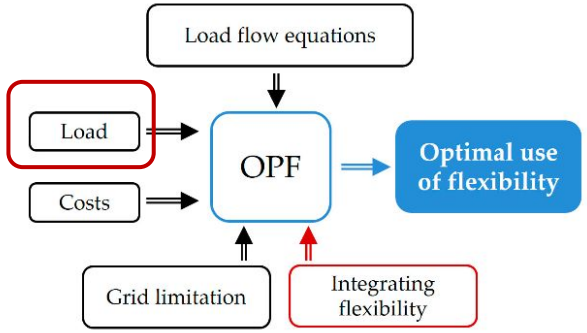
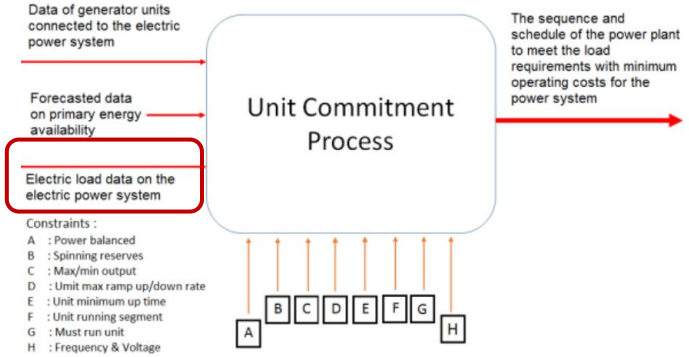
Deep Learning Based Load Forecasting for Electric Power Systems

Why we need load forecasting

- The supply and demand must be balanced in real time.

- Input to tasks such as Optimal power flow, unit commitment, etc.





A 1% reduction in forecasting error of a 10 GW utility can save up to $1.6 million annually.

Hobbs, Benjamin F., et al. "Analysis of the value for unit commitment of improved load forecasts." IEEE Transactions on Power Systems. 1999.
https://www.global.toshiba/content/dam/toshiba/migration/energysolution/ess/en/transmission/image/03-02-002.jpg
Steinle, Sina, et al. "Time-dependent flexibility potential of heat pump systems for smart energy system operation." Energies. 2020.
Rendroyoko, Ignatius, et al. "Integration method of unit commitment using PL-GA binary dispatch algorithm for intermittent RES in isolated microgrids system." International Journal on Electrical Engineering and Informatics . 2021.

Problem statement

Given the historical load and relevant features, learn a forecasting model to predict the future load demand:

$$\mathbf{y}_f = F(\mathbf{y}_h, \mathbf{x}_h^1, \cdots, \mathbf{x}_h^n, \mathbf{x}_f^1, \cdots, \mathbf{x}_f^n)$$

- $\mathbf{y}$ Load
- $\mathbf{x}^i$ Features such as temperature, holiday, etc.
- f Future value
- h Historical value

Short term forecasting (1 hour to 1 week)

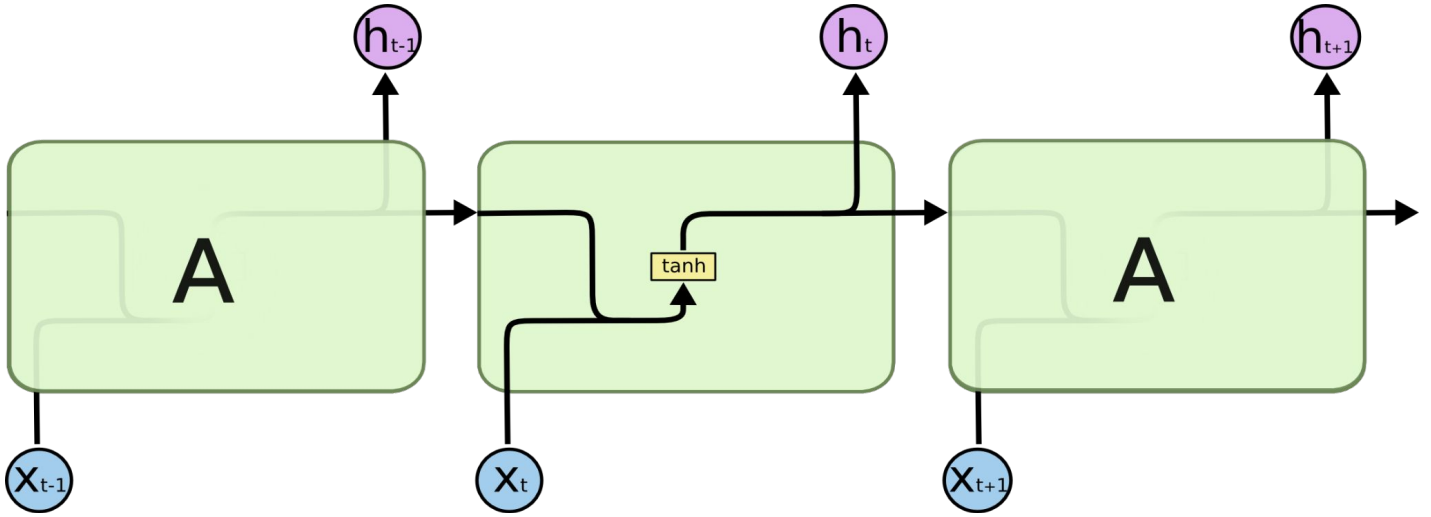
Provides the basis for economic dispatch of generation resources, market planning and energy trading, demand response programs, etc.

Prior work

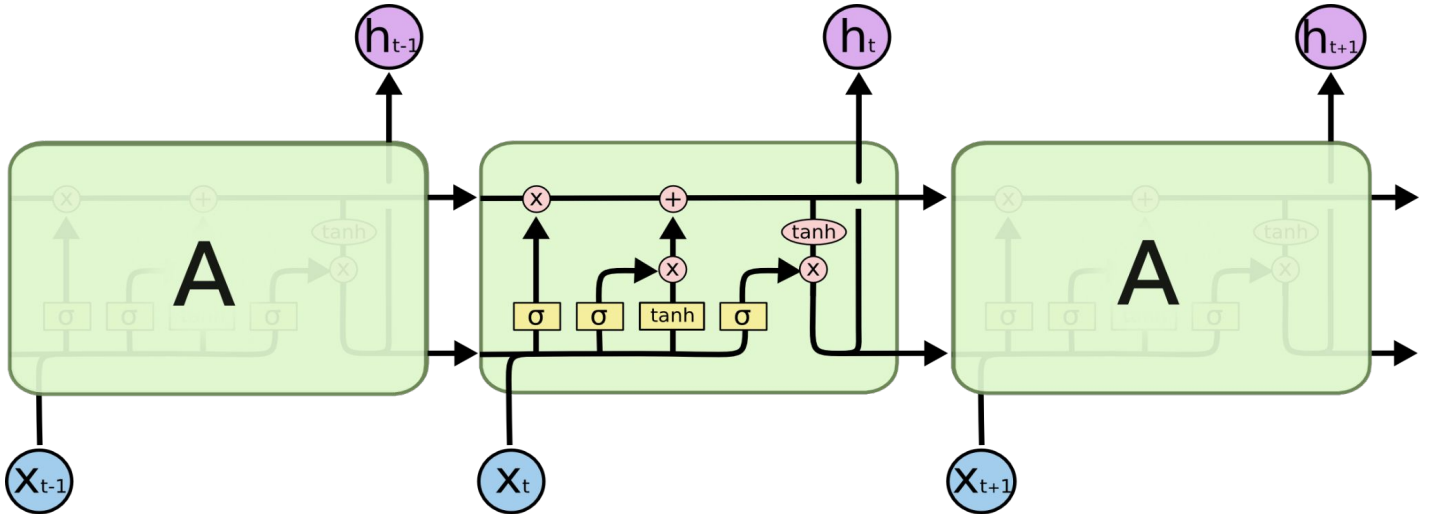
- Time series analysis
  - Autoregressive moving average (ARMA) [Chen et al. (1995)]
  - Autoregressive integrated moving average (ARIMA) [Juberias et al. (1999)]
- Classical machine learning algorithms
  - Support vector regression (SVR) [Ceperic et al. (2013)]
  - Random Forest (RF) [Dudek et al. (2015)]
  - Gradient boosting machine (GBM) [Taieb et al. (2014)]
- Deep learning models
  - Convolutional neural network (CNN) [Amarasinghe et al. (2017)]
  - Long short-term memory (LSTM) [Zheng et al. (2017)]
  - Deep residual network [Chen et al. (2018)]

Long short-term memory (LSTM)

- RNN



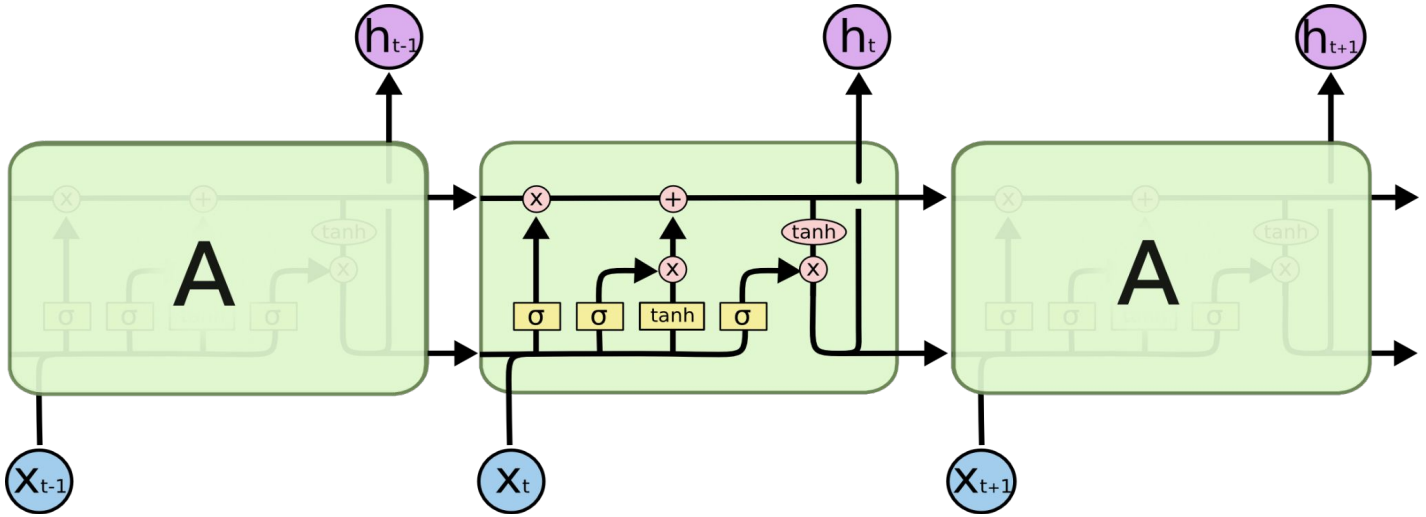
- LSTM



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long short-term memory (LSTM)

