

Article

Development of Robust and Physically Interpretable Soft Sensor for Industrial Distillation Column Using Transfer Learning with Small Datasets

Yu-Da Hsiao ¹, Jia-Lin Kang ^{2,*}  and David Shan-Hill Wong ^{1,*}¹ Department of Chemical Engineering, National Tsinghua University, Hsinchu 30031, Taiwan; s106032504@m106.nthu.edu.tw² Department of Chemical and Materials Engineering, National Yunlin University of Science and Technology, Douliu City 64002, Taiwan

* Correspondence: jlkang@yuntech.edu.tw (J.-L.K.); dshwong@gmail.com (D.S.-H.W.)

Abstract: In the development of soft sensors for industrial processes, the availability of data for data-driven modeling is usually limited, which led to overfitting and lack of interpretability when conventional deep learning models were used. In this study, the proposed soft sensor development methodology combining first-principle simulations and transfer learning was used to address these problems. Source-domain models were obtained using a large amount of data generated by dynamic simulations. They were then fine-tuned by a limited amount of real plant data to improve their prediction accuracies on the target domain and guaranteed the models with correct domain knowledge. An industrial C4 separation column operating at a refining unit was used as an example to illustrate the effectiveness of this approach. Results showed that fine-tuned networks could obtain better accuracy and improved interpretability compared to a simple feedforward network with or without regularization, especially when the amount of actual data available was small. For some secondary effects, such as interaction gain, its interpretability is mainly based on the interpretability of the corresponding source models.

Keywords: artificial neural network; distillation column; small datasets; soft sensor; transfer learning



Citation: Hsiao, Y.-D.; Kang, J.-L.; Wong, D.S.-H. Development of Robust and Physically Interpretable Soft Sensor for Industrial Distillation Column Using Transfer Learning with Small Datasets. *Processes* **2021**, *9*, 667. <https://doi.org/10.3390/pr9040667>

Academic Editor: Jesus Rafael Alcantara-Avila

Received: 24 February 2021

Accepted: 6 April 2021

Published: 10 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Soft sensors are virtual sensors instantly estimating hard-to-measure variables, such as concentration, which is traditionally measured online by low frequency laboratory analysis through inputting easy-to-measure variables, such as pressure, temperature, and flowrate. In the past few decades, soft sensors have been extensively studied and implemented in the process industries. Typically, soft sensors can be divided into two general classes: the model-driven (white box) and the data-driven (black box). The model-driven soft sensors were commonly based upon first-principle models, while the data-driven ones were usually based on regression techniques such as principal component analysis, partial least squares, neuro-fuzzy systems, support vector machines, and artificial neural networks (ANNs).

Recently, with the advanced progressions in deep learning, ANN variants once again caught the attention of process engineers due to their power in nonlinear regression ability. However, the ANN variants were black boxes, usually difficult to interpret by domain knowledge [1]. Such a drawback held scientists and engineers from further implementing ANNs on the systems they were focusing on, thus slowing down their popularizing rates. With these concerns, explainable artificial intelligence (AI), which aims to make AI interpretable and trustworthy, became a focusing field of machine learning [2]. For process engineering and control, it was also critical and necessary to implement interpretable models into the processes to make sure the predictions of these models were not merely accurate but also interpretable based on domain knowledge.

The quality of the data is key to training a good AI model. Udagama et al. [3] reported that the data of chemical plants requires four properties: volume, variety, velocity, and veracity. It is difficult to obtain an accurate model when lacking any one of the properties in data. For conventional machine learning methods, a large amount of such data was necessary for training robust models with both accuracy and interpretability. However, in process industries, most of the critical quality variables, such as concentration and viscosity, were measured by low frequency offline analysis in laboratories. Due to the low frequencies, the corresponding databases were usually small and required long periods of time to enlarge to be sufficient for training neural networks. Furthermore, for some newly started processes with a short operation history, it was impossible to gather big data. Small datasets are the inherent problem in soft sensor development [4]. To overcome the lack of data, one common approach is to use linear models such as partial least squares (PLS). However, industrial processes are nonlinear over a large range. For instance, distillation column operations in chemical plants have very nonlinear behavior for producing high-purity products. Hence, constantly updating linear models is required [5,6]. Nonlinear models such as ANN are commonly used to predict nonlinear systems such as distillation columns, but the generalization ability of ANN models must be checked using validation data and regularizations [7]. Our recent study showed that a simple validation test might not be sufficient to ensure the generalizability and physical consistency when the datasets are limited.

It should be pointed out that some form of prior knowledge must exist when we try to build a data-driven model. Hybrid models have been used to alleviate the problem of small datasets and improve soft sensor accuracy [8,9]. Prior knowledge may be in the form of data of a similar system or an approximate simulator based on the first-principle model. In machine learning, the technique of building a data-driven model for the current problem, a target domain, from a model of another similar system, the source domain, is known as transfer learning [10–12]. The purpose of this study was to present a new data-driven soft sensor development methodology that combines first-principle simulations and transfer learning methods to overcome the overfitting issue and ensure the interpretability of the soft sensor when there is only a limited data set. An industrial C₄ separation column was used to demonstrate the performance of this approach. Furthermore, gain consistency analysis was used to ensure the interpretability of soft sensors.

2. Methodology

2.1. Transfer Learning

One of the most common approaches to performing transfer learning is to fine-tune the parameters (weights and bias) of the networks [12], which was employed in this study. During the fine-tuning procedures, it was critical to determine which layer should be frozen (nontrainable). However, there is no common census to which layers should be frozen, but usually, the weights of the first or last few layers are changed.

