A Framework of Soft Sensor Systems with Machine Learning

Woo Young Moon
Digital Platform
SK Inc
SK u-Tower 9, Seongnam-si, Gyeonggi-do, Korea
moon.wooyoung@gmail.com

Soo Dong Kim*
School of Software
Sooongil University
369 Sangdo-Ro, Seoul, Korea
sdkim777@gmail.com

Abstract—A Soft-sensor is a means or a method to predict response variables that are difficult to predict by using the data of variables that can be easily obtained[1]. There have been increasing technical demands on improving the accuracy of soft sensors and reducing development complexity since the soft sensor is more cost-effective and easier to collect data than hardware sensors. However, few systematical methods to select the optimal set of features for building soft-sensor models have been proposed, although feature selection is the most essential factor to improve the development quality of the soft sensors.

Therefore, this thesis is to present a systematic method for generating soft-sensor models to enhance the model accuracy by measuring similarities of soft-sensor models and selecting the best feature set from the similarity analysis. The proposed method utilizes ML technologies to build soft-sensor models and presents an algorithm to build soft-sensor models by reusing existing and similar soft-sensor models to improve the accuracy of the soft-sensor models.

Keywords—soft sensor models, framework, ML

I. INTRODUCTION

Sensors are the main means to acquire observations on the environment, transform the observations into quantitative measures using numeric units, and deliver the measures. A soft sensor is defined as a software component that emulates some hardware sensors.

It is similar since it provides the measurement values like hardware sensors. However, soft sensors do not actually measure the status of the underlying environment, rather, they produce projected and computed values that are specially designed for special purposes. This becomes the key motivation for the Framework of Soft Sensors.

Soft sensors can work together with hardware sensors, to acquire meaningful and useful information for increasing task performance and reliability. They enable the real-time projection of sensor data, overcoming the time delays introduced by slow hardware sensors or lab experiments. Thus, they improve the performance of the control strategies. This becomes another key motivation for improving the performance of the Framework of Soft Sensors.

Hence, the benefits of the Framework of soft sensors:

- Methods for Automatic Feature Identification and Selection
- Effective Guidelines for Data Acquisition and its preprocessing

II. RELATED WORK

Yu’s work proposes a Bayesian inference based two-stage support vector regression (BI-SVR) approach to develop soft sensors in batch or fetch-batch processes[2]. The proposed approach effectively solves the issue of the significant model deviations resulting from the measurement uncertainty, but it can be elaborated by improving the accuracy of the soft sensor.

Deng's work presents the design of the soft-sensor modeling framework including the offline mode clustering for the unobserved multimode data, and the online mode identification for the query sample[3]. The approach can be improved in two aspects: The proposed method can be generalized to many other similar data-driven soft sensor modeling methods, and it does not reflect the state changes of the actual process.

Popli’s work proposes a process morning scheme using fundamental models and online measurements from a soft-sensor network[4]. The proposed scheme supports multivariate image data to measure real-time grading and recovery using vector machine classification and regression. Existing works on presenting soft sensor frameworks focus on a specific industry domain, and they consider how to extract features and how to select input variables, which affects the qualities of the soft sensors.

Sheng's work presents an active learning framework for ensemble Gaussian process regression models of soft sensor design[5]. The traditional ensemble modeling methods are highly dependent on labeled data samples. To improve the performance of the proposed framework, they selected the most representative and uncertain samples with additional process information. However, the framework can be improved by putting efforts to minimize the human effort associated with labeling.

Pan's work presents a soft sensor development framework for online quality prediction of nonlinear industrial processes[6]. In the proposed framework, input variables and quality information are taken to measure similarity. A data set
consisting of physical sensors and laboratory data may be sufficient to explore similarities. However, the proposed framework can be enhanced with more detailed methods for efficiently updating and maintaining the database and input selection as features in order to ensure modeling effectiveness on the common application in the industrial field.