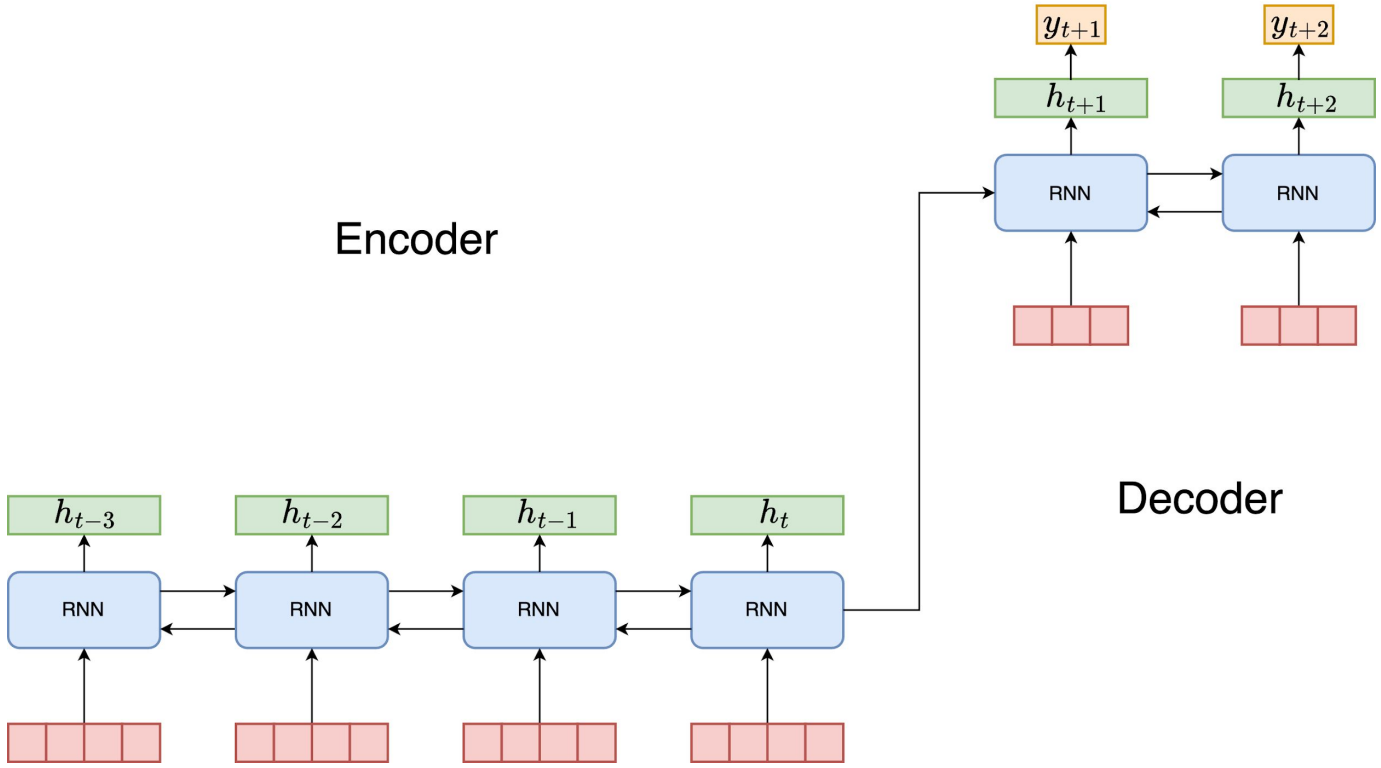
- LSTMs are specifically designed to address the long-term dependency problem in traditional RNNs.
- The gate structures (input, output, and forget gates) help in learning what to retain and what to discard.



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Encoder-decoder structure

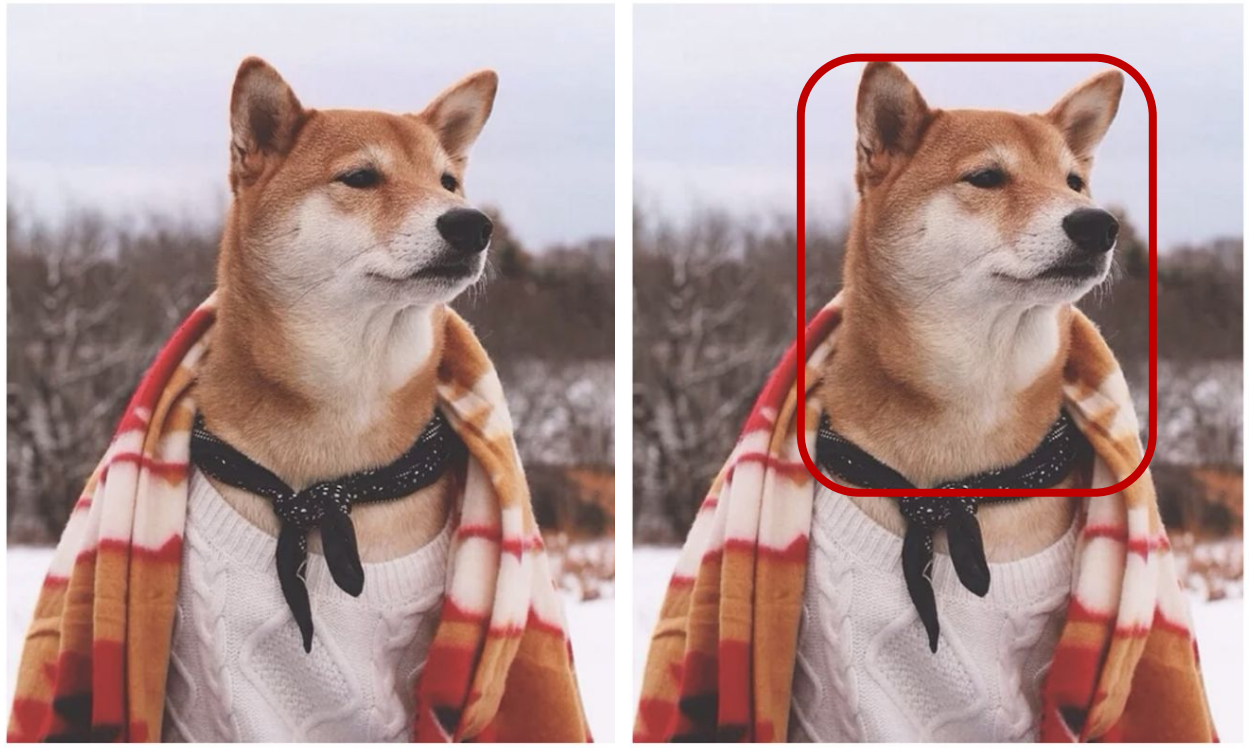
- Encoder transforms the input into a context-rich representation, capturing the essence of the input data.
- Decoder generate accurate and relevant output.



Attention

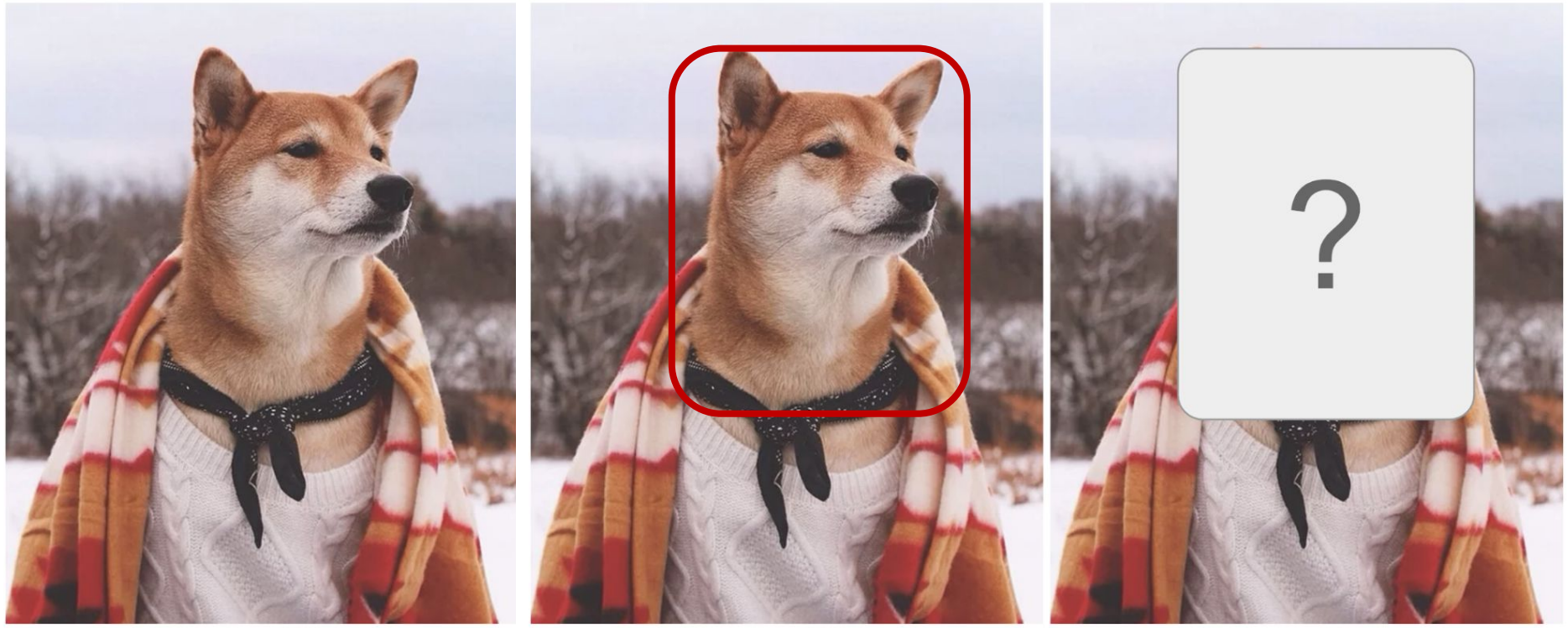
https://lilianweng.github.io/posts/2018-06-24-attention/

Attention



https://lilianweng.github.io/posts/2018-06-24-attention/

Attention



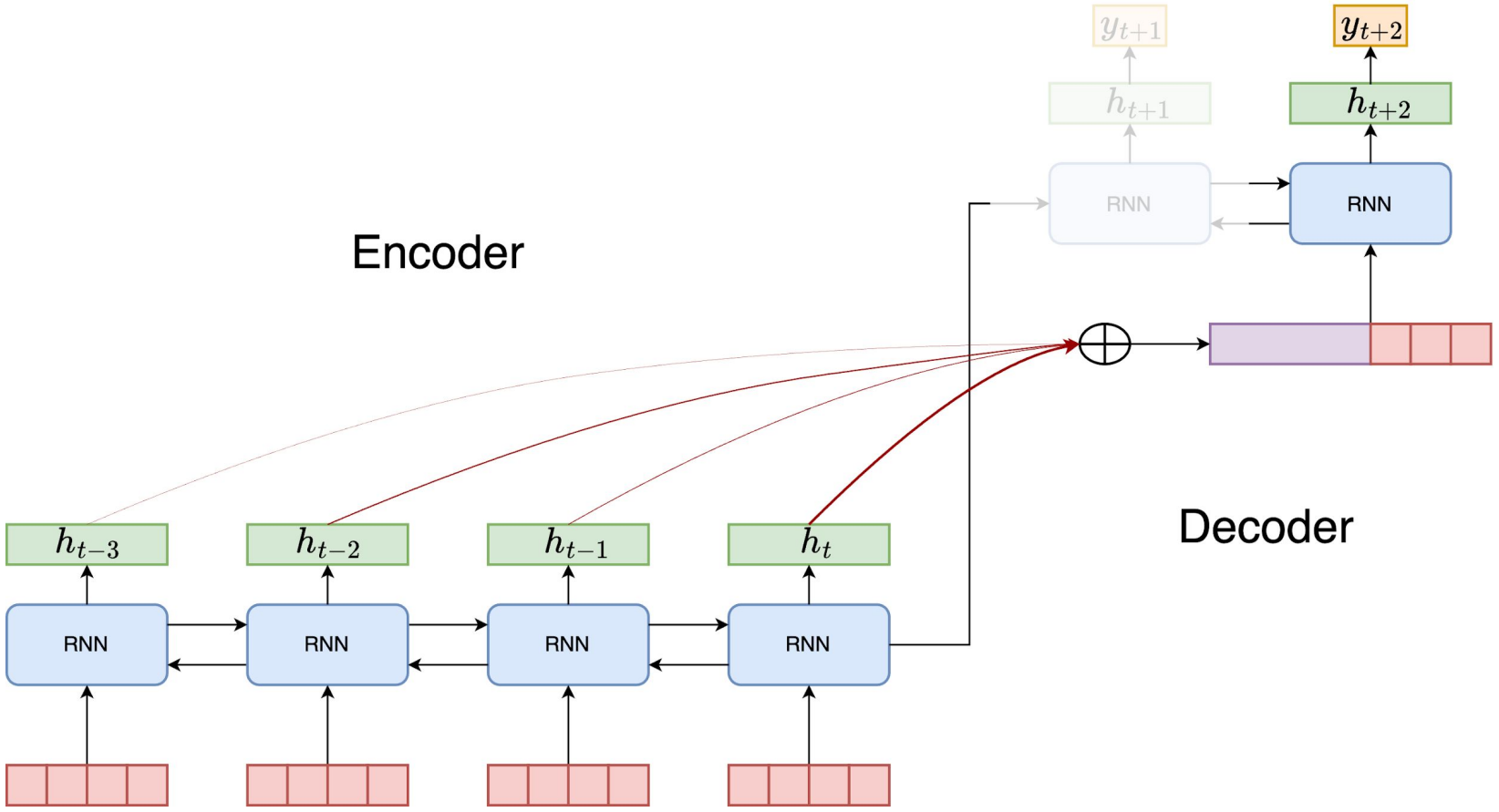
https://lilianweng.github.io/posts/2018-06-24-attention/

Attention

https://lilianweng.github.io/posts/2018-06-24-attention/

Attention for time series forecasting

- Focus on different parts of the input when generating the output.



