# Forecast the copper price that has structural changes

Tsaih, Rua-Huan

# Forecast the copper price

## Importance

- Copper is one of important raw materials for economic development (Liu et al., 2020).
- More than 90% of industrial products rely on copper (Chang et al., 2013; Astudillo et al., 2020).
- The copper prices account for 60% of total cost of the relevant copper manufacturing companies (Sharma et al., 2015).

## Characteristics of copper price data stream

- The copper price data is a time series data stream with structural change in statistics (Koitsiwe & Adachi, 2017, 2018), which is also known as concept drift (Tsymbal, 2004).
- The embedded fitting function form of the copper price data stream is non-linear (Seguel et al., 2015) and adaptive.

# structural change / concept drift

**Definition:**

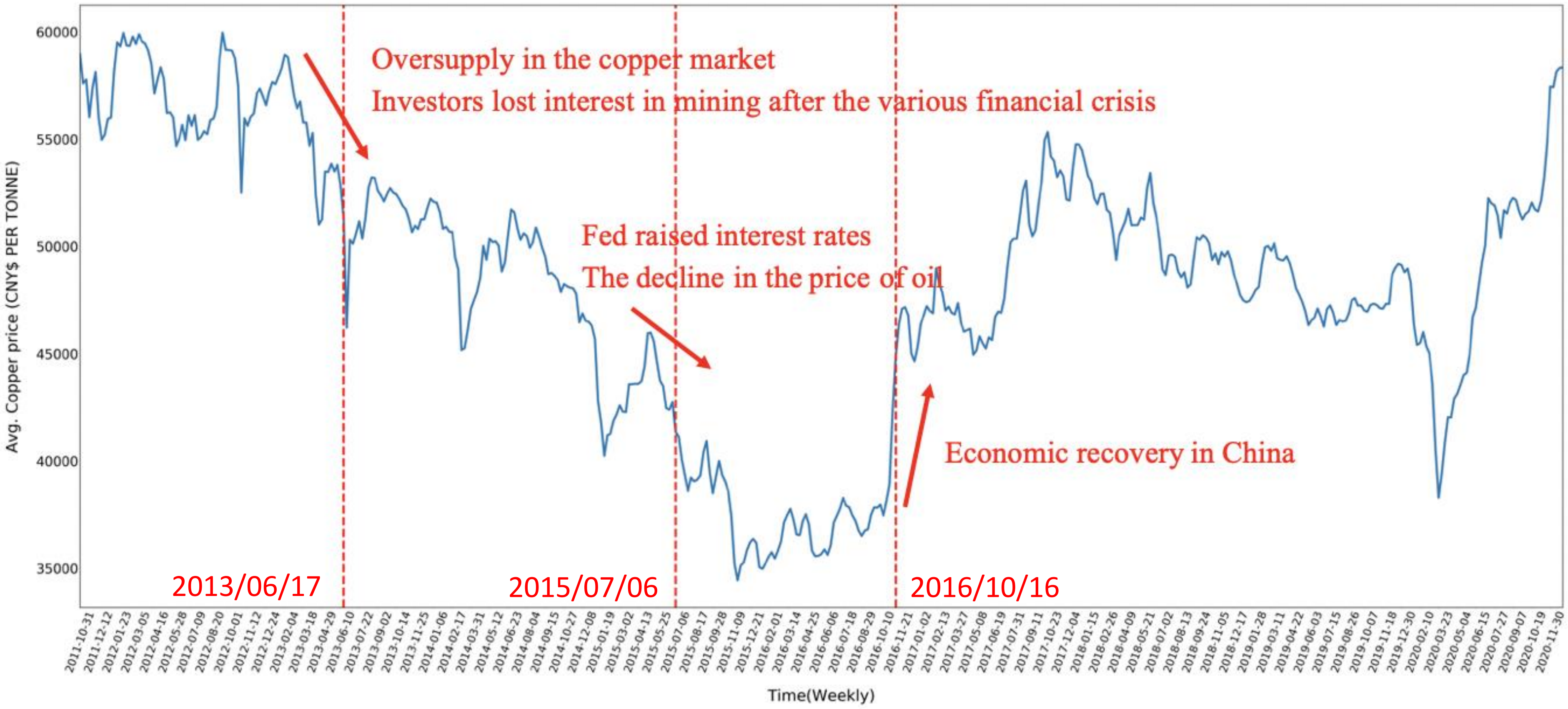
- In the econometrics and statistics literature, the structural change refers to the fundamental changes in the way a market or economy functions (Matsuyama, 2008).
- In the engineering and machine learning literature, the concept evolution emerges when the novel classes appear in the data stream (Tsymbal, 2004).

**Challenges:**

- Wang et al. (2003) argue that in the concept-drifting environment, the model built by the training dataset cannot be consistent with current concepts.
- The emerging concept evolution leads to the change of fitting function form embedded in the data stream (Baier et al., 2020).

# Results of structural change test of copper prices by the methodology of Bai and Perron (1998, 2003)

Three structural breakpoints are detected.



# Handling the structural change

- To handle time-series data stream with the structural change, the sliding window is an efficient strategy (Giannella et al., 2003; Chang & Lee, 2005; Li et al., 2005; Leung & Khan, 2006; Mozafari et al., 2008; Wang et al., 2019).
- Handling the concept-drifting data stream:
  - sequence-based moving window (Babcock et al., 2001)
  - instance weight (Klinkenberg, 2004)
  - monitoring two different time window distribution (Gama & Kosina, 2014)
  - detection concept change point (Kosina & Gama, 2015)
  - sliding window (Fornaciari & Grillenzoni, 2017)

# Neural Networks (NN)-based models

- The neural networks (NN)-based models are usually used to learn the data with an embedded non-linear fitting function form (Shokry & Espuña, 2018).

- The challenges associated with the NN-based model:

  - Gradient-vanishing: the learning algorithm basically using the gradient descent method converges to the local optimum or the saddle point of the loss function.

  - Overfitting: the complicated learned hypothesis may fit the training data very well, even noises (or outliers), but fail to generalize to out-of-samples.

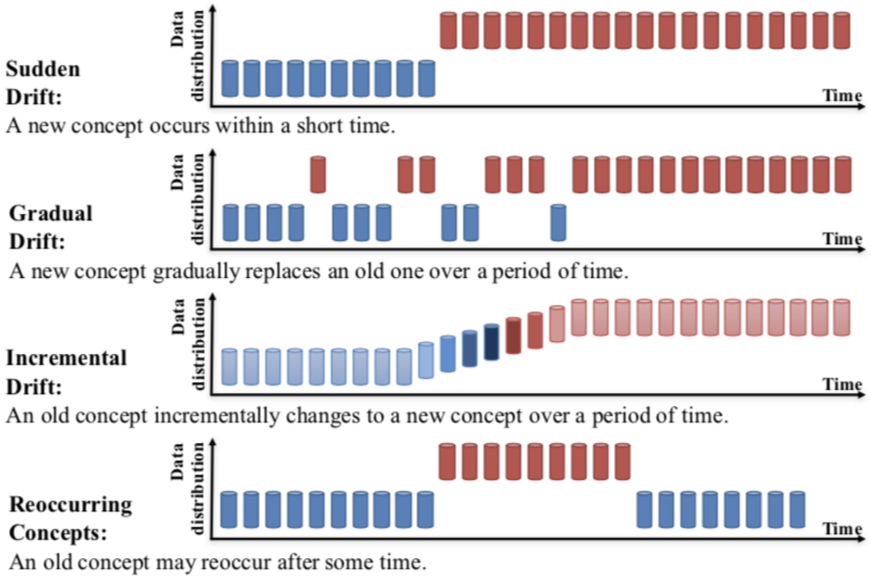
# Adaptive Single hidden-Layer Feedforward Networks (ASLFN)

Algorithms for systematically adapting the number of hidden nodes adopted in SLFN:

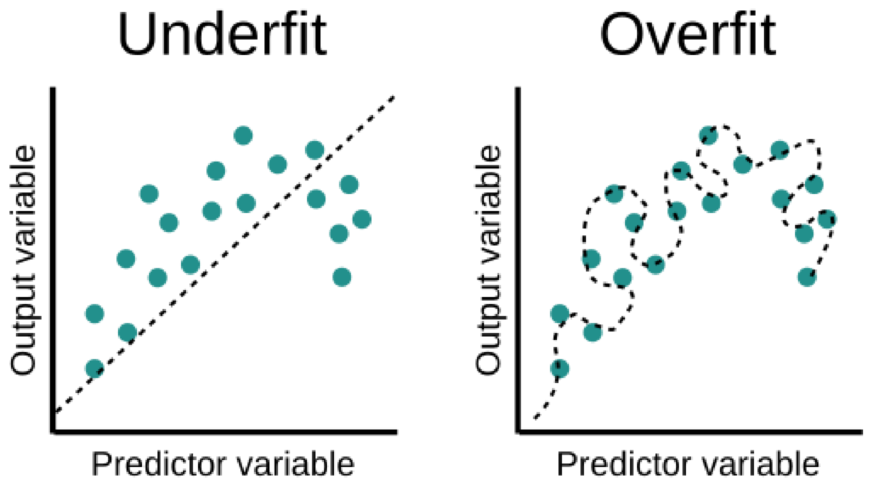
- Tiling algorithm (Mezard & Nadal, 1989)
- Cascade-correlation (CC) algorithm (Fahlman & Lebiere, 1990)
- Upstart algorithm (Frean, 1990)
- Softening learning procedure (Tsaih, 1993)
- W&S algorithm (Watanabe & Shimizu, 1993)
- CTN algorithm (Chen et al., 1994)
- Reasoning neural networks (Tsaih, 1998)
- CSI learning algorithm (Tsai et al., 2019)

# Theoretical gap

Rare modeling approaches can simultaneously deal with the aforementioned challenges (structural change, adaptive non-linear function form, gradient-vanishing, and overfitting) of copper price data stream.




structural change/concept drift




adaptive non-linear fitting function form, gradient-vanishing, and overfitting

# Modelling approaches

| Modelling approach | Examples |
| --- | --- |
| Statistical approaches (Li & Li, 2015; Sánchez et al., 2015) | AR, MA, ARMA, ARIMA, and SARIMA, TAR, ExpAR, SETAR, ARCH, GARCH |
| Learning-based approaches (Liu et al., 2017; Carrasco et al., 2018; Dehghani, 2018; Astudillo et al., 2020; Khoshalan et al., 2021; Zhang et al., 2021; Zhang et al., 2021) | Decision tree, KNN, GA, SVM, SVR, ANN, RNN |
| Hybrid approaches (Dehghani & Bogdanovic, 2018; García & Kristjanpoller, 2019; Alameer et al., 2019; Hu et al., 2020; Zhang et al.,2021) | The combination of multiple models into one framework |

# Statistical approaches

## Mechanism

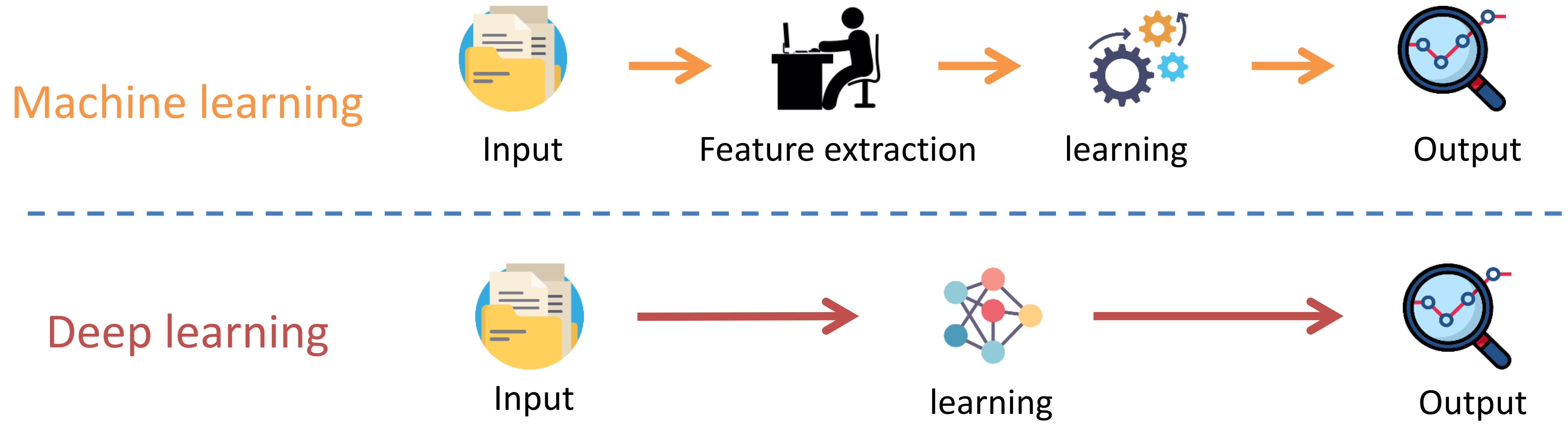
- The general form is a combination of serial correlation and heteroscedasticity in the errors.
- These models require less computational resources.

