Transformers and Multi-features Time2Vec for Financial Prediction

Bui Nguyen Kim Hai, Nguyen Duy Chien

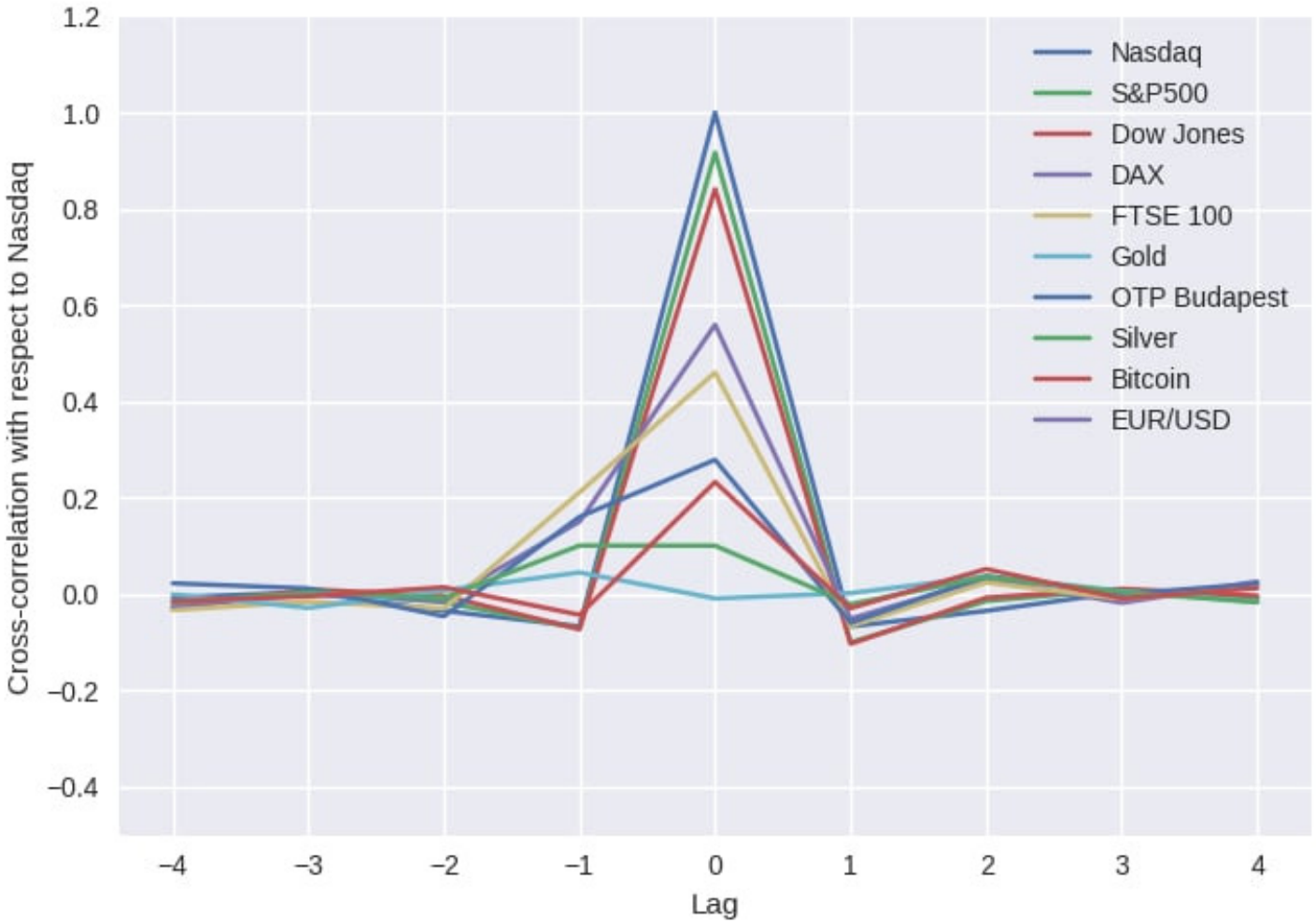
TDK CONFERENCE – IT SCIENCE SECTION, 2024 SPRING

