Design of Adaptive Soft Sensors using Recursive Just-in-Time Learning Framework with Non-linear Local Modeling Strategy for Quality Monitoring in Process Industries

THESIS

Submitted in partial fulfillment

of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

S. VENKATAVIJAYAN

Under the Supervision of

Dr. Ajaya Kumar Pani

and co-supervision of

Prof. Hare Krishna Mohanta

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

2023

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the thesis entitled **"Design of Adaptive Soft Sensors using Recursive Just-in-Time Learning Framework with Non-linear Local Modeling Strategy for Quality Monitoring in Process Industries"** and submitted by **S. Venkatavijayan** ID. No. **2016PHXF0017P** for the award of PhD Degree of the Institute embodies the original work done by him under our supervision.

Signature in full of the Supervisor _____

Name in capital block letters

Designation

DR. AJAYA KUMAR PANI
Assistant Professor
Department of Chemical Engineering
BITS-Pilani, Pilani Campus
Rajasthan

Signature in full of the Co-Supervisor _____

Name in capital block letters

Designation

PROF. HARE KRISHNA MOHANTA
Associate Professor
Department of Chemical Engineering
BITS-Pilani, Pilani Campus
Rajasthan

Date:

Acknowledgements

First, I like to express the gratitude towards my Supervisor Dr. Ajaya Kumar Pani (Assistant Professor, Department of Chemical Engineering) and Co-supervisor Prof. Hare Krishna Mohanta (Associate Professor, Department of Chemical Engineering) for their immense patience, constant encouragement and support throughout my PhD work.

I thank my Doctoral Advisor Committee members Prof. Bijay Kumar Rout (Professor, Department of Mechanical Engineering) and Prof. Pratik N. Sheth (Professor & Head, Department of Chemical Engineering) for giving me valuable suggestions and advices during semester presentations and thesis correction to improve my research work.

I thank my Examiners Prof. Kishalay Mitra, Professor & Head, Department of Chemical Engineering, Associated faculty, Department of Climate Change, Indian Institute of Technology, Hyderabad and Prof. Ramesh C Panda, Department of Chemical Engineering, Central Leather Research Institute, Council of Scientific and Industrial Research, Chennai for examine my PhD thesis and provide me critical comments to improve my research work for the award of PhD degree.

I am thankful to Prof. V. Ramagopal Rao, (Vice chancellor, BITS Pilani), Prof. Souvik Bhattacharyya (Former Vice chancellor, BITS Pilani), Prof. Sudhirkumar Barai (Director, BITS Pilani, Pilani campus), Prof. A. K. Sarkar (Former Director, BITS Pilani, Pilani campus) and Col. Soumyabrata Chakraborty (Registrar, BITS Pilani) for giving me the opportunity to carry out my research work.

I thank Prof. M.B. Srinivas (Dean, Academic - Graduate Studies Research Division, BITS-Pilani), Prof. Srinivas Krishnaswamy (Former Dean, AGSRD, BITS-Pilani), Prof. Shamik Chakraborty (Associate dean, AGSRD, BITS-Pilani, Pilani campus), Prof. Jitendra Panwar (Former Associate Dean, AGSRD, BITS-Pilani, Pilani campus) and Prof. Sanjay Kumar Verma (Dean, Administration, BITS-Pilani, Pilani campus) for their support.

I also thank former Heads of Department of chemical engineering: Prof. Arvind Kumar Sharma (for providing me critical comments in PhD thesis), Prof. Suresh Gupta and Prof. Banasri Roy for

their kind support and suggestions throughout my research work. I thank Prof. Srinivas Appari, Prof. Bhanuvardhan Reddy, Dr. Amit Jain and Prof. Krishna Etika, with whom I was associated as a teaching assistant in various phases of my PhD work. I thank Dr. Priya C. Sande, for providing me several suggestions and advices for improvement at various phases of my research work. I also thank other faculty members of the chemical engineering department, Prof. Pradipta Chattopadhyay, Prof. Smita Raghuvanshi, Dr. Somak Chatterjee, Dr. Sarbani Ghosh, Dr. Mohit Garg and Dr. Jay Pandey for their support and cooperation.

I thank Dr. Andrijic Ujevic Zeljka (Department of Measurements and Process Control, Faculty of Chemical Engineering and Technology, University of Zagreb, Croatia) for generously providing the naphtha splitter section datasets for this research work.

I thank Prof. Surekha Bhanot (Electrical & Electronics Engineering department, BITS-Pilani, Pilani campus), Prof. Kumar Neeraj Sachdev (Former Associate Dean, Student Welfare Division, BITS-Pilani, Pilani campus), Prof. P. Srinivasan (Former Dean, Practice School Division), Dr. G. Muthukumar (Associate Dean, Practice School Division) and Prof. Madhusree Kundu (Chemical Engineering department, NIT-Rourkela, Odisha) for their support.

I also thank faculty and staffs of AGSRD, AUGSD, SWD and PSD for their support and cooperation. I thank staffs of chemical engineering department: Mr. Babulal (Former staff), Mr. Suresh Kumar, Mr. Jangvir, Mr. Ashok, Mr. Jeevan lal and Mr. Sunder for their kind support. I thank Dr. Saswat, Dr. Arun, Dr, Sweta, Ms. Shailee, Mr. Abhishek, Ms. Ajita, Ms. Priya, Mr. Anil Kumar S, Mr. Ramakrishna, Mr. Rajesh, Mr. Anil Kumar K, Ms. Rachael, Mr. Amir Khurshid, Mr. Bharat, Ms. Soumya, Dr. Kiran, Dr. Manikandan and other research scholars, ME students of chemical engineering department, friends in civil and mechanical engineering department, BITS-Pilani, Pilani campus for their support.

I thank Dr. Anil Rai (Assistant Professor, Department of Humanities and Social Sciences) for teaching me sitar during my PhD studies. I thank Mrs. Munmun Ajaya Kumar Pani for her love and affection towards my family. I also thank Dr. Vidhi Vyas and Mrs. Manju Rani Sheoran for their kind support and affection towards my family. I also thank all my family members for giving me kind support. Finally, I thank supreme almighty for completion of this research work.

Abstract

Soft sensors are data-driven intelligent software programs which use statistical and/or artificial intelligence techniques to estimate the primary variables (product/effluent quality) from the knowledge of secondary variables (temperature, pressure, level and flow rates etc.). However, performances of conventional soft sensors degrade with time due to gradual/abrupt changes in process conditions, ambient conditions and/or feedstock quality. This leads to performance deterioration of the soft sensor which is deployed for estimation. In order to tackle the issue of time varying process conditions, the soft sensor model must have adaptation capability to maintain the prediction accuracy it is initially designed for. Prediction of primary quality variables in real time with adaptive nature for time varying process conditions is a critical task in process industries.

This work focuses on the design of adaptive soft sensors using recursive just-in-time learning (JITL) mechanism. In the JITL mechanism, at each prediction instance, a small subset of data (relevant dataset) are taken from a large database. Optimum relevant dataset size is determined by rigorous grid search method in this work. Samples of relevant dataset are chosen based on their similarity with a particular input sample (query data). Three types of similarity computation methods are investigated in this work: similarity based on Euclidean distance, similarity based on a combination of Euclidean distance & angle and similarity based on Mahalanobis distance. The relevant dataset is used to develop a local model that is used for output prediction for a particular query data. Various linear and nonlinear local models used for development of adaptive soft sensors are: locally weighted regression (LWR), multiple linear regression (MLR), partial least squares regression (PLS), support vector regression (SVR) and generalized regression neural network (GRNN). In

case of non-linear local modeling, effect of hyper parameters on model prediction accuracy and average model computation time per query sample are also investigated. The prediction accuracy is further improved by incorporating a bias update strategy in the adaptation framework. Further, after every prediction, the database of the JITL model is recursively updated. Finally, the recursive-JITL based adaptive soft sensors are also compared with only recursive adaptive model and sliding window based adaptive model.

The adaptive soft sensors are designed using three datasets from various processes in petroleum refinery, as mentioned below:

- Case study 1: Prediction of heavy naphtha initial boiling point (IBP) and end boiling point (EBP) in naphtha splitting unit.
- Case study 2: Prediction of butane concentration in the bottom stream of debutanizer column.
- Case study 3: Prediction of hydrogen sulfide (H_2S) and sulfur dioxide (SO_2) in the tail gas of sulfur recovery unit (SRU).

The designed model shows better accuracy as compared with other adaptive models reported in the literature. The important contributions of this research work are briefly mentioned below.

1. Adaptive soft sensor is developed for prediction of initial and end boiling point of heavy naphtha. As of now, for naphtha boiling point estimation, only steady state soft sensors are reported in the literature to the best of Author's knowledge.
2. Use of GRNN as a local modeling strategy in the JITL based adaptation framework is a novel contribution of this work.

3. Recursive Just-in-Time Learning algorithm is a unified approach by combining recursive and just-in-time learning frameworks based on nonlinear models (SVR and GRNN) for quality estimation.
4. In JITL approach, extensive investigation on effect of different similarity index, computation time on model prediction accuracy are compared and reported.

Keywords: *Adaptive soft sensor, just-in-time learning, locally weighted regression, multiple linear regression, partial least squares regression, support vector regression, generalized regression neural network*

Table of contents

List of Figures

List of Tables

Notations

Abbreviations

A & D	Angle and Euclidean distance
AI	Artificial Intelligence
ASTM	American Society for Testing and Materials
BC	Butane Content
CC	Correlation Coefficient
CDU	Crude Distillation Unit
D	Euclidean distance
DC	Debutanizer Column
EBP	End Boiling Point
EN ISO	International Organization for Standardization adopted by European Union
GPR	Gaussian Process Regression
GRNN	Generalized Regression Neural Network
H_2S	Hydrogen sulfide
IBP	Initial Boiling Point
ISDA	Iterative Single Data Algorithm
JITL	Just-in-Time Learning
LPG	Liquefied Petroleum Gas
LWR	Locally Weighted Regression
MAE	Mean Absolute Error
MD	Mahalanobis distance
MLR	Multiple Linear Regression
MW	Moving Window
PCA	Principal Component Analysis
PCR	Principal Component Regression
PLS	Partial Least Squares regression
R	Recursive
RDS	Relevant Dataset
RMSE	Root Mean Square Error
SMO	Sequential Minimal Optimization
SRU	Sulfur Recovery Unit
SVM	Support Vector Machines
SVR	Support Vector Regression
SW	Sliding Window
SO_2	Sulfur dioxide

Mathematical operators

b	Bias value or constant in SVR model
C	Regularization constant or penalizing factor in SVR model
d	Euclidean distance between query sample and each sample of the database
E, F	Error matrix in PLS model
$f\ (x, y)$	Joint probability density function in GRNN model

h	Bandwidth of kernel function
h_i	Activation function in GRNN model
K	Kernel function
L_ε	SVR model optimization function (ε-insensitive loss function)
Md	Mahalanobis distance between query sample and database sample
P_L	Loading matrix of input variables in PLS model
q_L	Loading vectors of output variables in PLS model
R	Correlation coefficient
R_{emp}	Empirical risk in SVR model
R_{reg}	Regression risk in SVR model
S_i	Similarity index
t	Time instance
w	Window length in sliding window
w_i	Weighting function
W	Weighting matrix
x	Input object
x_q	Input query object
y	Actual output
$\bar{y}$	Mean value of output
$\hat{y}$	Predicted output

Greek symbols

α_i^*, α_i	Lagrangian multipliers in SVR model
β	Regression coefficients of linear models
ε	Error bound or loss function parameter in SVR model
λ	Forgetting factor in recursive approach
λ	Weighting parameter in JITL approach
σ	Spread parameter in GRNN model
ϕ	Relevant dataset
$\varphi\,(x)$	Nonlinear function in SVR model
ξ_i^*, ξ	Positive slack variables in SVR model

Chapter - 1

Introduction

1.1 Motivation

A system is defined as a confined state of any chemical processes that can be described in terms of variables, which can be divided into primary and/or secondary variables. Secondary variables e.g. temperature, pressure, level, flow-rate etc. can be directly and easily measured with physical sensors or instruments. Conversely, primary variable e.g., product quality cannot be measured or difficult to measure in real time using external sensing devices.

Accurate real time information of primary variable is still an intricate step in many process industries. However, this is essential for achieving maximum productivity with enhanced safety. Quality of products (or effluent) is often determined by infrequent manual (offline laboratory) analysis. Though in some cases hardware sensors are available for continuous monitoring (e.g. gas chromatograph for continuous online measurement of composition), these instruments suffer from regular maintenance issues (mechanical failure, drift etc.,) and significant time delay.

The practical difficulties associated with delay in measurements and unreliable real time information of output variables due to drifts, fouling or accidental damage of process analyzers often causes deviation of output from its desired value. These aforementioned reasons make it difficult to achieve real time product quality control and often results in formation of off-grade products (Bhartiya & Whiteley, 2001). On the other hand, improved digitization of process industries has assisted in collection and storage of process variable information in the form of plant historical databases. These past historical plant data can be

utilized in developing soft or virtual sensors (Facco et al. 2009; Gonzaga et al. 2009) which in turn will ensure improved product quality.