Attention for short term load forecasting

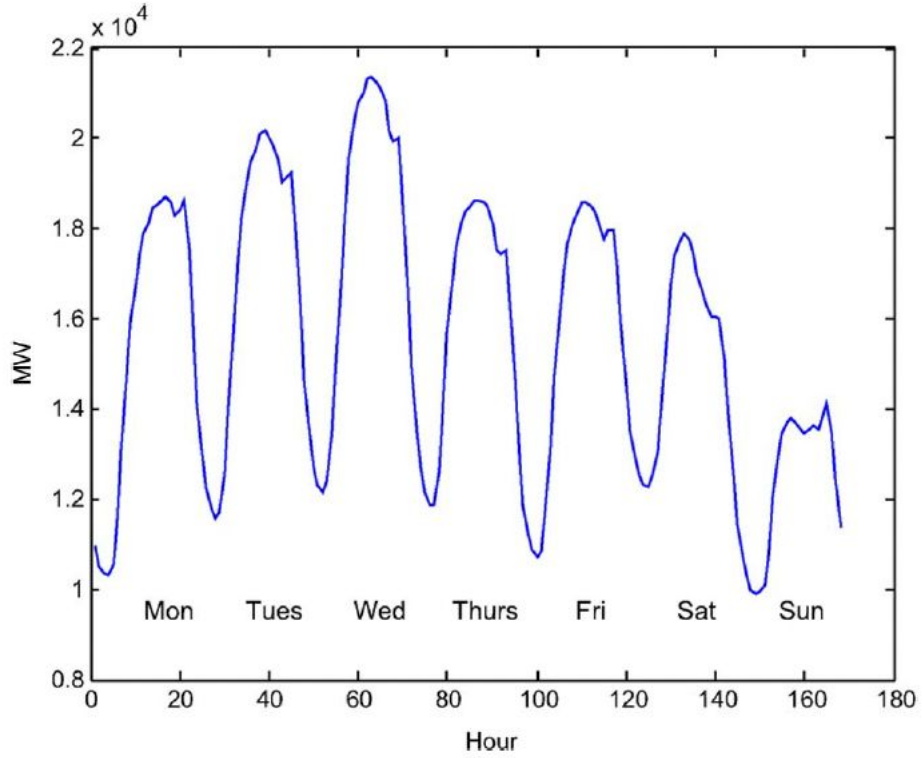


Fig: Weekly patterns of load in ISO-NE dataset

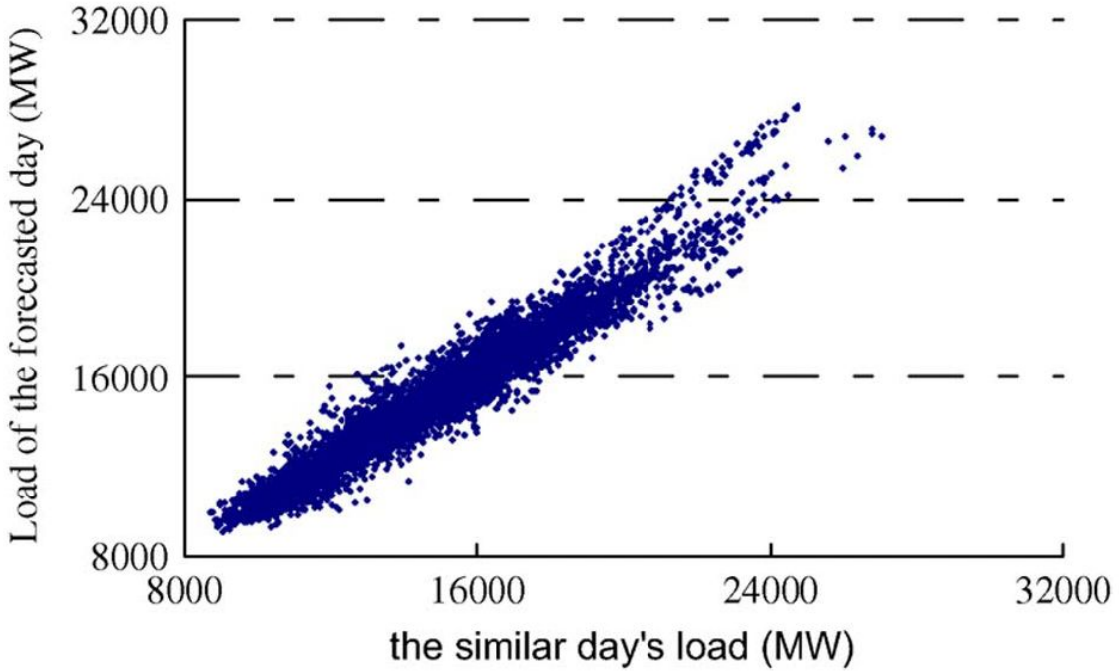
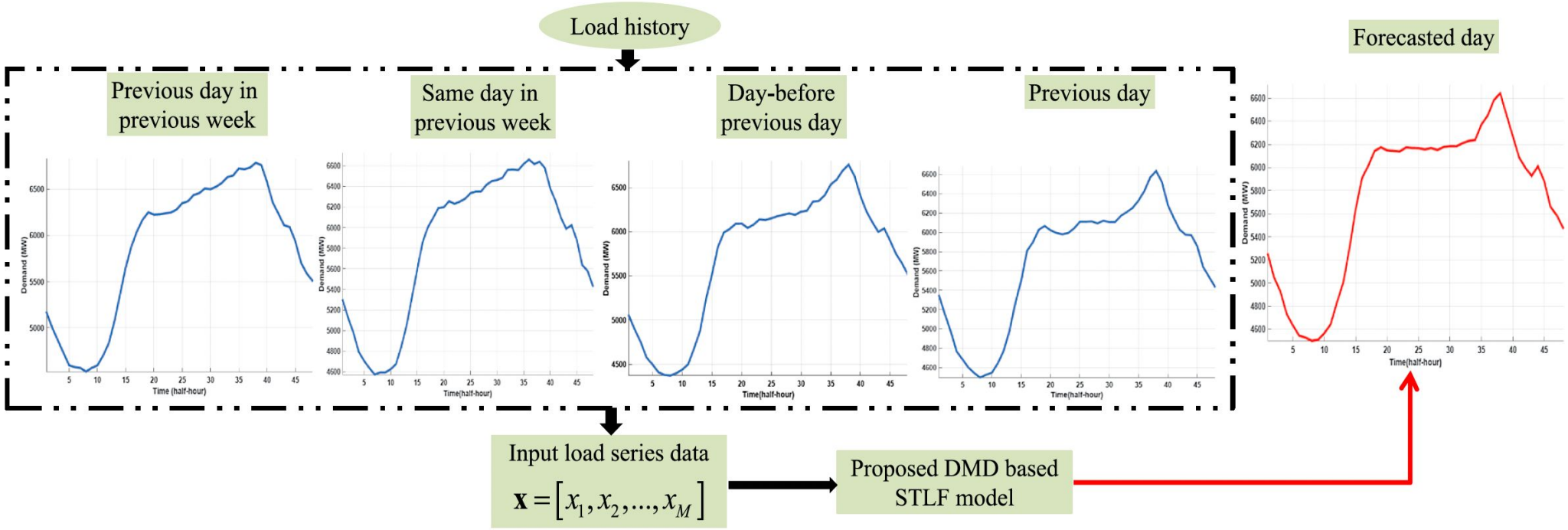


Fig: Weekly patterns of load in ISO-NE dataset

Chen, Ying, et al. "Short-term load forecasting: Similar day-based wavelet neural networks." IEEE Transactions on Power Systems 25.1 (2009): 322-330.

How similar day is used?

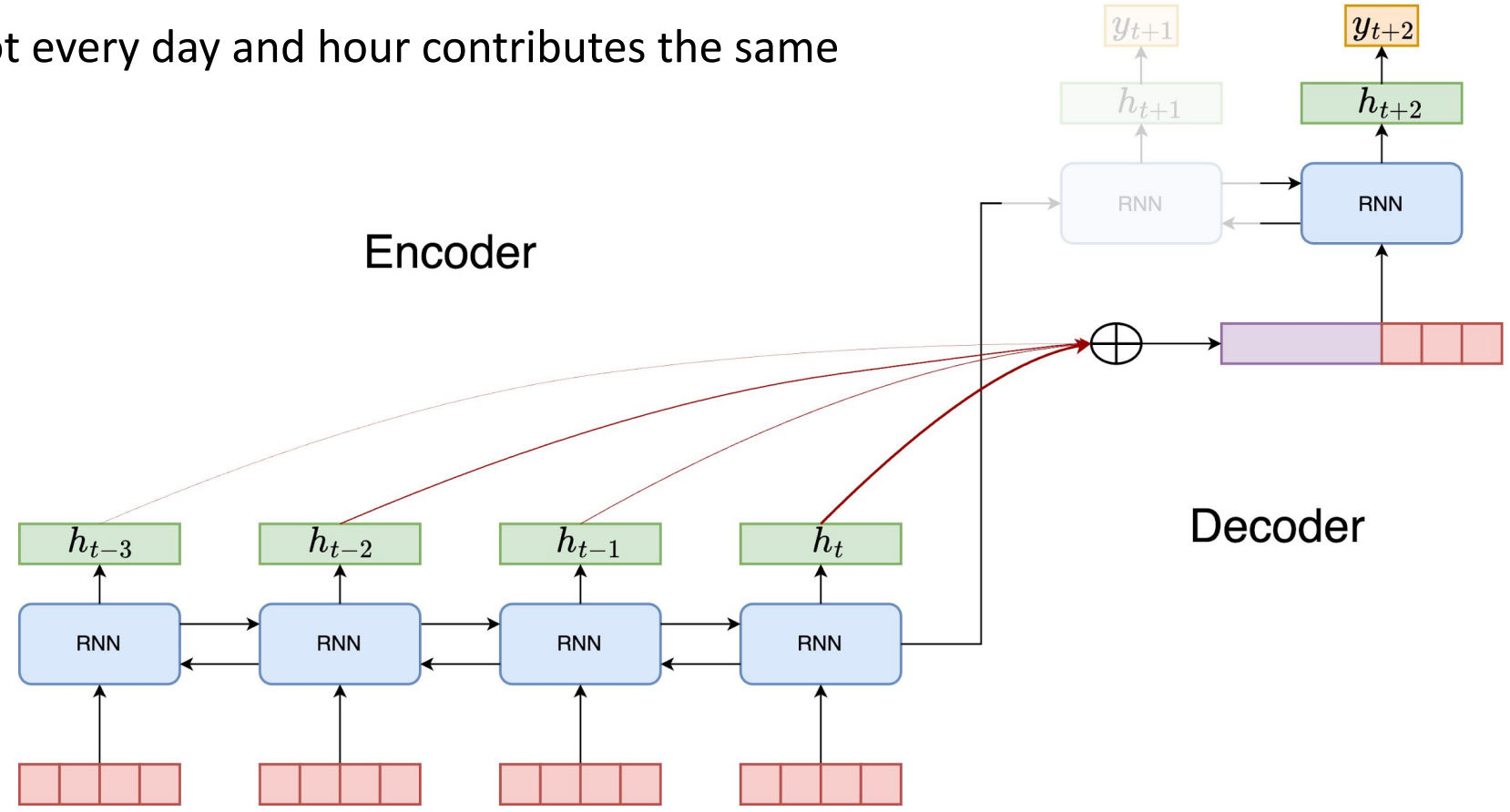
Modified the input to includes similar day data.