2.2. Process Simulator in Transfer Learning Framework

In process industries, process simulators serve as the core tools to calculate, analyze, and optimize the chemical and refining processes. These simulators provide engineers with reasonable results based on the first-principle theories and empirical correlations for operation decisions making. In the rapid progressions on computational abilities, process simulators, especially the dynamic ones, were ideal candidates to provide the dataset of source domains under transfer learning framework for the development of soft sensors. With the help of dynamic simulators, big datasets can be obtained within a short period of time.

To generate a dataset from first-principle simulators, the operating conditions of the first-principle simulators should be set to give periodic random variations within reasonable ranges such as feed stream conditions and controller setpoints. The methods of

data extraction from simulators were reported in [13,14]. In this study, we used MATLAB Simulink, connecting with simulators to extract and collect the simulation data.

3. Case Study

3.1. Process Description

In this study, an industrial C_4 separation column was used as an example to illustrate the effectiveness of this approach. The column separated the C_4 and C_5+ components of the reactor effluent. The main product, C_4 (over 90% of the feed) left as liquid distillate, while some noncondensable light impurities are left from the vapor distillate; C_5+ are left from the bottom. Quality control of the liquid distillate and bottom product was the top priority of this distillation operation, especially the quality of liquid distillate; namely, the concentration of C_5+ impurities at distillate and C_4 losses at the bottom should be controlled within an acceptable range. Hence, two soft sensors were built in this case study to monitor C_5+ impurities and C_4 losses at the distillate and bottom, respectively.

According to the domain knowledge of distillation unit operations, 14 critical process variables, including pressures, temperatures, and flow rates, were selected as the input variables for the soft sensors, as shown in Figure 1. The selected variables can be divided into two types: six manipulated variables (MVs) and eight sensor variables (SVs). MVs were manipulated by manual or automatic approaches, while the SVs were only the measured values, as shown in Table 1.

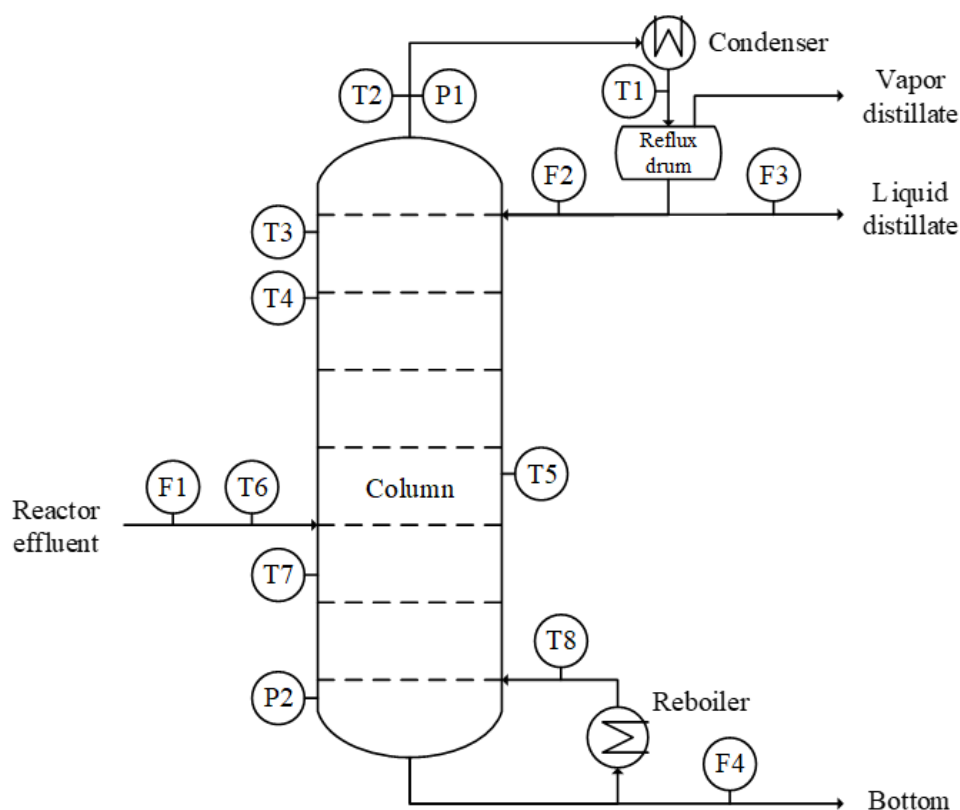


Figure 1. The process flow diagram of the industrial C_4 column.

3.2. Data Preprocessing

For soft sensors of distillate quality, there were 929 available samples for modeling, where 838 samples were used for learning (training and validation) and 91 samples for testing. For soft sensors of bottom quality, there were 453 available samples for modeling, where 414 samples were used for learning and 39 samples for testing.

Table 1. Selected input variables for soft sensors.

Tag	Variable Descriptions	Type	Unit
P1	Top pressure	MV	kg/cm ² -g
P2	Bottom pressure	SV	kg/cm ² -g
T1	Condenser temperature	SV	°C
T2	1st stage temperature	SV	°C
T3	2nd stage temperature	SV	°C
T4	5th stage temperature	SV	°C
T5	35th stage temperature	SV	°C
T6	Feed temperature	SV	°C
T7	49th stage temperature	SV	°C
T8	Reboiler temperature	MV	°C
F1	Feed flowrate	MV	m ³ /h
F2	Reflux flowrate	MV	m ³ /h
F3	Liquid distillate flowrate	MV	m ³ /h
F4	Bottom flowrate	MV	m ³ /h

The moving window method was applied to consider the dynamic behaviors of the process [15]. Window length (W) was 1-h backtracking from each sampling instant t , and each input variable was averaged every 10 min. The input-output relations of the soft sensor can be mathematically expressed as:

$$q_t = f(mv_t, mv_{t-1}, \dots, mv_{t-W}, sv_{t-1}, \dots, sv_{t-W}) \quad (1)$$

where subscript t represents time; W represents window length; mv represents the manipulated variables; sv represents the sensor variables.