1.2 Basics of soft sensors

The term soft sensor is a combination of the words ‘software’ and 'sensor'. Soft sensor is alternatively termed as inferential sensor or virtual sensor and is the most promising alternative to hardware sensor. Usually, the difficult-to-measure primary variable is a function of many easy-to-measure secondary variables. Therefore, it is highly possible to estimate the primary variable from the information of secondary variables, by developing a relationship between them (Kadlec et al. 2011; Jiang et al. 2020).

Soft sensors develop a relationship between input and output by making predictive process model for real-time estimation of primary variable with the help of statistical or artificial intelligence (AI) techniques through the information retrieved from real process (collected from industry) secondary (as input) and primary (as output) variables, which significantly reduces the measurement delay, installation and maintenance costs.

There are two different approaches for designing soft sensors: phenomenological/model based and data-driven approaches. The model-driven or phenomenological technique or physical modeling based on first principle models are often not possible for highly complicated or poorly understood processes. On the other hand, the advancement in data storage facilities has led to numerous process variables being monitored and stored in the plant database. The vast industrial data can be retrieved back and can be used for better monitoring, optimization and control of chemical processes. The data-driven based models provide the viable option for quality monitoring by detection and analysis of vast amounts of operating data. Soft sensor design for estimation of key product properties will be a

significant step forward in effective implementation of Industry 4.0 concept in process industries.

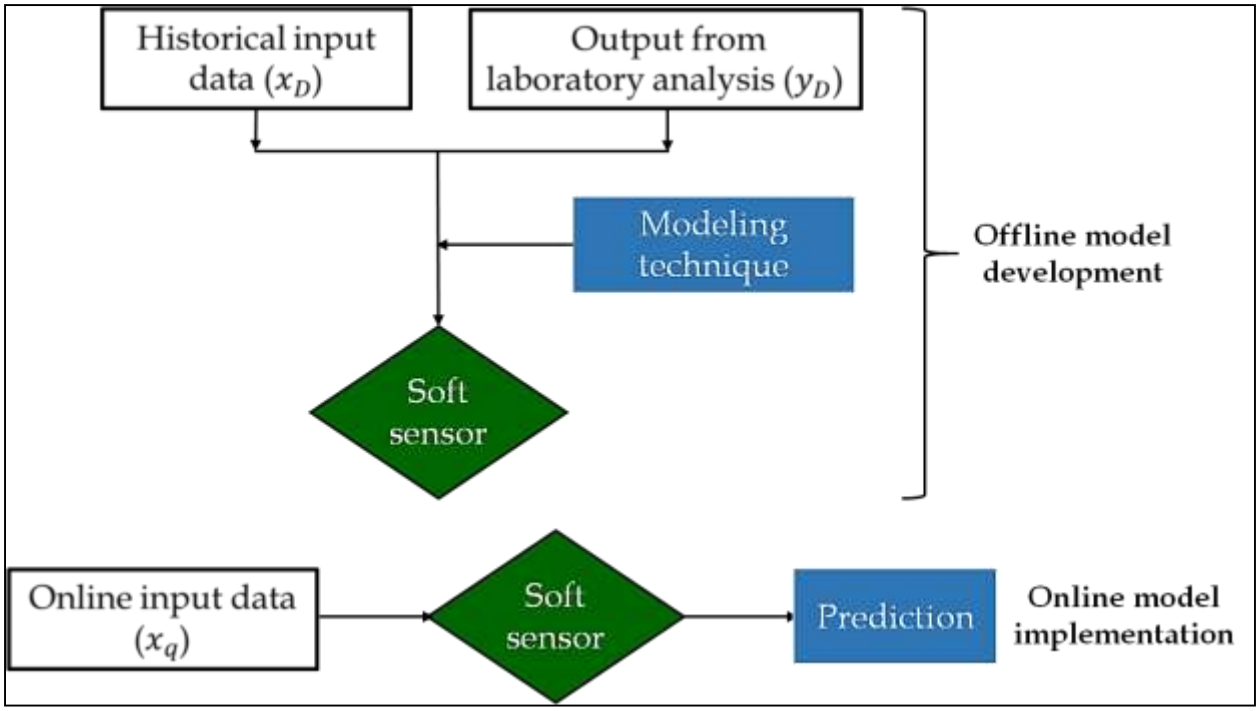


Figure 1.1: Conventional soft sensors

General scheme for development of soft sensor includes different steps such as data acquisition and identification, pre-processing, model selection, training, maintenance before deploying soft sensor into real-time estimation (Kadlec et al. 2009; Kadlec et al. 2011; Jiang et al. 2020). After acquisition of data, data preprocessing may be required prior to model development because industrial data may have the undesired characteristics such as missing values, collinearity, outliers, drifting, sampling rates and delays (Kadlec et al. 2009; Pani & Mohanta, 2011). The final model is developed based on either statistical or artificial intelligence (AI) techniques. The statistical methods may be simple linear regression, multiple linear regressions, principal component analysis, partial least squares (Abdi, 2003), kernel principal component analysis and Gaussian process regression etc. and AI techniques

include artificial neural network, neuro-fuzzy systems and support vector machines (Kadlec et al. 2011, Pani & Mohanta, 2011). Steps for conventional soft sensor development and online implementation is schematically explained in Figure 1.1.

In the past two decades, soft sensors have been reported for various process industries such as cement (Pani & Mohanta 2016), refinery (Shokri et al. 2015; Singh et al. 2019; Morey et al. 2019; Jalanko et al. 2021), polymer (Gonzaga et al. 2009; Kaneko & Funatsu, 2011a), metals (Radhakrishnan & Mohamed, 2000; Mitra & Ghivari, 2006; Kano & Nakagawa 2008; Inapakurthi et al. 2020), fermentation (Liu et al. 2012; Seo et al. 2021), rubber (Godoy et al. 2011) and pharmaceutics (Bosca & Fissore, 2011). Implementation of soft sensors in real life applications have also been reported in processes such as Bayer's process (Cregan et al. 2017), copper floatation regrind circuit (Napier & Aldrich, 2017), shaft furnace roasting system in mineral processing (Wu & Chai, 2010) and energy efficiency in buildings (Mattera et al. 2018).

Table 1.1: Survey of soft sensor applications in Japanese chemical & petroleum refining industries (Kano & Ogawa, 2010)

Process	Methods							
	Phys.	MRA	PLS	O.L.	ANN	JIT	Grey	Total
Distillation	20	256	41	6	0	5	3	331
Reaction	5	32	43	0	0	5	1	86
Polymerization	0	4	8	0	3	0	5	20
Others	0	1	1	0	0	0	0	2
Total	25	293	93	6	3	10	9	439

Phys. – Physical model; MRA – Multiple Regression Analysis; PLS – Partial Least Squares Regression; O.L. – Other Linear regression; ANN – Artificial Neural Networks; JIT – Just-in-Time model; Grey – Grey box model or hybrid model between physical model and statistical model

Table 1.1 presents a survey of soft sensor applications in Japanese chemical & petroleum refining industries (Kano & Ogawa, 2010). It can be observed from this survey that MRA and PLS models have found more number of applications than other modeling techniques. Use of non-linear techniques such as neural networks are very few in soft sensor applications. This contrast reveals the gap between theoretical and practical usage of soft sensors based on neural networks in process industries.

1.3 Limitations of conventional soft sensors

During the last decade of twentieth century and the initial years of twenty first century, most of the research works were based on design of first generation soft sensors. These are steady state soft sensors, which accept instrument values and predict the output at the same instant. They are applicable only for quality monitoring, when process conditions do not show any significant change.

Table 1.2: Problems associated with soft sensor applications (Kano & Ogawa, 2010)

Accuracy deterioration due to changes in process characteristics	29%
Burden (time/cost) of data acquisition	22%
Burden of modeling itself	14%
Burden of data preprocessing	7%
Inadequate accuracy since installation	7%
Inadequate accuracy due to change in operating conditions	7%
Difficulty in evaluating reliability	7%
Unjustifiable cost performance	7%

Table 1.2 summarizes the various problems associated with soft sensor applications in Japanese chemical & petroleum refining industries (Kano & Ogawa, 2010). The table clearly indicates that accuracy degradation due to changes in process characteristics and

changes in operating conditions account for more than 35% of all problems. This is because prediction performance of conventional soft sensors deteriorates under time varying process states, feed properties, environmental conditions (change of operating parameters), climatic conditions, catalyst deactivation due to aging and mechanical aging). In order to address these problems, the soft sensor after online implementation should have adaptation capability in order to address the time varying conditions.

1.4 Adaptive soft sensors and their various design approaches

The design of inferential sensors, having adaptation capability is the key to ensure good prediction accuracy for a longer period of time after online implementation. Therefore, presently, the attention is more towards the development of adaptive soft sensors.

Selected adaptive techniques reported in literature are recursive (Mu et al. 2006; Ahmed et al. 2009; Poerio & Brown, 2018), moving/sliding window (Liu et al. 2010; Ni et al. 2012; Kaneko & Funatsu, 2015; Liu et al. 2018; Kneale & Brown, 2018), just-in-time learning (Cheng & Chiu, 2004; 2005; Fujiwara et al. 2009; Liu, 2017), time difference (Kaneko & Funatsu, 2011a; 2011b; 2015; Xiong et al. 2017) and ensemble (Kaneko & Funatsu, 2016; Shao & Tian 2017; Kanno & Kaneko, 2020; Shi & Xiong, 2020) methods. Figure 1.2 shows the pictorial representation of online adaptation mechanism by the soft sensor.

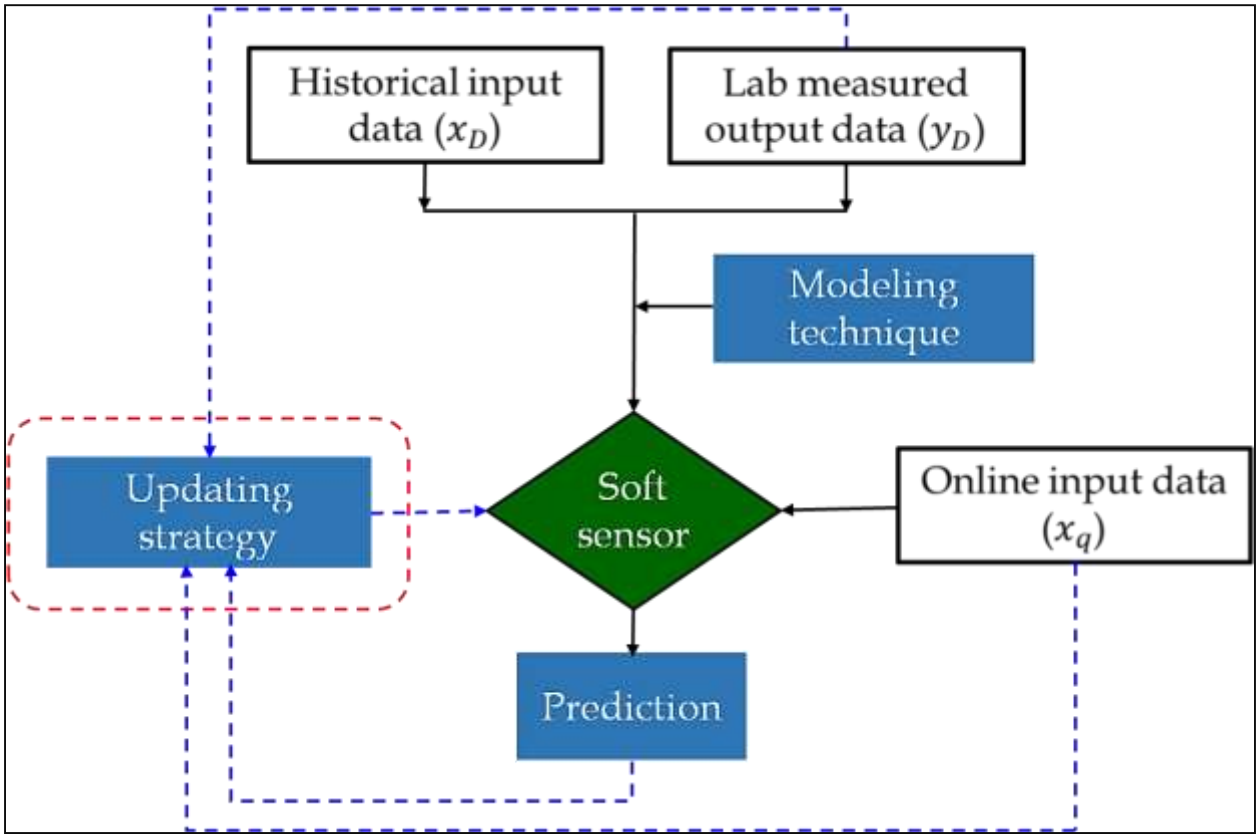


Figure 1.2: Soft sensor with adaptation mechanism (Chen et al. 2014)

1.5 Motivation for the present work and Research objectives

The prime focus of the proposed work is to design soft sensors with adaptation capability using data-driven techniques. During the design of adaptive soft sensors, it is planned to investigate the different aspects of adaptation mechanism and different modeling strategies as mentioned in Section 1.4.