## Limitation

- Since the associated data must be stable, these models do not take into account the varied structures and nonlinear patterns of the underlying data (Çinar, 1995; Stevenson, 2007; Wang et al., 2019; Schaffer et al., 2021).
- The new statistical approaches are still limited by many variations in the model (Abbasimehr et al., 2020).
- Traditional statistical approaches are unable to respond to the high volatility data (Hu et al., 2020).

# Learning-based approaches

## Mechanism



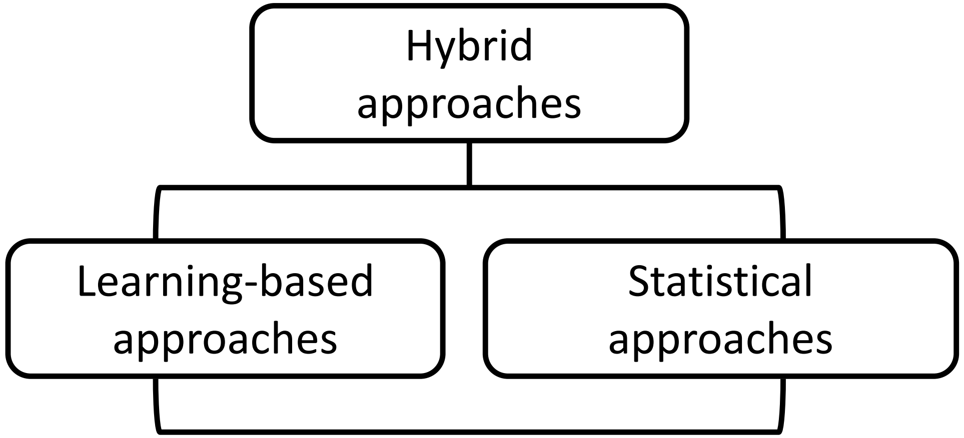

## Limitation

- The performance of the learning-based approach is sensitive to the hyperparameter setting (Domhan et al., 2015; Diez-Sierra & del Jesus, 2020).
- Issues: gradient-vanishing, overfitting, and huge computations (Ooyen & Nienhuis, 1992; Schalkoff, 2007).

# Hybrid approaches

## Mechanism

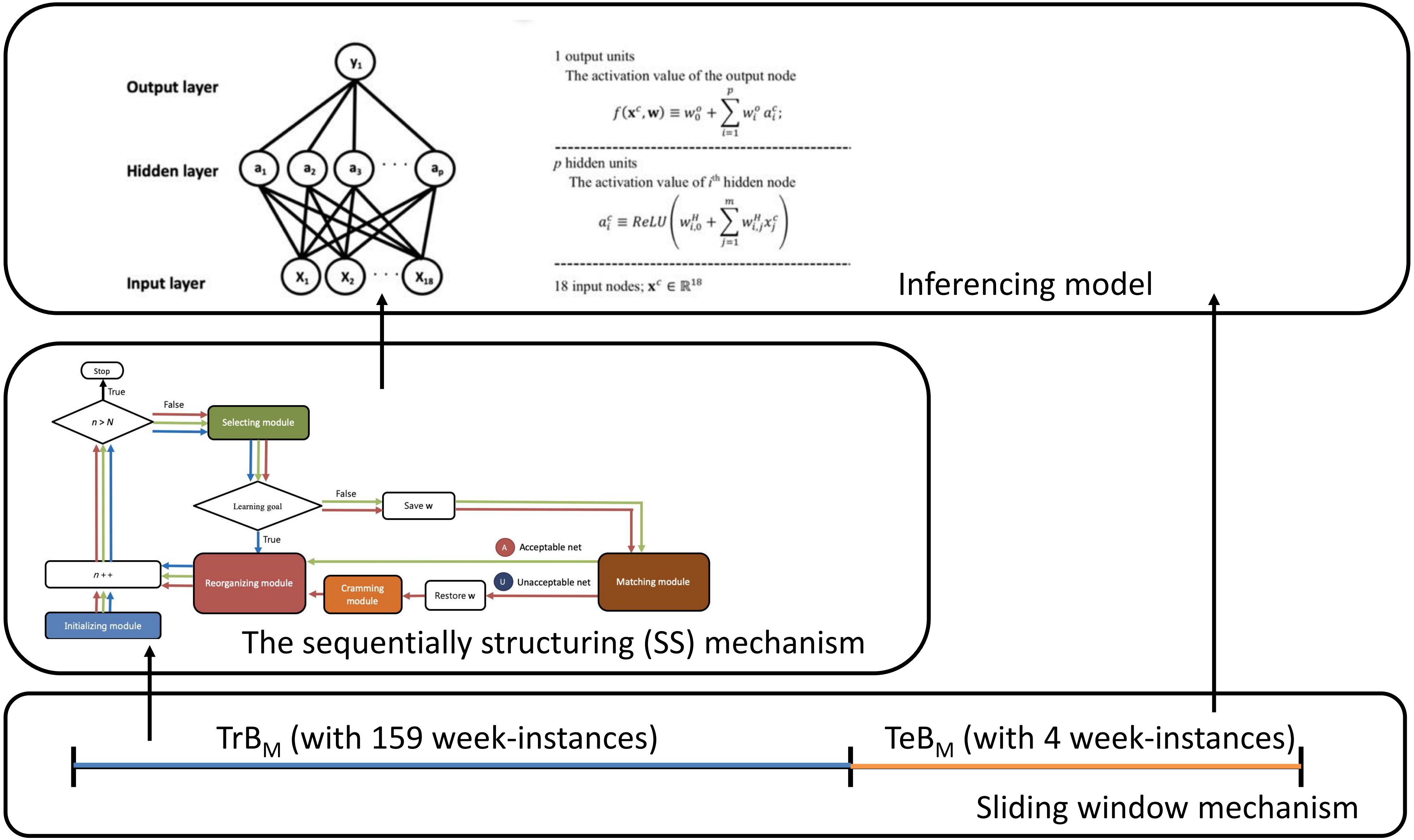
- The combination of multiple models into one framework (Alameer et al., 2019).
- The basic idea of the hybrid approach is to generate a synergetic effect through addressing the shortcomings of each single model.



# Study objectives

- To derive an AI system that (1) can deal with the copper price data stream with structural changes, and (2) contains a learning mechanism applicable to ASLFN and able of simultaneously coping with the issues of gradient-vanishing and overfitting.
- The proposed learning mechanism is named as the sequentially structuring (SS) mechanism.

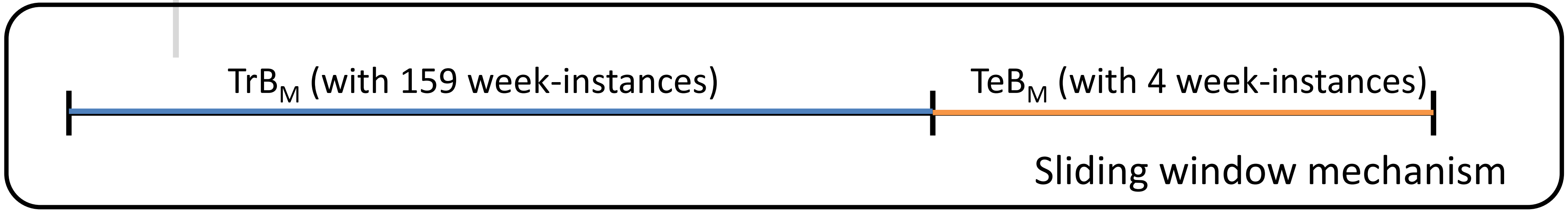
# The proposed AI system



# The proposed AI system



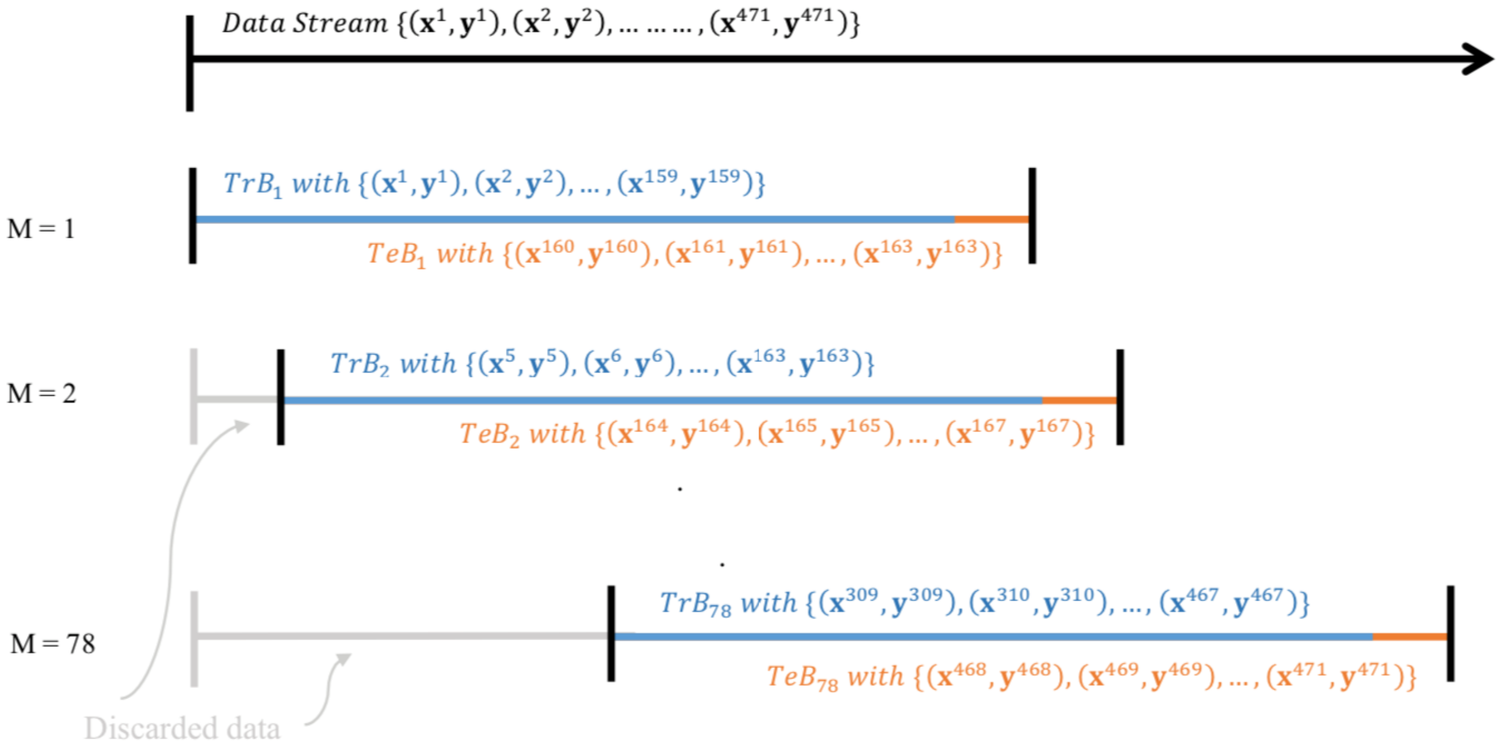

The sliding window mechanism (Fornaciari & Grillenzoni, 2017) is adopted to deal with the structural change in the copper price data stream.