Mohan, Neethu, et al. "A data-driven strategy for short-term electric load forecasting using dynamic mode decomposition model." Applied energy. 2018.

Attention for short term load forecasting

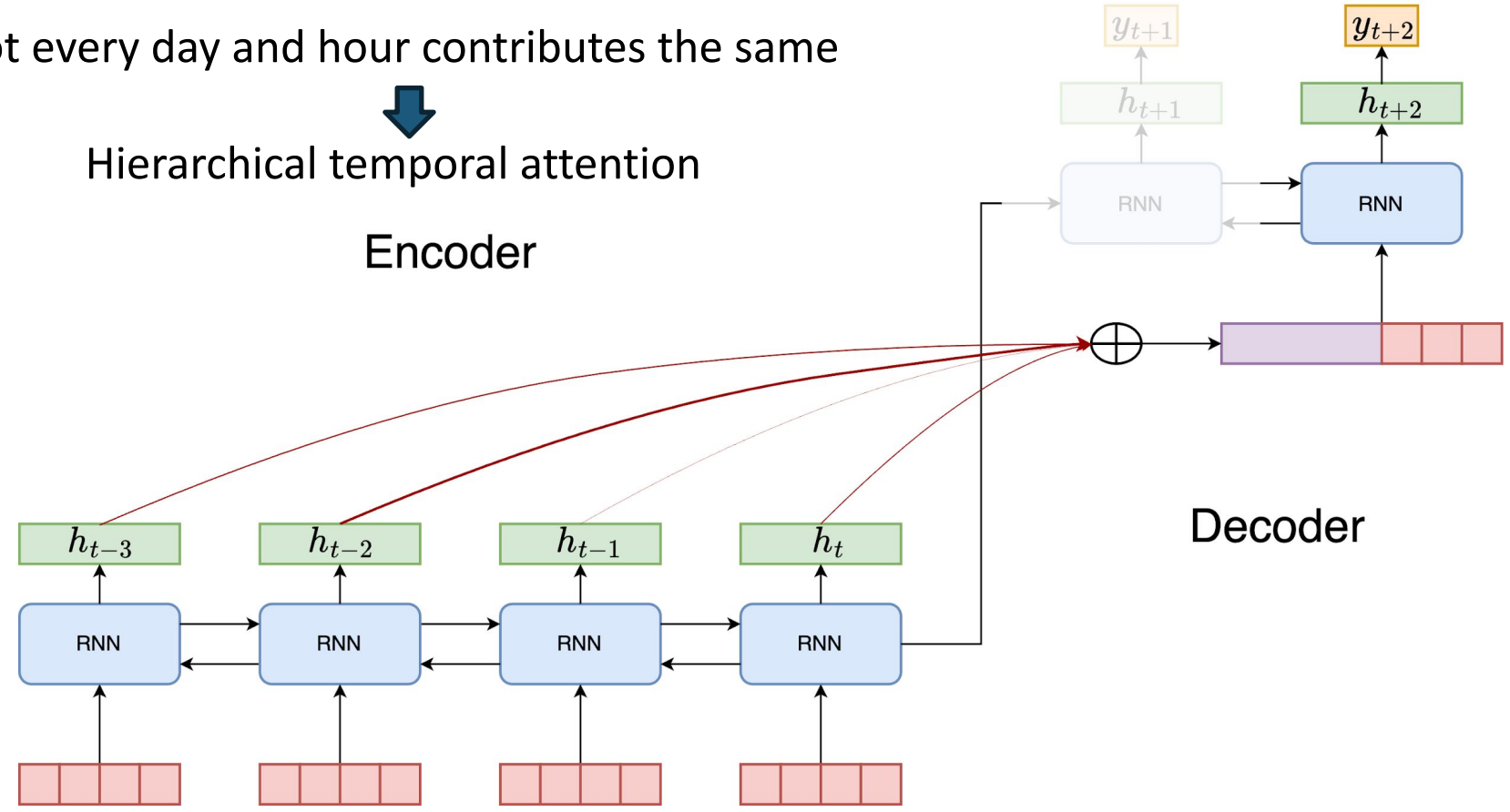
Not every day and hour contributes the same



Attention for short term load forecasting

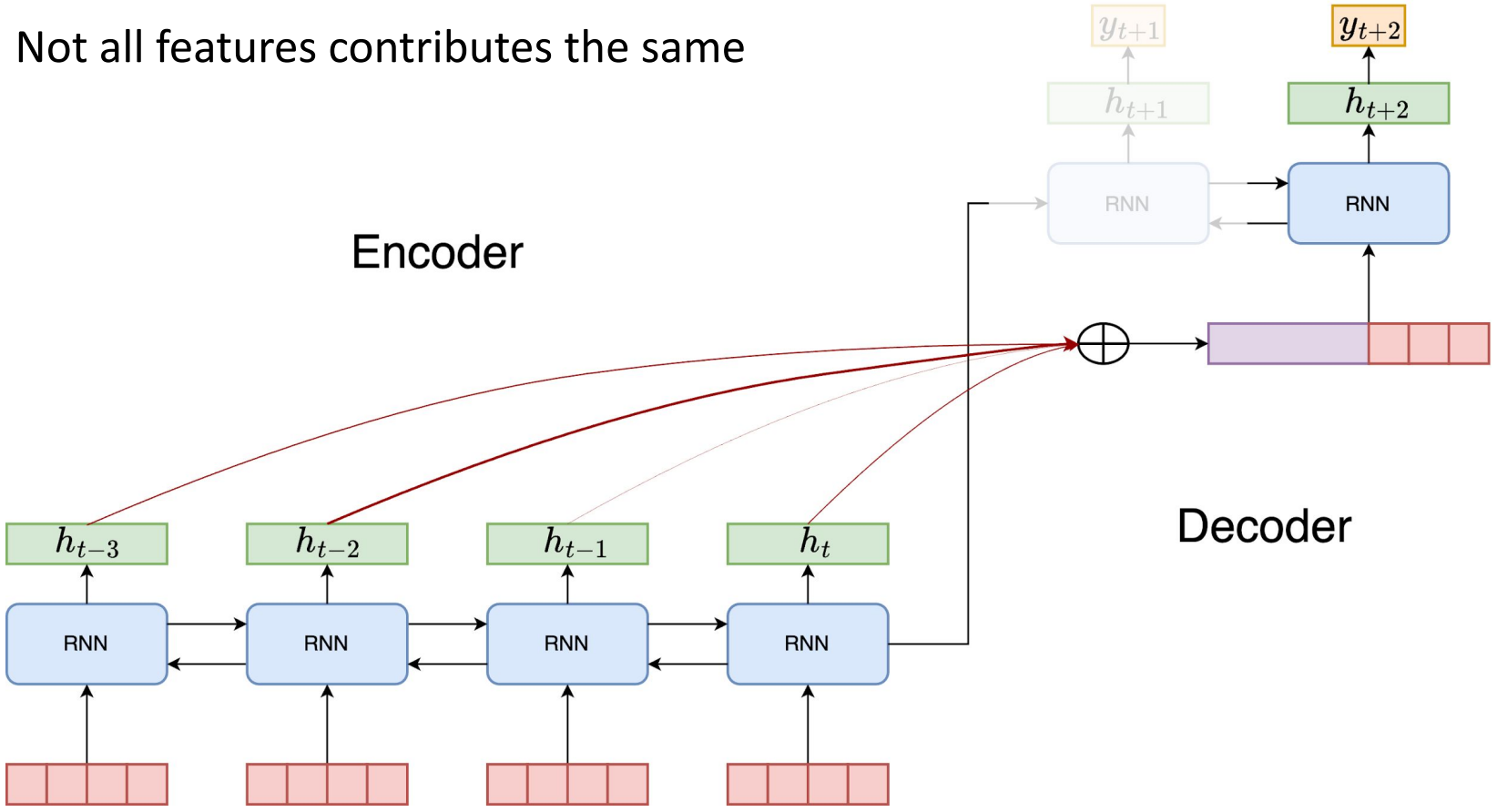
Not every day and hour contributes the same