Research objectives

- To design adaptive soft sensors for industrial processes using various adaptive frameworks.
- To investigate the effect of various hyper-parameters on predictive performance of developed adaptive schemes.

- To statistically evaluate the predictive performance of the designed soft sensors by various error metrics and residual analysis.

1.6 Organization of thesis

In this work, extensive investigation on the JITL based adaptation technique is performed followed by comparison of the JITL based soft sensor with two other commonly used adaptive soft sensors i.e. recursive and sliding window based soft sensors. Detailed literature review on use of these adaptive techniques for quality monitoring in various industrial processes along with its scope is presented in Chapter 2.

The different adaptation techniques also require a certain local modeling strategy (global models). These global models may be linear or non-linear. The different linear local modeling strategies explored in this work include multiple linear regression, locally weighted regression and partial least square regression. The non-linear modeling strategies investigated are support vector regression and generalized regression neural network. In Chapter 3, the underlying theoretical details of the aforementioned local models and the three adaptive approaches investigated for soft sensor development are presented.

Various adaptation techniques presented in Chapter 3 are applied for quality monitoring in three types of processes used in petroleum refining industry. The first case study involves estimation of initial and end boiling point of heavy naphtha in naphtha splitter unit. The second case study is to estimate the butane concentration in the bottom product coming from the debutanizer column used in the LPG recovery unit. The third case study involves effluent monitoring instead of product quality monitoring. This application involves estimation of H_2S and SO_2 concentrations in the tail gas which is emitted to the atmosphere from a sulphur recovery unit of petroleum refinery. Process description of all these three

industrial processes (naphtha splitter unit, debutanizer column and sulfur recovery unit) along with details of the process variables are presented in Chapter 4.

In Chapter 5, three types of statistical performance indices which help to corroborate the performance of all developed models, significance of model computation time and four plot analysis are briefly discussed. Then, the results of all the developed models using adaptive approaches for the respective industrial processes are tabulated and critically analyzed using prediction performance plots (Actual and Predicted). Finally, the model with best prediction performance is further validated using 4-plot analysis.

Concluding remarks and future scope are presented in Chapter 6.

Chapter - 2

Literature Review

From 2000-2022, there are extensive applications of soft sensors for quality monitoring during steady state operation. The significant list of literature based on 1st generation soft sensors found in applications such as splitter composition in crude distillation unit/ debutanizer (Park & Han, 2000; Dam & Saraf, 2006; Ujevic et al. 2011; Rogina et al. 2011, Napoli & Xibilia, 2011, Pani et al. 2016, Bidar et al. 2017), desulphurization unit (Fortuna et al. 2003; 2007; Shokri et al. 2016), clinker quality in cement (Pani et al. 2013, Pani & Mohanta, 2016), pharmaceuticals (Bosca & Fissore, 2011), chemicals in water system (Bowden et al. 2006; Heddam et al. 2011; Zhu et al. 2017), bioreactors (Kulkarni et al. 2004; Desai et al. 2006), concentration and purity in industrial sugar crystallization process (Damour et al. 2010), product quality in polymerization (Gonzaga et al. 2009; Nogueira et al. 2017), rubber quality (Godoy et al. 2011), metallurgical industry (Jia et al. 2011; Li et al. 2013) and mineral processing industry (Mitra & Ghivari, 2006; Ko et al. 2011; Napier & Aldrich, 2017; Inapakurthi et al. 2020).

However, when there is a deviation in steady state operation, these kinds of soft sensors fail to give accurate predictions (Kadlec et al. 2011). Changes in process conditions in an industry may arise due to different drifts, start-up/shut-down activities, grade changeover, change in input efficiency because of fouling, scaling on the walls of the pipelines (Kadlec & Gabrys 2011; Zhou et al. 2012). Therefore, there is a need for development of soft sensors which adapt to the changes in process conditions. Adaptive soft sensors take care of all process deviations by taking recent data points with exclusion of old ones. Moreover, it

helps to update the model parameters periodically according to the current process state by instance selection (moving window), instance weighting in the form of covariance matrix (recursive), locally weighted (just-in-time learning), time difference and ensemble methods.

A brief review of adaptive soft sensors reported in the literature in the past decade is presented here.

2.1 Just-in-Time learning based approach

Different types of local modeling approaches in the JITL framework are reported in the literature for soft sensor development.

Cheng & Chiu (2004) proposed an autoregressive exogenous model in just-in-time learning approach to predict the output in van de Vusse reactor and non-isothermal reactor with first order reaction. They incorporated combined angle and distance similarity measure in identifying the relevant dataset used for model development. Further, in the extended work reported by Cheng & Chiu (2005), the finite impulse response local model is integrated with principal component analysis for nonlinear static / dynamic systems.

Fujiwara et al. (2009) developed correlation based just-in-time learning model for investigation of reactant concentration (catalyst deactivation) in continuous stirred tank reactor system and estimating aroma concentration in cracked gasoline fractionator of the ethylene production process. They proposed correlation based similarity measure using Q and T^2 statistics. They found that the performance of correlation based just-in-time learning model is better than recursive partial least squares model.

An adaptive soft sensor based on three combinations of just-in-time learning approach along with partial least squares, support vector regression and least squares support vector regression was presented by Ge and Song (2010), to address the issues like change of

process conditions and nonlinearity. They found that the prediction performance of the least square support vector regression based adaptive model is efficient over other methods in the debutanizer column.

Chen et al. (2011) developed adaptive local kernel-based learning scheme to predict the end point of crude oil in fluidized catalytic cracking unit. Adaptive weighted least square support vector regression is employed as a local model. In their work, new similarity measure based on supervised locality preserving projection technique to utilize both input and output information to select the relevant samples. Also, iterative and recursive selection of trade-off parameters for the local model was incorporated.

Kadlec & Gabrys (2011) developed a two level adaptive local-learning based nonlinear soft sensor model to predict the activity of catalysts in a polymerization reactor. The drawbacks such as lack of adaptive nature, forgetting factors for parameter selection, length of adaptation window, storage and access to past data set were addressed by model adaptation at two levels by splitting global data into local partitions by k-means algorithm at first level and adapts performance mapping to update combination weights by sample-by-sample, to avoid storing of historical data at second level.

Liu et al. (2012) proposed a just-in-time learning soft sensor based on least square support vector regression model to estimate the concentration of active biomass and recombinant protein in streptokinase fed-batch fermentation process. They incorporated cumulative similarity factor criterion to determine the relevant dataset reasonably. Fast leave one out cross validation method is deployed to online optimization of parameters for least square support vector regression model.

To predict the melt index in multi-grade polyethylene process, Liu & Chen (2013) developed a just-in-time learning approach based on least square support vector regression. Probabilistic analysis is used to identify the steady state grade in which the mode/state of query belongs to the polymer process. After identifying the particular cluster, the models were built and estimate the melt index. For transitional modes, the proposed adaptive approach was used to predict the melt index.

Xie et al. (2013) proposed just-in-time learning approach based on non-Gaussian regression model to estimate the tail gas composition of SRU. Support vector data description is used to extract the independent components to calculate the D statistic. A new similarity measure was proposed by merging D statistic with T^2 statistics. This model is suitable for non-Gaussian processes.

Fan et al. (2014) designed partial least square just-in-time learning soft sensor to predict the butane concentration in debutanizer column. They incorporated gaussian mixture model based similarity measure for extraction of relevant data features to characterize the time varying and non-gaussian behavior of industrial processes.

Adaptive soft sensor based on kernel partial least square regression using multiphase just-in-time learning approach is developed by Jin et al. (2014) to predict the substrate concentration in industrial fed-batch chlortetracycline fermentation process. Bayesian inference strategy is used to select the phase samples for the particular query with maximum probability. In addition to that, hybrid similarity measure and database updating scheme also proposed in their work to enhance the predictive performance of the model.

Yuan et al. (2014) proposed an adaptive soft sensor just-in-time learning based locally weighted kernel principal component regression to define the high product quality, which is

validated using the debutanizer column process. They were trying to provide solutions to both nonlinear and time-variant problems, in which the root mean square errors for locally weighted kernel principal component regression is less compared to other combinations of models.

Chen et al. (2014) developed a recursive locally weighted partial least square regression approach to predict the final boiling point of diesel using Near infrared spectroscopy dataset collected from refinery process. They incorporated both space and time weight component in prioritizing samples for the incoming query sample. Information present in the database can be updated by moving window fashion. Nonlinearity and time varying issues are addressed in their work by merging recursive algorithm with JITL frameworks.

In order to tackle the regression problem, a supervised extraction method called supervised local and non-local structure preserving projections incorporated in just-in-time learning method was introduced by Shao et al. (2015) by considering both input and output data in sample dataset. This model was introduced in sulphur recovery unit to understand the discriminating and database monitoring abilities over local and non-local structure preserving projections.

Min & Luo (2016) designed a soft sensor by combining just-in-time learning and AdaBoost regression learner for calibration of output in pH neutralization process. Here, moving window learner is required to run the calibration method. This approach works well, when the feedback output values are unavailable, while the developed model is calibrated using re-estimated output values.

Hybrid just-in-time learning soft sensor based on back propagation neural network was proposed in Chen et al. (2017) to estimate the carbon efficiency in iron ore sintering process.

In their work, comprehensive carbon residue is taken as the measure of carbon efficiency. Initially, genetic algorithm-fuzzy C means clustering algorithm is used to extract the current feature parameters from the key frames obtained through discharge end of complex environment. Multi-task learning- Back propagation neural network is the offline learning sub-model which initially optimizes the initial weights and biases required for the online just-in-time learning back propagation neural network sub-model to estimate the output.

Yuan et al. (2017) proposed weighted gaussian regression model, in which the locally weighted gaussian model is built using the relevant samples for the particular query sample collected based on the joint density of inputs and outputs. The outputs were estimated by a probabilistic approach taking conditional distribution to reduce the prediction uncertainty.

Yeo et al. (2019) developed ensemble locally weighted independent component kernel partial least square regression to deal with nonlinear and non-Gaussian processes. The developed model is used to predict the concentration of total cyclin in eukaryotic cell cycle regulation and product concentration in continuous stirred tank reactor.

Kanno & Kaneko (2020) proposed an adaptive soft sensor based on ensemble just-in-time learning approach using a gaussian process dynamic model to predict the butane content in the debutanizer column and tail gas composition in the sulfur recovery unit. They found that the proposed approach provides better prediction than dynamic just-in-time learning gaussian process regression, dynamic locally weighted partial least squares regression and dynamic locally weighted principal component regression. For every query sample, factors such as process dynamics, autocorrelation, non-linearity, noise and changes in hyper parameters are taken into consideration for prediction of desired output by this approach.

Zhang et al. (2020) proposed a double-level locally weighted extreme learning machine model with two similarity indexes (mutual information at the first level and Euclidean distance at the second level) to predict the butane content in the debutanizer column. They found that the performance of the proposed model provides better performance than locally weighted, just-in-time learning and conventional extreme learning machine models.

Yamada & Kaneko (2021) proposed an adaptive soft sensor based on genetic algorithm based process variable and dynamics selection integrated with locally weighted partial least squares to predict the butane content in the debutanizer column and tail gas composition in the sulfur recovery unit. This work helps to improve the predictive capability of adaptive soft sensors in the presence of multiple process states. They incorporated selection of optimal process variables and time delay of output variables to construct a dynamic soft sensor.

Combination of moving window and just-in-time learning in transfer learning frame is developed by Alakent (2021a) to estimate the butane concentration in debutanizer column and tail gas composition in SRU. In their work, kernel ridge regression is used as predictive model for development of task transferred just-in-time learning approach. This model helps to combat heterogeneous concept drift phenomena occurring in process industries.

Alakent (2021b) proposed an adaptive soft sensor based on Lasso estimator by combining task transferred just-in-time learning with a moving window learner in a transductive learning setting to predict the butane concentration in debutanizer column, tail gas composition in SRU, fluoride concentration in waste water treatment plant, melt index in sequential reactor multi-grade polyethylene production process and concentration of lighter end products from fluidized catalytic cracking unit.

Abdolkarimi et al. (2022) proposed a double level similarity criterion and support vector machine and co-evolutionary particle swarm optimization method to predict the research octane number, benzene volume percentage and reid vapour pressure for indicating gasoline quality in isomerization reaction. Moreover, they compared the performance of proposed model with double locally weighted extreme learning machine, support vector regression and response surface methodology.

2.2 Recursive based approach

Mu et al. (2006) proposed recursive partial least squares soft sensor with dual updating strategy for prediction of average crystal size of a purified terephthalic acid purification process. They update both process model and model output offset simultaneously for each sample, thereby the performance of recursive partial least squares with dual updating strategy is found to be better than dynamic partial least squares model.