Most of these approaches are devised for specific industry domains. In addition, these works only address the part of the design of the soft sensor, implementation, and operation guidelines. There is a need to put more effort to enhance these soft sensors to maintain high qualities such as accuracy. To address these limitations, the thesis is to propose a framework for designing a soft-sensor, which can be generally applied across domains by realizing a feature extraction method for designing soft-sensors.

III. PROCESS TO GENERATE SOFT SENSOR MODELS

A process to generate machine learning models for soft sensors is defined by considering the typical machine learning process and the characteristics of soft sensors. The process consists of four steps as shown in Fig. 1, and each step is defined with an input, the output, and instructions, in this paper shows the instructions

- **Missing values**: any failure in a hardware sensor or unavailability of the sensor due to its maintenance or removal, communication problems to the database and errors in the database itself.

- **Measurement noise**: Measurement noise is another common symptom observed in industry data. Most approaches to soft sensor development try to deal with these measurement noises in the preprocessing stage of data processing.

To resolve the missing data issue, deleting a particular row and calculating the mean. Next, we need to normalize the data to make sure that the data have the same scale. The data in a sensor-based application database varies in size depending on the unit and process characteristics adopted. The learning accuracy relies on variables with a larger scale than variables with a smaller scale. The methods used in this paper are two general methods such as min-max and z-score normalization.

B. STEP 2. Select Features for Soft Sensor Models

The first thing to do for defining a feature set is to analyze the relevance of the available physical sensor dataset to the target soft sensors. After analyzing the relevance, only the data sets that are specific and relevant to target soft sensors are identified. That is because part of the data elements for the physical sensors is relevant to the process and characteristics of soft sensors. By doing this, we only include the essential features while keeping the size of the feature set minimal.

Typically, the data measured from the same kind of sensors have similar values. For example, the temperature result of an adjacent temperature sensor provides a similar collinear measurement. We must remove this collinear data to create more effective and well-performing soft sensor models.

In this paper, a Pearson Correlation Coefficient (PCC)][8][9] is used to find the relationship between the physical sensor dataset and the target soft sensors, and Variance Inflation Factor (VIF) is used to rank the importance of the features. After evaluating their importance with these methods, the Incremental Feature Selection (IFS) approach is applied to determine the optimal number of features.

**TABLE I.** shows the algorithm for evaluating feature importance by using PCC and VIF.

<table>
<thead>
<tr>
<th>TABLE I. ALGORITHM FOR FINDING FEATURE IMPORTANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Preprocessed the physical sensor data</td>
</tr>
<tr>
<td><strong>Output:</strong> [ \mathbf{X, Y} ] feature set</td>
</tr>
<tr>
<td><strong>SSR:</strong> the sum of squared residuals, <strong>SST:</strong> the total sum of squares</td>
</tr>
<tr>
<td><strong>// Step 2-1. Initialize.</strong></td>
</tr>
<tr>
<td>1. Create Physical Sensor List ( P )</td>
</tr>
<tr>
<td>2. Create Selected Soft Sensor Features ( D )</td>
</tr>
<tr>
<td>3. Set ( SSR, SST, VIF, i = 0 )</td>
</tr>
<tr>
<td>4. Set physical sensor ( X ), label ( Y )</td>
</tr>
<tr>
<td><strong>// Step 2-2. Compute PCC and VIF.</strong></td>
</tr>
<tr>
<td>5. While not compute do</td>
</tr>
<tr>
<td>6. Get data ( X ) and ( Y ) from database using ( Pi )</td>
</tr>
<tr>
<td>7. Compute ( sd(X), sd(Y), covariance(X,Y) )</td>
</tr>
<tr>
<td>8. Compute ( SSR = \text{the sum of squared residuals}, SST = \text{total sum of squares} )</td>
</tr>
<tr>
<td>9. Compute Pearson correlation coefficient(( \rho )) and VIF</td>
</tr>
<tr>
<td>10. If ( \rho &gt; 0.75 ) and VIF &lt; 10 then store related data set to ( Di )</td>
</tr>
<tr>
<td>11. If additional Pi is necessary, continue ( i++ );</td>
</tr>
<tr>
<td>12. End while</td>
</tr>
</tbody>
</table>

C. STEP 3. General Soft Sensor Model

First, a suitable machine learning algorithm is selected to generate soft sensor models. We use the Ensemble model
combining SVM and RF models since the quality or performance of this Ensemble model is generally higher than other algorithms.