# The sliding window mechanism

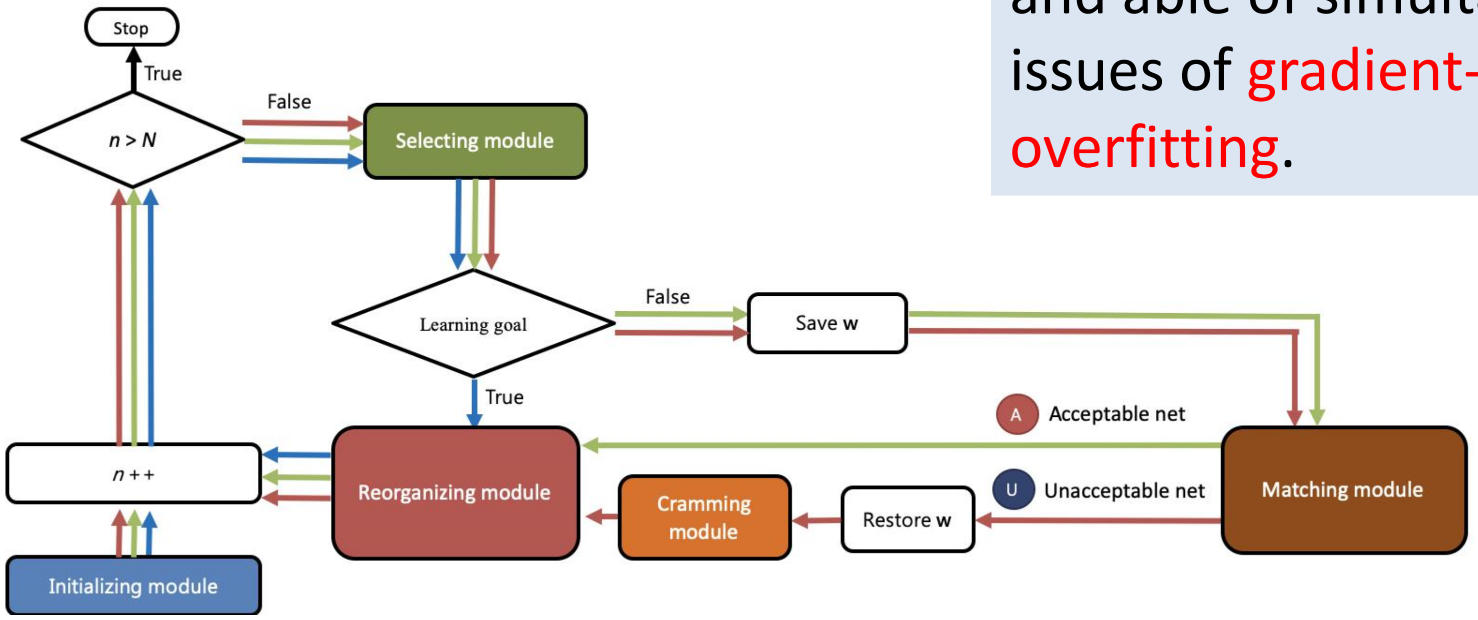
In each sliding window:

- The training block ($TrB_M$) consists of 159 week-instances.
- The testing block ($TeB_M$) consists of 4 week-instances.



# The proposed sequentially structuring (SS) mechanism

The SS mechanism is applicable to ASLFN and able of simultaneously coping the issues of gradient-vanishing and overfitting.



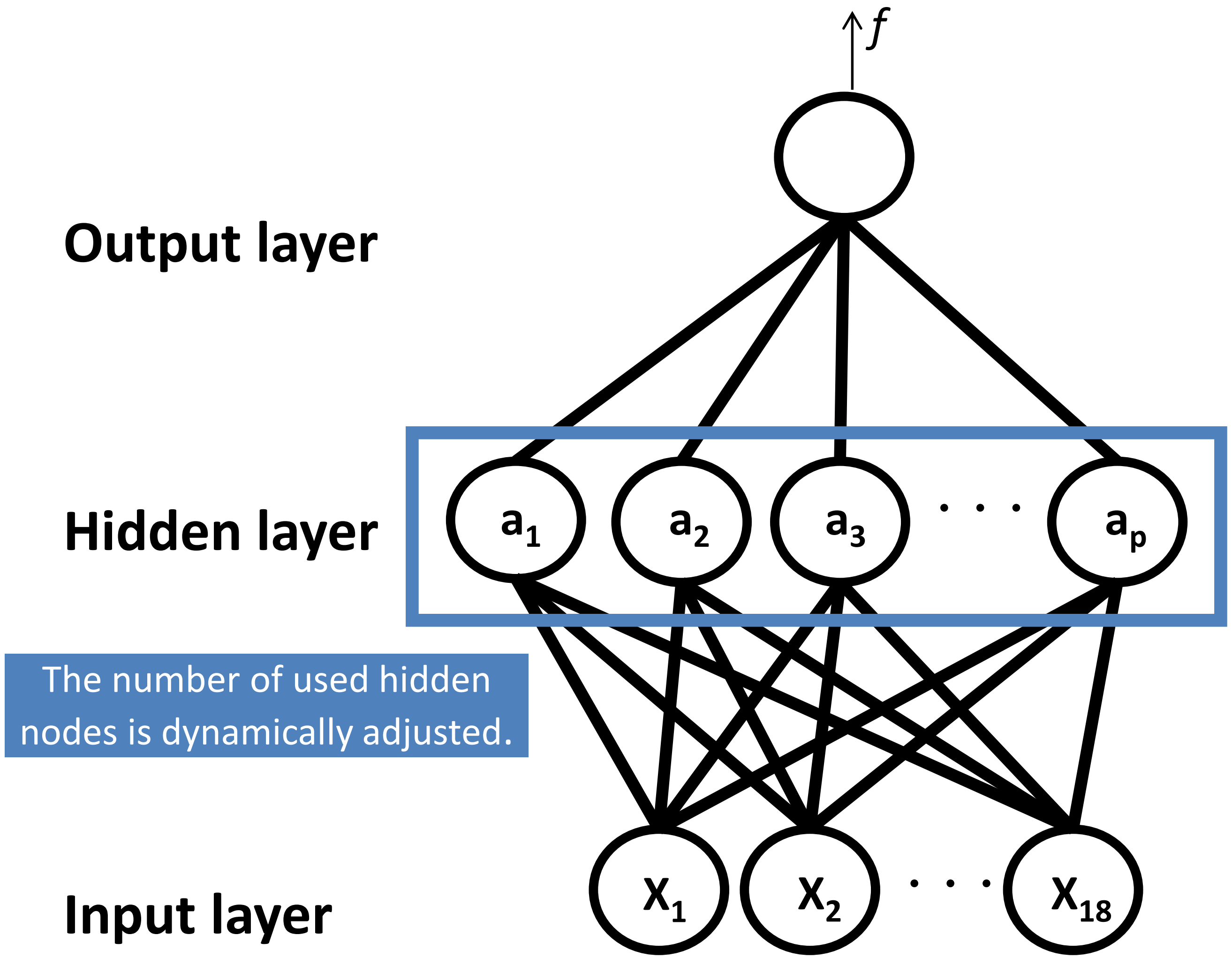

Sequentially structuring (SS) mechanism

$TrB_M$(with 159 week-instances)

$TeB_M$(with 4 week-instances)