Ahmed et al. (2009) developed recursive partial least squares soft sensor for prediction of melt index during grade change operations in high density polyethylene plant. In their work, they proposed two schemes 1 and 2. Scheme 1 approach is analogous to the work proposed by Mu et al. (2006), with dual updating strategy to minimize the prediction error. To maximize prediction power and minimize time taken by the model, scheme 2 was developed to prevent the predictive model from unnecessary update by incorporating threshold criterion.

Xianghua et al. (2009) proposed a recursive partial least squares soft sensor for online estimation of para-xylene concentration at the reactor outlet in an industrial isomerization unit. Transformation of ortho-xylene, meta-xylene and ethylbenzene to para-xylene is the main process occurring in the refinery. The performance of recursive partial least squares

and regular partial least squares were compared and found that the recursive partial least squares possess better estimation and tracking capability.

Jia et al. (2011) designed a hybrid soft sensor by combining simplified first principle model with recursive partial least square regression model to predict the raffinate concentrations of copper extraction process in cobalt hydrometallurgy. In their work, they address the problem of time variant feature of copper extraction process. They employed model rectification strategy using bias update procedure in adaptive soft sensor to correct the final output.

Ni et al. (2011) proposed recursive gaussian process regression along with autoregressive with exogenous inputs model for prediction of melt flow rate in polypropylene polymerization process. They incorporated dynamics in the model and the bias for dynamic model can be updated by bias updating scheme to improve the prediction accuracy. They found that the predictive performance of this approach better than the recursive gaussian process regression (static) and gaussian process regression models.

To address the issues of forgetting factor and sensitivity of variable scaling in recursive partial least squares algorithm, Ni et al. (2012b) proposed localized and adaptive recursive partial least squares model. In their work, they incorporated two levels of local adaptation (local model and local time regions) and three levels of adaptive strategies (mean and variance, forgetting factor and time regions) in the structure of recursive partial least squares algorithm to predict the catalyst activity in polymerization process, reactor temperature in gas phase polyethylene process and melt flow rate in polypropylene polymerization process.

Galcia et al. (2012) proposed a recursive reduced order dynamic partial least square regression model to predict the Kappa number in industrial Kamyr pulp digester. This model

helps to address the frequent process changes in pulp digester. Online data scaling procedure was incorporated in their work and finally the effectiveness of this soft sensor was demonstrated in closed loop study.

For effective tracking of process dynamics, Ni et al. (2014) proposed new localized adaptive soft sensor, the extension of the previous model localized and adaptive recursive partial least square regression to predict the melt flow rate in polypropylene polymerization process, catalyst activity in polymerization process and reactant concentration in continuous stirred tank reactor. For online prediction, moving window fashion is introduced in local learning framework instead of forgetting factor. In addition to that, the need for choosing the weighting factor in advance manner is avoided by introducing the averaged bias updating strategy.

A soft sensor model by combining recursive fixed-memory principal component analysis and least squares support vector machines was reported in Qiao & Chai (2012) to guide the calciner temperature setting which relates the final quality (percentage of calcium oxide, ferric oxide and particle size) of products by decomposition rate in raw meal calcination process.

Matias et al. (2015) proposed an online sequential extreme learning machine based on a recursive partial least squares soft sensor to predict the butane content of the debutanizer column. In their work, they improved the existing model online sequential extreme learning machine based on recursive least squares to recursive partial least squares to mitigate the ill-conditioned problem associated with the estimation of output weights in hidden layer neurons due to the presence of redundant input variables.

Shao & Tian (2015) proposed an adaptive soft sensor based on selective ensemble of local partial least squares model by combining local learning and selective ensemble learning (through bayesian inference) strategy for prediction of residual acid gas (H_2S and

SO_2) concentration from tail gas in industrial sulfur recovery unit. Finally, they compared root mean square errors and mean absolute deviation for a selective ensemble of local partial least squares models with other models such as recursive PLS, locally weighted PLS, moving window PLS, localized adaptive soft sensor and found that proposed model provides significant results over other available models.

Shao & Tian (2017) used new distance to model criterion in semi-supervised selective ensemble learning strategy as adaptive soft sensing technique to solve the constraints for processing number of labeled samples, non-linearity and limitation of k-nearest neighbor method in case studies such as debutanizer column and sulfur recovery unit with high performance of prediction.

2.3 Moving/Sliding window based approach

Ni et al. (2012a) proposed a moving window gaussian process regression soft sensor incorporated with bias update and dual preprocessing step (for noise) using Savitzky-Golay filter for prediction of catalyst activity in multi-tube polymerization reactor and melt flow rate in polypropylene polymerization process.

Abusnina & Kudenko (2013) proposed an adaptive soft sensor based on moving window gaussian process regression to predict the catalyst activity in multi-tube polymerization reactor. Moreover, the proposed approach provides better performance than moving window partial least squares regression. Reduced training subset were retrieved based on maximal information coefficient criterion from the pool of input variables. They used four different validation measures to interpret the prediction accuracy of developed models: mean square error, mean relative standard error, relative variance tracking precision and Theil's inequality coefficient.

Kaneko & Funatsu (2013) discussed characteristics of adaptive soft sensors (moving window, time difference and just-in-time learning approaches based on partial least squares model) on the basis of the degradation phenomena due to state shifting in industrial distillation unit. Adaptive models were subjected to predict the concentration of bottom product having lower boiling point in distillation unit.

A non-linear adaptive principal component analysis was developed by Salah et al. (2015) to detect the breakout phenomenon in continuous casting process. The adaptive version (sliding window and recursive technique) of principal component analysis deals with evaluating and updating the model parameters for improving the prediction accuracy. This work claimed that adaptive principal component analysis has better prediction capability as compared with traditional principal component analysis, neural network and support vector regression models.

Yuan et al. (2016) proposed probabilistic just-in-time learning soft sensor by adopting variational Bayesian principal component analysis with symmetric Kullback-Leibler divergence as similarity measure to predict the butane concentration in debutanizer column. However, they tested the prediction performance of the developed model using different missing ratio of output values and finally compared with deterministic just-in-time learning model.

Yao & Ge (2017a) designed selectively integrated moving window weighted supervised latent factor analysis method to estimate the oxygen content in the furnace of primary reformer (hydrogen manufacturing unit in Ammonia synthesis process). The proposed model helps to address the degradation of soft sensor due to process nonlinearity and state shifting issues.

In order to encounter process characteristics drifts such as state shifting and non-linearity, Yao and Ge (2017b) proposed a high performance adaptive soft sensor with moving window based weighted supervised latent factor analysis. They took samples from the primary reformer unit in the hydrogen manufacturing unit (from raw methane) of the urea synthesis process and considered the similarity between the training and query data samples for improving model adaptation capability, which further decreases the computation time for retrieving samples from the database.

Xiong et al. (2017) developed combined moving window and time difference approach based on gaussian process regression model to handle the time delay and drifts in debutanizer column. They incorporated fuzzy curve analysis to solve the time delay mismatch in the output prediction, thereby retrieving the reliable estimation from the soft sensor.

Kneale & Brown (2018) proposed an adaptive soft sensor based on recursive partial least squares, moving window partial least squares, moving window random forest regression, mean moving window and random forest partial least squares regression ensemble to predict the butane content in debutanizer column and tail gas composition in sulfur recovery unit. They found that the random forest partial least square regression ensemble is more robust against other models with lowest prediction errors. This can find application in chemical processes which lack historical databases or frequently changing process states.

Liu et al. (2018) proposed a soft sensor based on an adaptive framework such as moving window, time difference and locally weighted regression using Bayesian network to predict

the butane content of the debutanizer column. The proposed model has the capability to handle model degradation due to drifts or state shifting and missing values.

Strategy for selective use of adaptive models was designed by Yuge et al. (2018) for estimation of tail gas composition in SRU. In their work, moving window and just-in-time learning frameworks based on partial least square regression and random forest model was taken in to consideration. Initially, process state indexes (mean distance, mean square error, T^2 and Q statistics) were calibrated and then prediction efficiencies (mean absolute error, root mean square error) for each model were calculated. Finally, the soft sensor with best predictive performance is selected based on the prediction efficiency and deployed for prediction of output.

Chen et al. (2019) proposed a dynamic adaptive soft sensor based on ensemble regularized local finite impulse response model, in which multiple local domains was created to ensure that the prediction depends upon the optimal window belonging to the same process states. The coefficients for the proposed model were determined using stable kernel based regularized least squares. This work helps to predict the tail gas composition of the sulfur recovery unit and measurement of arc current in the ladle furnace transformer during the smelting process.

Urhan & Alakent (2020) proposed a soft sensor based on integrative moving window and just-in-time learning adaptive approach using relevance vector machine model to predict the butane content in the debutanizer column.

Alakent (2020a) proposed an adaptive soft sensor based on Lasso (least absolute shrinkage and selection operator) estimator combining just-in-time learning and moving window using a transductive inference to predict the butane content in debutanizer column,

tail gas composition in sulfur recovery unit and melt index in sequential reactor multi grade polyethylene production process.

Alakent (2020b) proposed an adaptive soft sensor based on Lasso (least absolute shrinkage and selection operator) estimator combining task transferred just-in-time learning with a transductive moving window learner to predict the butane content in debutanizer column, tail gas composition in sulfur recovery unit, fluoride concentration in wastewater treatment plant, melt index in sequential reactor multi grade polyethylene production process and concentration of gasoline, light diesel oil and liquefied petroleum gas in fluidized catalytic cracking unit.

Li et al. (2021) developed variable exponentially weighted moving window partial least square regression algorithm to estimate the emission of NO_x concentrations in flue gas of coal fired power plants. This model helps to overcome the difficulties associated with moving window approach such as selection of window size and forgetting factor without prior process knowledge.

Yamakage & Kaneko (2022) proposed an adaptive soft sensor based on Bayesian optimization approach to predict the butane content in the debutanizer column and tail gas composition in the sulfur recovery unit. The adaptive mechanisms deployed in their work were moving window, just-in-time learning and time difference based on partial least squares, locally weighted partial least squares, elastic net, least absolute shrinkage and selection operator, support vector regression and gaussian process regression. The proposed approach automatically optimizes the types of adaptive mechanism (along with hyper parameters) and its regression methods (along with hyper parameters), which provides the high predictive capability of adaptive soft sensor combination for the given process.

2.4 Gaps in existing research

Literature survey of adaptive soft sensors proposed so far reveals few limitations exist in the available research work. The aim and scope is identified based on the available gaps and they are examined in this research work.

Gaps in identified process

Petroleum Refinery involves highly complex multicomponent distillation process, where petroleum is fractionated into several lighter and heavier ends. Naphtha is the lighter end liquid distillate, which is the main feedstock for various petrochemical products such as ethylene and propylene, hydrogen production etc. It acts as a solvent for elastomer, diluent in paints and varnishes. In literature, very few predictive model is available to estimate the initial and end boiling point of naphtha. The available predictive models are mainly constructed based on conventional methods. Adaptive model for estimation of quality variables is rarely reported in literature.

Gaps in methodology

There have been several local modeling strategies in adaptive soft sensor development were proposed so far. Neural networks are rarely used as a local modeling strategy. Nevertheless, generalized regression neural networks is still yet to be explored in literature. Also, linear (PLS) were explored in all adaptive frameworks. Very few nonlinear models (GPR) were explored in one or more adaptive frameworks. Moreover, availability of unified approach (two or more adaptive frameworks) is available for only linear (PLS) model. For nonlinear models, unified approach is still not explored.

2.5 Scope of the present work

The focus of this research work is development of adaptive soft sensor for quality estimation in refinery applications. Three processes are identified where there is a requirement for online monitoring of product quality.

First, naphtha splitter unit in crude distillation unit, where end boiling point of splitter bottom fraction should not exceed 204^0C because the rate of deactivation of platinum catalyst increases while processing through catalytic reforming units. Also, the initial boiling point of splitter boiling point of splitter bottom fraction should maintain between 75^0-100^0C, such that it prevents the formation of precursors for undesirable benzene above this range in catalytic reforming units (Ujević et al. 2011). Therefore, real time estimation of these parameters will be of great assistance in maintenance of naphtha quality.

In debutanizer column, to achieve better performance of the process, the main objective is to minimize the butane content (C_4) in the column bottom stream. Butane concentration is indirectly measured by online gas chromatograph, which is located at the top of the de-isopentanizer column to quantify the butane content in the bottom output flow of the debutanizer column. Further, there is a significant time delay in assessing the butane content of the process stream through gas chromatograph. Measuring cycle of gas chromatograph is 15 minutes. The location of analyzer is far away such that the time delay of the range 30-75 minutes is needed to obtain the real time concentration values of butane. Real time estimation of butane concentration in C_5 is required for better control of debutanizer column.