Budapest, Hungary
May 29, 2024

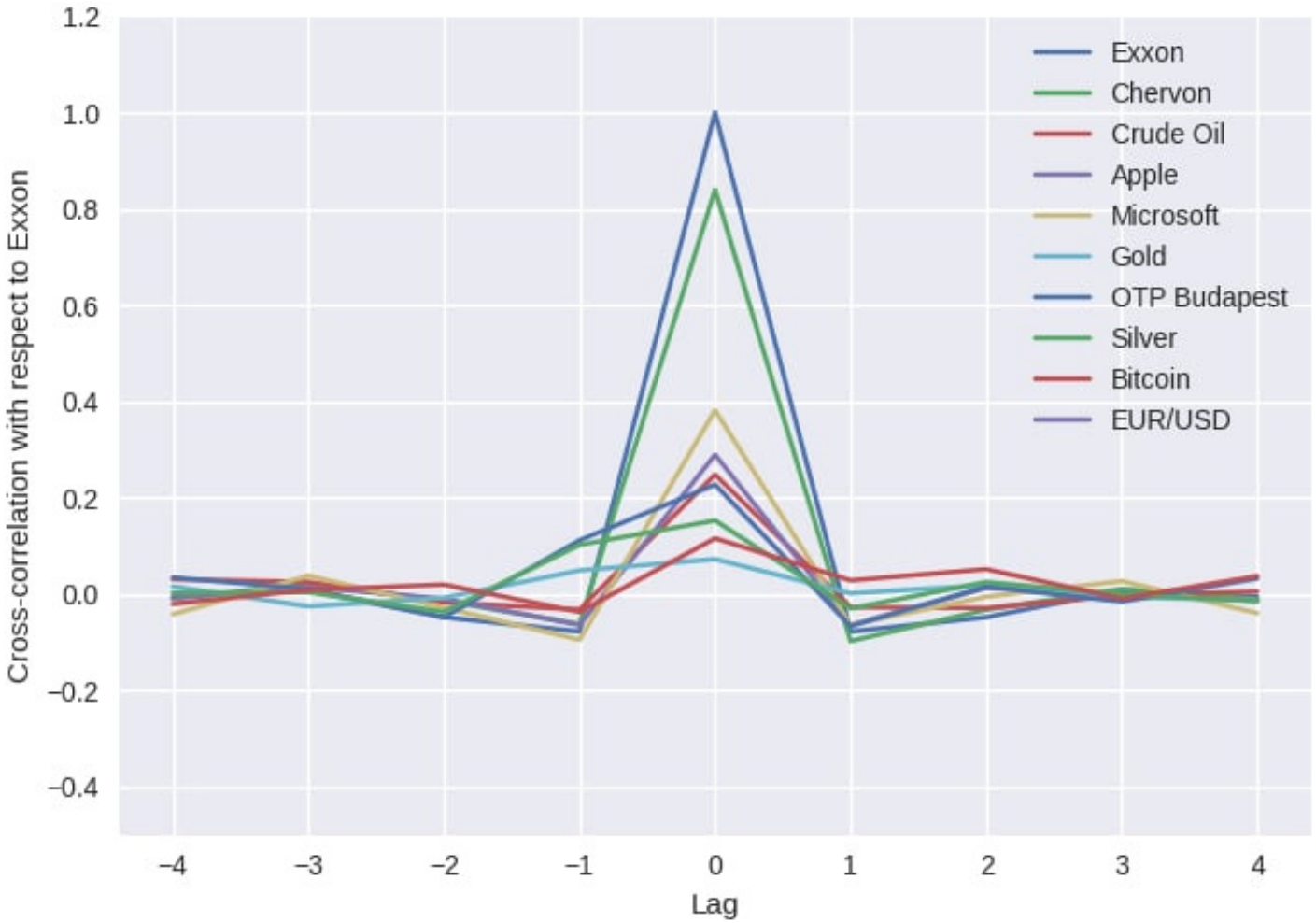
Outline

Outline

Motivation: Cross-correlation to NASDAQ



Motivation: Cross-correlation to Exxon Mobil



Motivation

By other works

- Researchers try to combine Time2Vec with CNN, RNN, LSTM, and Attention mechanism
- For instances:
  - Aeroengine Risk Assessment
  - Predicting Production in Shale and Sandstone Gas Reservoirs
  - Stock Price Forecasting

In finance area

- Studies primarily rely on one dataset

By observing trends

- Stock's trend is a Markov process
- Historical data offers limited foresight
- Stocks having similar trend is more promising

Related work

ARIMA

Making one-step-ahead predictions

RNN

Handling temporal problems in sequential data and time-series analysis.

LSTM

Using gates, LSTM enables network to learn long-term dependencies and prevent the vanishing gradient problem.

Transformer

The SOTA architecture that works well in many area such as NLP, and time-series

Time2Vec

Use to embed the time-series data to vector

Outline

Data collection

Where to collect?

Yahoo Finance

What will be collected?

Date, Open, High, Low, Close, Volume columns

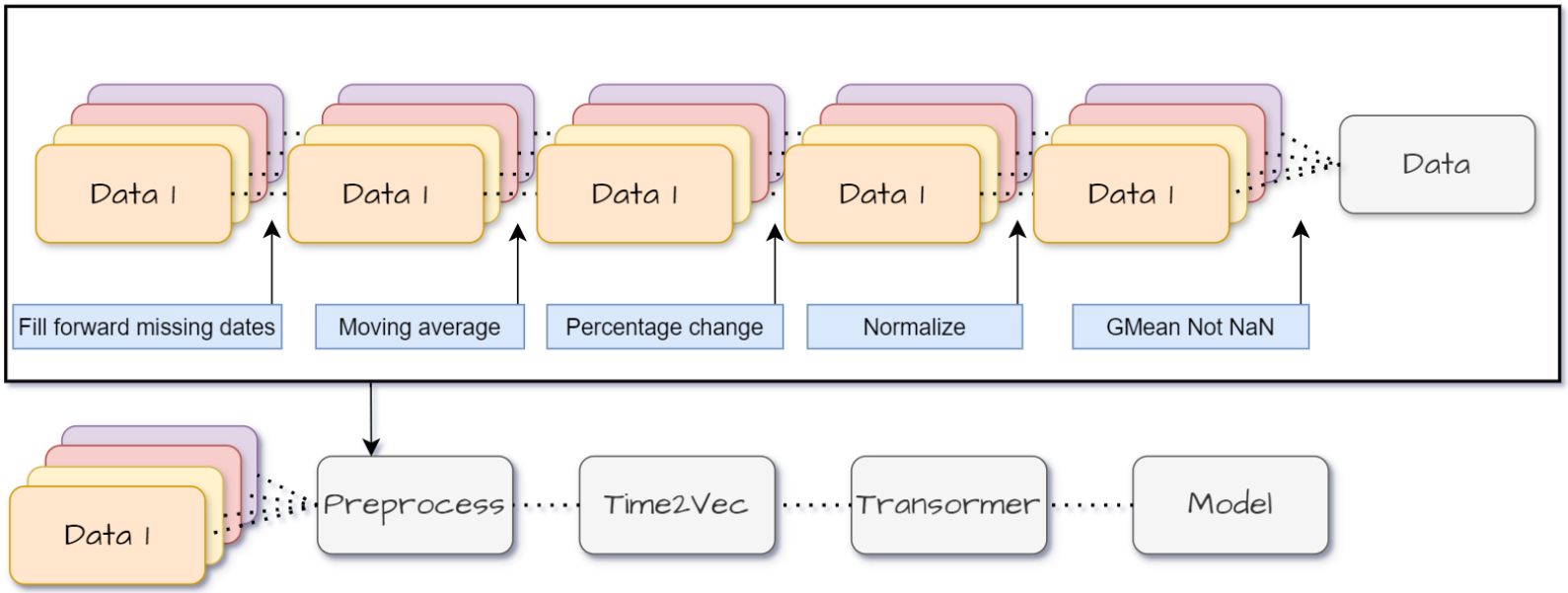
How many datasets should we collect?