Sliding window mechanism

# The ASLFN



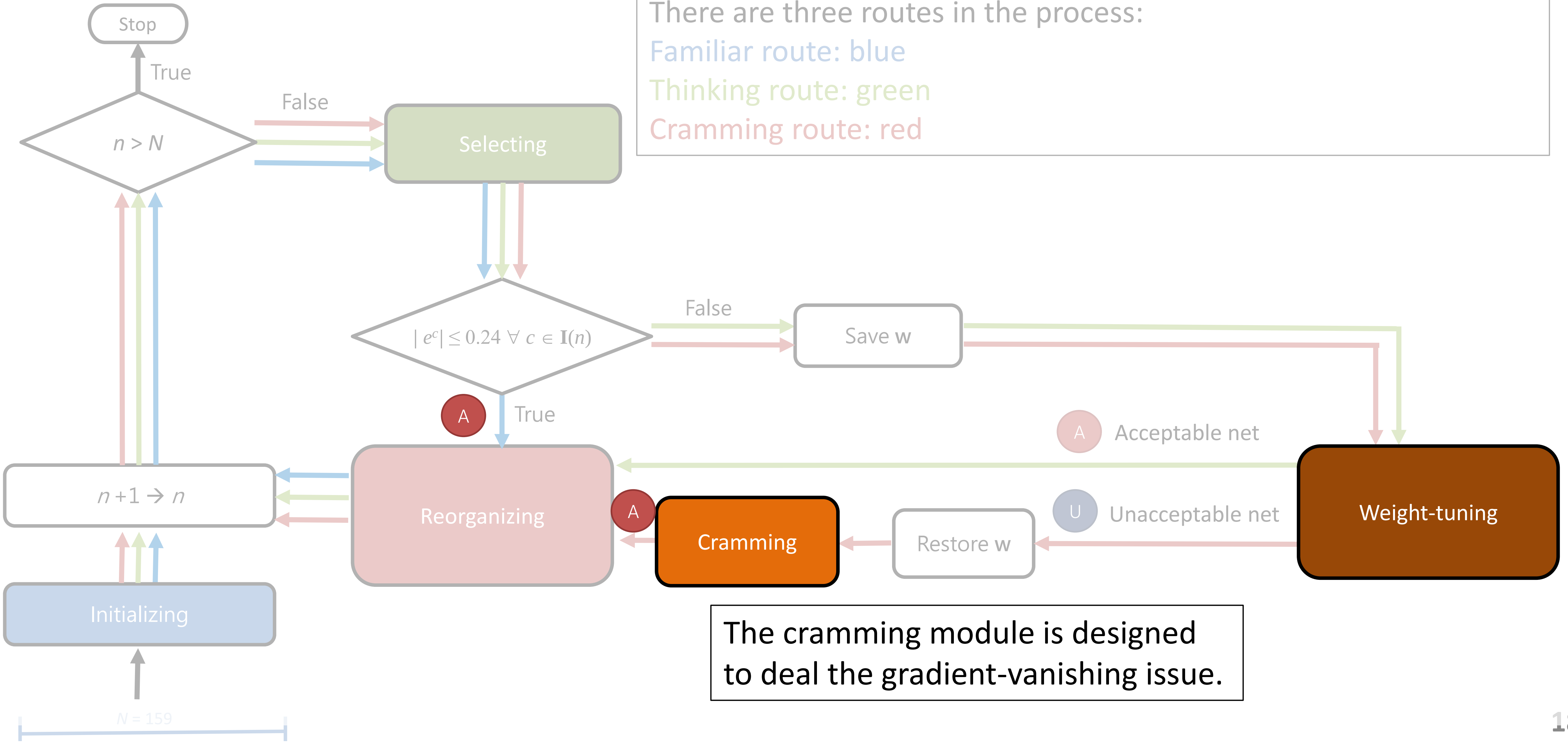

$$f(\mathbf{x}^c, \mathbf{w}) \equiv w_0^o + \sum_{i=1}^{p} w_i^o \, a_i^c$$

$$a_i^c \equiv ReLU\left( w_{i,0}^H + \sum_{j=1}^{m} w_{ij}^H x_j^c \right)$$

18 input nodes; $\mathbf{x}^c \in \mathbb{R}^{18}$

# The design concept of the SS mechanism

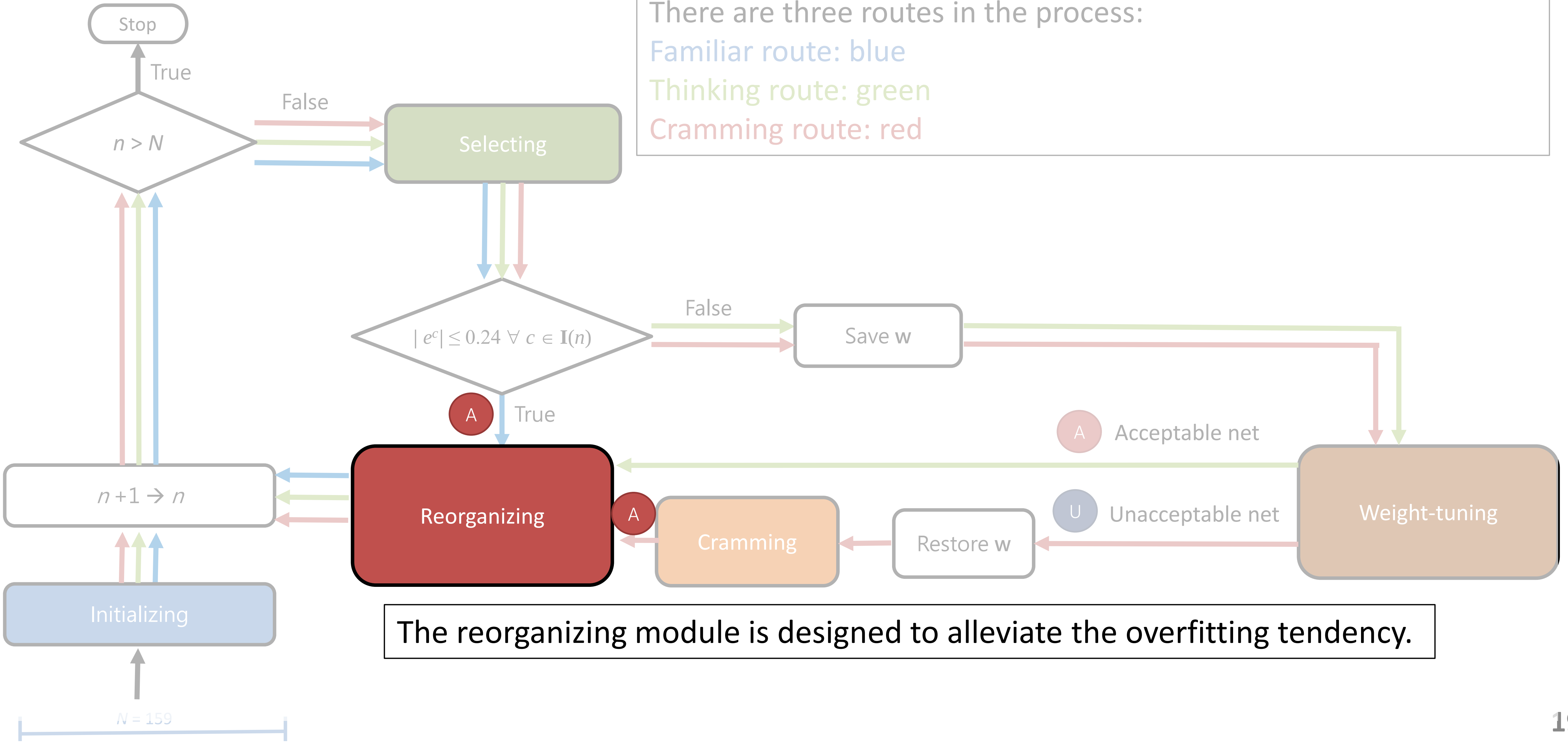
There are three routes in the process:
Familiar route: blue
Thinking route: green
Cramming route: red




The cramming module is designed to deal the gradient-vanishing issue.

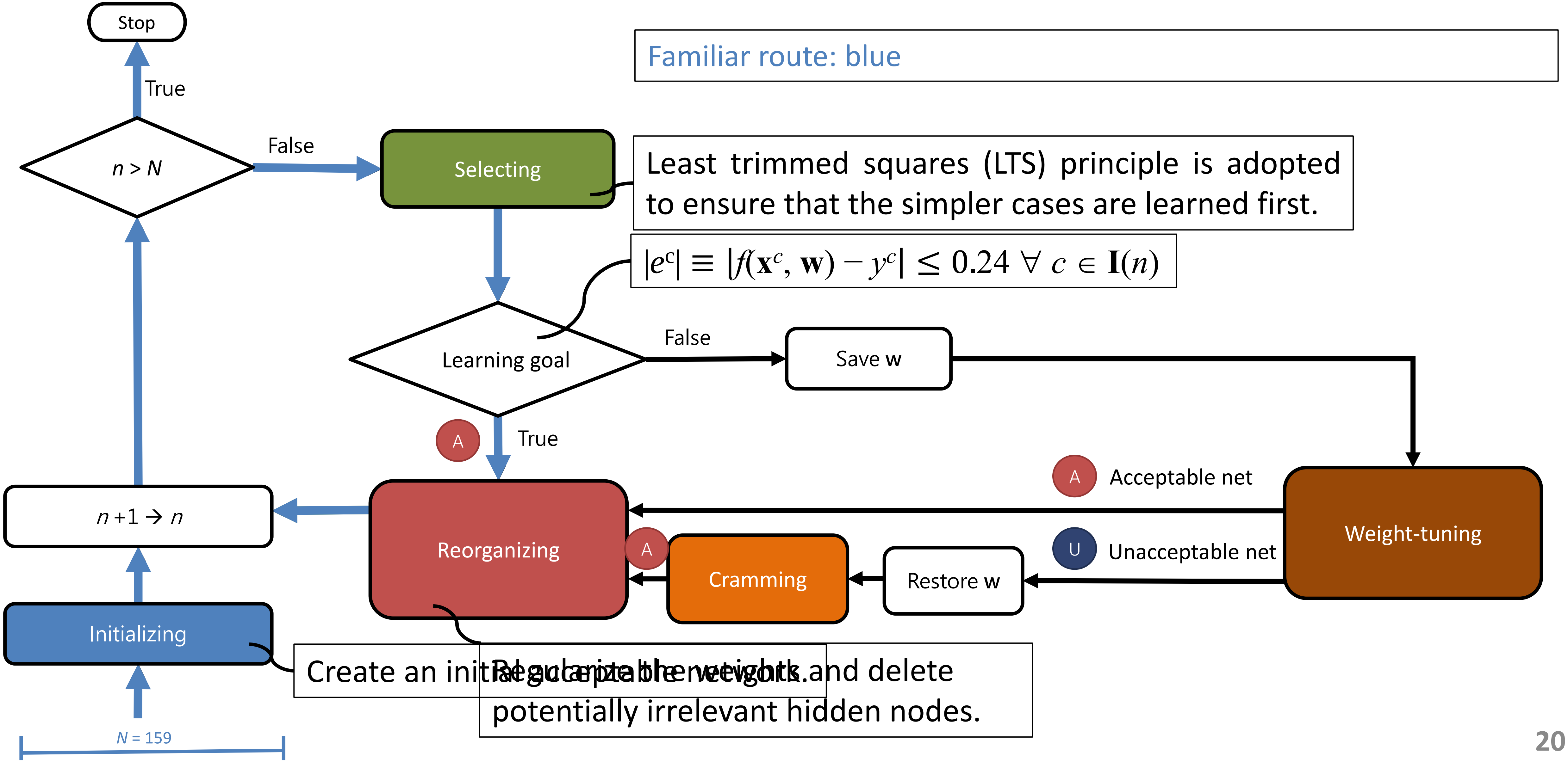
# The design concept of the SS mechanism

There are three routes in the process:
Familiar route: blue
Thinking route: green
Cramming route: red

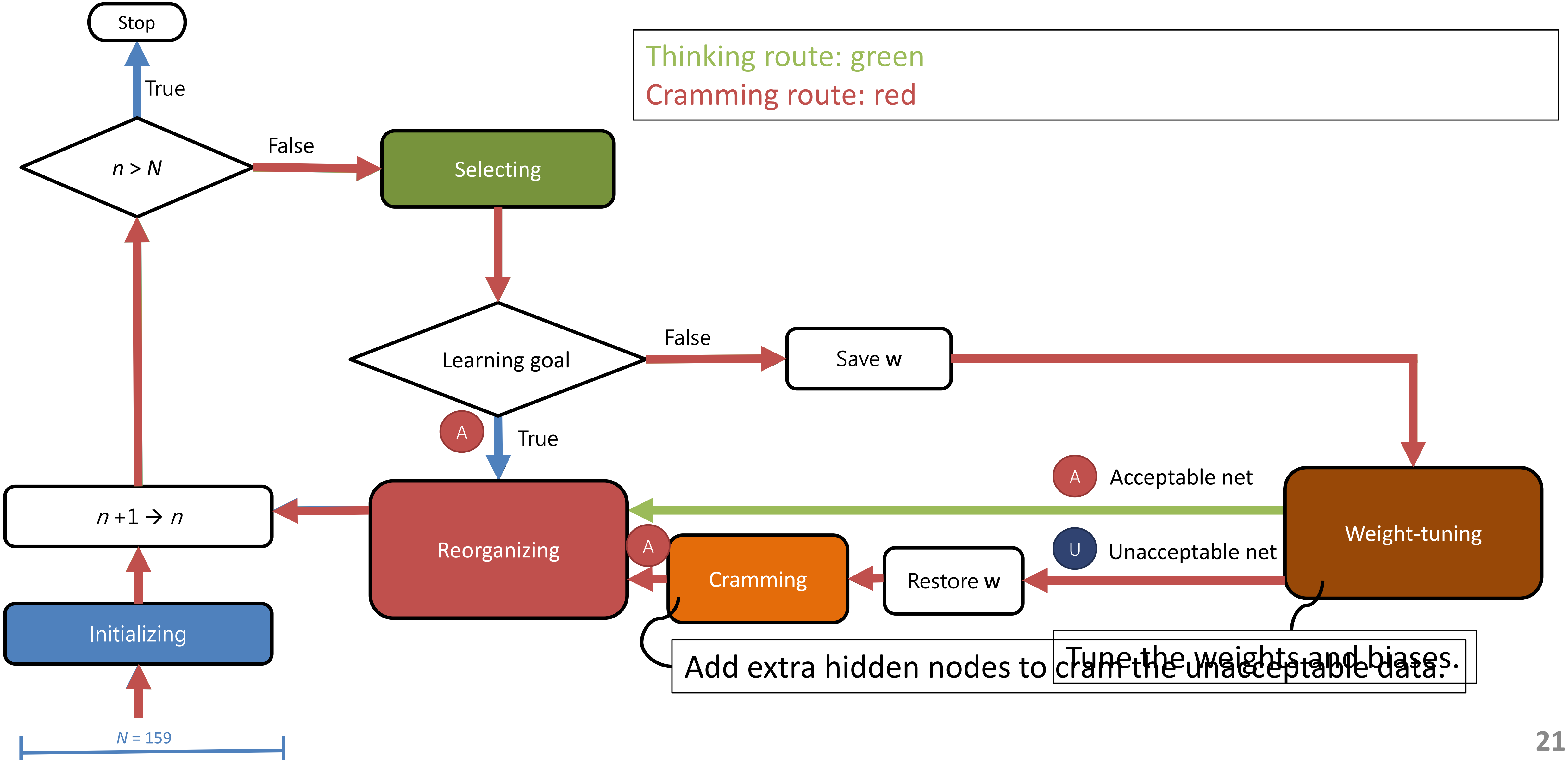



The reorganizing module is designed to alleviate the overfitting tendency.

# Sequentially structuring (SS) mechanism



# Sequentially structuring (SS) mechanism



# Validate and evaluate the AI system

- There is no mathematical proof to validate the proposed AI system.
- Instead, it is validated and evaluated with a real-world AI application problem.

**AI fundamental research**

- The corresponding learning process of the SS mechanism does take three alternative routes.
- The cramming module helps cope with the gradient-vanishing issue, while the reorganizing module helps alleviate the overfitting tendency.

**AI application research**

- The proposed AI system have better accuracy than other modelling tools, and the total training time is acceptable.

# Description of copper price data stream

- A total of 471 weekly copper prices of Yangtze River nonferrous metals from October 31, 2011 to December 21, 2020 are used.
- The four-week ahead forecasting is adopted in this study. Astudillo et al. (2020) state that medium-term (monthly) forecasts are the most practiced in many literatures compared to short-term (daily and weekly) forecasts, which are more complex but more helpful.