Acid gases are the main causes for the formation of acid rain. The presence of H_2S in air prevents the cells of human body from breathing phenomena. Acid gas stream is major source of sulphur composed of H_2S and SO_2, from gas sweetening and sour water stripping

process. Elemental sulphur which is highly valuable in the commercial market, is produced as by-product by oxidation of H_2S in desulphurization or gas sweetening unit, which is known as Claus process. In this process, multi-stage thermal step followed by catalytic step to increase the overall sulphur yield is performed at high temperature, results in gaseous elemental sulphur. In this work, soft sensors were developed to measure the concentrations of acid gases (H_2S and SO_2) in the tail stream of sulfur recovery unit. They were designed to predict the tail gas composition at real time in SRU, using suitable set of input/output measurements of the process.

Chapter - 3

Methodology

In this chapter, the difference between global models and local models is briefly discussed. Then, adaptive frameworks that are developed in this research work, description about local models that are taken for adaptive soft sensor development and integration of local model in the adaptive frameworks are discussed.

Conventional data-driven modeling methods primarily focused on global modeling strategy (principal component regression, partial least squares regression, support vector regression, neural networks etc.). There are difficulties in specifying model structure associated with complex optimization problems for developing robust global models using large datasets. Moreover, there is a serious deterioration in the global model prediction of quality variables, when the process parameters move away from the nominal operating condition (Park & Han, 2000). On the other hand, local modeling strategy approximate the complex process system using simple models valid for a particular operating regimes. Adaptive soft sensor comprises of two parts: external adaptive framework (database technology) and internal local predictive model (Cheng & Chiu, 2004). External adaptive framework helps to identify the samples from the space of predictor variables which are highly pertinent to current process dynamics in the system. Next, the local predictive model is used to estimate the regression surface for the current query in the space. Integration of external adaptive framework with local predictive model is the core idea that comes from the combination of local modeling strategy and database technology.

3.1 Adaptation techniques used in this work

Industrial processes exhibit time varying behavior due to change in process characteristics. The process states variation exhibited by chemical processes cannot be captured by static model. To avoid those limitations, online adaptation of the soft sensors to time varying process condition is required to accurate estimation of primary quality variables.

3.1.1 Just-in-Time learning (JITL) approach

Just-in-Time learning is developed as an alternative hybrid (local and global) modeling approach which helps to adapt to the changing process environment. It is otherwise referred to as model-on-demand, instance-based learning, locally weighted model or lazy learning approach (Aha et al. 1991; Cybenko, 1996; Atkeson et al. 1997; Birattari et al. 1998; Braun et al. 2001). In the JITL approach, a certain number of input-output data values are stored in a database. Adaptive model for output is developed by using a fraction of this database which are the closest neighborhood values of the incoming query data. The neighborhood is selected based on a weighting procedure. Similarity index criterion is applied to compute the distance of the query data from each data sample present in the database. Higher weights are assigned to data points which are close to the query sample. The most similar samples chosen from the database form the relevant data set (RDS) for the particular query data and a local input-output model is developed from this RDS. This local model is used to predict the output for the query data. At the next sampling instance, a new local model is developed following this procedure.

The step by step guide for implementation of JITL technique is presented below.

1. Initially, the search for closest neighborhood points for the incoming query data is initiated in the database when the estimation of output is requested for an incoming query data measured from process stream.
2. After the search is over, a vector of closest neighborhood (for the incoming query point) samples is created by sorting each object of the database in descending order from the query data, based on the similarity index. Higher weights are assigned to data points which are closer to the query. Consequently, the farther ones acquire lower weight. Then, the weighted least squares were used to build a local function. Herewith, the effect of outlier in the incoming query data from the real plant data can be minimized by using weighting method (Park and Han, 2000).
3. In this work, we have used three different approaches for assessing similarity: Euclidean distance, Mahalanobis distance and angle combined with Euclidean distance. The method for finding the similarity index using each of these three approaches is presented below.

Euclidean distance: The Euclidean distance is computed according to Equation 3.1.

$$\boldsymbol{d\left(x_q, x_i\right) = \|x_q - x_i\|_2} \quad \textbf{(3.1)}$$

Equation 3.1 represents Euclidean norm, which is applied to compute distance between an object of the database x_i and query data x_q.

Mahalanobis distance: The expression for Mahalanobis distance is represented as:

$$\boldsymbol{Md^2 = \left(x_q - x_i\right) S^{-1} \left(x_q - x_i\right)^T} \quad \textbf{(3.2)}$$

Md^2 - Squared Mahalanobis distance; S^{-1} – Variance-Covariance matrix of training database samples; x_i – Training database samples; x_q - Query sample.

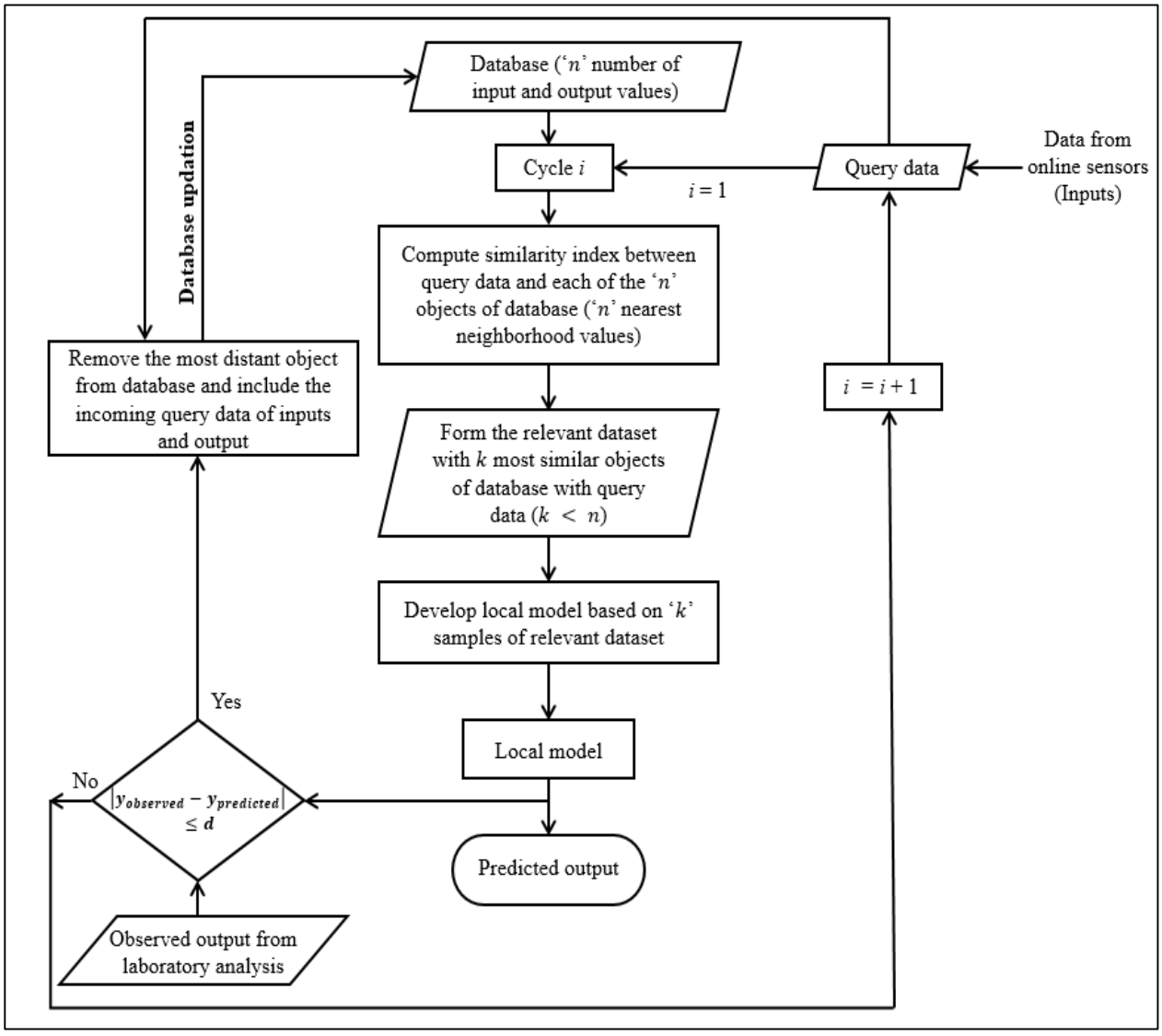


Figure 3.1: Flow chart of recursive Just-in-Time learning technique

After sorting each object of the database in descending order, a similarity index vector is created. The initial step in getting a similarity index involves computation of the weighting function corresponding to each distance according to Equation 3.3 given below.

$$\boldsymbol{w_i = \sqrt{K\,(d(x_q, x_i)/h)}} \quad \textbf{(3.3)}$$

w_i – weighting function; d or Md – distance between query sample and each sample of the database; K – Kernel function; h – Bandwidth of kernel function.

Commonly used kernel functions for weighting purposes are linear, Gaussian (or radial basis function) and polynomial. Among these, the Gaussian function maps nonlinear complex features effectively so as to minimize the residual error due to under-fitting problems by bias addition (Wang et al. 2016).

$$\boldsymbol{K\,(d) = e^{-d^2}} \quad \textbf{(3.4)}$$

Where, $d^2(x_q, x_i) = (x_q - x_i)^T\,(x_q - x_i)\;;\; i = 1, 2, \ldots n$

It may be noted that, when the distance from the query point to neighborhood point decreases, the weight function, w_i increases. Based on this, the similarity index (S_i) is computed between query data and its neighborhood in the database.

$$\boldsymbol{S_i = \sqrt{e^{-d^2\,(x_q, x_i)}}} \quad \textbf{(3.5)}$$

Angle combined with Euclidean distance: In this procedure, the above steps (Equations 3.9, 3.10 and 3.11) are repeated and subsequently, the similarity index for combined angle and distance method is computed as per Equation 3.6.

$$\boldsymbol{S_i = \lambda\sqrt{e^{-d^2\,(x_q, x_i)}} + (1 - \lambda)cos\,(\theta_i)} \quad \textbf{(3.6)}$$

Here, λ is the weight parameter constrained between 0 and 1; θ_i is the angle between Δx_q and Δx_i which are defined as: $\Delta x_q = x_q - x_{q-1}$ and $\Delta x_i = x_i - x_{i-1}$

$$\boldsymbol{\theta_i = cos^{-1}} \left[\left(\left(\Delta \boldsymbol{x_q}.\Delta \boldsymbol{x_i}\right)/\left(\left|\Delta \boldsymbol{x_q}\right|\left|\Delta \boldsymbol{x_i}\right|\right)\right)\right] \quad \textbf{(3.7)}$$

If S_i approaches 1, then the selected closest neighborhood resembles query data. The cosine function, $cos\ (\theta_i)$ should be greater than or equal to zero. Only positive cosine function is subjected to further calculation. If the cosine function is less than zero (negative), then the corresponding neighborhood in the database is discarded and cannot be used in subsequent calculations. Negative cosine function indicates that the neighborhood and query data are dissimilar. If the value of λ approaches zero, Equation 3.6 becomes distance only method. If λ = 1, then it becomes the weighting based on cosine function only. So, the value of λ should be constrained between 0 and 1 (Cheng & Chiu, 2004; 2005).

4. Similarity index computed by any of the aforementioned techniques is sorted in descending order and converted into a diagonal matrix. Then the diagonal matrix (weighting matrix, W) is multiplied with the database samples to generate training data for local model development. For generalization, the weighting matrix for all the proposed methods in this work is taken as 1.
5. The local model is then used to predict the output by simulating the local model with the query data. The local model is discarded after prediction of output for a particular query sample and for the next query sample a new local model is constructed.

In addition to method of similarity index computation and size of relevant data set, choice of local model also plays a key role in model's prediction accuracy. Figure 3.1 represents the flowchart for JITL technique.

3.1.2 Recursive approach

Recursive algorithm is developed from old model using past run dataset and includes new incoming data from the process stream (Qin, 1998). In recursive algorithm, adaptation to current process states can be done using model update. In model update strategy, the original training model can be updated, which is the classical approach. To account for change in process states using recursive algorithm, the dataset for modeling continuously updated with incoming query sample and their estimated output as shown below.

$$\boldsymbol{x_{new}} = \begin{bmatrix} \boldsymbol{x_D}^T \\ \boldsymbol{x_q}^T \end{bmatrix};\ \boldsymbol{y_{new}} = \begin{bmatrix} \boldsymbol{y_D}^T \\ \boldsymbol{\hat{y}}^T \end{bmatrix} \qquad \textbf{(3.8)}$$

Where, x_D − Training inputs; x_q − Query data; y_D − Training outputs; $\hat{y}$ − Estimated output.