Two, three, four ..., as long as they are highly correlated to each other

Collected datasets

- Group1: NASDAQ, S&P500, DJI, DAX
- Group2: Exxon Mobil, Chervon

Preprocessing data: The pipeline

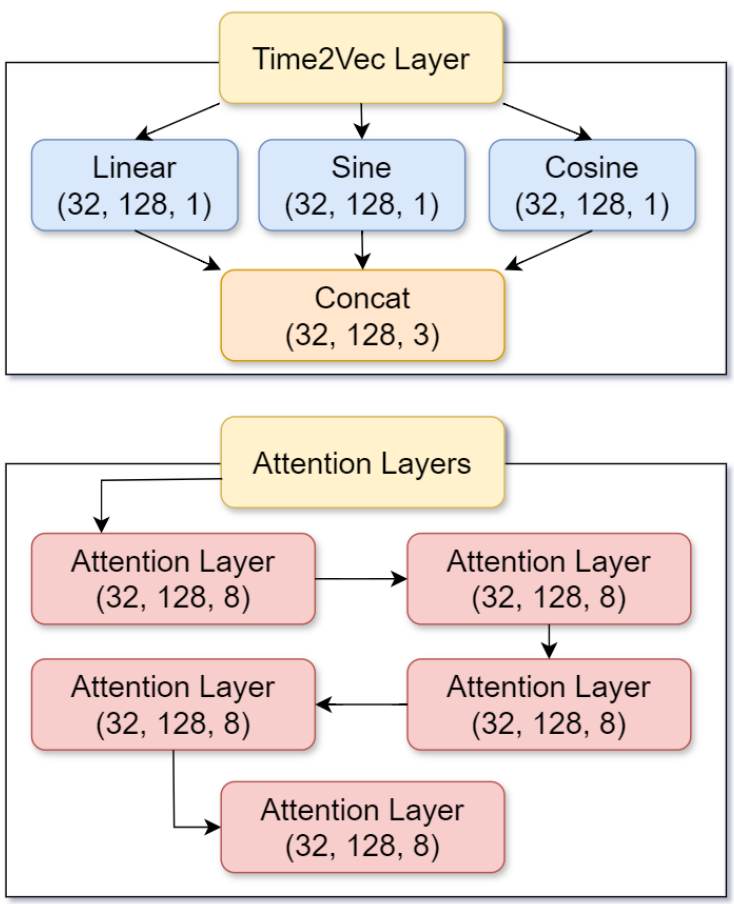


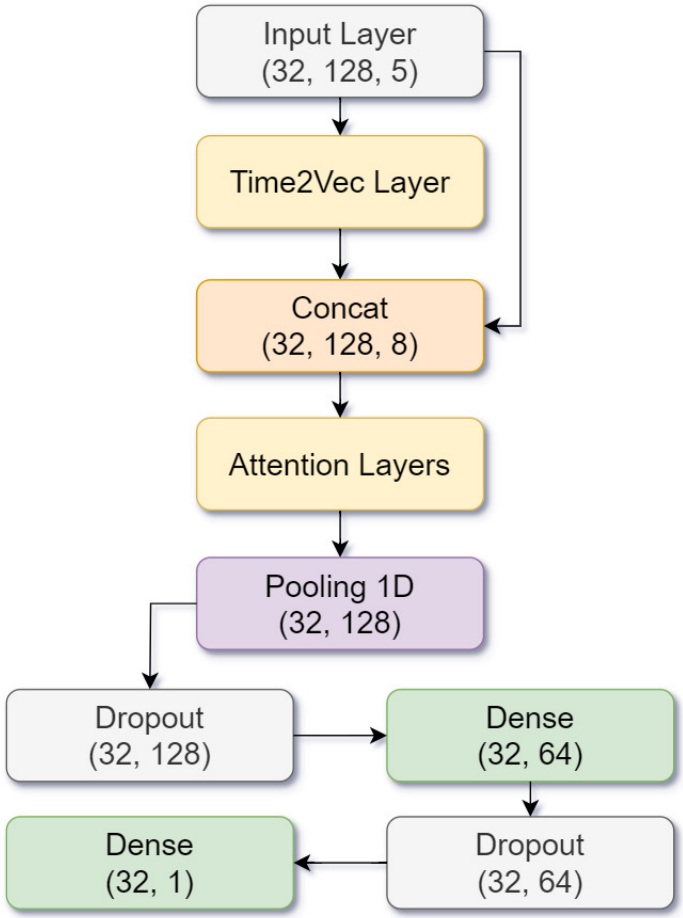
The preprocessing data pipeline.

Techniques

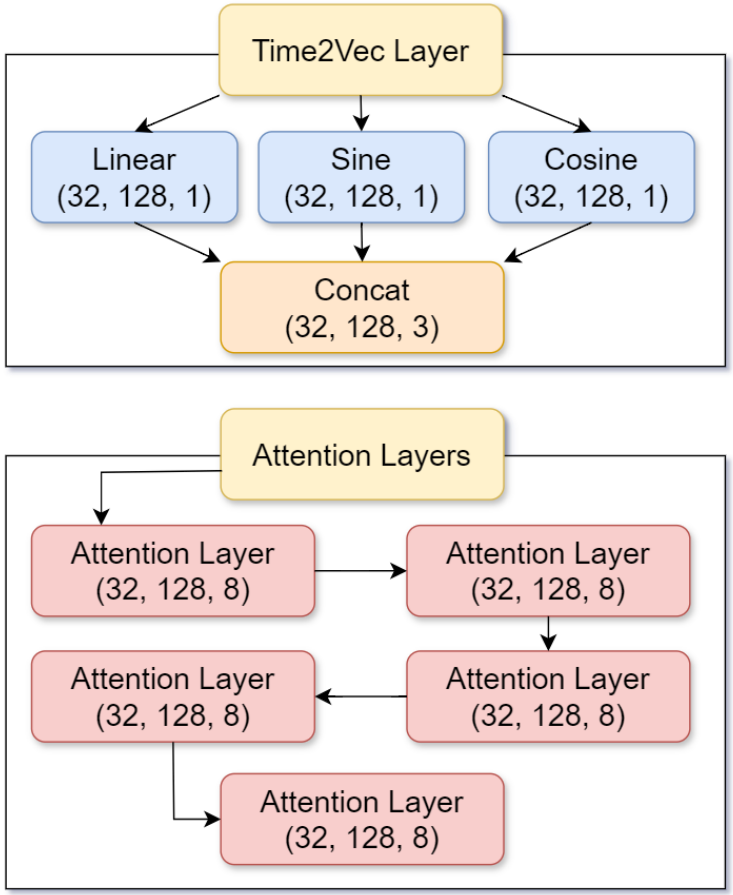
- Fill-forward: Filling missing data in dataset
- Moving Average: Smoothing dataset by averaging data
- Percentage Change: Compute the difference in the data
- Min-Max Normalization: Normalizing dataset
- Geometry Mean Not NaN (GMNN): Combining multiple datasets