# Literature of independent variables

| Category | Independent variable |
|---|---|
| Energy source (Jiang & Adeli, 2005; Behmiri et al., 2015; Liu et al., 2017; Alameer et al., 2019) | Crude oil prices |
| Macroeconomics (Cologni & Manera, 2008; Orlowski, 2017; Alameer et al., 2019) | Inflation rates of US and China |
| Exchange rates (Chen, 2010; Wets & Rios, 2015; Ciner, 2017; Zheng et al., 2017; Alameer et al., 2019; Wang & Wang, 2019) | USD/CLP, USD/PEN, USD/CNY, USD/EURO |
| Related metals prices (Morales & Andreosso-O'Callaghan, 2011; Liu et al., 2017; Dehghani, 2018; Alameer et al., 2019) | The prices of gold, silver, nickel, aluminum, zinc, iron |

# Variable description

| variables | Description |
|---|---|
| $x_1^t$ | The weekly crude oil price of New York Mercantile Exchange in period $t$. |
| $x_2^t$ | The weekly copper spot price of Yangtze River nonferrous metals in period $t$. |
| $x_3^t$ | The weekly copper spot price of Yangtze River nonferrous metals in period $t$-1. |
| $x_4^t$ | The weekly copper spot price of Yangtze River nonferrous metals in period $t$-2. |
| $x_5^t$ | The weekly copper spot price of Yangtze River nonferrous metals in period $t$-3. |
| $x_6^t$ | The weekly copper spot price of London Metal Exchange in period $t$. |
| $x_7^t$ | The weekly gold spot price of FX Broker in period $t$. |
| $x_8^t$ | The weekly silver spot price of FX Broker in period $t$. |
| $x_9^t$ | The weekly nickel spot price of London Metal Exchange in period $t$. |
| $x_{10}^t$ | The weekly aluminum spot price of London Metal Exchange in period $t$. |
| $x_{11}^t$ | The weekly zinc spot price of London Metal Exchange in period $t$. |
| $x_{12}^t$ | The weekly iron spot price of London Metal Exchange in period $t$. |
| $x_{13}^t$ | Inflation rates of US in period $t$. |
| $x_{14}^t$ | Inflation rates of China in period $t$. |
| $x_{15}^t$ | The weekly USD/CLP dollar exchange rate in period $t$. |
| $x_{16}^t$ | The weekly USD/PEN dollar exchange rate in period $t$. |
| $x_{17}^t$ | The weekly USD/CNY dollar exchange rate in period $t$. |
| $x_{18}^t$ | The weekly USD/EURO dollar exchange rate in period $t$. |
| $y_1^t$ | The weekly copper spot price of Yangtze River nonferrous metals in period $t$+4. |

# Data preprocessing

All (training and test) data are normalized in the range [0, 1] by the min-max normalization method with Eq. (1).

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

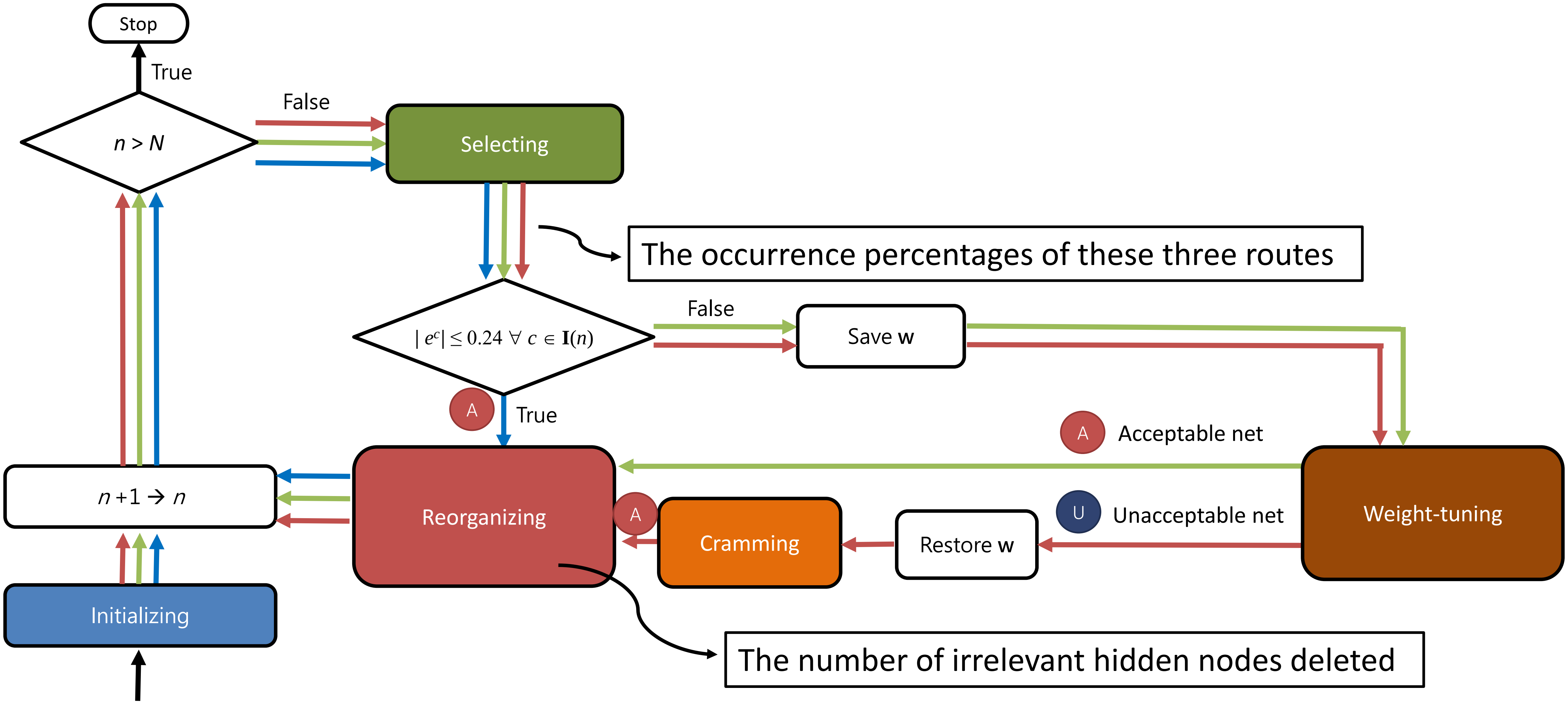
# Experimental design

- For the validation purpose, there are four versions of the proposed SS mechanism.
- They are different in the selecting module and/or the maximum number of iterations of the regularizing module.

| Version | The selecting module | Maximum number of iterations of the regularizing module |
|---|---|---|
| SW-PO-100 | PO[a] | 100 |
| SW-LTS-0 | LTS | 0 |
| SW-LTS-100 | LTS | 100 |
| SW-LTS-500 | LTS | 500 |

[a]The pre-order (PO) principle merely follows the original sequence to select training data.

# Validate the proposed SS mechanism



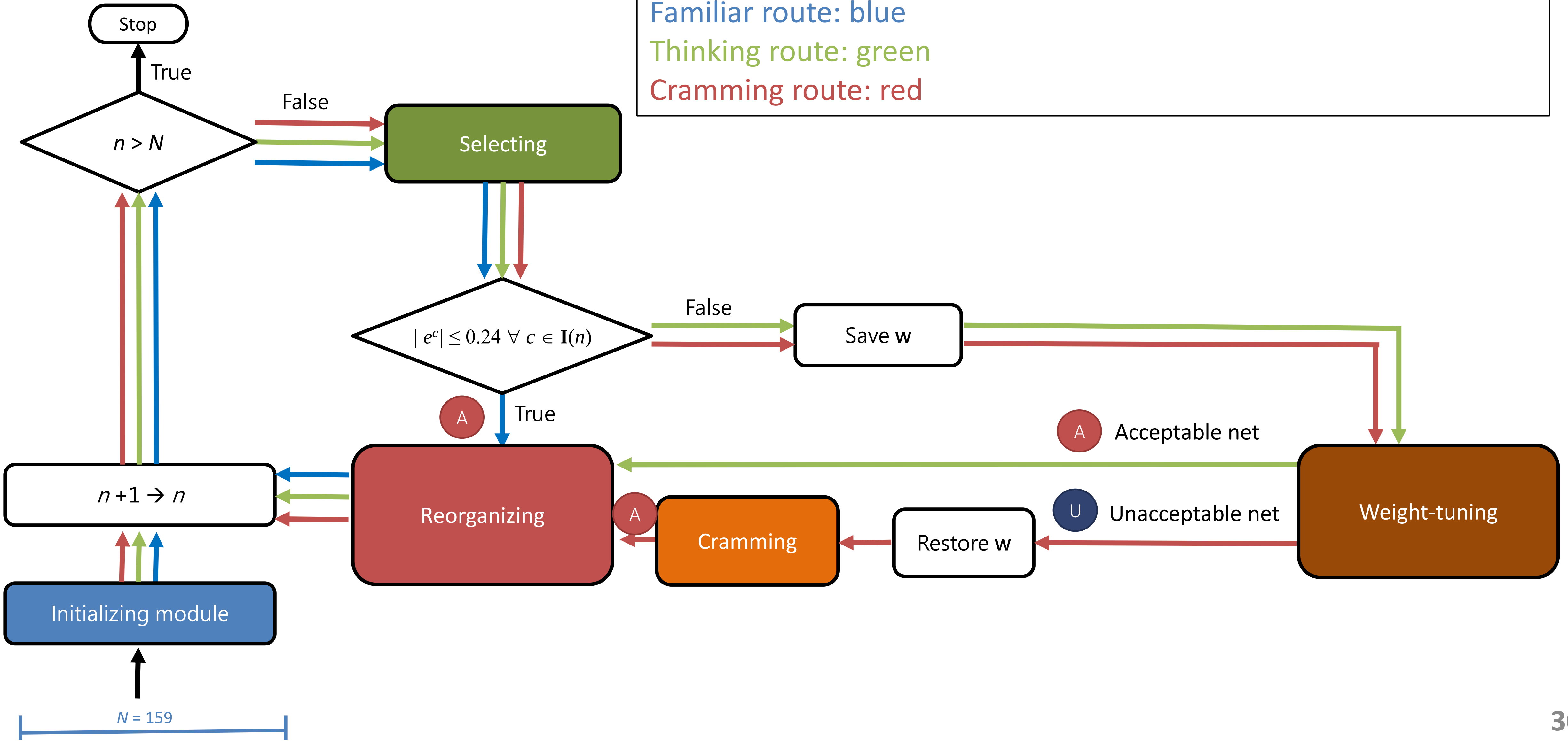
# The performance measurements

- The performance measurements:
  - mean absolute error (MAE)
  - mean absolute percentage error (MAPE)
  - root mean square error (RMSE)

- The baseline models are:
  - seasonal ARIMA model (SARIMA)
  - support vector regression (SVR)
  - single-hidden layer feedforward neural network (SLFN)
  - recurrent neural network (RNN)
  - long short-term memory (LSTM)
  - gated recurrent unit (GRU)

# Three learning routes

Familiar route: blue
Thinking route: green
Cramming route: red



# The occurrence percentages of each route regarding each version

### SW-PO-100

Familiar route > Cramming route > Thinking route

| | Familiar route | Thinking route | Cramming route |
|---|---|---|---|
| Mean | 88.14% | 3.60% | 8.26% |
| S.D. | 12.51% | 2.43% | 12.13% |
| Min | 30.82% | 0.00% | 0.00% |
| Max | 100.00% | 10.69% | 64.78% |

### SW-LTS-0

Familiar route > Thinking route > Cramming route

| | Familiar route | Thinking route | Cramming route |
|---|---|---|---|
| Mean | 84.41% | 11.33% | 4.26% |
| S.D. | 10.44% | 9.33% | 7.92% |
| Min | 47.80% | 0.63% | 0.00% |
| Max | 98.11% | 35.85% | 51.57% |