3.3. Network Structure and Hyperparameters

A feedforward network (FFN) was the simplest neural network containing a multilayer perceptron (MLP). In this study, fully-connected FFNs with five hidden layers based on five different models were tested and compared. The five different models in this study were listed as follows:

1. Simple FFN
2. Regularized FFN (R-FFN) with Level 2 regularization (L_2 -norm) [16]
3. Three Fine-Tuned FFNs (FT-FFNs) with L_2 -norm, which transferred from different source-domain models

For FFN, R-FFN, three of FT-FFNs, and the source-domain models, the number of inputs was 76 features, and the number of the output was one. The number of parameters for all models was 32,161. To train from scratch for FFN and R-FFN, the Glorot uniform initialization [17] (the default option of Keras library) was applied. The regularization rate, λ , penalty weighting of L_2 -norm of parameters were fixed to the value of 0.01 for both R-FFN and FT-FFNs.

According to the universal approximation theorem, the rectified linear unit (ReLU) [18], a commonly used activated function, activated width-bounded deep networks with $N + 4$ neurons per layer can approximate any Lebesgue-integrable function, where N is the number of features [19]; in this case, there were 76 features ($mv_t, mv_{t-1}, \dots, mv_{t-W}, sv_{t-1}, \dots, sv_{t-W}$) as the inputs, and there were 80 neurons for each layer. The algorithm for gradient descent optimization was Adam [20]. Additionally, to avoid overfitting during modeling with extremely small datasets, the L_2 -norm was implemented to penalize the loss function. All the modeling works were done in Python language using Keras library.

3.4. Metrics of Performance

For the development of robust and interpretable neural models, both predictive accuracy and interpretability (descriptive accuracy) should be cautiously considered. In

this study, Root-mean-square error (RMSE) was used as the metric of accuracy for the soft sensors and was calculated with the following equation.

$$RMSE_{qv} = \sqrt{\frac{\sum_{n=1}^N (qv_n^{pred} - qv_n^{real})^2}{N_{\text{testing samples}}}} \quad (2)$$

Alongside the metrics for predictive accuracy, the models were also interpreted using post hoc analysis. Post hoc interpretability is a concept and approach of interpretable machine learning [2,21]. It aims to interpret the black boxes globally or locally using domain-knowledge-based models. Local interpretation targets to identify the contributions of each feature in the input towards a specific model prediction and usually attributes a model's decision to its input features (Du et al., 2018). Such interpretation is usually done by posing perturbation on certain features in the input.

For soft sensors regressing input-output relationships of chemical processes, the responding behaviors, which were usually called the process gains, of outputs to the disturbances (perturbations) of inputs should be physically consistent with the chemical engineering domain knowledge. The dynamic process gain (K_{ij}^{dyn}) of quality variables i posed by manipulated variable j can be defined as:

$$K_{ij}^{dyn} = \left(\frac{\Delta qv_{i,t}}{\Delta u_{j,t}} \right)_{u_t | u_{j,t}, u_{t-1}, sv_{t-1}, \dots, u_{t-W}, sv_{t-W}} \quad (3)$$

where $\Delta u_{j,t}$ is the perturbation of manipulated variable j at sampling instant t . For soft sensors of distillation columns, the inputs are the manipulated variables, including reflux rate and reboiler temperature, and the outputs are the qualities of distillate and bottom products. Thus, there should be four process gains, including two main gains ($i = j$) and two interaction gains ($i \neq j$).

It was reasonable, based on the common sense of distillation unit operation, to believe that the sign of dynamic and steady-state process gains to be consistent. Therefore, the percentage of testing samples whose signs of dynamic and steady-state process gains were consistent was defined as the gain consistency (Con_{ij}), as follows:

$$Con_{ij} = \frac{\sum_{n=1}^N \left(Hv \left(sgn \left(K_{ij}^{dyn} \right) sgn \left(K_{ij}^{ss} \right) \right) \right)}{N} \times 100\% \quad (4)$$

where Hv is the Heaviside function. The interpretable soft sensors should be at least with high gain consistency to reasonably respond to the change of manipulated variables.

3.5. Degree of Freedom

There is often an issue of the degree of freedom (DoF), whether the networks were over-parameterized, which implies overfitting. Intuitively, the parameter-to-data ratio, which was conventionally calculated in the form of Equation (5), was an appropriate way to estimate the DoF of networks. However, some works of literature [22,23] provided that the equivalent DoF of the multilayer FFNs only related to the units in the highest hidden layer, with the other layers performing only geometric transformations of data. Thus, instead, we considered the DoF of networks using Equation (6).

$$\frac{\text{Parameter}}{\text{Data}} = \frac{N_{\text{parameters of the network}}}{N_{\text{sample}}} \quad (5)$$

$$\frac{\text{Parameter}}{\text{Data}} = \frac{N_{\text{parameters of the highest layer}}}{N_{\text{sample}}} \quad (6)$$

To consider the effect of the number of learning samples, the neural networks were trained with 360, 270, 180, 90, and 45 samples (with the parameter-to-data ratio of 0.225, 0.3, 0.45, 0.9, and 1.8, respectively), 20% of these samples were used as the validation set during learning.