Model architecture: Proposed model





Model architecture: Role of layers



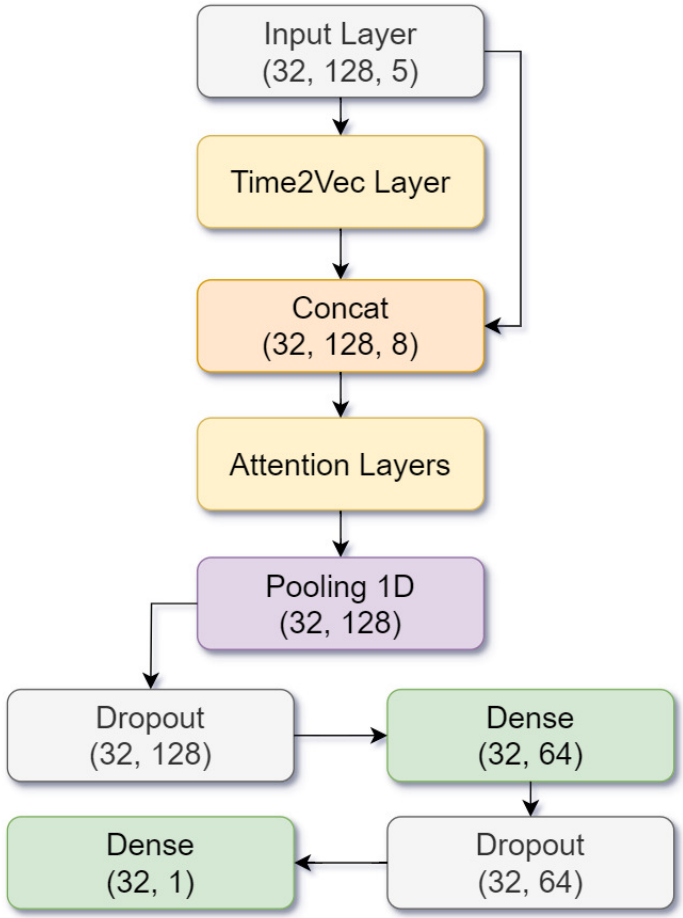
Roles

- Time2Vec
  - Linear: Capturing linear trends
  - Sine, Cosine: Encoding positions and capturing periodic behaviors
  - Concat: Concatenating above three layers
- Attention Layers
  - To study the trend from different aspects, positions

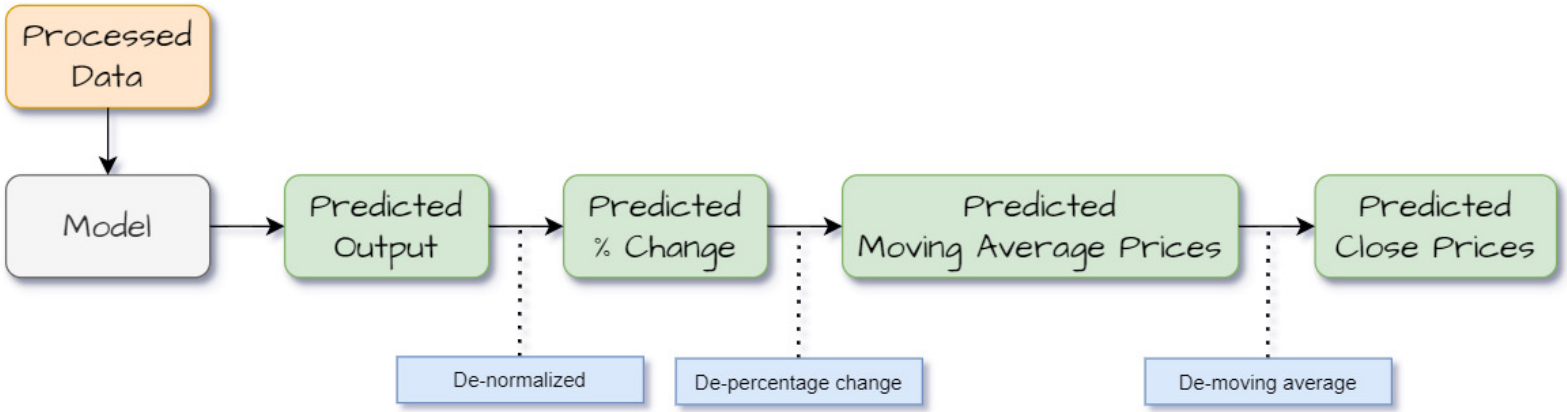
Model architecture: Role of layers

Roles

- Time2Vec: Catch continuous attribute of time
- Concat: Apply Residual Connection
- Attention: Deep understanding trend movements
- Pooling: Reducing dimension
- Dropout: Prevent over-fitting
- Dense: Apply activation functions (ReLU)



Decoding engineering



The decoding pipeline.

Techniques

- De-normalized
- De-percentage change
- De-moving average

Why don't we use De-GMNN step?