### SW-LTS-100

Familiar route > Cramming route > Thinking route

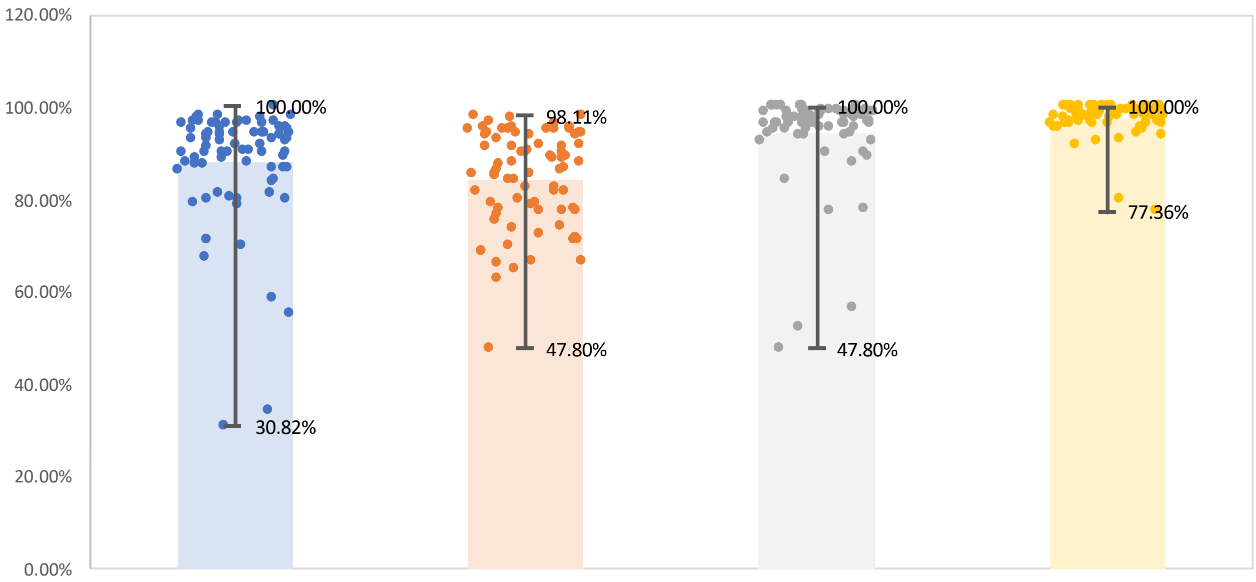
| | Familiar route | Thinking route | Cramming route |
|---|---|---|---|
| Mean | 94.56% | 1.31% | 4.13% |
| S.D. | 9.54% | 1.65% | 9.47% |
| Min | 47.80% | 0.00% | 0.00% |
| Max | 100.00% | 8.81% | 51.57% |

### SW-LTS-500

Familiar route > Cramming route > Thinking route

| | Familiar route | Thinking route | Cramming route |
|---|---|---|---|
| Mean | 97.46% | 0.79% | 1.75% |
| S.D. | 3.61% | 1.10% | 3.29% |
| Min | 77.36% | 0.00% | 0.00% |
| Max | 100.00% | 5.03% | 21.38% |

# The occurrence percentages of the *familiar route* regarding each version

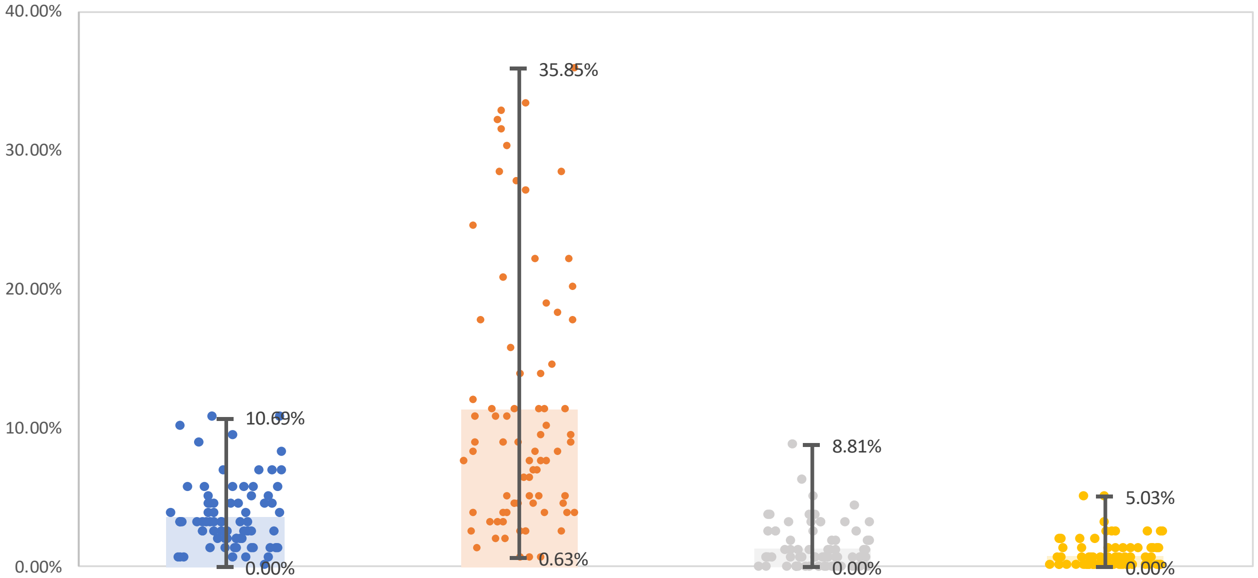



| | SW-PO-100 | SW-LTS-0 | SW-LTS-100 | SW-LTS-500 |
|---|---|---|---|---|
| Mean | 88.14% | 84.41% | 94.56% | 97.46% |
| S.D. | 12.51% | 10.44% | 9.54% | 3.61% |
| Min | 30.82% | 47.80% | 47.80% | 77.36% |
| Max | 100.00% | 98.11% | 100.00% | 100.00% |

**Finding:**

- The occurrence percentage of the *familiar route*: SW-LTS-500 > SW-LTS-100 > SW-PO-100 > SW-LTS-0.
- The versions with both the LTS principle and the regularizing module have a higher occurrence percentage of the *familiar route*.
- The more the maximum number of iterations of the regularizing module is, the higher the occurrence percentage of the *familiar route* is.

# The occurrence percentages of the *thinking route* regarding each version

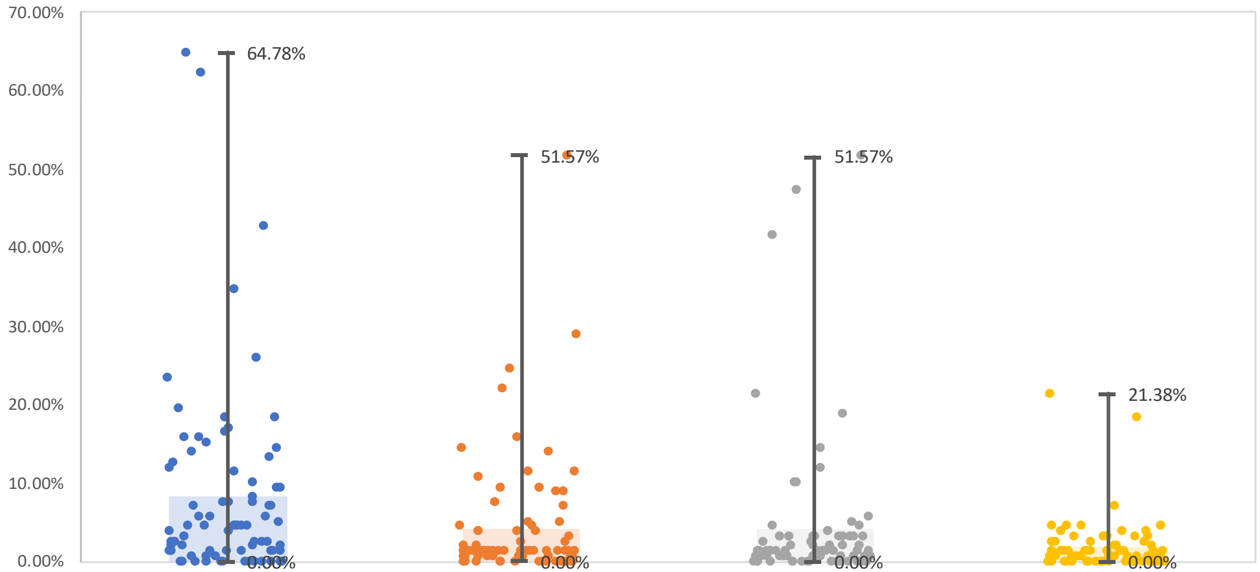



| | SW-PO-100 | SW-LTS-0 | SW-LTS-100 | SW-LTS-500 |
|---|---|---|---|---|
| Mean | 3.60% | 11.33% | 1.31% | 0.79% |
| S.D. | 2.43% | 9.33% | 1.65% | 1.10% |
| Min | 0.00% | 0.63% | 0.00% | 0.00% |
| Max | 10.69% | 35.85% | 8.81% | 5.03% |

**Finding:**

- The occurrence percentage of the *thinking route*: SW-LTS-0 > SW-PO-100 > SW-LTS-100 > SW-LTS-500.
- With the LTS principle, the higher the maximum number of iterations of the regularizing module is, the less the occurrence percentage of the *thinking route* is.

# The occurrence percentages of the *cramming route* regarding each version




| | SW-PO-100 | SW-LTS-0 | SW-LTS-100 | SW-LTS-500 |
|---|---|---|---|---|
| Mean | 8.26% | 4.26% | 4.13% | 1.75% |
| S.D. | 12.13% | 7.92% | 9.47% | 3.29% |
| Min | 0.00% | 0.00% | 0.00% | 0.00% |
| Max | 64.78% | 51.57% | 51.57% | 21.38% |

**Finding:**

- The occurrence percentage of the *cramming route*: SW-PO-100 > SW-LTS-0 > SW-LTS-100 > SW-LTS-500.
- The version with the PO principle has a highest occurrence percentage.
- With the LTS principle, the higher the maximum number of iterations of the regularizing module is set, the less the occurrence percentage of the *cramming route* is.

# The final number of adopted hidden nodes regarding each version

The final number of adopted hidden nodes
SW-PO-100 > SW-LTS-0 > SW-LTS-100 > SW-LTS-500

| | SW-PO-100 | SW-LTS-0 | SW-LTS-100 | SW-LTS-500 |
|---|---|---|---|---|
| Mean | 36.97 | 17.76 | 15.99 | 7.85 |
| S.D. | 53.77 | 34.97 | 31.67 | 14.94 |
| Min | 1 | 1 | 1 | 1 |
| Max | 286 | 245 | 197 | 102 |

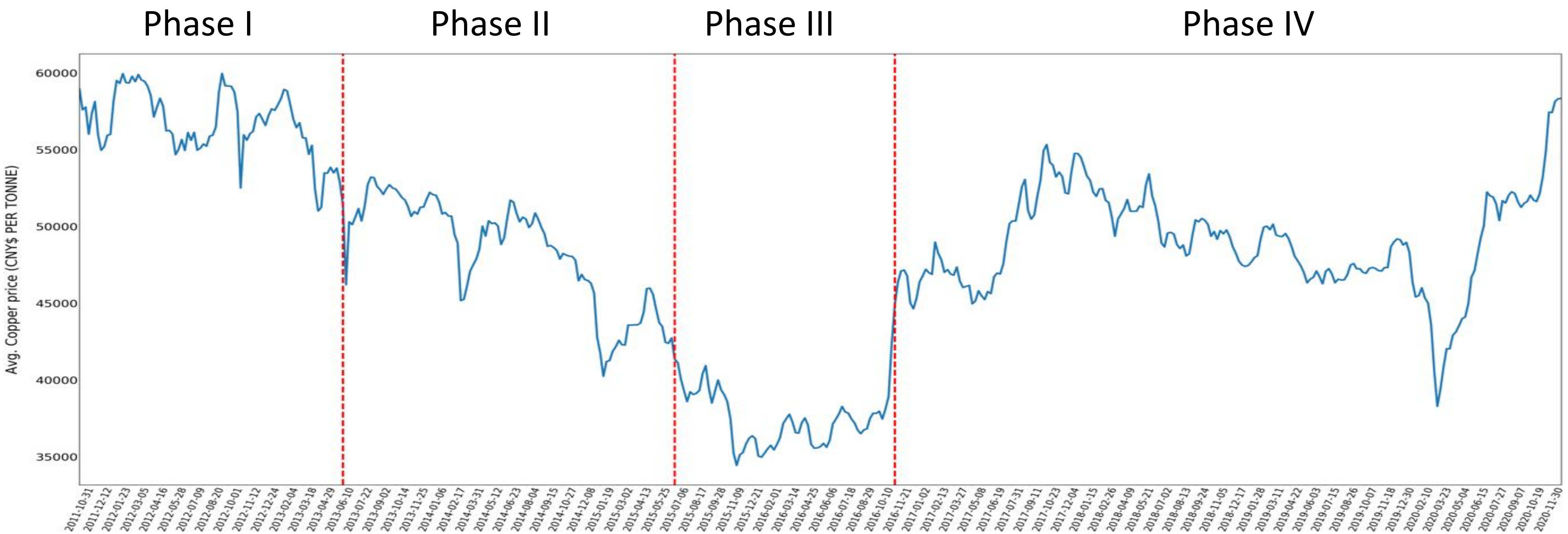
Largest

Smallest

The occurrence percentage of cramming route
SW-PO-100 > SW-LTS-0 > SW-LTS-100 > SW-LTS-500

| | SW-PO-100 | SW-LTS-0 | SW-LTS-100 | SW-LTS-500 |
|---|---|---|---|---|
| Mean | 8.26% | 4.26% | 4.13% | 1.75% |
| S.D. | 12.13% | 7.92% | 9.47% | 3.29% |
| Min | 0.00% | 0.00% | 0.00% | 0.00% |
| Max | 64.78% | 51.57% | 51.57% | 21.38% |