3.6. Source-Domain Models

Three ASPEN Plus dynamics simulators were constructed to serve as the source domains in this study. The first source domain (D_1) mimicked the actual plant, a debutanizer. The number of trays, type of trays, location of feeds and draws, and feed flowrates were similar to the actual column. The hardware parameters such as actual column diameter, sump size, size of accumulators were obtained using auto-sizing in ASPEN Plus. It should be noted that it is tedious, time-consuming, and somewhat unrealistic to build a rigorous simulation that dynamically matches the real process exactly. Furthermore, we showed that such simulation should be unnecessary with transfer learning techniques in the following discussion. The second (D_2) was also a debutanizer found in the literature [24,25]. The third source domain D_3 was a methanol/water splitter, also given in the literature [24,25]. Since the processes were the same, D_1 and D_2 used the same thermodynamic models, the Peng–Robinson equation of states. For the methanol/water splitter, a UNIQUAC model was used.

These source domains were the process of separation towers. The three source models had the same MVs shown in Table 1. The temperature sensors in D_1 were located as those in the plant. D_2 and D_3 had different numbers of trays, and hence the corresponding SVs were the temperature of trays selected by the relative positions with respect to the condenser and the reboiler. The quality output of D_2 and D_3 was set as the light or heavy component.

In general, the qualities of the distillate and bottom were affected by reflux flowrate and reboiler temperature. Thus, in this study, we focused on the interaction of the distillate (qv_1) and bottom (qv_2) to the reflux flowrate (u_1) and the reboiler temperature (u_2) to the gain signs analysis. For these source domains, the corresponding steady-state process gains shared the same sign. Namely, the main gain signs (K_{11}^{ss} and K_{22}^{ss}) were all negative, and the interaction gain signs (K_{12}^{ss} and K_{21}^{ss}) were all positive, as shown in Table 2. With the datasets generated by these source domains, three source-domain neural network models were pretrained, and their gain consistencies were calculated. There were two potential factors that affect the result of transfer learning: (1) domain similarity between source domains and target domain, and (2) gain consistency of source-domain models. Both effects were considered in this case study. For observation of gain consistency effect, source-domain models with low gain consistencies were intentionally chosen; namely, the Con_{12} for D_3 model was 0% consistent, shown in Table 3.

Table 2. The sign of steady-state gains of source domains.

	$Sgn(K_{11}^{ss})$	$Sgn(K_{22}^{ss})$	$Sgn(K_{12}^{ss})$	$Sgn(K_{21}^{ss})$
D1	−	−	+	+
D2	−	−	+	+
D3	−	−	+	+

Table 3. The gain consistency of source-domain models.

	Con_{11}	Con_{22}	Con_{12}	Con_{21}
D1	100%	100%	100%	100%
D2	99%	100%	89%	81%
D3	100%	100%	0%	100%

3.7. Fine-Tuning Recipe

There is no general criterion for neural network fine-tuning [26]. The most common practices are done by fine-tuning deep layers while freezing shallow ones [12]. However, Li et al. [27] stated that shallow layers also had some effects during domain adaptation. Thus, to obtain better fine-tuning results, the trial-and-error method was used to figure out the best one before performing fine-tuning.

The trial results are shown in Figure 2. As the figure shows, the shallowest layer gave the most significant contribution to minimizing the RMSE in fine-tuning procedures of soft sensors of the distillate and bottom. Note that the six digits in tuning recipes represented positions of the hidden layers and output layer; 1 represented trainable on the specific layer, and 0 represented nontrainable. Generally, the recipes freezing the shallowest layer performed worse than the ones updating their weights, and the ones freezing the intermediate layers performed better than the ones updating their weights. Compared with the shallowest layer, the deeper layers contributed much less effect during fine-tuning, but the deeper layers still gave the contribution to reducing the RMSE. Thus, in this study, the recipe “100011” was chosen, marked by the red arrow in Figure 2, where the output layer, the shallowest hidden layer, and the deepest hidden layer were fine-tuned, while the intermediate ones were frozen. The number of trainable parameters with this recipe was 12,721, and the number of nontrainable parameters was 19,440.

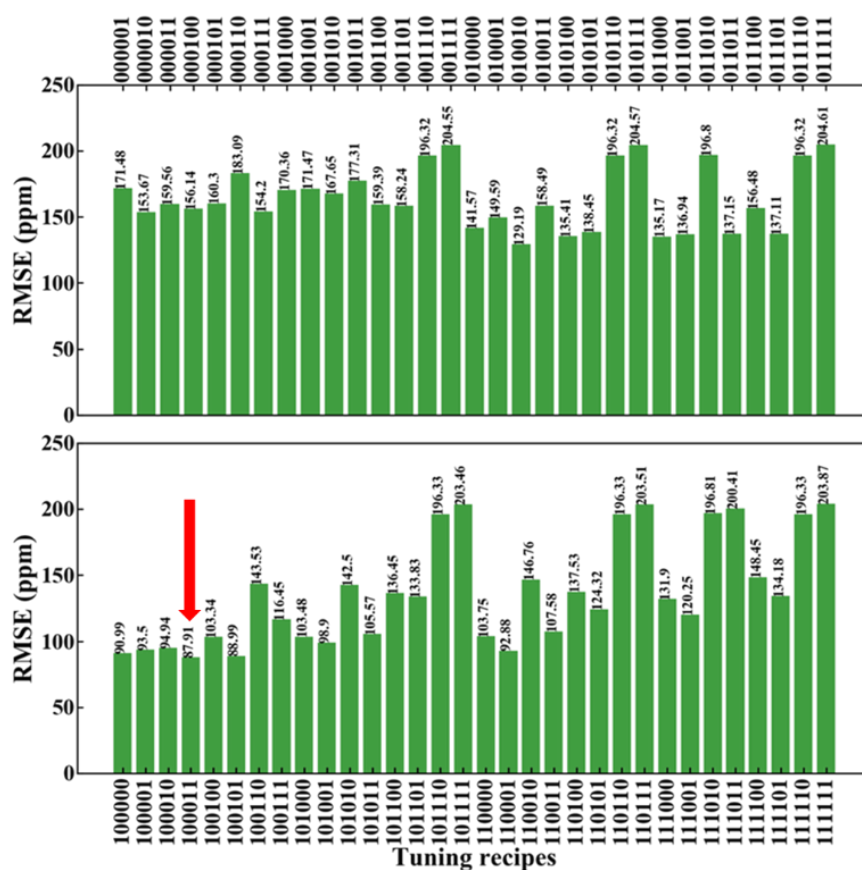


Figure 2. The trial-and-error results for different recipes based on the C₅+ impurities at distillate.