- Output is **normalized** (Invariant)
- Target is **one** dataset, output only reflects that one

Outline

Conclusion

Conclusion

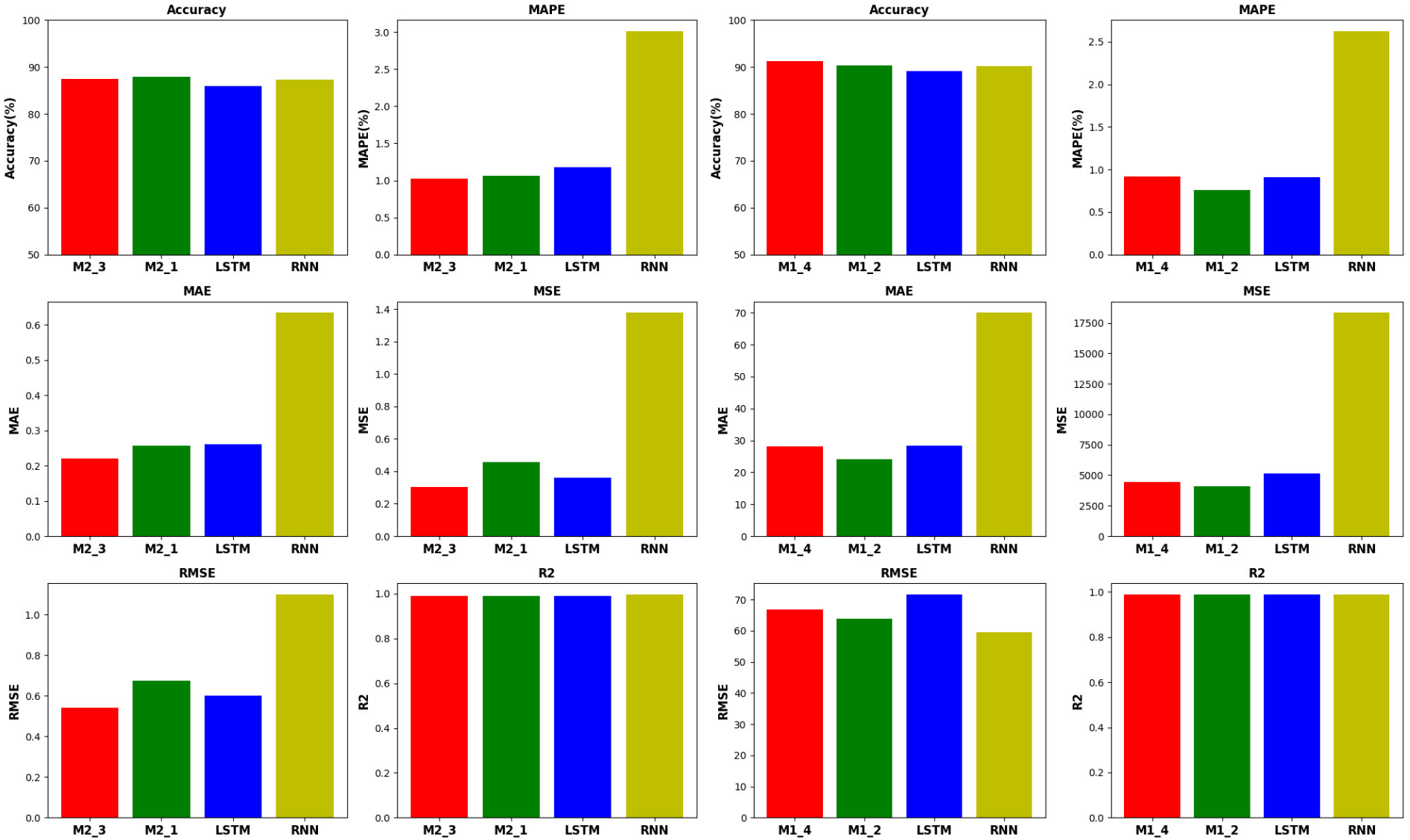
By leveraging multiple criteria to evaluate the proposed model such as

- MAE, MAPE, RMSE, MSE, R2-score (price prediction task)
- Accuracy (trend forecasting task)

We can proudly say that, the multi-feature model

- **Outperforms** the single-feature one in most cases and they are **extremely close** to each other in other scenarios.
- Usually yields **better** result than the SOTA in almost every contexts.

Results



Comparing 6 metrics with respect to Exxon (Left), NASDAQ (Right)

Outline

Summary

Summary

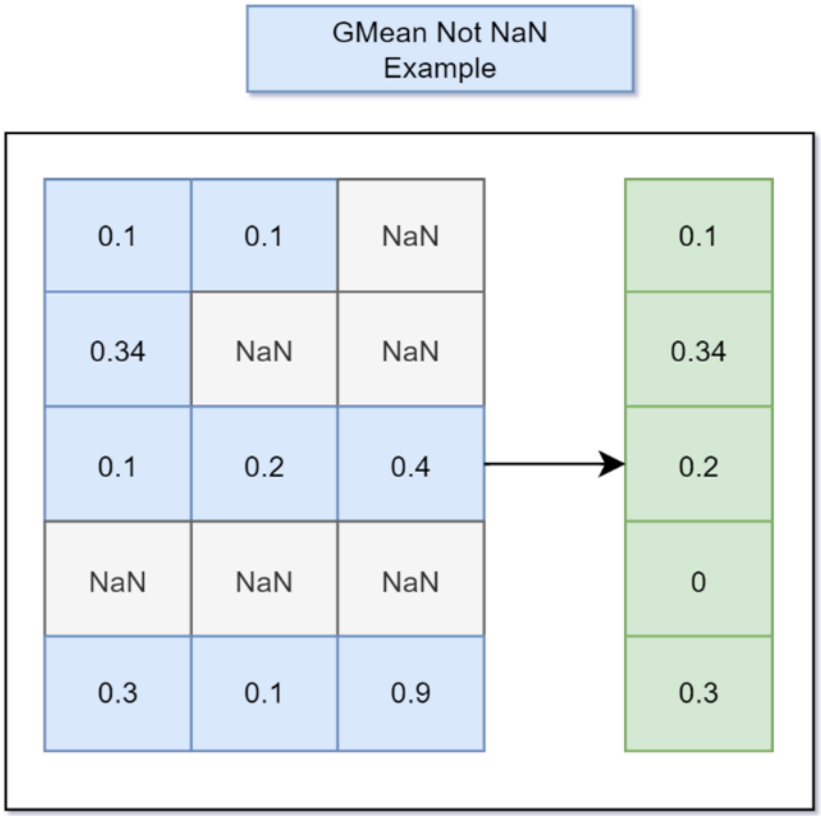
- We explore deep learning for challenging stock price prediction
- Paving the way for new feature studies and applications in various deep learning models
- Demonstrates correlation-based features and innovative neural networks improve stock price prediction

Further Research

- Fine-tuning the architecture
- Continuing improving processing methods
- Comparing to other SOTA neural networks like KAN
- Applying the architecture to other areas

Thank you for your attention!