# The number of adopted hidden nodes in each phase regarding each version

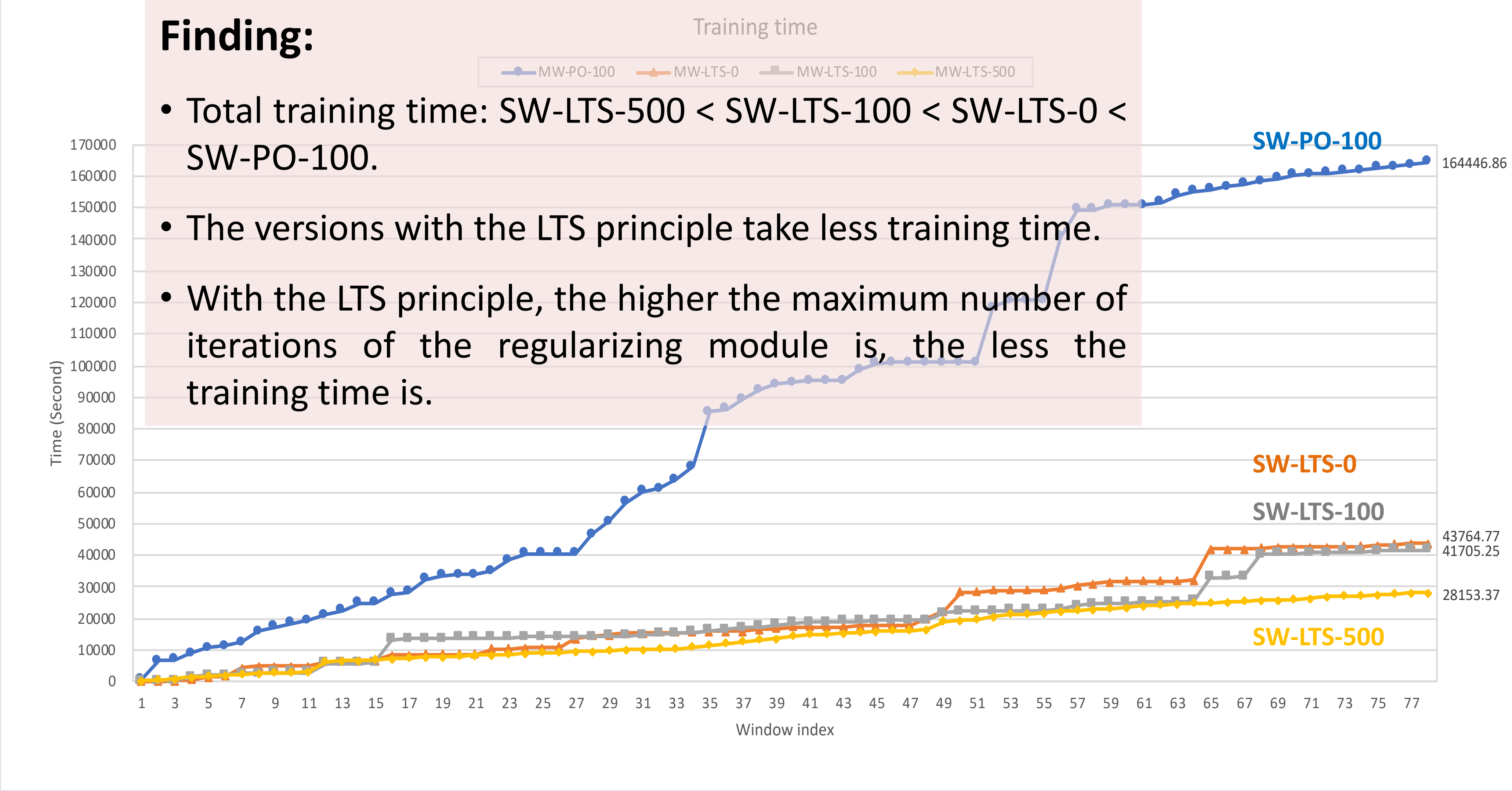



| | SW-PO-100 | SW-LTS-0 | SW-LTS-100 | SW-LTS-500 |
|---|---|---|---|---|
| Phase I (1st-21st window) | 40.00 | 19.45 | 23.70 | 9.80 |
| Phase II (22nd-48th window) | 38.48 | 11.81 | 8.44 | 5.33 |
| Phase III (49th-66th window) | 47.67 | 34.28 | 20.89 | 11.67 |
| Phase IV (67th-78th window) | 15.50 | 4.92 | 14.00 | 5.08 |

# Total training time

**Finding:**

- Total training time: SW-LTS-500 < SW-LTS-100 < SW-LTS-0 < SW-PO-100.
- The versions with the LTS principle take less training time.
- With the LTS principle, the higher the maximum number of iterations of the regularizing module is, the less the training time is.



# The average training time of a sliding window

(Units: second)

| | SW-PO-100 | SW-LTS-0 | SW-LTS-100 | SW-LTS-500 |
|---|---|---|---|---|
| Mean | 2108.29 | 561.09 | 534.68 | 360.94 |
| S.D. | 3650.71 | 1388.54 | 1457.91 | 443.63 |
| Min | 35.31 | 10.37 | 45.29 | 161.73 |
| Max | 19790.97 | 9912.40 | 7685.57 | 3150.11 |

**Finding:**

- Average training time of a sliding window: SW-LTS-500 < SW-LTS-100 < SW-LTS-0 < SW-PO-100.
- The versions with the LTS take less training time.
- Under the LTS principle, the higher the maximum number of iterations of the regularizing module is, the less the training time is.

# Relationship between the cramming route and the average training time of a sliding window

The average training time

SW-PO-100 > SW-LTS-0 > SW-LTS-100 > SW-LTS-500

| | SW-PO-100 | SW-LTS-0 | SW-LTS-100 | SW-LTS-500 |
|---|---|---|---|---|
| Mean | 2108.29 | 561.09 | 534.68 | 360.94 |
| S.D. | 3650.71 | 1388.54 | 1457.91 | 442.63 |
| Min | 35.31 | 10.37 | 45.29 | 161.73 |
| Max | 19790.97 | 9912.40 | 7685.57 | 3150.11 |

Largest

Smallest

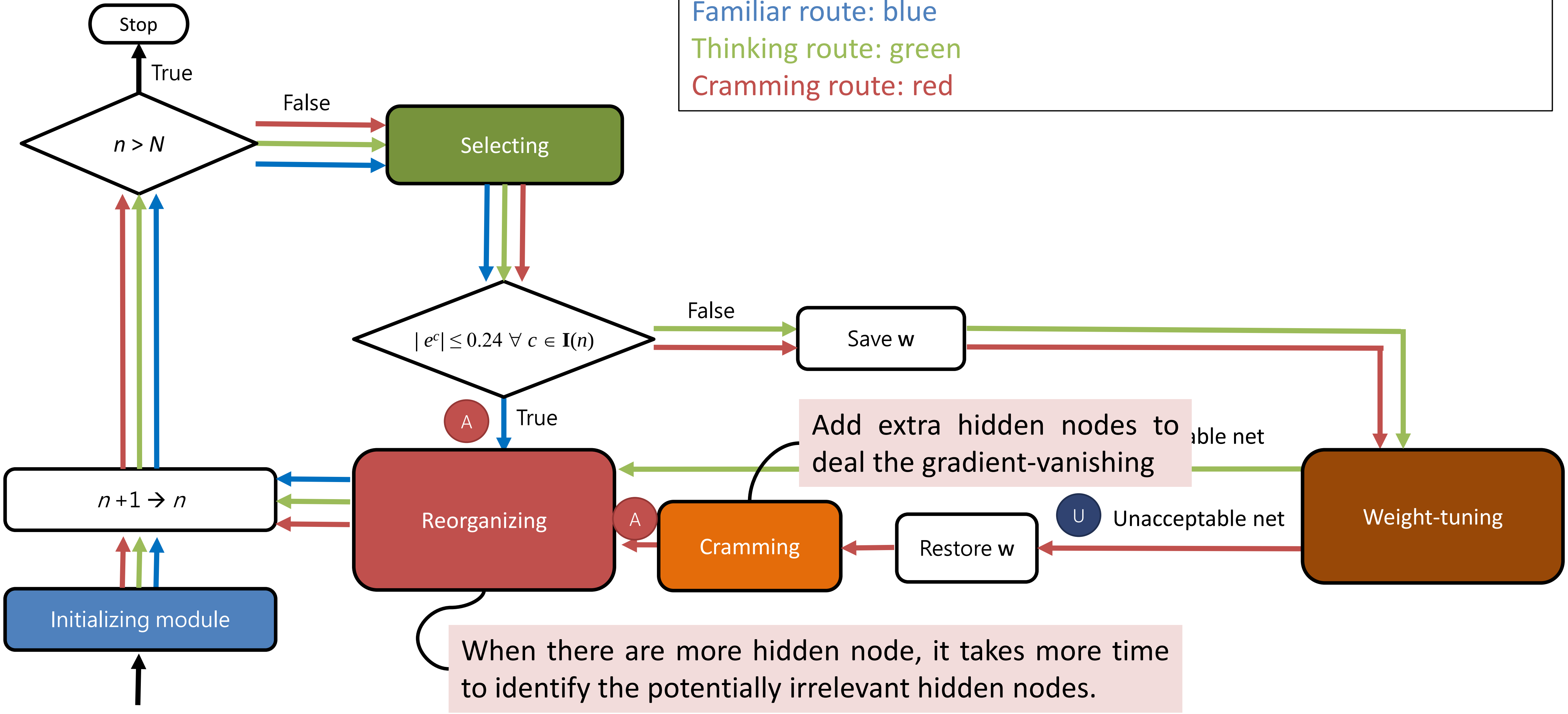
The occurrence percentage of the cramming route