Hierarchical temporal attention



Attention for short term load forecasting

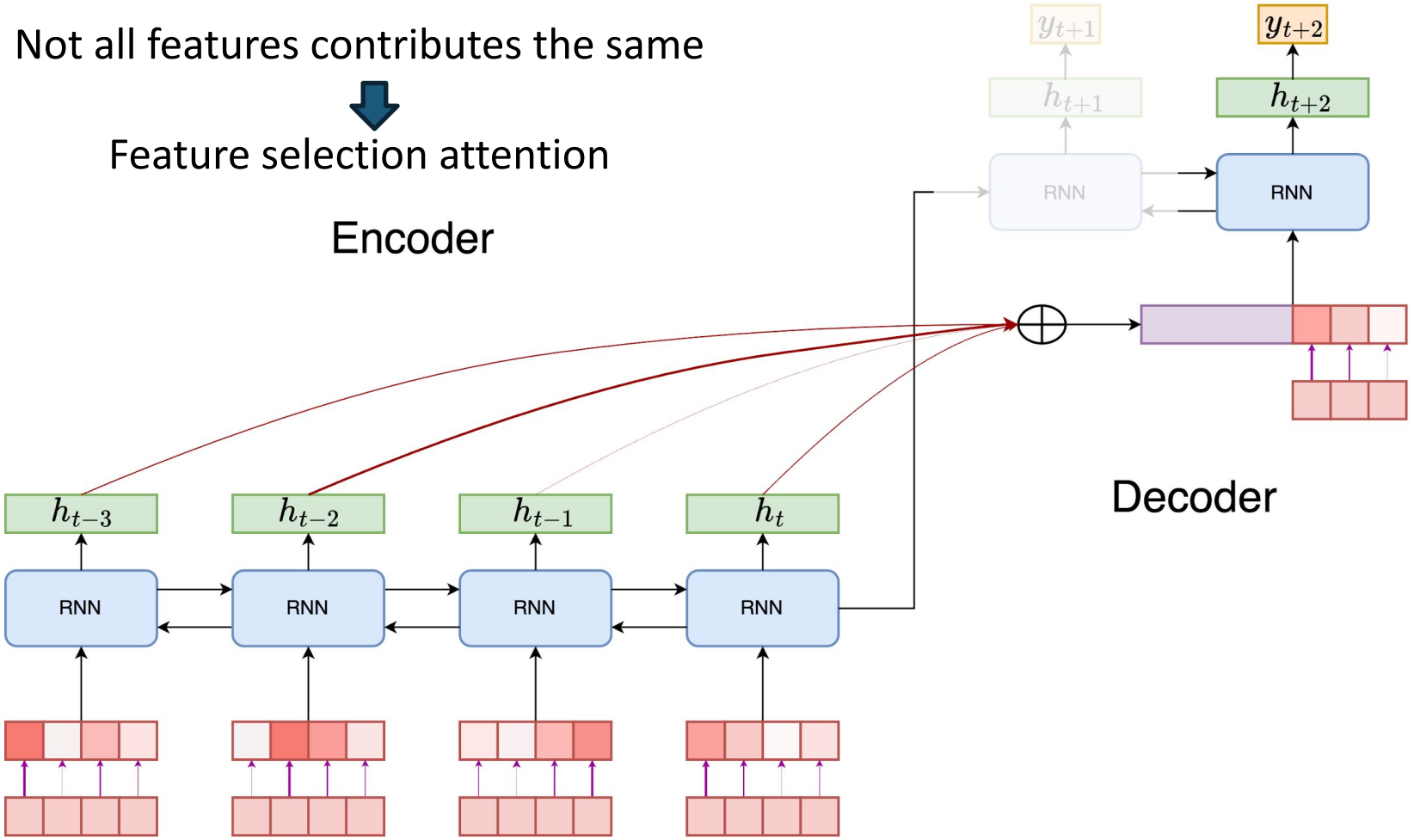
Not all features contributes the same



Attention for short term load forecasting

Not all features contributes the same

Feature selection attention



Architecture

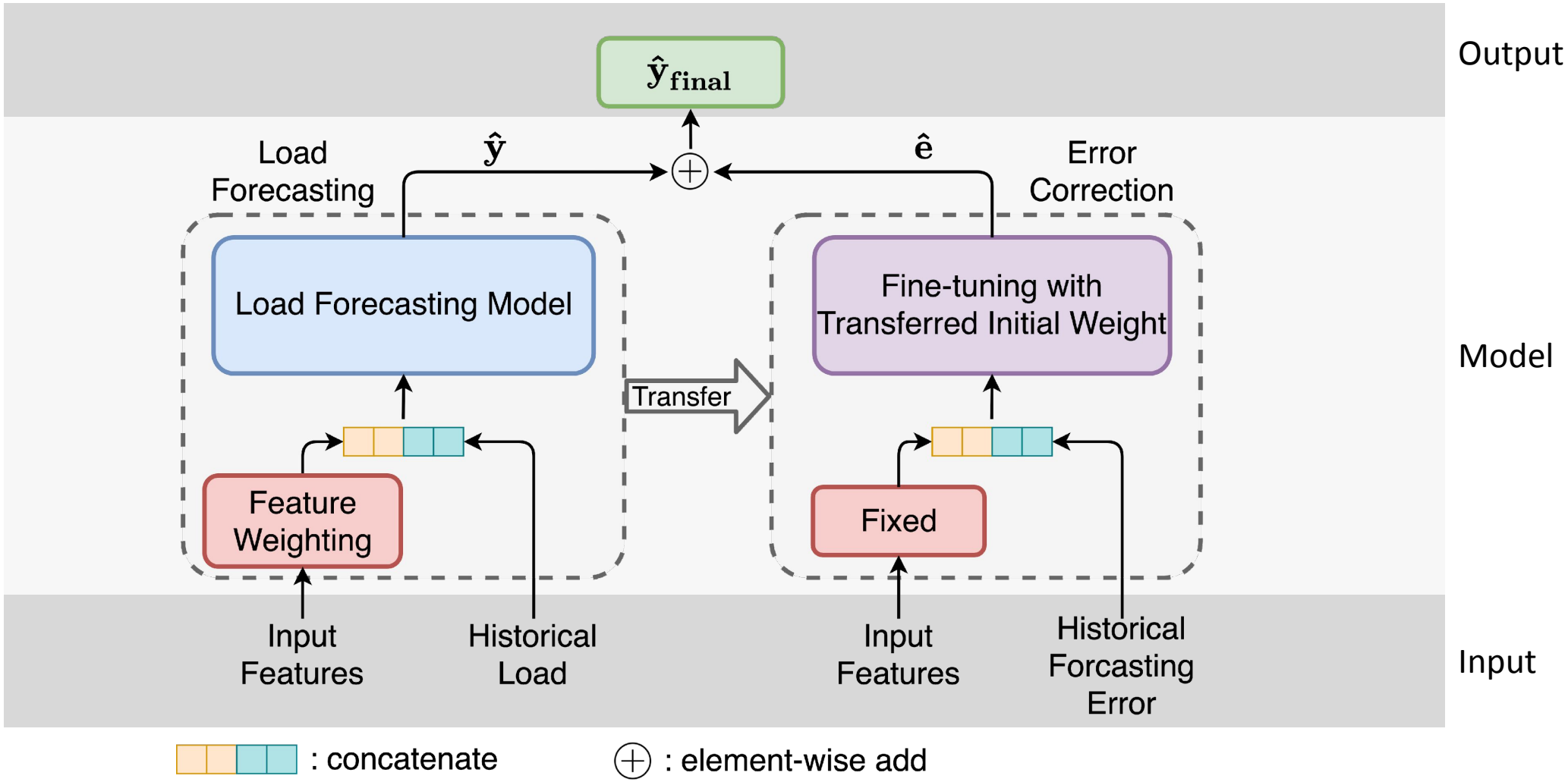


Fig. 4 The architecture of the proposed load forecasting framework

Feature weighting structure

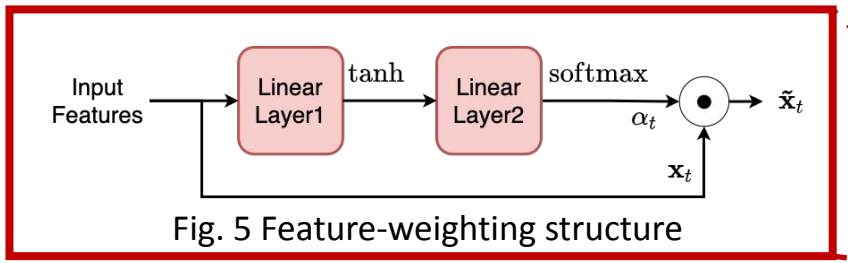
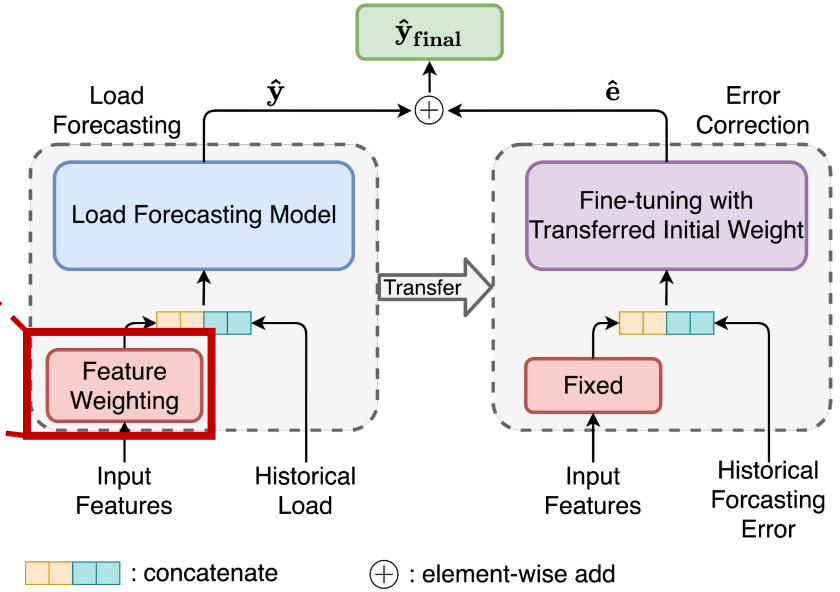


Fig. 5 Feature-weighting structure



- The feature weight is given by:

$$\mathbf{h}_t = \mathbf{V}_\alpha \tanh\left(\mathbf{W}_\alpha \mathbf{x}_t^\top\right)$$

$$\alpha_t^k = \frac{\exp\left(h_t^k\right)}{\sum_{i=1}^{n} \exp\left(h_t^i\right)},\ k = 1, 2, \ldots, n$$

- The weighted feature input is updated by:

$$\tilde{\mathbf{x}}_t = \alpha_t \odot \mathbf{x}_t$$

Load forecasting structure

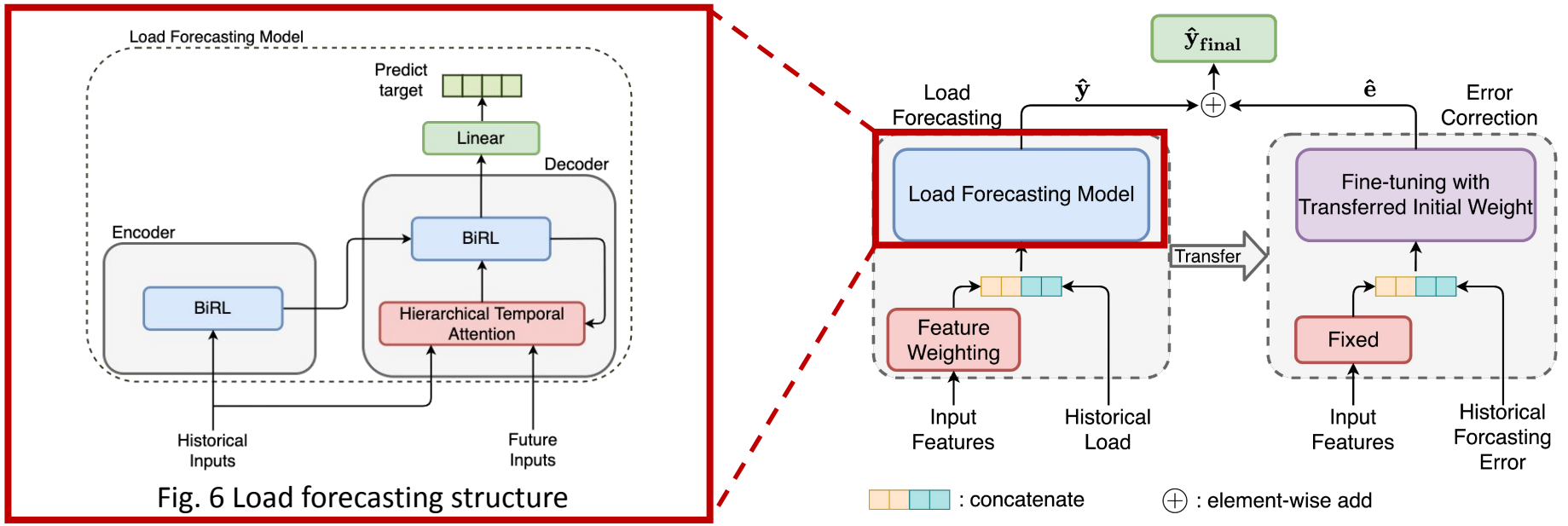


Fig. 6 Load forecasting structure

- The hidden state of the encoder is updated as:

$$\mathbf{h}_t^f = \mathrm{RL}^f(\mathbf{h}_{t-1}^f, [\tilde{\mathbf{x}}_t; y_t]^\top)$$

$$\mathbf{h}_t^b = \mathrm{RL}^b(\mathbf{h}_{t+1}^b, [\tilde{\mathbf{x}}_t; y_t]^\top)$$

$$\mathbf{h}_t^e = \left[\mathbf{h}_t^{f\top}; \mathbf{h}_t^{b\top}\right]^\top,$$

- The hidden state of the decoder is updated as:

$$\mathbf{h}_t^f = \mathrm{RL}^f(\mathbf{h}_{t-1}^f, [\tilde{\mathbf{x}}_t; \mathbf{a}_t^\top])$$

$$\mathbf{h}_t^b = \mathrm{RL}^b(\mathbf{h}_{t+1}^b, [\tilde{\mathbf{x}}_t; \mathbf{a}_t^\top])$$

$$\mathbf{h}_t^d = \left[\mathbf{h}_t^{f\top}; \mathbf{h}_t^{b\top}\right]^\top.$$

- The output is given by:

$$\mathbf{y}_f = \mathbf{V}_y \mathrm{ReLU}(\mathbf{W}_y \mathbf{h}_{t+1:t+T_f}^d)$$

- Recurrent Layer (RL) is implemented as LSTM and GRU in this work.