4. Results and Discussion

4.1. Predictive Accuracy

Figure 3 plots the predictive accuracy of the testing set of soft sensors at the distillate. Both R-FFN and three FT-FFNs largely improved the accuracy compared with the simple FFN even when the parameter-to-data ratio was larger than one (the red backgrounded area). FT-FFNs based on different source domains performed slightly better than R-FFN,

and they had similar accuracy, which indicated there was no need for the domain to be very accurate to the target domain. With narrower statistical distribution, FT-FFNs showed better robustness than their pure data-driven counterparts.

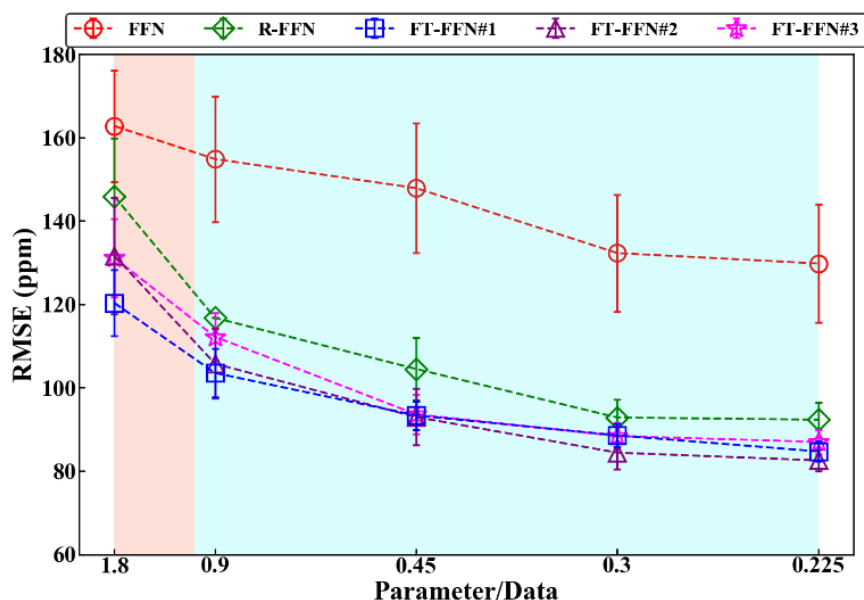


Figure 3. The predictive accuracy of distillate soft sensors.

Figure 4 plots the predictive accuracy of the testing set of soft sensors at the bottom. FT-FFNs still performed better than simple FFN and R-FFN regardless of the size of datasets. However, R-FFN performed better than FFN only when small datasets were used, and it failed to improve accuracy when bigger datasets were available. Such phenomena were called under-fitting, which occurred when λ parameter of L_2 -norm was too strict that ended up with over-penalized. The three FT-FFNs showed a similar order of accuracy when the size of the data set was small. When there is more data, the performance of FT-FFN#3 becomes inferior. The introduction of some useful prior knowledge, regardless of its accuracy, will be very helpful when there is not enough data. As more data are available, a more accurate source domain is required.

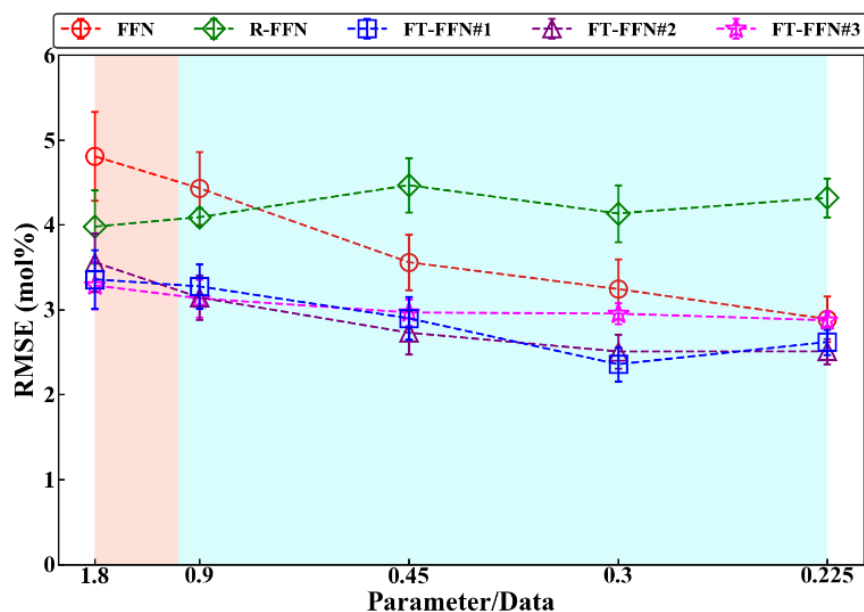


Figure 4. The predictive accuracy of bottom soft sensors.

Figure 5a–d shows the C_4 losses and C_5 losses predictions for R-FFN and FT-FFN#1 with 0.225 of the parameter-to-data ratios. Since the predictions of FT-FFN#2 and #3 were similar to the prediction of FT-FFN#1, only the result of FT-FFN #1 was displayed.

