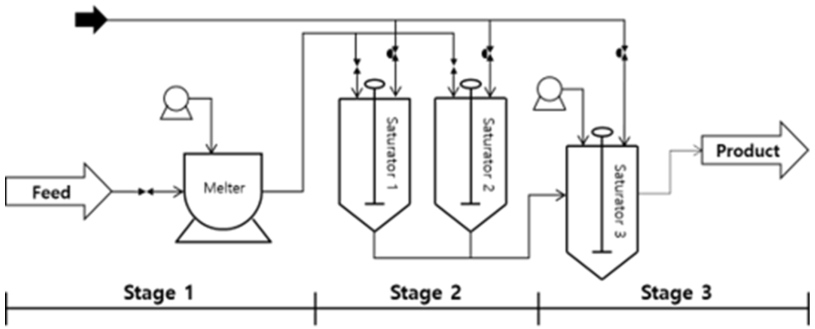
Conclusion

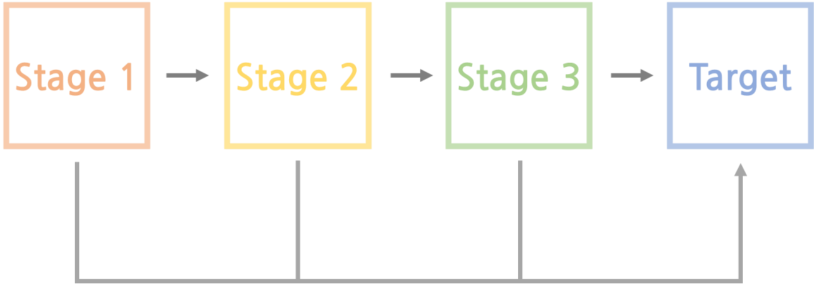
Apply traditional machine learning and recent deep learning methods

- Improve understanding of the recent deep learning models.

Try various methods on new data formats that we have never encountered

- Various stages of a manufacturing process are common, and learning sequential characteristics is critical.
- We proposed a new model framework that takes into account multi-stage beyond simple time-series forecasting.
- Analyze the backpropagation process, consider and implement various models.
- **When multi-step processes are factored directly into the model, we can see that performance can actually improve.**





Future Work

Need to develop a model that can automatically preprocess the time lags

- In our case, we had prior knowledge of the time lag and used it after preprocessing, but in the field, there is more often no prior knowledge, so a model that can automatically preprocess the time lags for this is needed.

Need a model that can take more temporal information into account

- It's hard to say that our model doesn't capture temporal information, but we'll need to apply models that can handle time series more sophisticatedly, such as Transformer, RNN, and TCN, to the Multi-stage model to better capture temporal information.

Thank You ☺