Error correction

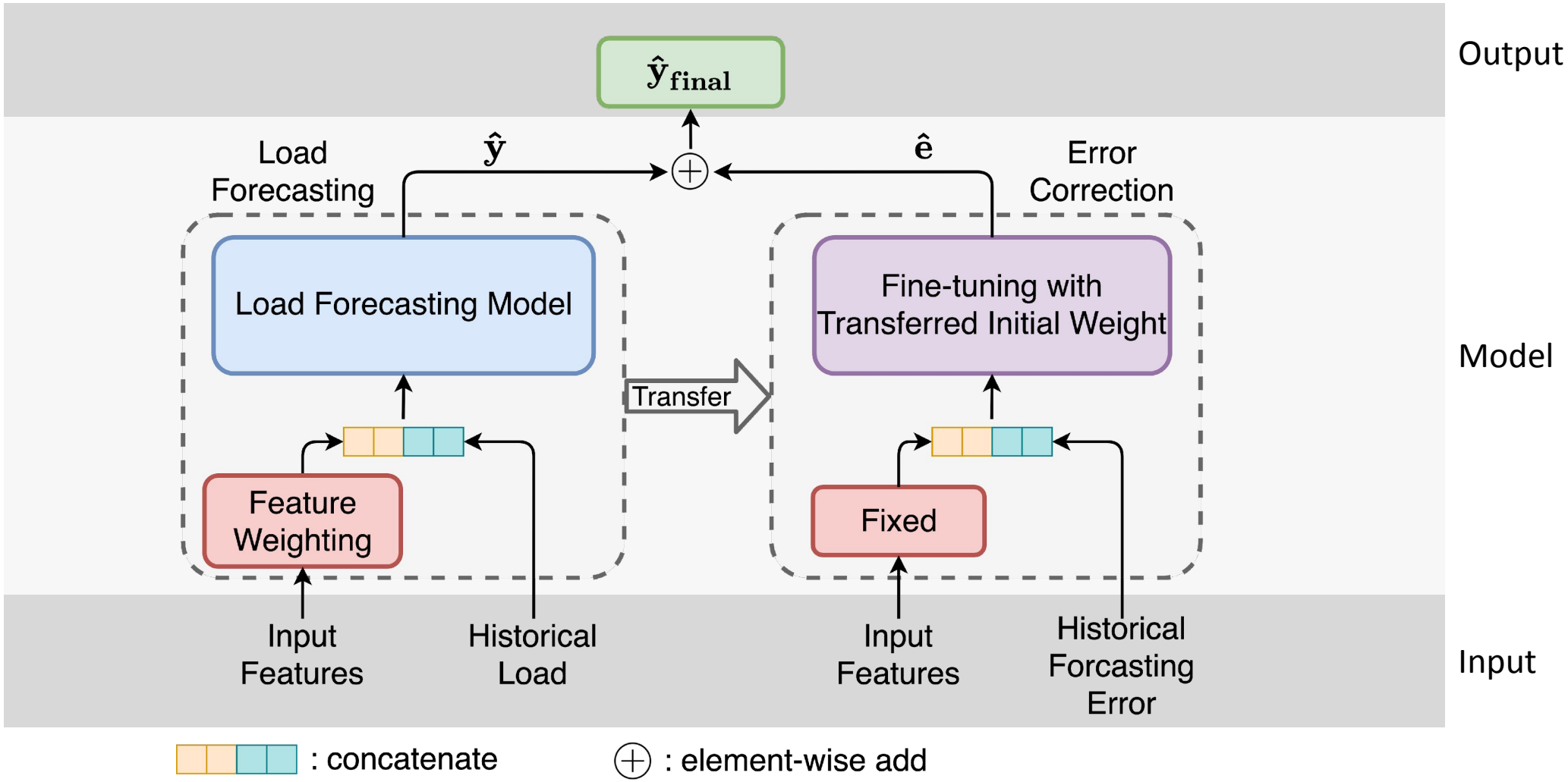
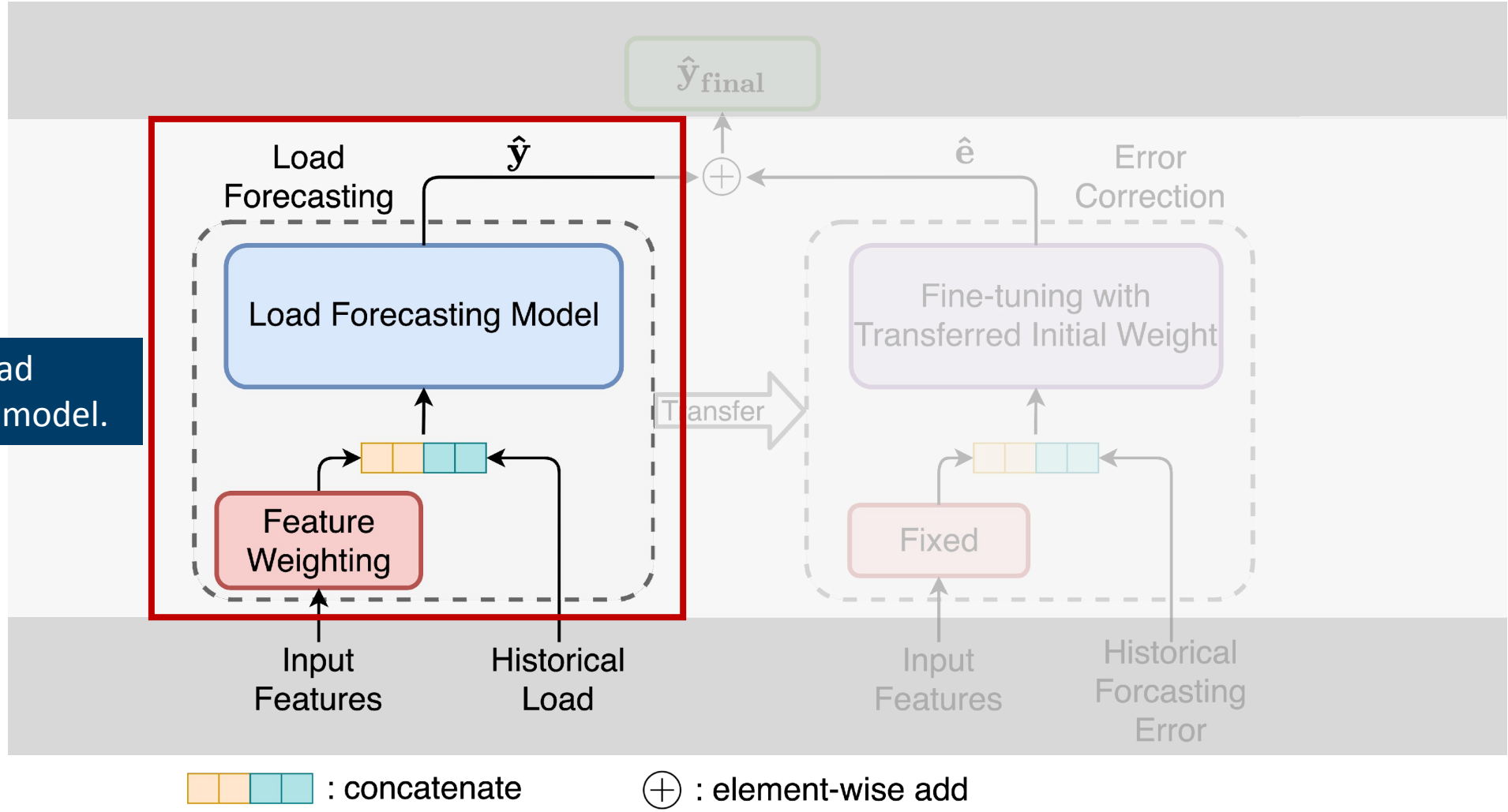


Fig. 4 The architecture of the proposed load forecasting framework

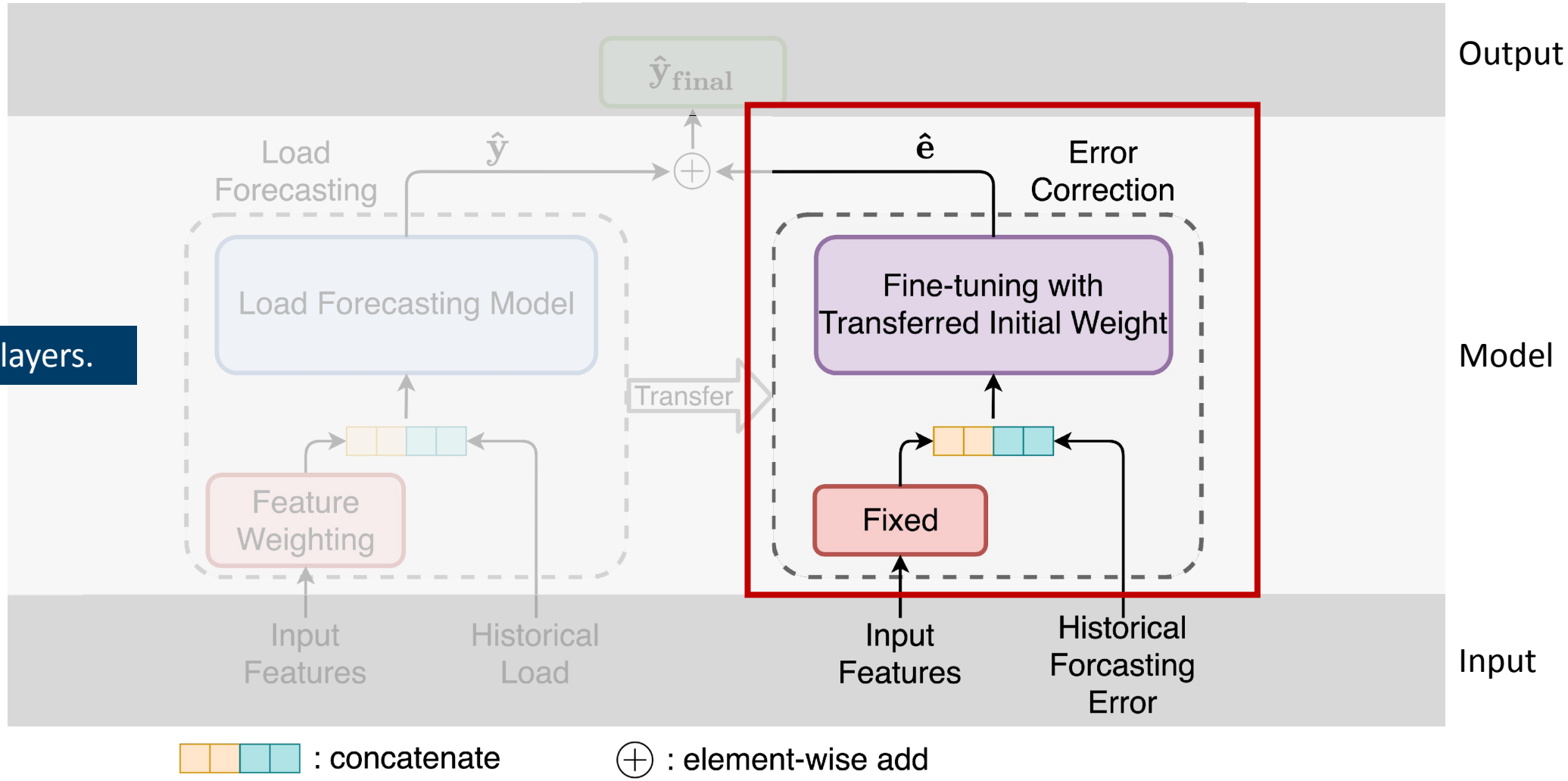
Error correction steps

- Train the load forecasting model.