As an example, a soft sensor can be designed by considering physical sensor set A and laboratory measurement set B. Those measurement sets consist of physical sensors and laboratory experiment results. With a machine learning algorithm like Random Forest (RF), the proposed framework builds soft sensor models from the training dataset. As a result of training soft sensor models with machine learning algorithms, the framework generates a model that is most likely matched to a given experiment result set. We present a soft sensor model. The representation of a soft sensor model is the result of independent physical sensor p(t) and the laboratory prediction results. This schema of the soft sensor model is shown in Fig. 2.

![Fig. 2. Schematic Diagram of a Soft Sensor model](image)

Here, the candidate model is designed as a black box, and it returns predictions based on the physical sensor and laboratory measurements of an industrial process. During the modeling process, the relationship between the physical sensors and laboratory data of the sensor-based application database is emphasized while ignoring the complex internal structures. To train a candidate model for the online soft sensors, it is needed to have a new algorithm for measuring the accuracy of the candidate models and the accuracy of the individual models on data. shows the algorithm for evaluating feature importance. TABLE II. Shows the algorithm for generating an online candidate model.

**TABLE II. ALGORITHM FOR GENERATING AN ONLINE CANDIDATE MODEL**

| Input: candidate data set D={(x_t, y_t) t=1}^T, divided into M candidate model each on of size m; model candidate input p(); an model learner; maximum number of model N, model error measure e(); |
| Output: candidates' model |

```
// Initialize.
Set the candidate as γ = 0, n=1
// Obtain a candidate model.
f_n ← get a candidate model trained with Dn;
get the input features of f_n on Dn using the input methods p();
get the error of f_n on Dn using the error function e();
set γ ← γ ∪ {f_n}, and n ← n+1
While n ≤ M do
get the output prediction of γ based on the model’s weights;
get the physical sensor list using p();
get the prediction error of all the soft sensor on Dn using e();
Weight all the soft sensor based on their calculated errors;
Retain all the current soft sensor using Dn;
Training a new soft sensor f_n with Dn;
```

D. STEP 4. Evaluate Soft Sensor

This step is to determine the best fit model for the soft sensors. We first need to choose the right evaluation metrics. The model evaluation task is to select an algorithm that is valid for the soft sensor model developed through the data set. The most common evaluation methods are the Mean Absolute Error (MAE), the Root Mean Squared Error (RMSE), and the Mean Absolute Percentage Error (MAPE). The predefined performance index is defined based on the characteristics of each output value. Next, the soft sensor models trained in the previous step are evaluated with the determined evaluation metrics. In the context of Soft Sensors, accuracy and error rates are enough for performance evaluation because the laboratory experiments are targets. Fig. 3 shows an example of selecting the best trained model which prediction result is the closest to the given laboratory results.

![Fig. 3. Select most appropriate for soft sensor design](image)

IV. DESIGN AND IMPLEMENTATION OF THE FRAMEWORK

This section presents a framework design with proposed tactics and solutions to technical challenges in framework development. And we show how the soft sensor design is implemented.

A. Architecture of the Framework

Architecture is a representation of a structural aspect of the system. The proposed context diagram of the Soft sensor design framework can be shown in Fig. 4.

![Fig. 4. Context diagram of the Framework](image)
we adopt the MVC style, one of the most representative architecture styles applied in complex systems to ensure high QoS, as shown in Fig. 5.