But... What is GMNN?



GMNN Attributes

- Union: Handling length difference when combining datasets
- Invariant: Keeping the data stays normalized
- Representation: The output reflects the whole datasets

A simple sample of applying GMNN transformation

But... What is Time2Vec[1]?

Time2Vec

An approach providing a model – agnostic vector representation for time

Time2Vec Function

$$\mathbf{t2v}(\tau)[i] = \begin{cases} w_i\tau + \varphi_i & \text{if } i = 0 \\ F\left(w_i\tau + \varphi_i\right), & \text{if } 1 \leq i \leq k \end{cases}$$

w, φ: learnable parameters τ: time

F: periodic activation function (eg. $\sin, \cos$)

Time2Vec Attributes

- Capturing both periodic and non periodic patterns
- Being invariant to time re-scaling
- Being simple enough so it can be combined with many models

[1] Seyed Mehran Kazemi et al. "Time2Vec: Learning a Vector Representation of Time" (2019)

Splitting data

Can we just put the processed data into model straight forward?

No, we can not feed the whole data straight forward into the model, because that action will cause **over-fitting** problem which no one want it to be happened when training model

Do we need to shuffle the data before splitting?

No, we don't want to shuffle the data because it has **continuous** attribute of time then shuffling will make the data lose this special and important property which cause a big problem for the model to learn the pattern

How will we feed input data to our model?

We will split the data into three categories:

- Train: The first 80% data of the input
- Validation: The next 10% data
- Test: The last 10% data

Applications

Can it be used in production?

Of course, but to be more accurate and precise, the architecture must be fine-tuned to fit the expectation

Is it easy to set up and use in production?

Yes it is, developer just need to find the appropriate datasets and train the model

Can it be used in other areas?

Yes it can be used in other areas, we just need to find the dimension sizes and appropriate hyper-parameters to fit the area expectation

Can it predict the tomorrow stock price?

Yes it can, but we need to apply some more techniques to decode back to real values

Questions

From the reviewer

- Correlation method: We use the default one in pandas (Pearson)
- Source of Equations and formulas in Decoding part: They are just applying the reverse logic of how we coding it. That's why we don't use any citation for these in our thesis

Explain correlation method

Pearson

Considering two time series $y(t)$ and $x(t)$ with an equal length of T $(1 < t < T)$, the Pearson's correlation coefficient between the series is defined as

$$\rho = \frac{\sum_{t=1}^{t=T}(x(t)-\bar{x})(y(t)-\bar{y})}{\sqrt{\sum_{t=1}^{t=T}(x(t)-\bar{x})^2}\sqrt{\sum_{t=1}^{t=T}(y(t)-\bar{y})^2}}$$

where $\bar{x}$ and $\bar{y}$ are average values over the whole series

A higher value of Pearson's correlation coefficient ρ positively corresponds with the stronger correlation degree, thereby indicating a stronger mutual interpretation ability between $y(t)$ and $x(t)$

However, when time series are nonstationary and nonlinear, the Pearson's correlation coefficient cannot represent a reliable correlation degree because the observational values in the time series probably rely on each other.

Explain used techniques: De-normalized

De-normalized step

$$x_{pct} = x_{nor} \times (max - min) + min$$

Is from

$$x_{standarlized} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Where:

- x_{pct} has the same role with x
- x_{nor} has the same role with $x_{standardlized}$
- max is stored value of $\max(x)$
- min is stored value of $\min(x)$

But... why?

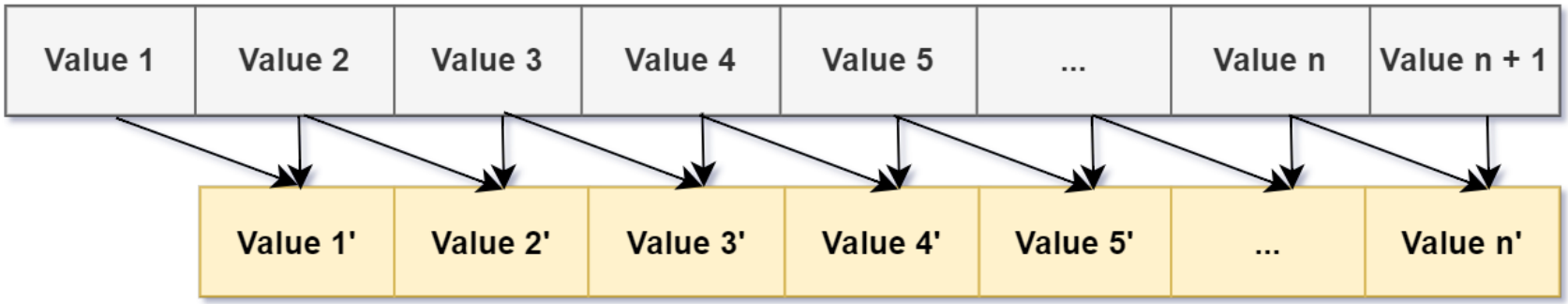
Because

- Before standardlizing, our x is currently percentage-changed

Explain used techniques: De-percentage change - Problem

De-percentage change step

This is the way we **create** to get high accuracy predict values by pairing with real values to decode
Why? Let's see how we encode percentage change