Forgetting factor is usually chosen as $0 < \lambda_n \leq 1$. In this work, λ_n is fixed at 0.9. Whenever, both query and predicted output for the respective query are available, the database samples get updated until the next query and output is defined. Weight matrix (β) is used to down-weighting the oldest sample from the training database. Weight matrix holds forgetting factor (λ_n) as diagonal elements, which assigns higher weight to the current sample and lower weight to the past instant samples. Forgetting factor reflects the changing rate of regression coefficients/parameters for every instance of query sample (Ciochina et al., 2009). $\beta = diag\ (\lambda_1\ , \lambda_2\ , \ldots . \lambda_p)$ with $p = n \times m$

After simplification, it becomes

$$\boldsymbol{\beta = diag\ (1 \ldots . \lambda_p^{n-1})} \qquad \textbf{(3.9)}$$

Where, n = number of rows; m = number of columns in training database;

By multiplying with weight matrix, the computational load of output estimation reduces significantly for every incoming query sample. In this way, the model adapts with current process events and partially retains the past events of process. Addition of forgetting factor helps proposed model to adapt the process characteristics very rapidly, which is explained as:

$$\boldsymbol{x_{new}} = \begin{bmatrix} \boldsymbol{\beta x_D}^T \\ \boldsymbol{x_q}^T \end{bmatrix};\ \boldsymbol{y_{new}} = \begin{bmatrix} \boldsymbol{\beta y_D}^T \\ \boldsymbol{\hat{y}}^T \end{bmatrix} \quad \textbf{(3.10)}$$

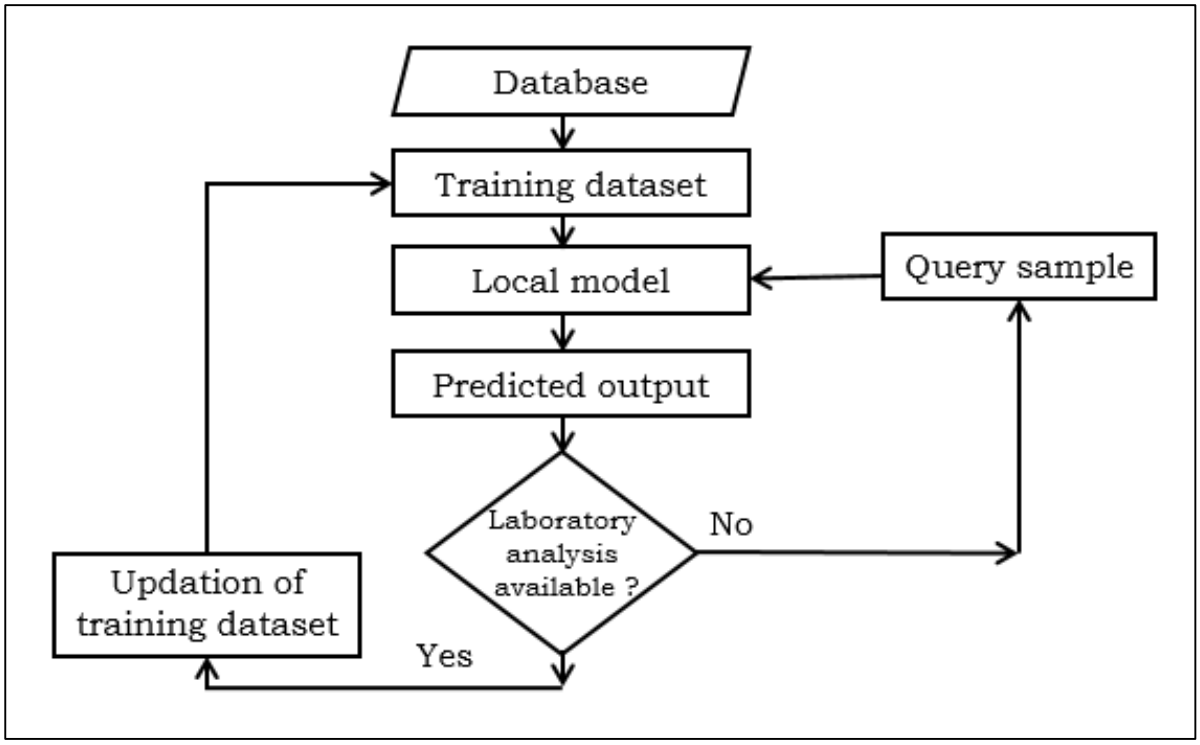


Figure 3.2: Flowchart of recursive approach

In R-GRNN algorithm, initially a generalized regression neural network model is developed from the available input-output dataset and the determination of spread parameter is discussed in Section 3.2.2.2. This model is used for predicting the output for the incoming

query sample along with simultaneous update of database using current query and its output. The step-by-step approach for design of R-GRNN is presented below followed by the algorithm proposed in literature (Mu et al. 2006; Ahmed et al. 2009).

a. Initially, the initial training (database) data are formulated.
b. Then, the GRNN architecture is trained by defining the hyper-parametric values based on the entire training data.
c. Once the model is available, prediction of output is performed for the incoming process sample (query).
d. After prediction, current query sample is included in the training database.
e. Then, GRNN is retrained with the updated training dataset (database).
f. Steps b-e are repeated for the entire incoming data sample to be predicted.

3.1.3 Moving/sliding window approach

Moving/sliding window approach is a sequence based or time-stamp based adaptive learning algorithm, where the learning of the model depends on the information contained in the set of observations stored in the particular window. Moving/sliding window approach takes either the current window or window chosen by the user, yet recursive algorithm takes the entire training dataset for further estimation. Updating of hyper-parametric values is common in both approaches.

Moving/sliding window algorithm has the flexibility to change either the window size or step size to achieve the model performance. This approach encompasses inclusion of incoming query along with its predicted output and excludes the oldest sample from the database. Moving/sliding window is used to model non-linear process systems. In the past, several linear systems (Principal component regression, Partial least squares regression) and

non-linear systems (Gaussian process regression) based on moving window approach were modeled without increasing the size of data matrix (Ni et al., 2012; Jaffel et al. 2016).

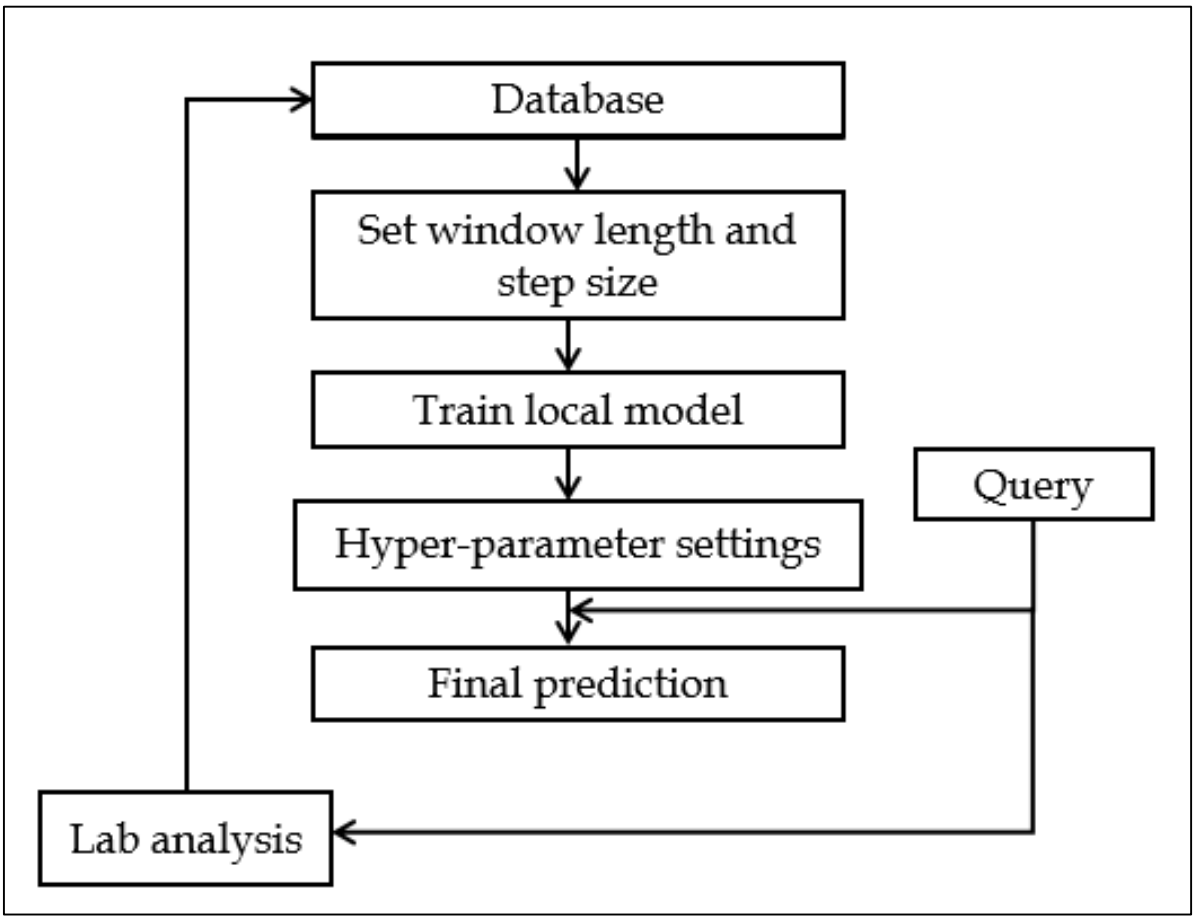


Figure 3.3: Flowchart of sliding window approach

Moving/sliding window framework helps to update the mean and variance online and keep the past sample information in a window to track the characteristics of dynamic process (Ni et al., 2012). Slightly different from the above approach, this article focuses on 'one step ahead' prediction feature of sliding window approach, which involves concurrent up-dating and down-dating of database samples (Jaffel et al., 2016).

For window length of w and at time t, the initial training data matrix $x_{D,t}$ is

$$\boldsymbol{x_{D,t}} = \begin{pmatrix} \boldsymbol{x_t} \\ \boldsymbol{x_{t+1}} \\ \vdots \\ \boldsymbol{x_{t+w-1}} \end{pmatrix} \in \mathbb{R}^{\boldsymbol{w \times m}} \quad ; \quad \boldsymbol{y_{D,t}} = \begin{pmatrix} \boldsymbol{y_t} \\ \boldsymbol{y_{t+1}} \\ \vdots \\ \boldsymbol{y_{t+w-1}} \end{pmatrix} \in \mathbb{R}^{\boldsymbol{w \times 1}} \qquad \textbf{(3.11)}$$

Down-dating strategy: This strategy helps to down-date the past information i.e., oldest sample from the first row and entire column of the training database. The resultant database excludes the past information for every successive training, thereby adapts to the current process states.

At time $t+1$, new query sample is available along with its predicted output with/without bias update. Old data sample in the time sequence is removed from the initial training data matrix, $x_{D,t}$

$$\boldsymbol{x_{D,t+1}} = \begin{pmatrix} \boldsymbol{x_{t+1}} \\ \boldsymbol{x_{t+2}} \\ \vdots \\ \boldsymbol{x_{t+w-1}} \end{pmatrix} \in \mathbb{R}^{\boldsymbol{w-1 \times m}} \; ; \; \boldsymbol{y_{D,t}} = \begin{pmatrix} \boldsymbol{y_{t+1}} \\ \boldsymbol{y_{t+2}} \\ \vdots \\ \boldsymbol{y_{t+w-1}} \end{pmatrix} \in \mathbb{R}^{\boldsymbol{w-1 \times 1}} \qquad \textbf{(3.12)}$$

Updating strategy: This strategy consists of adding new incoming query sample (x_q) to the last row and entire column of the database. Thereby, a new sample from the process stream, which carries the current process characteristics to be sequentially updated.

After new query sample, x_q or $x_{D,t+w}$ is added to the above matrix, the transformed training data matrix, $x_{D,t+1}$ becomes,

$$\boldsymbol{x_{D,t+1}} = \begin{pmatrix} \boldsymbol{x_{t+1}} \\ \boldsymbol{x_{t+2}} \\ \vdots \\ \boldsymbol{x_{t+w-1}} \\ \boldsymbol{x_{t+w}} \end{pmatrix} \in \mathbb{R}^{\boldsymbol{w \times m}} \; ; \; \boldsymbol{y_{D,t}} = \begin{pmatrix} \boldsymbol{y_{t+1}} \\ \boldsymbol{y_{t+2}} \\ \vdots \\ \boldsymbol{y_{t+w-1}} \\ \boldsymbol{y_{t+w}} \end{pmatrix} \in \mathbb{R}^{\boldsymbol{w \times 1}} \qquad \textbf{(3.13)}$$

In this way, GRNN is implemented in the SW framework for prediction of outputs. The step by step procedure for developing SW-GRNN model is presented below:

a. Initially, the historical data are used to train the neural network model based on the defined hyper-parametric values.
b. Once the sample from the process stream is measured online, the developed neural network model is subjected to estimation of final output. After the prediction, the window of training database slides down to include the newly predicted input/output sample and exclude the oldest sample.
c. Then the existing model is retrained based on the current window of the historical sample. The algorithm is repeated until the entire test sample in the dataset is predicted.