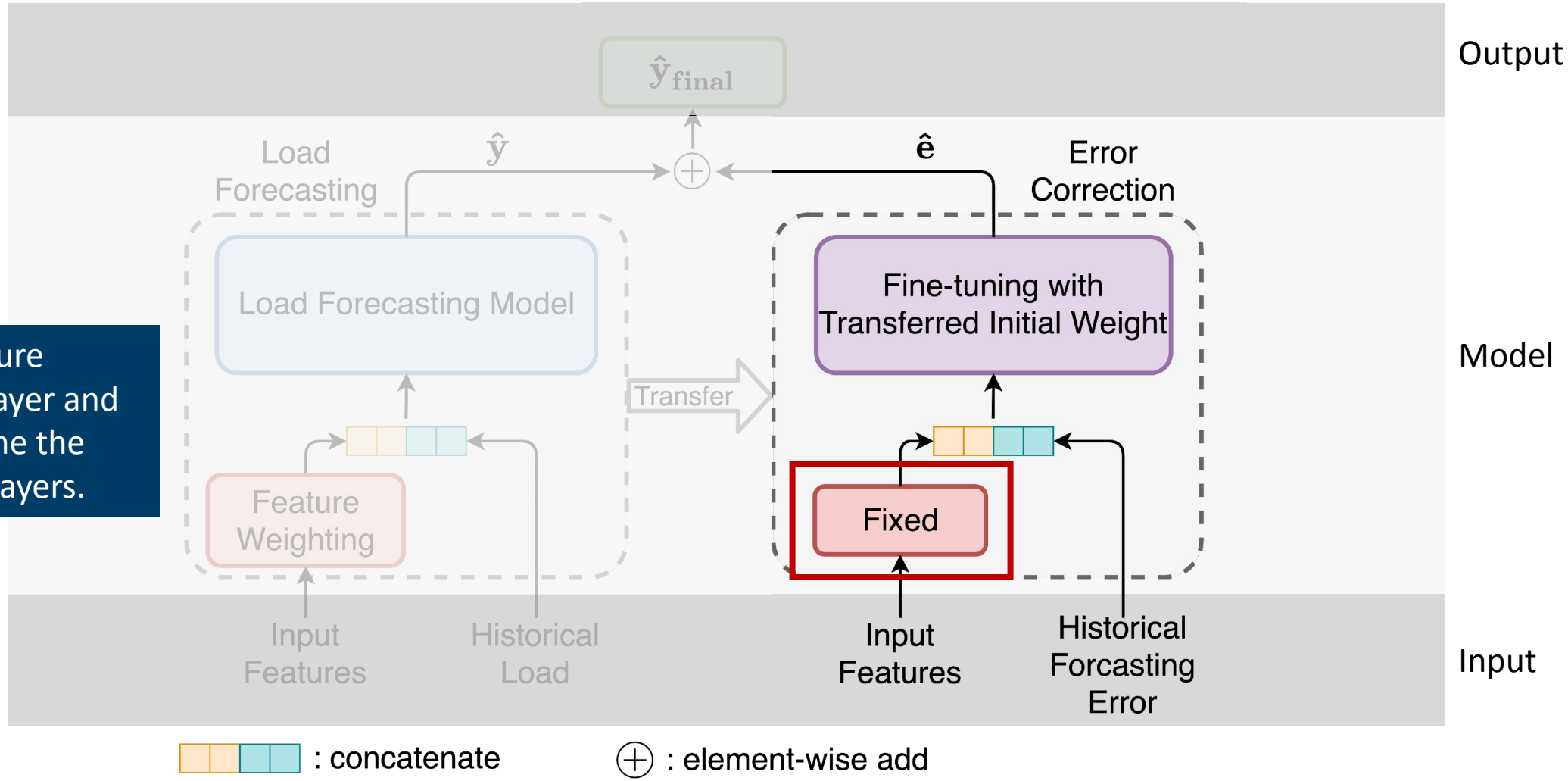
Error correction steps

- Transfer all layers.



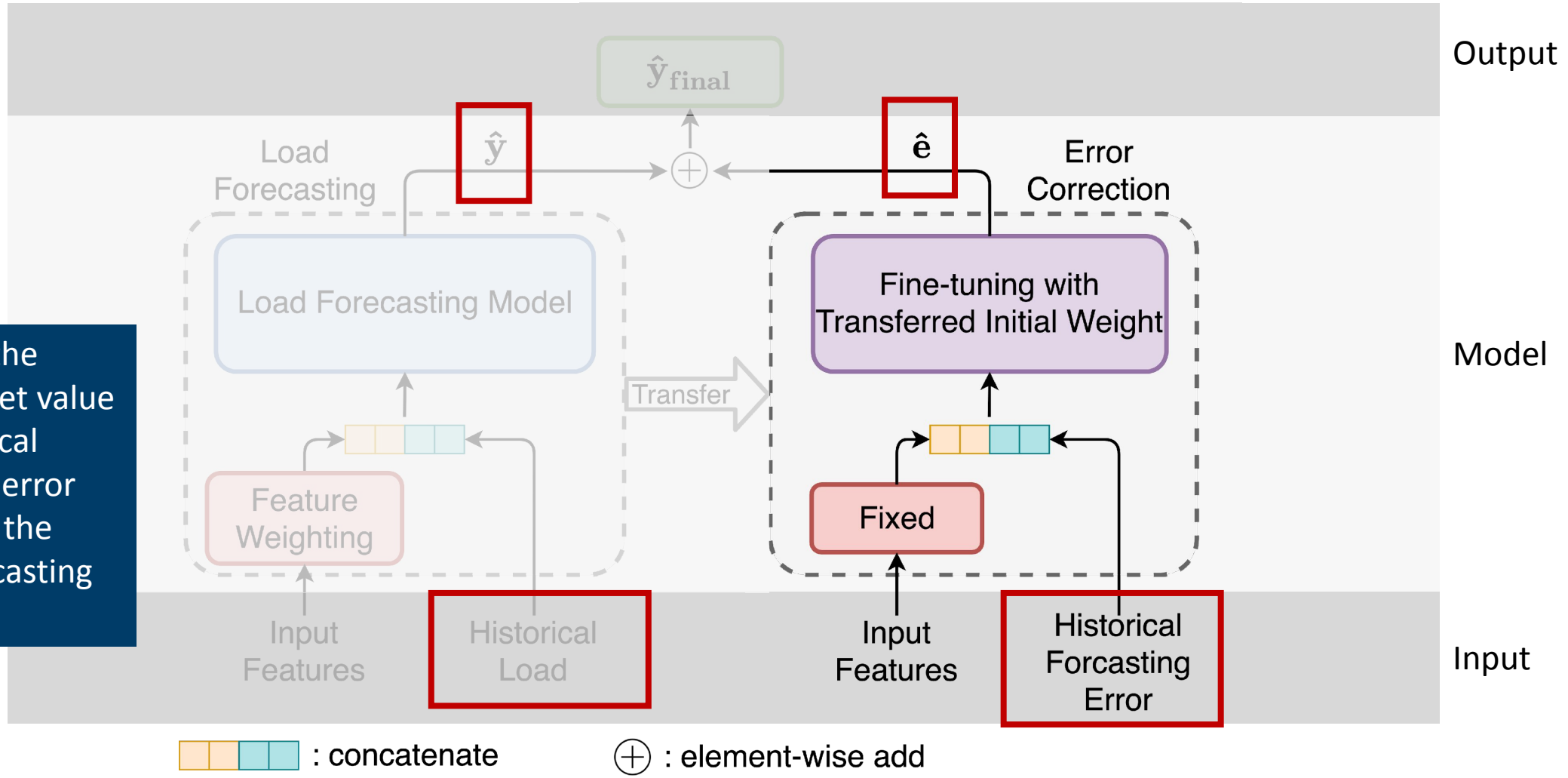
Error correction steps

- Fix the feature weighting layer and and fine tune the rest of the layers.



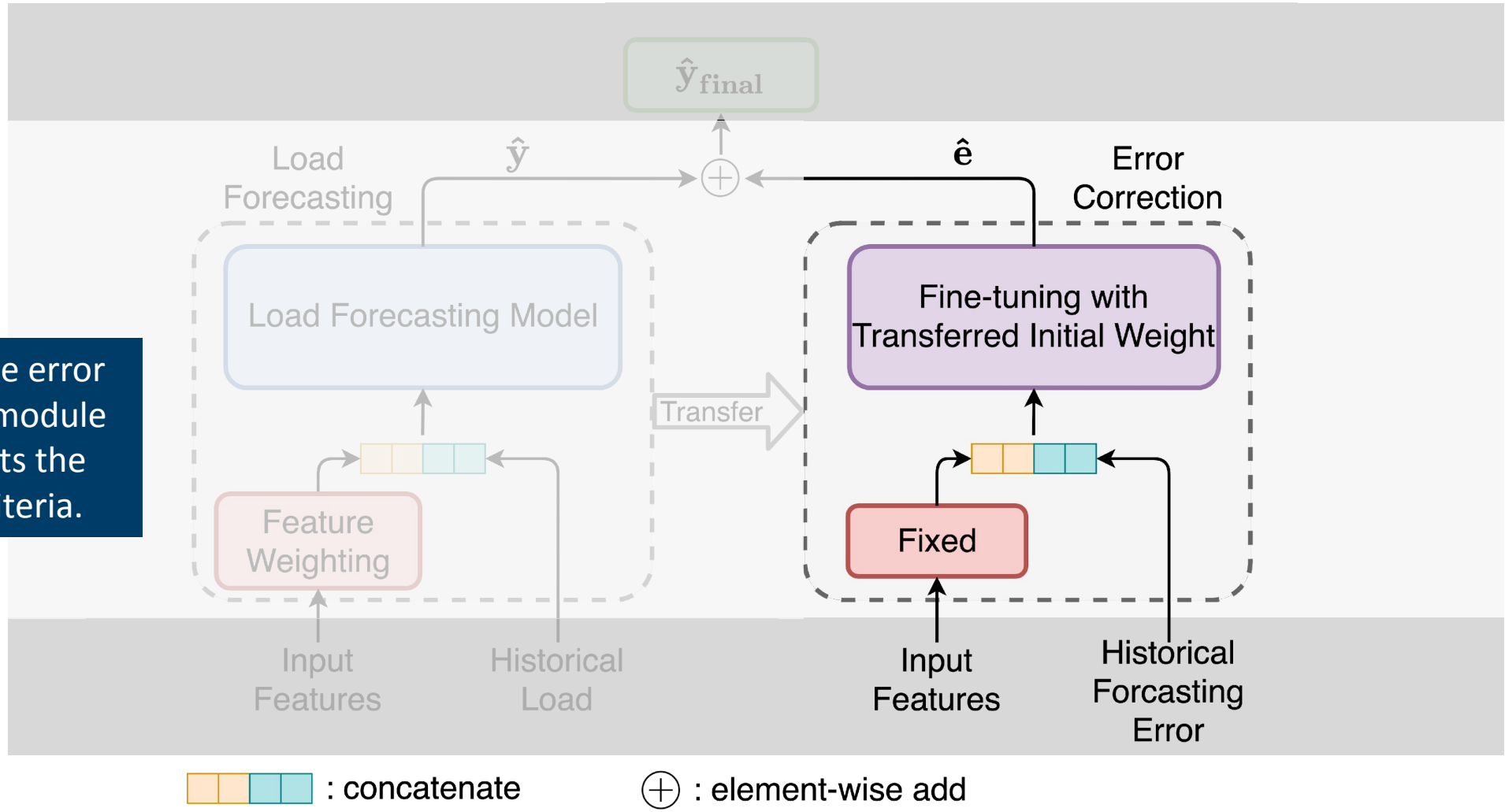
Error correction steps

- Substitute the history target value with historical forecasting error and output the future forecasting error.