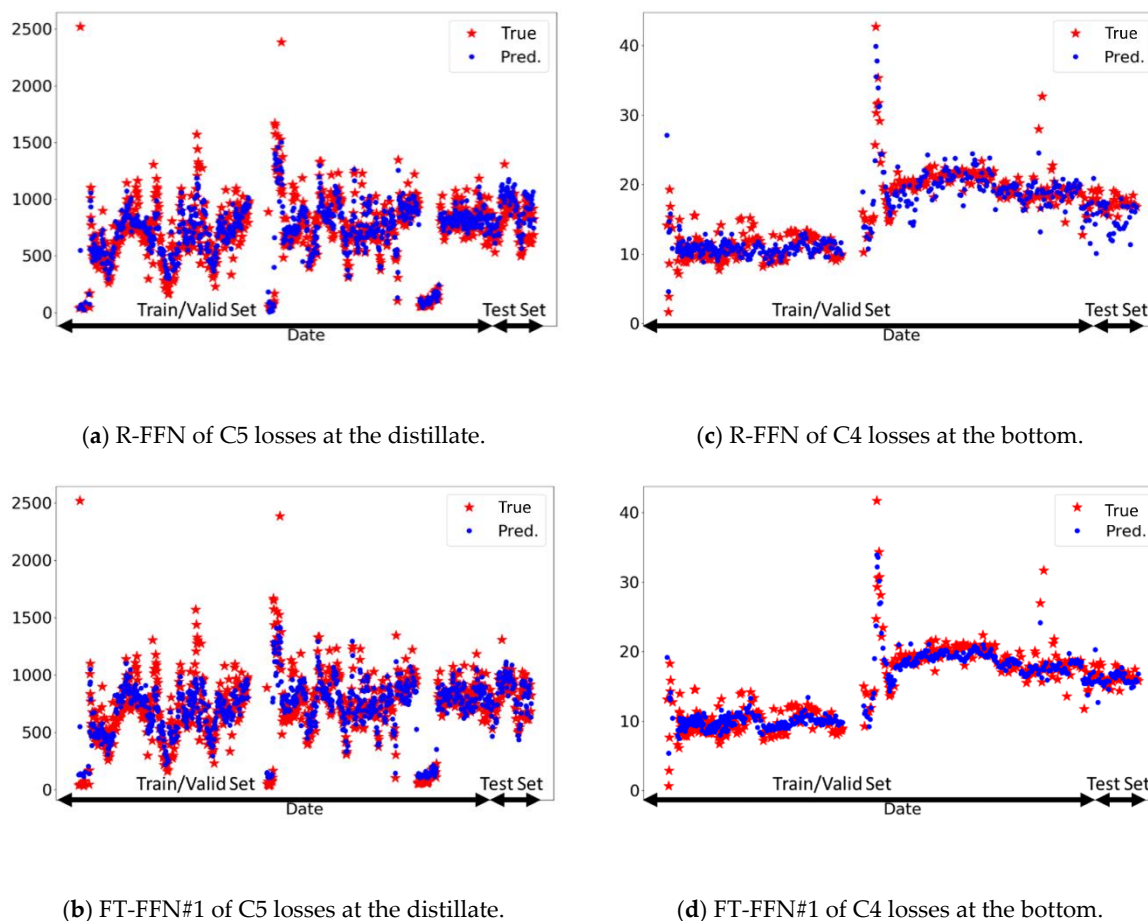


Figure 5. Predictions of regularized feedforward network (R-FFN) and fine-tuned feedforward network (FT-FFN) models for C_4 losses and C_5 losses: (a) R-FFN of C_5 losses, (b) FT-FFN#1 of C_5 losses, (c) R-FFN of C_4 losses, (d) FT-FFN#1 of C_4 losses.

4.2. Physically Interpretability

4.2.1. Main Gain Consistency

Figures 6 and 7 plot the main gain consistency Con_{11} and Con_{22} , respectively, for the different modeling methods. Both R-FFN and FT-FFNs improved the gain consistency even when the parameter-to-data ratio was larger than one; namely, only small datasets were available. Just like predictive accuracy, domain similarity also displayed few effects on obtaining correct directions of main process gains.

4.2.2. Interaction Gain Consistency

Figures 8 and 9 show the interaction gain consistency Con_{12} and Con_{21} , respectively, for the different modeling methods.

The results showed that FT-FFNs improved gain consistency when the source-domain models had gain consistency. In Table 3, the Con_{12} of D_1 and D_2 source-domain models were 100% and 89%, respectively, which ensured the target models based on them to be high gain consistent. Contrarily, the Con_{12} of D_3 source-domain model was 0%, which led to the low gain consistent target models. Additionally, although R-FFN improved predictive accuracy (shown in Figure 3), it not only failed to improve, but low down the Con_{12} instead, leading to low model interpretability.

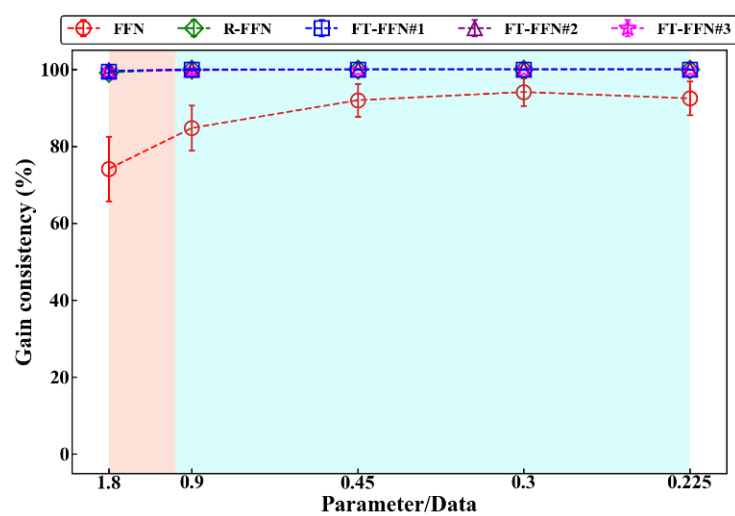


Figure 6. The Con_{11} of different modeling methods.