3.2 Local modeling strategy

3.2.1 Linear models

In this section, we discuss about various linear models that are used for adaptive soft sensor development.

3.2.1.1 Multiple linear regression

Dependent variable is the linear combination of all independent variables along with its unknown parameters ($\beta_0, \beta_1, \beta_2$) and linear coefficients are computed to minimize the prediction error within the training dataset. The expression for multiple linear regression with 'k' regressor variables can be described as:

$$\hat{\boldsymbol{y}} = \boldsymbol{\beta_0} + \boldsymbol{\beta_1 x_1} + \boldsymbol{\beta_2 x_2} + \cdots + \boldsymbol{\beta_k x_k} + \boldsymbol{\epsilon} \quad \textbf{(3.14)}$$

Where, $\hat{y}$ is a predicted output or dependent variable; x is an independent variables; β_j $(j = 0, 1, 2, \ldots k)$ - regression coefficients (expected change in response with respect to unit change in x_j, such that the remaining independent variables are held constant.

The described model represents the hyperplane in k-dimensional space of regressor variables $\{x_j\}$. Direct calculation of vector, β by taking inverse of x is not possible. x is not a square matrix and ill-conditioned? Linear coefficient is formulated as:

$$\boldsymbol{\beta} = (\boldsymbol{x}^{\boldsymbol{T}}\, \boldsymbol{W}\, \boldsymbol{x})^{-\mathbf{1}}\, \boldsymbol{x}^{\boldsymbol{T}}\, \boldsymbol{W}\, \boldsymbol{y} \qquad \textbf{(3.15)}$$

Where, W – Weighting Matrix (as mentioned in Section 3.1.1)

Also, in addition to first order model described in Equation 3.14, interaction term can also be incorporated to accompany combined effect of variables (interaction) in the developed model (Montgomery, 2017).

3.2.1.2 Locally weighted regression

Locally weighted regression is a memory based approach which performs regression with the nearby data using weighting concept by taking training criteria for local model or data used for modeling. It is derived from standard regression procedures with least square training criterion in which the local models are linear with the unknown parameters (Atkeson et al. 1997).

For locally weighted regression, the coefficient is calculated using:

$$\boldsymbol{\beta} = (\boldsymbol{Z}^{\boldsymbol{T}}\, \boldsymbol{Z}\,)^{-\mathbf{1}} \boldsymbol{Z}^{\boldsymbol{T}}\, \boldsymbol{v} \qquad \textbf{(3.16)}$$

Where, x and y were relevant dataset and its corresponding output data matrices, W – Weighting Matrix; β – Model parameter; $Z = W\, x$; $v = W\, y$.

3.2.1.3 Partial least square regression

PLS is widely used as a soft sensor in many applications, because of its ability to tackle collinearity in the datasets. It finds components from predictor variables, x, which are relevant to that of dependent variables, y.

Consider the input and output variables as $x_i = \{x_1, x_2, \dots x_m\}^T$ and $y_i = \{y_1, y_2, \dots y_n\}^T$ such that $X \in \mathbb{R}^{k \times m}$; $Y \in \mathbb{R}^{k \times n}$ where, k – number of samples. Every samples in the matrix are mean centered and scaled appropriately (Qin, 1998; Abdi, 2003).

In PLS, both X and Y, decomposed simultaneously into

$$\boldsymbol{X = TP^T + E} \quad \textbf{(3.17)}$$

$$\boldsymbol{Y = TQ^T + F} \quad \textbf{(3.18)}$$

with $T^T T = I$ (Identity matrix)

Where, T – latent variable (score) matrix; P – loading matrix of X; Q – loading matrix of Y; E & F – error matrix

The regression coefficients using PLS models is calculated as:

$$\boldsymbol{\beta = W_{pls} (P_L^T W_{pls})^{-1} q_L^T} \quad \textbf{(3.19)}$$

Where, W_{pls} – Weight matrix (for PLS model); P_L– Loading matrix of input variables; q_L – Loading vectors for output variables.

3.2.2 Nonlinear models

3.2.2.1 Support vector regression

Support vector regression (SVR) models have become attractive alternatives to neural network models for nonlinear processes. For solving convex quadratic optimization problems, SVR model develops linear relationship with output by projecting the predictor variables nonlinearly into the higher dimensional feature space, $\boldsymbol{F}$ (Cortes & Vapnik 1995; Vapnik 1999; Herceg et al. 2019).

Support vector regression is a regressor form of support vector machines, used to develop nonlinear models by applying kernel trick along with support vector machines. Using kernel functions (linear, polynomial, radial basis functions), the input vectors are

mapped in the higher dimensional feature space, $\boldsymbol{F}$. All kernel functions must obey the mercer theorem. In conventional statistical/machine learning techniques, risk minimization approach is employed to enhance the performance of developed models. Two types of risk minimization approaches are followed: structural risk minimization and empirical risk minimization. SVR follows the structural risk minimization approach, where the prediction error and model complexity are simultaneously kept minimal. However, empirical risk minimization concept focused only on minimizing the training prediction error.

Consider a linear estimation function to solve nonlinear regression problems:

$$\boldsymbol{f(x) = \left(w.\varphi\,(x)\right) + b} \qquad \textbf{(3.20)}$$

Where, w – weight parameter; $\varphi\,(x)$ – feature function; b – bias value or constant; $\left(w.\varphi\,(x)\right)$ – dot product in the feature space, $\boldsymbol{F}$ such that $\Phi: x \rightarrow \boldsymbol{F},\ w \in \boldsymbol{F}$

In SVR, nonlinear problem in the lower dimensional input space (x) is transformed into linear regression problem in the high dimensional feature space, $\boldsymbol{F}$. The formulation of SVR comprises of empirical risk and complexity term $\|w\|^2$, which minimizes the regularized risk function that is represented as Equation 3.21 to avoid overfitting phenomenon.

$$\boldsymbol{R_{reg}[f] = R_{emp}[f] + \frac{1}{2}\|w\|^2} \qquad \textbf{(3.21)}$$

Where, R_{reg} – regression risk; $\|\cdot\|$ - Euclidean norm; R_{emp}- empirical risk

Regularization term, $\frac{1}{2}\|w\|^2$ provides the trade-off between model complexity and performance without compromising the generalization capability of developed model. The complexity of linear function can be controlled by keeping the value of w, as small. Regularized risk function is similar to cost function along with standard decay term employed for developing artificial neural network model possessing generalization capability.

Cost functions which are employed to formulate the SVR model are Laplacian, Huber's, Gaussian and ε-insensitive; in which ε-insensitive is mostly preferred.

$$\boldsymbol{L_{\varepsilon}\,(f(x)-y)} = \begin{cases} \boldsymbol{|f(x)-y|-\ \varepsilon} & \boldsymbol{for} \quad \boldsymbol{|f(x)-y|\ \geq\ \varepsilon} \\ \boldsymbol{0} & \boldsymbol{otherwise} \end{cases} \tag{3.22}$$

Where, ε – precision parameter or tube radius located around the regression function, $f(x)$.

The region which is enclosed inside the tube is known as ε-insensitive zone, where the loss function assumed to be zero value. Moreover, if the loss function within this region found to be smaller than ε, predictions error assumed to be negligible. Slack variables (ξ_i and ξ_i^*) are used to measure the deviation ($y_i - (fx_i)$) from the boundaries of the tube (ε-insensitive zone). Empirical risk minimization using symmetric loss function can be expressed by adding the slack variables (ξ_i and ξ_i^*) $i = 1 \ldots p$ together with set of linear constraints in regression risk function.

Primary form of support vector machine with L1 soft-margin formulation can be written as optimization problem using ε-insensitive loss function along with regularization constant, C:

$$\boldsymbol{Minimize}: \quad \boldsymbol{\frac{1}{2}\|w\|^2 + C\ \sum_{i=1}^{p}(\xi_i + \ \xi_i^*)} \tag{3.23}$$

Subject to,

$$\begin{Bmatrix} \boldsymbol{\left(w\ .\ \Phi(x_i)\right) + b - \ y_i\ \leq\ \varepsilon + \ \xi_i^*} \\ \boldsymbol{y_i - \left(w\ .\ \Phi(x_i)\right) - b\ \leq\ \varepsilon + \ \xi_i^*} \\ \boldsymbol{\xi_i, \xi_i^*\ \geq 0 \quad for \quad i = 1, \ldots . . p} \end{Bmatrix} \tag{3.24}$$

Here, C is known as regularization constant or penalizing factor, which controls the tradeoff between model complexity and training errors. The significance of penalizing constant is to define the tolerance limit in which the error propagates beyond $\pm\varepsilon$-insensitive region. If the magnitude of C is large, SVR minimizes the empirical risk alone regardless of

model complexity. For low C value, it gives rise to more generalization of fitting model by assigning insignificant weightage to training data points.

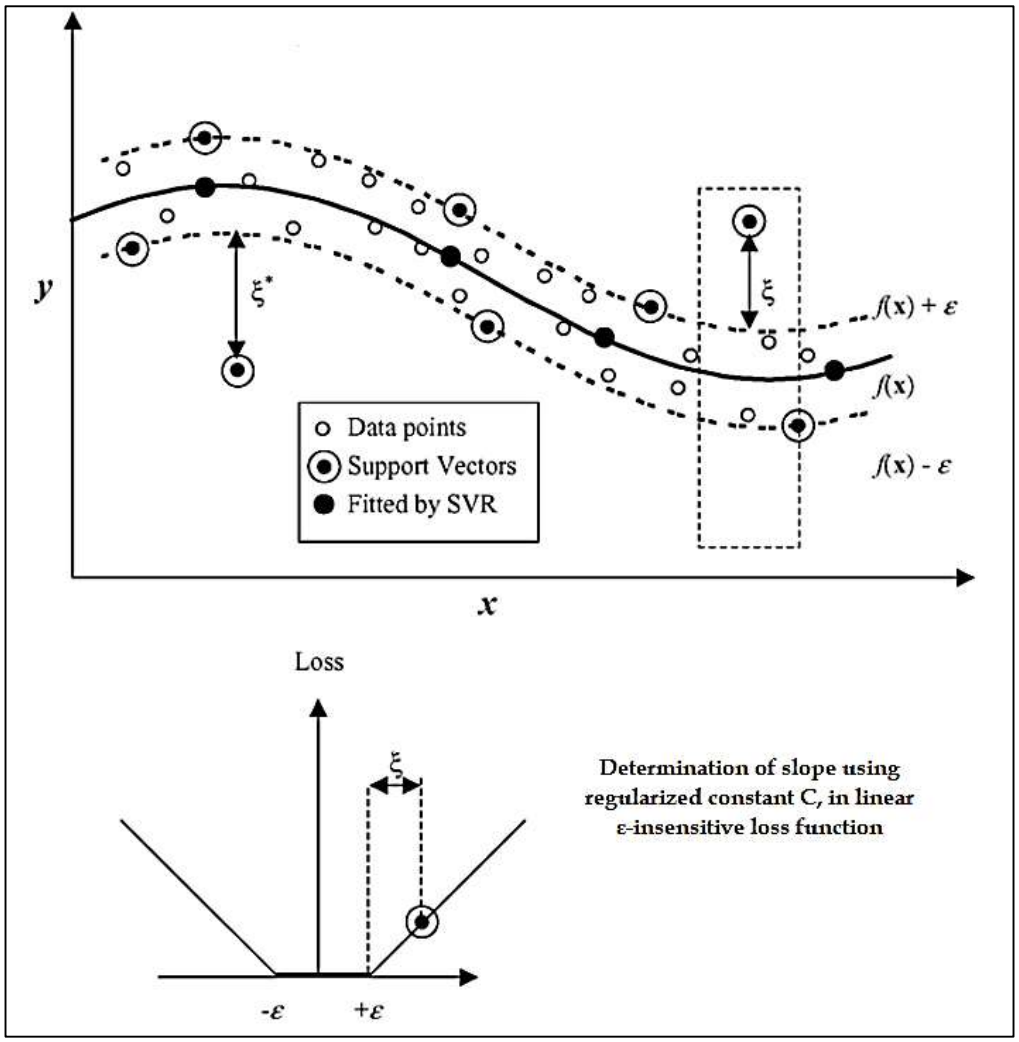


Figure 3.4: Pictorial representation of ε-insensitive loss function in SVR model (Desai et al. 2006)

SVR optimizes the position of the tube (ε-insensitive) around the data points by performing structural risk minimization as shown in Figure 3.4. $f(x)$, represents the fitted function using training data points. Using optimization criterion, the outputs of the respective training points are penalized, which are lies ε distance away from the fitted

function, $f(x)$. The excess positive and negative deviations from ε-insensitive is mentioned in the form of slack variables ξ_i and ξ_i^* respectively. Beyond the $[\varepsilon, -\varepsilon]$ insensitive range, slack variables have non-zero values. By minimizing $[\xi_i, \xi_i^*]$ and $\|w\|^2$ value, the training set errors and model over-complexity of SVR model was reduced significantly. This also avoids overfitting or under fitting phenomena, when the function $f(x)$ is fitted using training dataset (Desai et al. 2006).