SW-PO-100 > SW-LTS-0 > SW-LTS-100 > SW-LTS-500

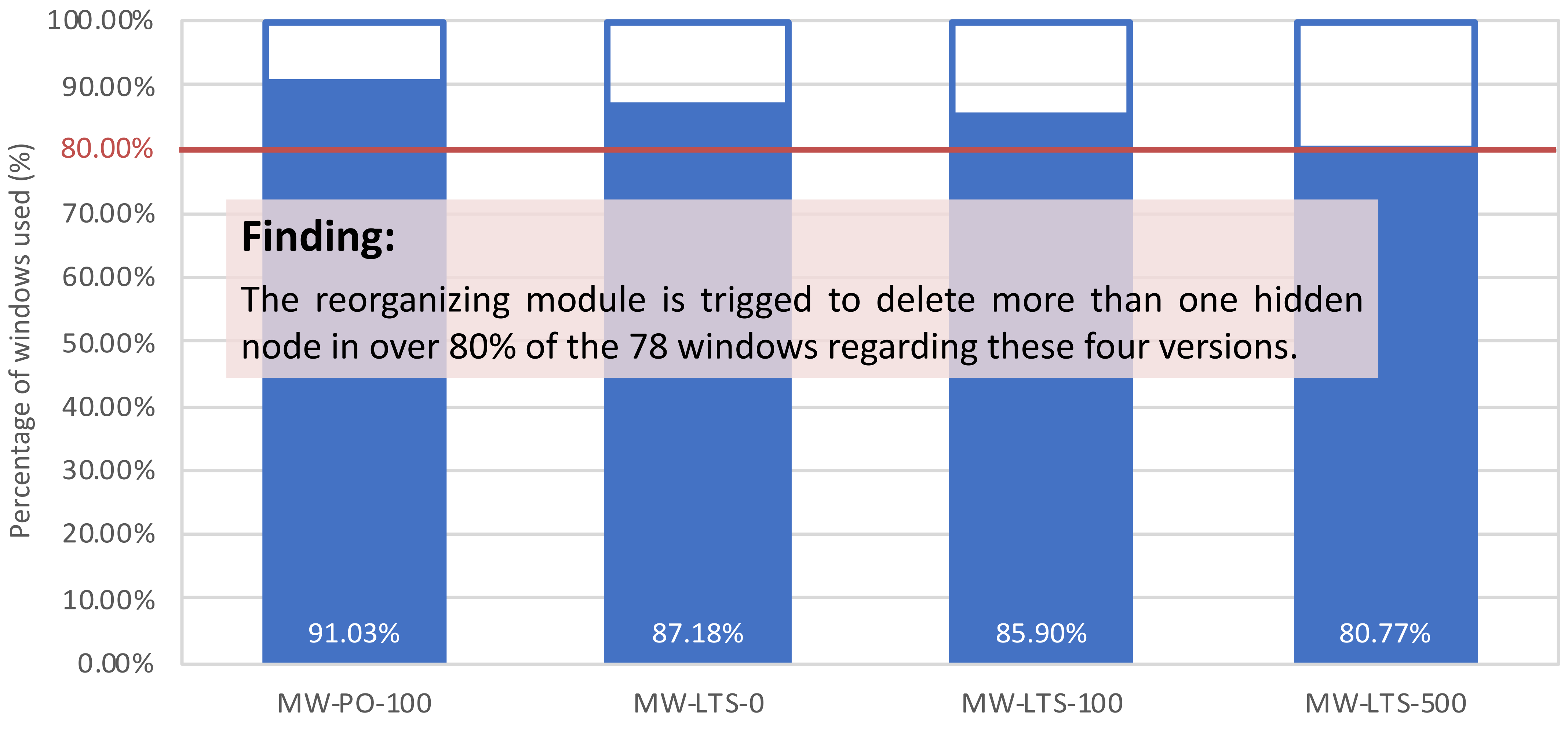
| | SW-PO-100 | SW-LTS-0 | SW-LTS-100 | SW-LTS-500 |
|---|---|---|---|---|
| Mean | 8.26% | 4.26% | 4.13% | 1.75% |
| S.D. | 12.13% | 7.92% | 9.47% | 3.29% |
| Min | 0.00% | 0.00% | 0.00% | 0.00% |
| Max | 64.78% | 51.57% | 51.57% | 21.38% |

# The training time

Familiar route: blue
Thinking route: green
Cramming route: red



# Validation for the reorganizing module




**Finding:**

The reorganizing module is trigged to delete more than one hidden node in over 80% of the 78 windows regarding these four versions.

# The number of deleted hidden nodes

| | SW-PO-100 | SW-LTS-0 | SW-LTS-100 | SW-LTS-500 |
|---|---|---|---|---|
| Mean | 39.01 | 18.94 | 16.31 | 8.17 |
| S.D. | 54.72 | 35.44 | 31.59 | 15.07 |
| Min | 0 | 0 | 0 | 0 |
| Max | 300 | 250 | 196 | 103 |

**Finding:**

- In a sliding window, on average at least 8 hidden nodes are deleted regarding these four versions.
- The version with the PO principle deletes more hidden nodes than the versions with the LTS principle.
- The higher the maximum number of iterations of the regularizing module is, the less the amount of deleted hidden nodes is.

# Relationship between the number of adopted hidden nodes and the number of deleted hidden nodes

The number of adopted hidden node

SW-PO-100 > SW-LTS-0 > SW-LTS-100 > SW-LTS-500

| | SW-PO-100 | SW-LTS-0 | SW-LTS-100 | SW-LTS-500 |
|---|---|---|---|---|
| Mean | 36.97 | 17.76 | 15.99 | $\boxed{7.85}$ |
| S.D. | 53.77 | 34.97 | 31.67 | 14.94 |
| Min | 1 | 1 | 1 | 1 |
| Max | 286 | 245 | 197 | 102 |

The number of deleted hidden nodes

SW-PO-100 > SW-LTS-0 > SW-LTS-100 > SW-LTS-500

| | SW-PO-100 | SW-LTS-0 | SW-LTS-100 | SW-LTS-500 |
|---|---|---|---|---|
| Mean | 39.01 | 18.94 | 16.31 | $\boxed{8.17}$ |
| S.D. | 54.72 | 35.44 | 31.59 | 15.07 |
| Min | 0 | 0 | 0 | 0 |
| Max | 300 | 250 | 196 | 103 |

# Validate the LTS principle

| Models | Training data | | | Test data | | |
|---|---|---|---|---|---|---|
| | MAE | MAPE | RMSE | MAE | MAPE | RMSE |
| SW-PO-100 | 1379.93 | 3.02 | 1824.07 | 3305.56 | 7.40 | 3435.47 |
| SW-LTS-100 | 1153.27 | 2.52 | 1510.34 | 2427.44 | 5.50 | 2562.26 |

# Validate the regularizing module

| Models | Training data | | | Test data | | |
|---|---|---|---|---|---|---|
| | MAE | MAPE | RMSE | MAE | MAPE | RMSE |
| SW-LTS-0 | 1820.24 | 4.02 | 2314.32 | 3271.44 | 7.43 | 3450.17 |
| SW-LTS-100 | 1153.27 | 2.52 | 1510.34 | 2427.44 | 5.50 | 2562.26 |
| **SW-LTS-500** | **1112.11** | **2.43** | **1453.04** | **1923.63** | **4.19** | **2071.42** |

# Validate the proposed AI system

- SLFN
- SW-LTS-500 with the sliding window mechanism and the SS mechanism
- SS-SLFN with only the SS mechanism
- SW-SLFN with only the sliding window mechanism

| | | With sliding window mechanism | |
|---|---|---|---|
| | | Yes | No |
| With SS mechanism | Yes | SW-LTS-500 | SS-SLFN |
| | No | SW-SLFN | SLFN |

# Validate the proposed AI system

| Model | MAE | MAPE | RMSE |
|---|---|---|---|
| SW-LTS-500 | 1576.39 | 3.33 | 2080.75 |
| SS-SLFN | 2361.18 | 4.87 | 3367.10 |
| SW-SLFN | 2436.76 | 4.99 | 3270.72 |
| SLFN | 3649.32 | 7.77 | 4570.24 |

**Finding:**

- From the comparison of SS-SLFN and SLFN, the SS mechanism helps reduce the prediction error.
- From the comparison of SW-SLFN and SLFN, the sliding window mechanism also helps reduce the prediction error.
- The SW-LTS-500 has the best performance on forecasting the copper price.

# Evaluate the proposed AI system with six popular modelling methods in the literature

| Model | MAE | MAPE | RMSE |
|---|---|---|---|
| SW-LTS-500 | 1576.39 | 3.33 | 2080.75 |
| SARIMA | 2873.98 | 6.02 | 3764.86 |
| SLFN | 3649.32 | 7.77 | 4570.24 |
| SVR | 3359.37 | 6.85 | 4318.43 |
| RNN | 2646.85 | 5.50 | 3686.23 |
| LSTM | 2584.56 | 5.46 | 3319.30 |
| GRU | 3354.81 | 6.87 | 4228.00 |

**Finding:**

- The results show that the proposed SW-LTS-500 has the best MAE, MAPE, and RMSE performances.

# Summary

- This study derives an AI system that (1) can deal with the copper price data stream with structural changes, and (2) contains a learning mechanism applicable to ASLFN and able of simultaneously coping with the issues of gradient-vanishing and overfitting.

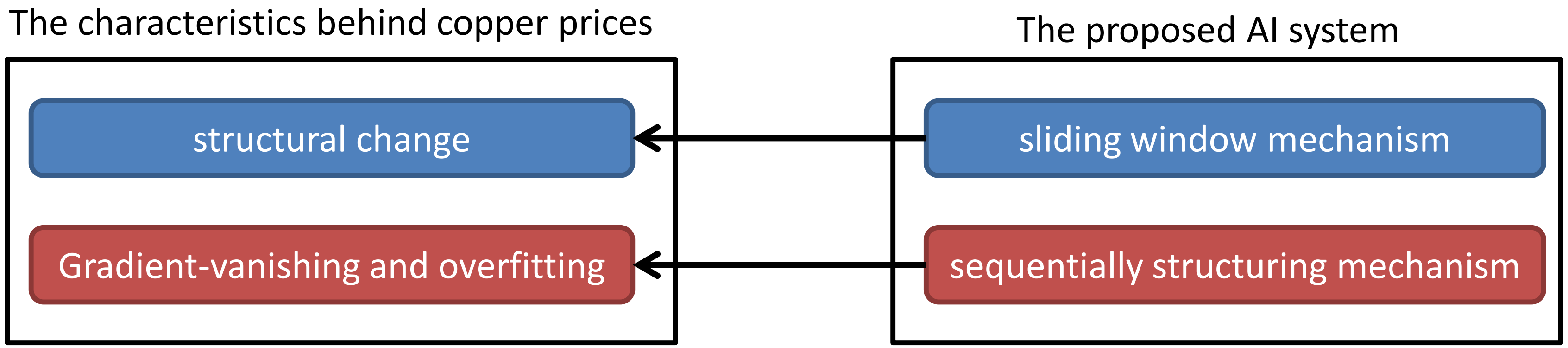
- The experiment results confirm that the corresponding learning process of the SS mechanism does take the proposed three alternative routes. It seems that all modules designed in the SS mechanism are effective.

- The SS mechanism helps cope with the issues of gradient-vanishing and overfitting.

- The proposed AI system has a better forecasting performance than the modelling tools of SARIMA, SLFN, SVR, RNN, LSTM ,and GRU.

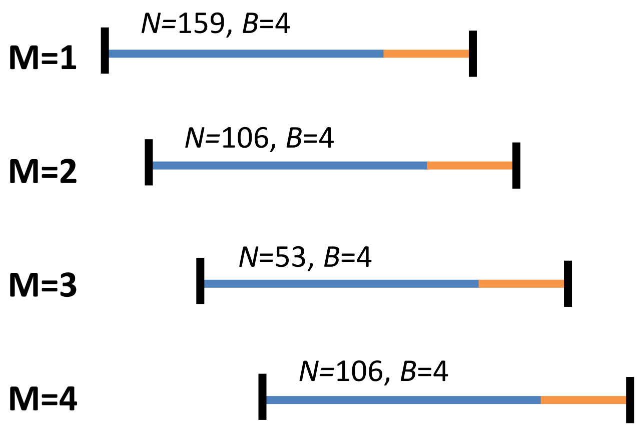
# Contributions

- Theoretical contributions

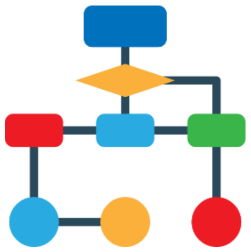


# Future work

## Modelling and learning



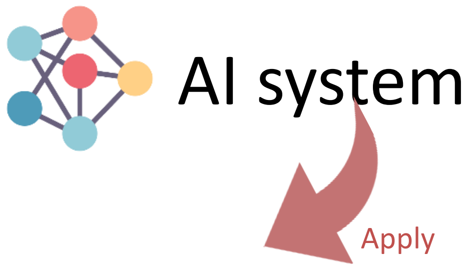

Adaptive window size



More effective arrangement for
- cramming module
- reorganizing module

Fine-tune modules

## Application




Other data streams similar with the copper price.

Apply to other data streams

# Thanks for listening.

# Q&A