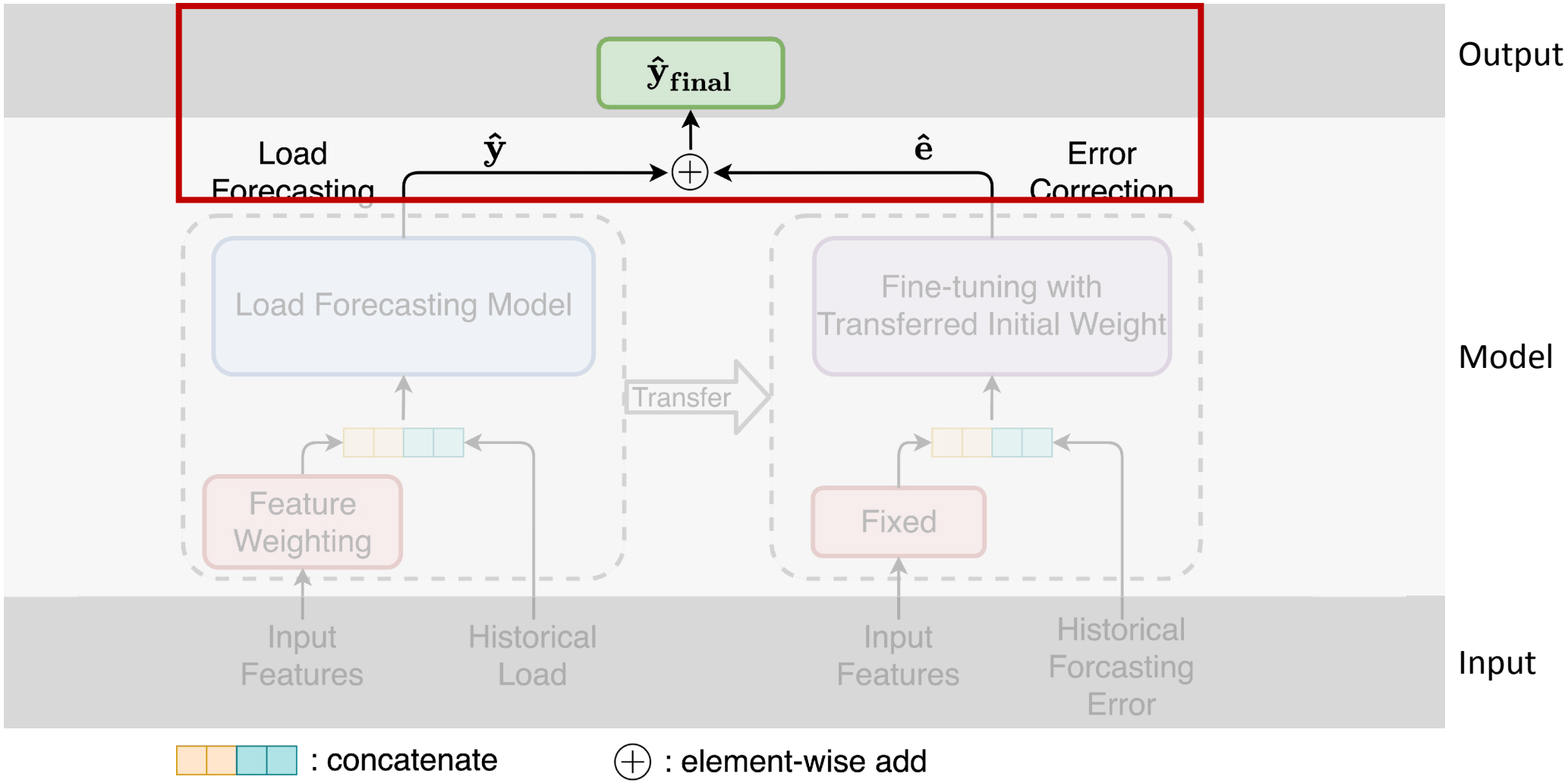


Error correction steps

- Finetune the error correction module until it meets the stopping criteria.



Error correction steps



Loss function

- Adding sparsity-promoting L1 regularizer to enhance feature selection/weighting.

$$\mathbb{L}_r = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \lambda \|\alpha\|_1$$

y_i	True value
$\hat{y}_i$	Predict value
α	Feature weight

- No L1 regularization for the error correction modular because the feature weighting layer is fixed. Thus, the loss function is:

$$\mathbb{L}_e = \frac{1}{N}\sum_{i=1}^{N}(e_i - \hat{e}_i)^2$$

Dataset

- Time resolution: 1 hour
- ISO New England Dataset (ISO-NE)
  - Training: 2015-2017
  - Validation: 2018
  - Testing: 2019

Table 3.1: Inputs of ISO-NE dataset.

Input	Size	Description
$\mathbf{y}_h$	168×1	Historical target values
DaDemd	192×1	day ahead demand
DryBulb	192×1	dry bulb temperature
DewPnt	192×1	dew point temperature
Weekday	192×1	Weekday or weekend indicator
Holiday	192×1	Holiday or non-holiday indicator
Season	192×4	One-hot encoding
Hour of Day	192×24	One-hot encoding
Day of Week	192×7	One-hot encoding
Month of Year	192×12	One-hot encoding

Dataset

- Time resolution: 1 hour
- ISO New England Dataset (ISO-NE)
  - Training: 2015-2017
  - Validation: 2018
  - Testing: 2019
- North-American Utility Dataset (NAU)
  - Training: 1987-1989
  - Validation: 1990
  - Testing: 1991

Table 3.2: Inputs of The North-American Utility Dataset.

Input	Size	Description
$\mathbf{y}_h$	168×1	Historical target values
Temperature	192×1	Historical and future temperature
Holiday	192×1	Holiday or non-holiday indicator
Season	192×4	One-hot encoding
Hour of Day	192×24	One-hot encoding
Day of Week	192×7	One-hot encoding
Month of Year	192×12	One-hot encoding

Baseline models

Classic Machine learning	• Gradient Boosting Machine(GBM) • Random Forest (RF) • Support Vector Regression (SVR)
DBN-based models	• Deep Belief Network (DBN) • Rough autoencoder (RAE)
RNN-based models	• Convolutional Neural Network + LSTM (CNN-LSTM) • Attention-based load forecasting (ANLF)
Transformer-based model	• Informer[1] (Best paper in AAAI'21)

1 Zhou, Haoyi, et al. "Informer: Beyond efficient transformer for long sequence time-series forecasting." Proceedings of AAAI. 2021.

Results

- Case 1: Ablation Study and Discussion
- Case 2: Load Forecasting Module Comparison
- Case 3: Generalization Capability

Ablation Study and Discussion

NAU dataset: Ablation study for the proposed framework

MI [84]	RF [25]	FW	TA	SDA	BL [27]	EC	MAE	MAPE (%)
-	-	-	-	-	-	-	89.20	3.93
✓	-	-	-	-	-	-	59.93	2.57
-	✓	-	-	-	-	-	63.65	2.76
-	-	✓	-	-	-	-	56.65	2.45
-	-	-	✓	-	-	-	69.10	2.96
-	-	-	✓	✓	-	-	64.29	2.78
-	-	✓	✓	✓	-	-	48.96	2.15
-	-	✓	✓	✓	✓	-	46.36	2.09
-	-	✓	✓	✓	-	✓	**45.70**	**2.00**

Acronyms:

MI: mutual information feature weight

RF: random forest feature weight

FW: feature weighting attention

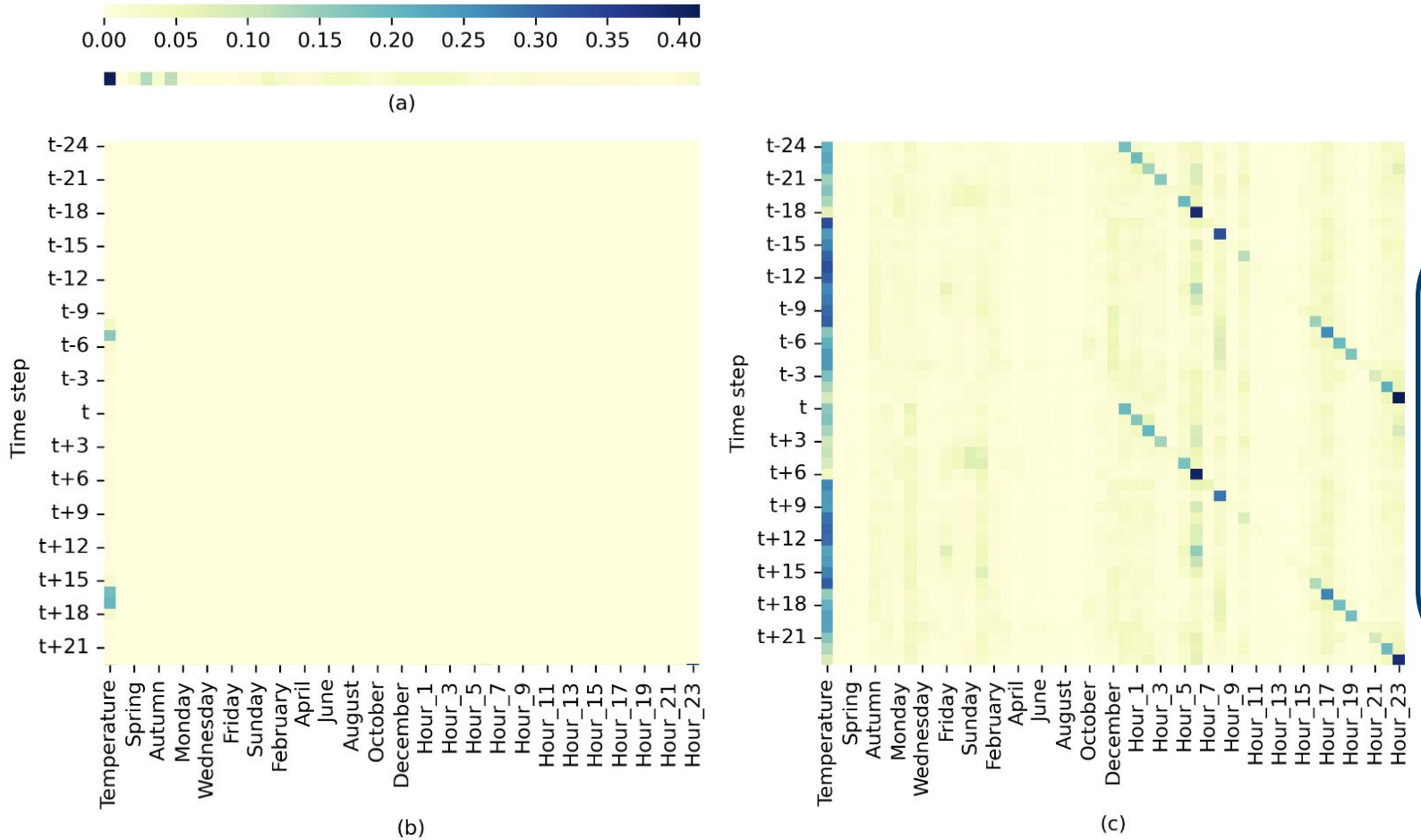
TA: temporal attention

SDA: similar day attention

BL: baseline ARIMA error correction

EC: the proposed error correction

Ablation Study and Discussion



Feature weight with different approaches:

(a) mutual information

(b) random forest

(c) the proposed feature weighting attention

NAU dataset: Two-day feature weight visualization

Load Forecasting Comparison

Forecasting errors over the year 2019 for the ISO-NE dataset

Model	SVR	RF	GBM	DBN	RAE	CNN-LSTM	ANLF	Informer	PM-LSTM	PM-GRU
MAE	317.49	393.58	277.54	326.45	308.36	309.06	258.70	256.89	**229.47**	**231.84**
MAPE (%)	2.33	2.87	2.00	2.38	2.22	2.27	1.88	1.89	**1.66**	**1.67**

Forecasting errors over the year 1991 for the NAU dataset

Model	SVR	RF	GBM	DBN	RAE	CNN-LSTM	ANLF	Informer	PM-LSTM	PM-GRU
MAE	67.97	99.91	83.05	68.33	63.55	61.85	58.65	57.13	**48.96**	**46.42**
MAPE (%)	2.98	4.40	3.61	2.99	2.79	2.73	2.58	2.49	**2.15**	**2.03**

Universality of FW & EC

ISO-NE dataset: Ablation study for Informer

FW	EC	MAE	MAPE (%)
-	-	256.89	1.89
✓	-	249.17	1.81
-	✓	239.35	1.76
✓	✓	**239.23**	**1.74**

Acronyms:

FW: feature weighting attention

EC: the proposed error correction

Contribution

- The proposed framework offers a modular and plug-and-play functionality that can be adapted to different types of data and setups in STLF.

- Error correction modular avoids the necessity of new model design and inherits the learnt feature knowledge via transfer learning.