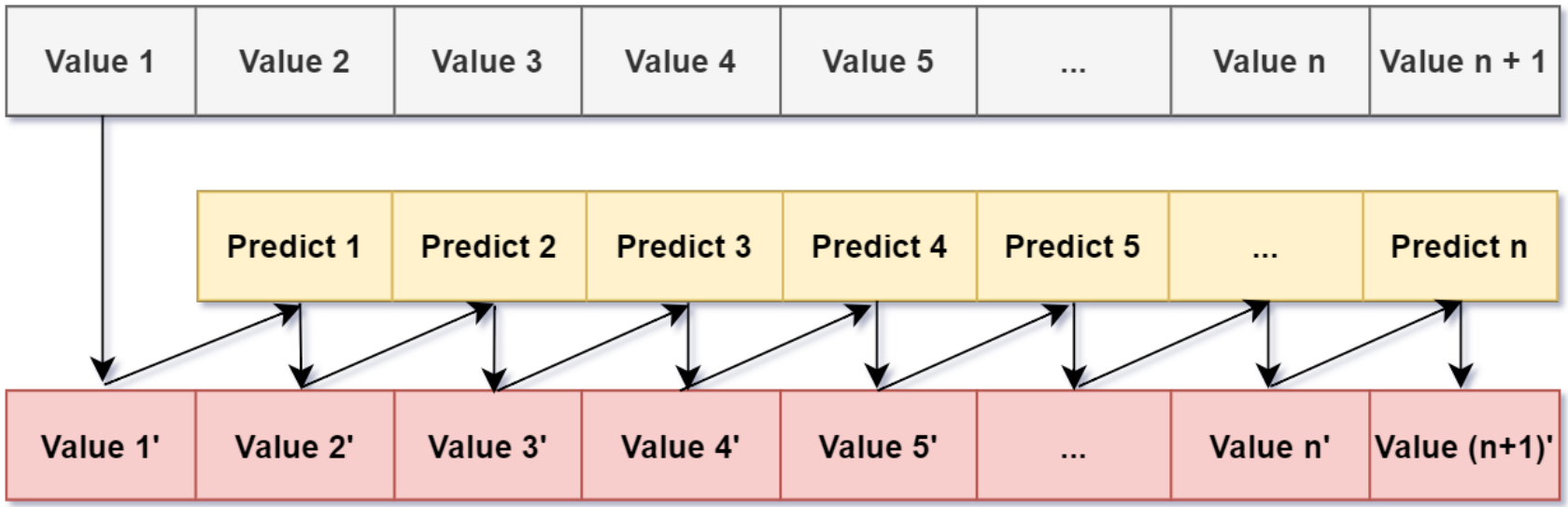
This lead to a problem!

We lost one data cell

Explain used techniques: De-percentage change - Plan A

Plan A to fix

This plan will cause a very big mistake when converting back to real prices because it one predict is not good, the next ones will be affected too!



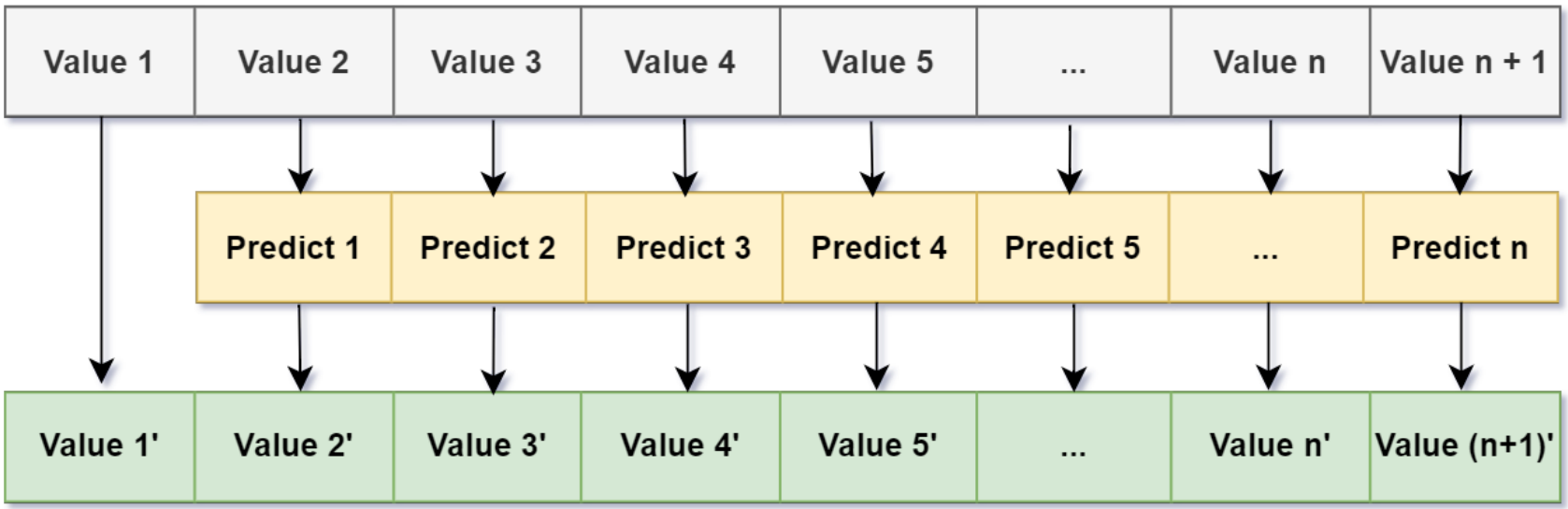
Any plan else?