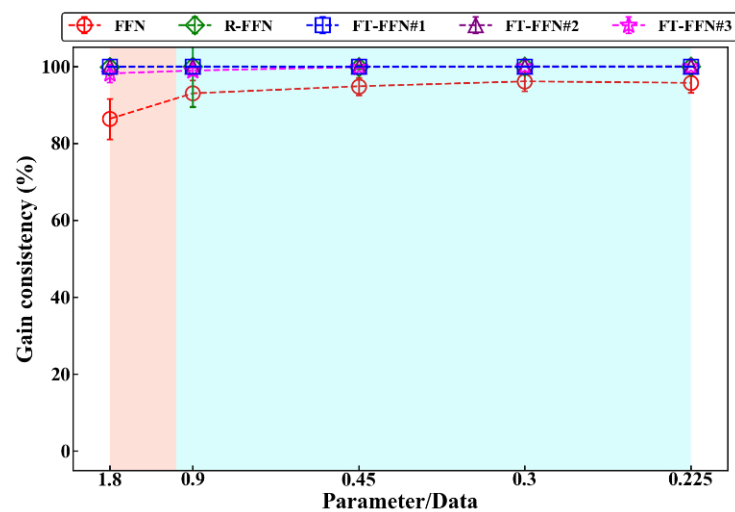


Figure 7. The Con_{22} of different modeling methods.

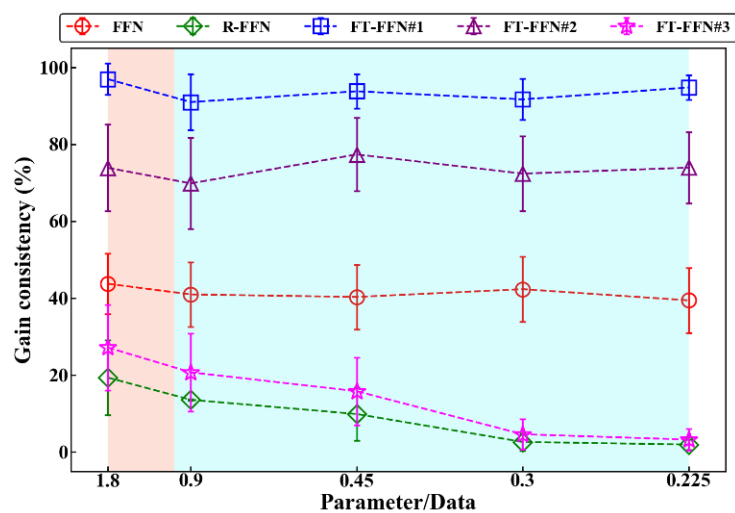


Figure 8. The Con_{12} of different modeling methods.

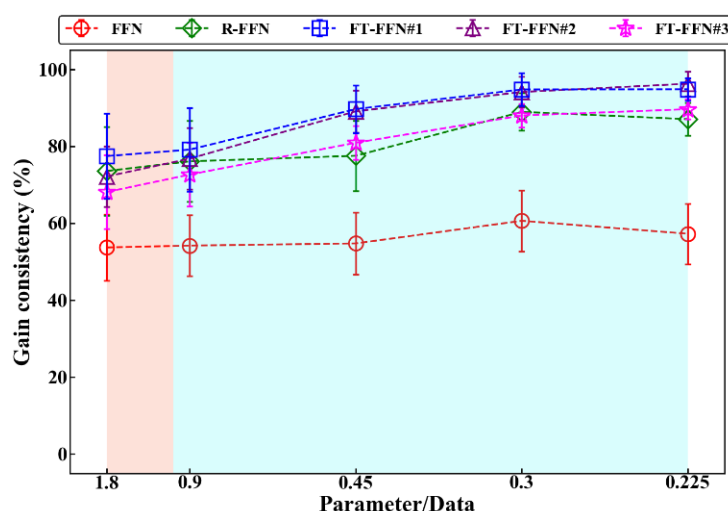


Figure 9. The Con_{21} of different modeling methods.

In Figure 9, three FT-FFNs improved the gain consistency due to the high value of Con_{21} of three source-domain models. R-FFN improved the Con_{21} , but it failed to provide high predictive accuracy when bigger datasets were available (shown in Figure 4).

5. Conclusions

In this paper, a new methodology combining the first-principle simulation and transfer learning was proposed to address the potential problems of overfitting and low interpretability posed by small available datasets often taking place in industrial processes. The method was applied to a real distillation process. It showed its advantages in enhancing both predictive accuracy and physical interpretability over the other conventional deep learning methods, especially when the amount of available real data was small compared to the number of network parameters. Transfer learning was implemented by fine-tuning the weights of networks, freezing inner layers, and updating outer layers. Through fine-tuning, the input-output relationships were modified to accomplish adaptation from the source domains into the target domain. The result showed that the similarities between source and target domains had nearly no effect on fine-tuning results, while the gain consistency of target models was strongly determined by the gain consistency of their corresponding source-domain models. Additionally, the concept and definition of gain consistency were used as the metrics to quantify the physical interpretability of the networks.