**Fig. 5. Layers architecture of the Soft Sensor Framework**

View Layer offers UIs to soft sensor designers depending on the roles of the soft sensor framework. Control Layer coordinates the workflow performed by a soft sensor framework. Model Layer manages entity-type objects such as sensor, parameter, session, and learning results, and it interacts with the soft sensor framework Database which stores entity-type objects.

**B. Design for Functional View**

In this section, we present the functional view of the framework by describing a component diagram with some essential components and their relationships. With the consideration of the separation of concerns principle, we derive 10 components and their interfaces considering the group of functionalities. These interfaces are used to resolve variability. The identified interface in each component performs the same function regardless of internal changes. The functional view, represented with a UML component diagram, is depicted as shown in Fig. 6.

**Fig. 6. Component diagram of the framework**

There are 10 derived components and their descriptions are specified as follows: Sensor Connector, Sensor Data Acquirer, Preprocessor, Learning Engine, Source Manager, Sensor Manager, Parameter Manager, Result Manager, Online Analyzer, Adaptive manager.

**C. Common Functionality of the Framework**

In the soft sensor design framework, there are common and variable tasks for each step. Soft Sensor Framework must be developed with commonality and variability in order to increase the reusability and applicability[9]. Applying machine learning techniques in soft sensor design domains commonly follows the procedure of 1) Pre-processing physical sensor data, 2) Feature selection for designing soft sensor, 3) Learning physical sensor data for soft sensor design, 4) Model evaluation and operation. TABLE III. shows Commonality and Variability analysis results of Soft Sensor Framework.

**TABLE III. COMMONALITY AND VARIABILITY**

<table>
<thead>
<tr>
<th>Component</th>
<th>Commonality</th>
<th>Variability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Connector</td>
<td></td>
<td>Sensor Type &amp; Implementation Configuration</td>
</tr>
<tr>
<td>Sensor Data Acquirer</td>
<td>✓</td>
<td>Dynamic Update and implementation</td>
</tr>
<tr>
<td>Preprocessor</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Learning Engine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source Manager</td>
<td>✓</td>
<td>Algorithm Type,</td>
</tr>
<tr>
<td>Sensor Manager</td>
<td>✓</td>
<td>Hyper parameters configuration</td>
</tr>
<tr>
<td>Parameter Manager</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Result Manager</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Adaptive Manager</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Among Commonality, the Preprocessor component and Sensor Data acquirer component must consider reusability and applicability in designing common functions.

**D. Variable Functionality of the Framework**

To customize soft sensors effectively and efficiently, the framework needs to learn and evaluate physical sensor data. The goal of the soft sensor system is to maintain or improve process conditions in the industry such as the chemical and bio through monitoring. The proposed framework covers the tasks of designing soft sensors by using machine learning techniques, as shown in TABLE III. Techniques for dynamically customizing machine learning models are used to assign external transformation points to the framework's transformation points. To support the variability of the framework, we provide the functionality for learning physical sensor data and the functionality for analyzing new sensor observations.

**E. Design for Behavior View**

In this section, the behavior view is focused on how to design for handling the third technical challenge of developing the framework. Fig. 7 shows an activity diagram for acquiring data set for learning. First, the framework acquires raw data for Soft Sensor design. After the physical data and laboratory data have different sampling cycles, so the framework transfers and merges them into a record based on an appropriate frequency. After the framework performs preprocessing (normalization) of the acquired sensor data. After normalizing data, the framework detects outliers. Then, it stores the results of preprocessing.
Fig. 7. Activity Diagram for Acquire data set for Soft Sensor Design

Fig. 8 shows the activity diagram for selecting features and generated used for Soft Sensor Models.

Fig. 8. Activity Diagram for A) select features for Soft Sensor Models B) generate Soft Sensor Models

Fig. 9 shows the activity diagram for selecting features used for Soft Sensor Models. First, the framework selects the last trained model.