Yes, we will see plan B - which will be better, and that is also the one we proposed in the thesis

Explain used techniques: De-percentage change - Plan B

Plan B to fix

This plan yields very good result in converting back to real values, because each cell is only related to the real one and/or the predict one - which is more realistic than plan A



Explain used techniques: De-percentage change - Formula

Formula for plan B

$$x_{mva} = \begin{cases} v_{mva}, i = 0 \\ v_{mva} \times (1 + x_{pct}), otherwise \end{cases}$$

Is from

$$x_{pct} = \frac{x_b - x_a}{x_a}$$

Where:

- x_a, x_b: is two consecutive moving average values (x_b after x_a)
- x_{pct} has the same meaning in two equation
- x_{mva} has the same role with x_a, x_b
  (x_{mva} is a vector; x_a, x_b are numbers)
- v_{mva} is real moving average values (from encoding part)

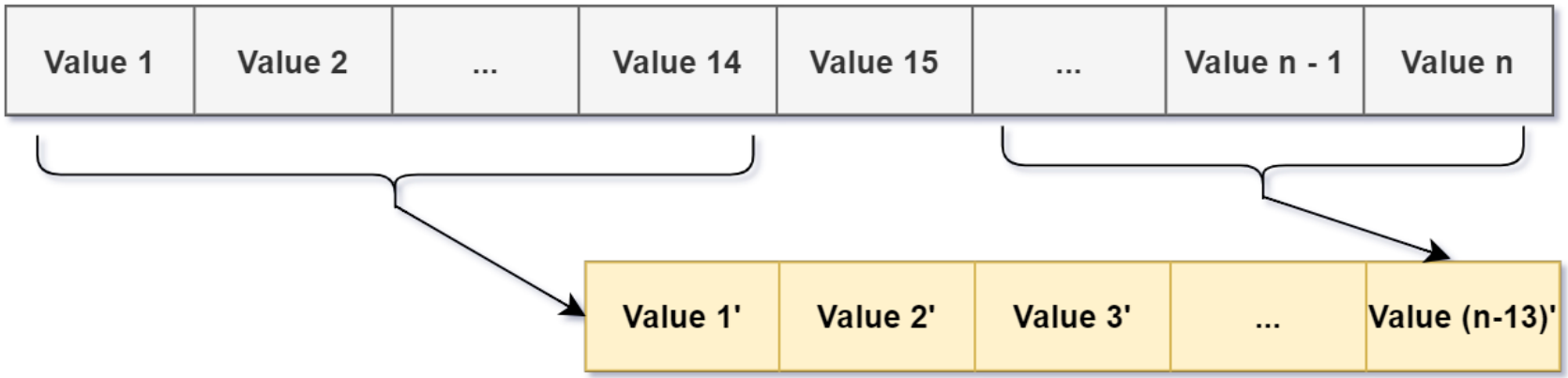
Disclaimer

We have a typo in our thesis in this step. We messed up between v_{vma} with v_{pct}

Explain used techniques: De-moving average - Problem

De-moving average step

This is the way we **came up with** to get high accuracy predict values by pairing with real values to decode
Why? Let's see how we apply moving average with step equals 14 days



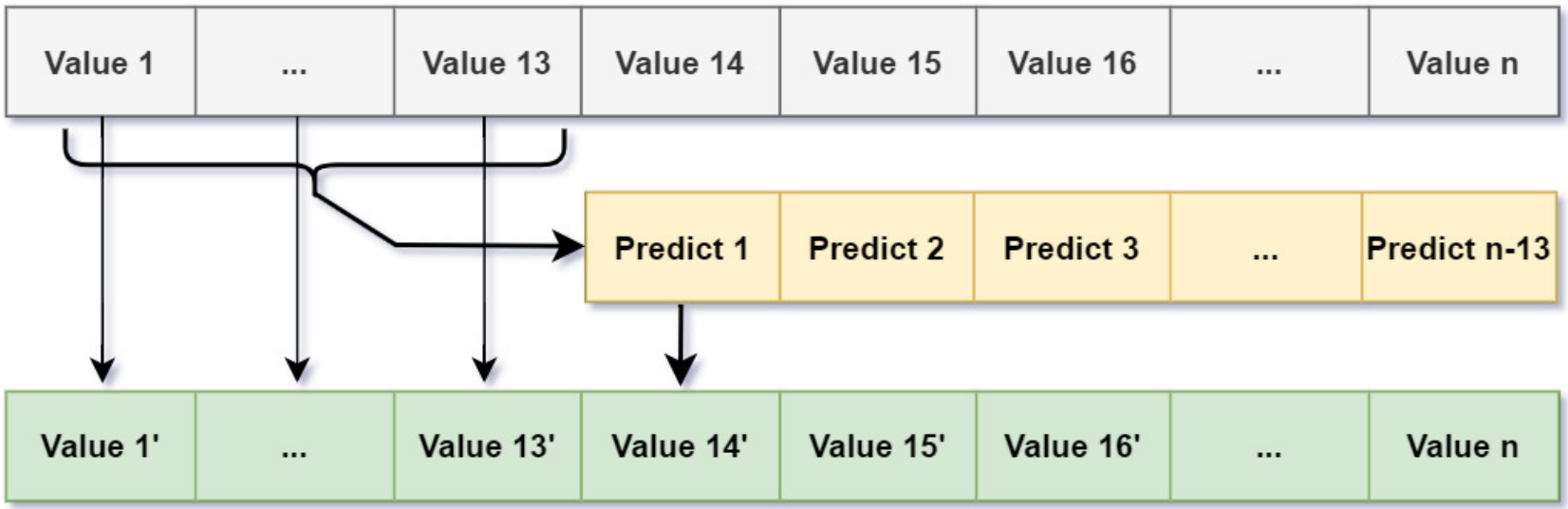
This lead to a problem!

We lost the first 13 data cells

Explain used techniques: De-moving average - Fix the problem

De-moving average step

This method yields very good result in converting back to real values, because each cell is only related to the real one and/or the predict one



Explain used techniques: De-moving average - Formula

Formula

$$x_{close} = \begin{cases} v_{close} & , i \leq 13 \\ x_{mva} \times 14 & - \sum_{k=i-14}^{i-1} v_{close}, i \geq 14 \end{cases}$$

Is from

$$mva = 1/14 * \sum_{0}^{13} close$$

Where:

- x_{vma}: has the predict moving average prices (from the De-percentage change step)
- mva: real moving average prices
- v_{close}, $close$: has the same meaning in two equation, real close prices