Author Contributions: Conceptualization, J.-L.K. and D.S.-H.W.; methodology, J.-L.K. and D.S.-H.W.; software, Y.-D.H.; validation, Y.-D.H., J.-L.K. and D.S.-H.W.; formal analysis, Y.-D.H.; investigation, Y.-D.H.; resources, D.S.-H.W.; data curation, Y.-D.H. and D.S.-H.W.; writing—original draft preparation, Y.-D.H.; writing—review and editing, J.-L.K. and D.S.-H.W.; visualization, Y.-D.H.; supervision, J.-L.K. and D.S.-H.W.; project administration, J.-L.K. and D.S.-H.W.; funding acquisition, J.-L.K. and D.S.-H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ministry of Science and Technology, grant number MOST 109-2636-E-224 -001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kadlec, P.; Gabrys, B.; Strandt, S. Data-driven soft sensors in the process industry. *Comput. Chem. Eng.* **2009**, *33*, 795–814. [CrossRef]
2. Du, M.; Liu, N.; Hu, X. Techniques for interpretable machine learning. *Commun. ACM* **2019**, *63*, 68–77. [CrossRef]

3. Udugama, I.A.; Gargalo, C.L.; Yamashita, Y.; Taube, M.A.; Palazoglu, A.; Young, B.R.; Gernaey, K.V.; Kulahci, M.; Bayer, C. The role of big data in industrial (bio)chemical process operations. *Ind. Eng. Chem. Res.* **2020**, *59*, 15283–15297. [\[CrossRef\]](#)
4. Fortuna, L.; Graziani, S.; Xibilia, M.G. Comparison of soft-sensor design methods for industrial plants using small data sets. *IEEE Trans. Instrum. Meas.* **2009**, *58*, 2444–2451. [\[CrossRef\]](#)
5. Kadlec, P.; Gabrys, B. Local learning-based adaptive soft sensor for catalyst activation prediction. *AIChE J.* **2010**, *57*, 1288–1301. [\[CrossRef\]](#)
6. Fujiwara, K.; Kano, M.; Hasebe, S.; Takinami, A. Soft-sensor development using correlation-based just-in-time modeling. *AIChE J.* **2009**, *55*, 1754–1765. [\[CrossRef\]](#)
7. Bishop, C.M. *Neural Networks for Pattern Recognition*; Oxford University Press, Inc.: New York, NY, USA, 1995.
8. Pan, C.; Dong, Y.; Yan, X.; Zhao, W. Hybrid model for main and side reactions of p-xylene oxidation with factor influence based monotone additive SVR. *Chemom. Intell. Lab. Syst.* **2014**, *136*, 36–46. [\[CrossRef\]](#)
9. Dong, Y.; Yan, X. Hybrid model of industrial p-xylene oxidation incorporated fractional kinetic model with intelligent models. *Ind. Eng. Chem. Res.* **2013**, *52*, 2537–2547. [\[CrossRef\]](#)
10. Pan, S.; Yang, Q. Une enquête sur l'apprentissage par transfert. *Trans. IEEE Ingénierie Connaiss. Données* **2010**, *22*, 1345–1359.
11. Bengio, Y. Deep learning of representations for unsupervised and transfer learning. In Proceedings of the ICML Workshop on Unsupervised and Transfer Learning, Edinburgh, UK, 26 June–1 July 2012; pp. 17–36.
12. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *Adv. Neural Inf. Process. Syst.* **2014**, *7*, 3320–3328.
13. Qiu, P.; Huang, B.; Dai, Z.; Wang, F. Data-driven analysis and optimization of externally heat-integrated distillation columns (EHIDiC). *Energy* **2019**, *189*, 116177. [\[CrossRef\]](#)
14. Robinson, P.J.; Luyben, W.L. Simple dynamic gasifier model that runs in aspen dynamics. *Ind. Eng. Chem. Res.* **2008**, *47*, 7784–7792. [\[CrossRef\]](#)
15. Kaneko, H.; Funatsu, K. Moving window and just-in-time soft sensor model based on time differences considering a small number of measurements. *Ind. Eng. Chem. Res.* **2015**, *54*, 700–704. [\[CrossRef\]](#)
16. Luo, X.; Chang, X.; Ban, X. Regression and classification using extreme learning machine based on L1-norm and L2-norm. *Neurocomputing* **2016**, *174*, 179–186. [\[CrossRef\]](#)
17. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Playa Blanca, Spain, 9–11 April 2018.
18. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
19. Lu, Z.; Pu, H.; Wang, F.; Hu, Z.; Wang, L. The expressive power of neural networks: A view from the width. *Adv. Neural Inf. Process. Syst.* **2017**, *6*, 6231–6239.
20. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference Learn & Represent (ICLR), San Diego, CA, USA, 5–8 May 2015.
21. Murdoch, W.J.; Singh, C.; Kumbier, K.; Abbasi-Asl, R.; Yu, B. Interpretable machine learning: Definitions, methods, and applications. *arXiv* **2019**, arXiv:190104592.
22. Bartlett, P.L. The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Trans. Inf. Theory* **1998**, *44*, 525–536. [\[CrossRef\]](#)
23. Ingrassia, S.; Morlini, I. Neural network modeling for small datasets. *Technometrics* **2005**, *47*, 297–311. [\[CrossRef\]](#)
24. Wood, R.; Berry, M. Terminal composition control of a binary distillation column. *Chem. Eng. Sci.* **1973**, *28*, 1707–1717. [\[CrossRef\]](#)
25. Gani, R.; Ruiz, C.; Cameron, I. A generalized model for distillation columns—I: Model description and applications. *Comput. Chem. Eng.* **1986**, *10*, 181–198. [\[CrossRef\]](#)
26. Guo, Y.; Shi, H.; Kumar, A.; Grauman, K.; Rosing, T.; Feris, R. SpotTune: Transfer learning through adaptive fine-tuning. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA, 2019; pp. 4800–4809.
27. Li, Y.; Wang, N.; Shi, J.; Liu, J.; Hou, X. Revisiting batch normalization for practical domain adaptation. *arXiv* **2016**, arXiv:160304779.