Fig. 9. Activity Diagram for evaluate and maintain Soft Sensor Models

After the framework evaluates the trained model using performance evaluation methods. After the framework selects the most appropriate soft sensor model and maintains soft sensor models. This framework evaluates the accuracy of the soft sensor models. The Adaptive Manager determines whether online (or real-time) changes are necessary. If online changes are needed, Learning Engine builds a new model and deploys the model via the provider.

Fig. 10 shows the control flow of soft sensor model training and operation. When the framework receives a request for soft sensor design, the framework trains the models, the model evaluator evaluates the accuracy of the models and checks whether online changes are needed or not. If online changes are needed, the framework builds a new model and deploys the model through the provider. If not, it transfers soft sensor estimation results to the operator.

F. Architectural Tactics for Handling Dynamic Model Change

The first tactic is realized by the adaptive manager. The proposed framework adopts a strategy pattern to dynamically load components and apply new algorithms according to the adaptive manager's request. The Adaptive Manager plays one of the important roles in the framework design. Strategy pattern is applied to dynamically change functions as shown in Fig. 11.

It is necessary to dynamically change and execute the diversity of algorithms at runtime. To do this, the analysis class used to be designed to dynamically load the right classes using a reflection technique. The framework provides the ability to inspect and modify the runtime behavior of the applications. This approach solves the problem of evaluating and applying the soft sensor models with an adaptive algorithm with a strategy pattern.

Fig. 11. Dynamic Algorithm change based on the strategy pattern
G. Architectural Tactics for a High-Accuracy Soft Sensor

Predicting results accurately is one of the most important and intrinsic requirements in designing soft sensors. If the accuracy of the model falls below a certain criterion, it should be replaced with another model with higher accuracy. To do this, the adaptive manager and learning engine are designed to select one of the several algorithms or to be ensemble so that an algorithm with high accuracy can be selected.

If the soft sensor is inaccurate due to a new physical sensor, process change, or misuse of the model, the proposed framework is designed to adapt a new model. The model that returns the optimal result is deployed to the operating model as shown Fig. 12. If the accuracy of the soft sensor is lower than the observer’s expectation, the adaptive manager evaluates the model and deploys the new model.

![Fig. 12. Dynamic component adaptation for a high-accuracy soft sensor](image)

V. EXPERIMENTS AND LESSONS LEARNED

In this section, a topic addressed from the theoretical point of view in Chapter 3 and Chapter 4 is reconsidered by taking an experiment and assessment. Some of the methods discussed will be applied to data referring to the process described in the previous section.

A. Experiment Scenarios and Environments

The purpose of the experiment is to show that the process of the soft sensor is well-designed and provides high accuracy. In addition, we aim to examine whether the proposed framework of soft sensors can overcome the technical challenges of the soft sensor. We evaluate the accuracy of a soft sensor with proposed algorithms based on the soft sensor development, to see whether the soft sensor framework can yield better performance.

We hypothesized that the soft sensor model would have good performance. The hypothesis was tested by performing soft sensor modeling. The physical sensor’s data set in the sensor-based application database include measurements of the flow meter, level gauge, thermometer, and pressure gauge. The number of sensor data is about 2,544 per physical sensor and the number of laboratory data is about 1,400 per sample.

The laboratory results for physical sensor contents are provided daily, so the soft sensor provides an average value on a daily basis and evaluates them. The summary of experiment sets of physical sensors for designing the soft sensor is represented in TABLE IV.

<table>
<thead>
<tr>
<th>Data Set for Paraffin property Soft Sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
</tr>
<tr>
<td>Physical Sensor</td>
</tr>
<tr>
<td>Laboratory data</td>
</tr>
</tbody>
</table>

The physical sensor’s data set in the sensor-based application database include measurements of the flow meter, level gauge, thermometer, and pressure gauge. The number of sensor data is about 2,544 per physical sensor and the number of laboratory data is about 1,400 per sample. We have performed experiments to verify the performance and evaluation of the proposed algorithm. The experiments were confirmed based on the following processes. We have conducted the tests through data preprocessing, feature selection, and model evaluation. In the experiment, we investigated the effectiveness of the proposed design and the accuracy of the soft sensor models. Soft sensor history data is used for soft sensor modeling. After preprocessing of these raw data, typical data sets are chosen to apply as shown in TABLE V.

| TABLE V. EXAMPLE OF EXPERIMENT DATASET FOR SOFT SENSOR |
|-------------|-------------|
| $x$ | $y$ |
| $x_1$ | $x_2$ | $x_3$ | ... | $y_1$ | $y_2$ |
| 8/2/201 | 6 5:00 | 67.11 | 24.63 | 42.10 | ... | 58.4 | 12.3 |
| 8/3/201 | 6 5:00 | 7434 | 191 | 42.17 | ... | 57.6 | 13.1 |
| 8/4/201 | 6 5:00 | 66.96 | 24.42 | 42.11 | ... | 57.6 | 13 |

B. Applying Steps for Analytics and Interpretations

First, we eliminated invalid data elements such as missing values and data outliers. The data elements are collected from Sensor Connectors and Sensor Data Acquirer during the system operations. When the system stops running due to various reasons such as system maintenance, the measured data values would become displayed as zero. During the operations, their values would be zero or higher. And next, the set of essential features for generating soft sensor models is identified. The selection of the most appropriate features is important for at least 3 reasons. First, the right set of features can yield highly performing-machine learning models. Secondly, the right selection of features can also contribute to reducing the complexity of models. It can also lead to the selection of a minimal set of essential soft sensors for a given application. Thirdly, the right selection of features provides vital information or indications in choosing the minimal set of essential sensors. It is a well-known fact that using a smaller number of sensors to get the environmental status is always better than doing the same work with a larger number of sensors.
And next, the Online Analyzer of the Soft Sensor framework evaluates the accuracy of the soft sensor prediction results and determines if a model change is needed or not. There exists a variety of different soft sensor models and hence their evaluation metrics could also be various. Hence, evaluation metrics specific to the target soft sensors should be determined and applied. An example of evaluating soft sensors for processing plants is shown in Fig. 13.

![Fig. 13. Result of Evaluating Soft Sensor Model](image1)

C. Assessment of the framework

The trend of the soft sensor value is consistent with the actual value, as shown in Fig. 14. In other words, paraffin soft sensors based on ML models can estimate product quality variables. We could provide reliable quality property data to operators in real time. It allows operators to improve the product quality by tuning the chemical process conditions based on predicted values from the Soft Sensor.

![Fig. 14. Real-time property estimation examples using Soft Sensor](image2)

After measuring the quality of the product for one month from January 6th to February 5th, comparing the quality characteristics and the predicted values, we figure out that the prediction accuracy is pretty high. When analyzing estimated errors, about 70% of all data have error rates that are mostly lower than 0.1 as shown in Fig. 15.

![Fig. 15. The Error Distribution of Soft Sensor](image3)

VI. CONCLUSION

To reduce the time and cost of Soft Sensor development and to let developers just focus on the performance of the soft sensor, it is necessary to automate all steps in the design of the soft sensors or support them for easy development.

The proposed framework is presented to cover the soft sensor’s design process and to automate the process. The main technical challenge of the framework is to improve applicability or reusability so as to make the framework applied to designing various soft sensors and maximize the accuracy of the soft sensor predictions. We address these through the Soft Sensor development process and the framework design through the experiment and assessment.

However, there are limitations to the estimated results due to various data characteristics. In the future, it is required to review the applicability of the improvement and optimization aspects through expansion and linkage of the estimation model and to secure a methodology for estimating the process/quality changes according to the changes in operating conditions and establishing operation guidelines with case studies for the utilization scenario.

REFERENCES