The regularized risk function defined in Equation 3.21 can be minimized by the function holding finite number of parameters has the form:

$$\boldsymbol{f(x, \alpha, \alpha^*) = \sum_{i=1}^{p} (\alpha_i - \alpha_i^*)(\Phi(x_i) . \Phi(x)) + b} \quad \textbf{(3.25)}$$

Where, α_i^*, α_i – Lagrangian multipliers related to input vectors; $\alpha_i^*, \alpha_i \geq 0$ with $\alpha_i^* \alpha_i = 0$; $(\Phi(x_i) . \Phi(x))$ - inner product of two elements of Hilbert space

To find the value of lagrangian multipliers or coefficients α_i^* and α_i, it is necessary to solve the quadratic optimization problem based on Karush-Kuhn-Tucker conditions:

Maximize the function

$$\boldsymbol{R(\alpha^*, \alpha) = -\varepsilon \sum_{i=1}^{p} (\alpha_i^* + \alpha_i) + \sum_{i=1}^{p} y_i (\alpha_i^* - \alpha_i) - \frac{1}{2} \sum_{i,j=1}^{p} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) \left(\Phi(x_i), \Phi(x_j)\right)} \quad \textbf{(3.26)}$$

Subject to constraints

$$\boldsymbol{\sum_{i=1}^{p} (\alpha_i^* - \alpha_i) = 0, \quad 0 \leq \alpha_i, \alpha_i^* \leq C, \quad i = 1, \ldots p.} \quad \textbf{(3.27)}$$

Where, x_i – support vectors

To evaluate the inner product, $\left(\Phi(x_i), \Phi\left(x_j\right)\right)$ in Hilbert space, the following expression is deployed.

According to Hilbert space theory,

For a symmetric function, $K\ (u, v)$, the expression can be written as:

$$\boldsymbol{K\ (u, v) = \sum_{k=1}^{\infty} \alpha_k \phi_k(u) \phi_k(v)} \quad \textbf{(3.28)}$$

having positive coefficients $\alpha_k > 0$.

Further, $K\ (u, v)$ is the inner product in the feature space ϕ, and the required conditions should be valid for any nonzero function, g in the Hilbert space (Vapnik et al. 1996).

$$\boldsymbol{\int K(u, v)\, g(u) g(v) du dv > 0} \quad \textbf{(3.29)}$$

After simplification, Equation 3.29 can be written as,

$$\boldsymbol{f(x, \alpha_i, \alpha_i^*) = \sum_{i=1}^{p} (\alpha_i^* - \alpha_i) K(x, x_i) + b} \quad \textbf{(3.30)}$$

Where, $K(x, x_i)$ – Kernel of support vectors; $K(x, x_i)$ - inner product of $\left(\Phi(x_i), \Phi\left(x_j\right)\right)$

The coefficients α_i^* and α_i can be found by solving the quadratic optimization problem,

Maximize the function

$$\boldsymbol{R(\alpha^*, \alpha) = -\varepsilon \sum_{i=1}^{p} (\alpha_i^* + \alpha_i) + \sum_{i=1}^{p} y_i\ (\alpha_i^* - \alpha_i) - \frac{1}{2} \sum_{i,j=1}^{p} (\alpha_i^* - \alpha_i)\left(\alpha_j^* - \alpha_j\right) K(x_i, x_j)} \quad \textbf{(3.31)}$$

Subject to constraints

$$\boldsymbol{\sum_{i=1}^{p} (\alpha_i^* - \alpha_i) = 0, \quad 0 \le \alpha_i, \alpha_i^* \le C, \quad i = 1, \ldots p.} \quad \textbf{(3.32)}$$

Bias parameter, b can be found as

$$\boldsymbol{b = \begin{cases} y_i - f(x_i)_{b=0} - \varepsilon & for \quad \alpha_i \in (0, C) \\ y_i - f(x_i)_{b=0} + \varepsilon & for \quad \alpha_i^* \in (0, C) \end{cases}} \quad \textbf{(3.33)}$$

Lagrangian multipliers (α_i and α_i^*) are forces tend to push the regression function ($f(x_i)$) towards the desired output, y_i. Each training data has a strong association with a pair of lagrangian multipliers (α_i and α_i^*). Some regression coefficients($\alpha_i - \alpha_i^*$) possess non-zero values, in which training data with non-zero coefficients are called support vectors ($|f(x_i) - y_i| \geq \varepsilon$).

Support vectors are crucial points for determining the SVR approximated function. These vectors are represented as points as shown in Figure 3.1 on the surface and outside of ε-insensitive tube. If the number of support vectors are smaller, more generalized approximate function is obtained. Also, prediction of new unknown output from the obtained function requires less computation. Moreover, regression function approximates the data points which are present inside the ε-insensitive tube. There is no influence in the solution of regression task, when the regression coefficients ($\alpha_i - \alpha_i^*$) of training data points possess zero value. The tube width parameter ε, is inversely proportional to the number of support vectors used to build the regression function. The decrease in the value of ε, leads to increase in support vectors associated with poor generalization and over- fitting of training data points due to complex model. However, better generalization of model associated with high training set error resulted due to large magnitude of ε.

In SVR, two optimization approaches sequential minimal optimization (SMO) algorithm and iterative single data algorithm (ISDA) are used to solve large quadratic programming problems. SMO (Platt 1999; Smola & Schölkopf, 2004) is based on the rule of second-order iterative selection algorithm that uses two Lagrangian multipliers as a reference to solve optimization problems faster than the existing quadratic programming in SVR. ISDA (Kecman et al. 2005) works by classical Gauss-Seidel iterative algorithm

updating the single Lagrangian multiplier every-time for a huge dataset to converge rapidly. Here, the selection of optimum value of loss function (ε) for modeling applications is very important to select the best performing models with reasonable prediction (Shokri et al. 2015; Yan et al. 2004).

3.2.2.2 Generalized regression neural network

Local models used in the adaptive framework can be based on linear or nonlinear techniques. Since most industrial processes are inherently nonlinear in nature, it is highly desirable to develop robust nonlinear local models which can cope with high-dimensional and nonlinear data. The various nonlinear local models reported in the adaptive framework include: support vector regression, least square support vector regression (Ge & Song, 2010), kernel principal component regression (Yuan et al, 2014), Gaussian process regression (Shi & Xiong, 2020) and back propagation neural networks (Chen et al. 2017). These techniques require optimum values of various hyper-parameters to approximate the given system. In case of support vector regression, kernel parameter, regularization parameter, error bound; for Gaussian process regression, signal variance, noise variance, length scale (Shi & Xiong, 2018) and for kernel PCR, the kernel type and parameters are to be specified (Pani, 2021). Similarly, use of back-propagation neural networks requires knowledge of number of hidden layers, number of neurons in hidden layers, weights, and transfer function for hidden and output layer prior to model development (Chen et al. 2017; Singh et al. 2019).

Generalized Regression neural network (GRNN) is another special case of probabilistic learning algorithms with single pass and parallel memory structure in the family of neural networks. GRNN was originally proposed by Specht (1991) for prediction and control of

processes by understanding the plant dynamics. Unlike other aforementioned nonlinear techniques, GRNN trains itself in a shorter duration and requires the least number of hyper-parametric values (i.e. spread parameter) to be specified. There has been only limited application of GRNN technique for predictive modeling of quality parameters in industries. A hybrid integrated Principal component analysis based GRNN approach was proposed by Kulkarni et al. (2004) for nonlinear fed-batch fermentation process involving biosynthesis of penicillin and protein synthesis. The technique of GRNN is further applied in prediction of ethane content in Ethane-Ethylene distillation column, butane content in Debutanizer column, tail gas composition in Sulphur recovery unit, prediction of chlorine residuals in water distribution system, rate of coagulation dosage in drinking water treatment plant, particle size in cement mill and cement clinker quality (Singh et al. 2019; Bowden et al. 2006; Heddam et al. 2011; Pani et al. 2012; 2013). Review presented above indicates some application of GRNN for industrial quality monitoring. However, all these applications are steady state models.

GRNN uses theory of non-linear regression for defining the control function so as to reduce the residual squared error for estimation of output (Specht, 1991; Kulkarni et al. 2004). It also provides better prediction by mapping the non-linear systems with less number of training samples. GRNN monitors and estimates the relationship between sparse variables in multidimensional space. This parallel architecture can be effectively used in real time estimation and control of process by learning the dynamics of systems.

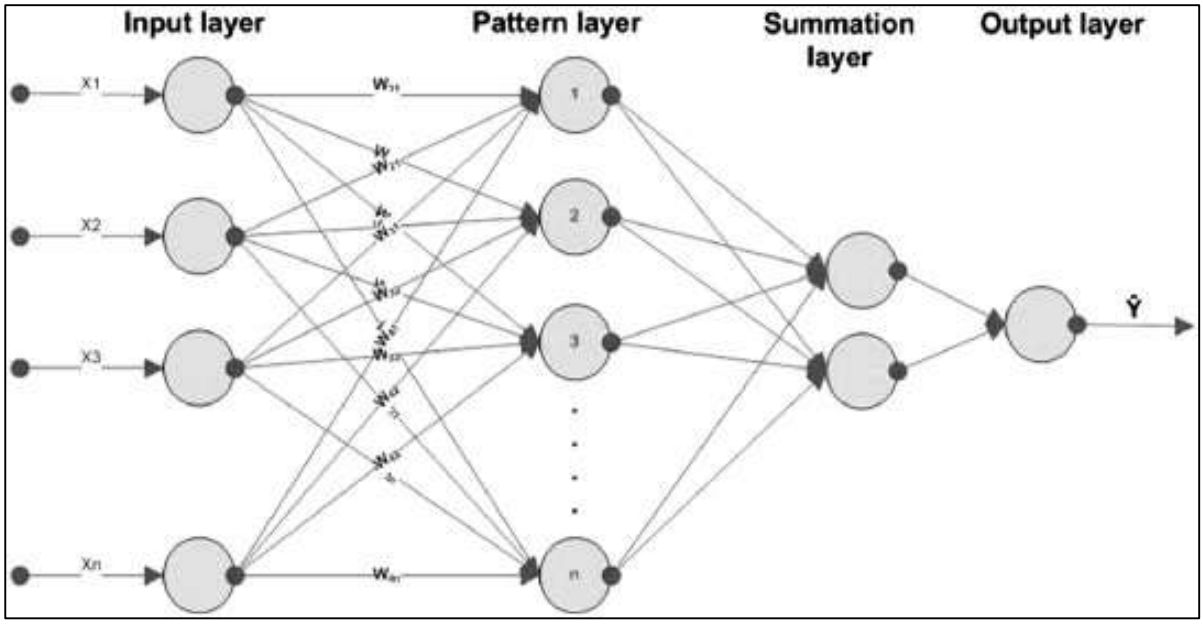


Figure 3.5: Generalized Regression Neural Network (Pani et al. 2013)

The neural architecture of GRNN comprises four distinct layers which are input layer, pattern layer, summation layer and output layer.

The GRNN function (conditional mean) can be described as:

$$\boldsymbol{E[y|x]} = \frac{\int_{-\infty}^{\infty} \boldsymbol{y}\, \boldsymbol{f(x,y)dy}}{\int_{-\infty}^{\infty} \boldsymbol{f(x,y)dy}} \qquad \textbf{(3.34)}$$

Here, x and y are input and output vectors. $E[y|x]$ is output expected for the given input, $f(x,y)$ is the joint continuous probability density function of x and y.

$$\widehat{\boldsymbol{f}}\,\boldsymbol{(x,y)} = \frac{\mathbf{1}}{\mathbf{2\pi^{(p+1)/2}\,\sigma^{(p+1)}}} \times \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{exp}\left[-\frac{\boldsymbol{(x-u_i)^T\,(x-u_i)}}{\mathbf{2\sigma^2}}\right] \times \boldsymbol{exp}\left[-\frac{\boldsymbol{(y-y_i)^2}}{\mathbf{2\sigma^2}}\right] \qquad \textbf{(3.35)}$$

The joint probability density function $f\,(x,y)$ can be estimated through the observations x and y. The function $\hat{f}(x,y)$ which is a probability function estimator, is found from the sample values u_i and y_i from the pool of observations x and y in a random manner. This estimator is also known as non-parametric Parzen window estimator. Substitution of probability function estimator in conditional mean provides desired

conditional mean of y_i for a given x. Here, n represents number of sample observations while p corresponds to dimensions of variable x.

The exponential form of activation function (Gaussian radial basis function, h_i) found in pattern layer can be written as:

$$\boldsymbol{h_i = e^{-I_j^2/2\sigma^2}} \quad \textbf{(3.36)}$$

The distance can be found by pattern neurons through square of difference across all weights. This can be clearly described by net input for j^{th} pattern neuron,

$$\boldsymbol{I_j = \sum_{i=1}^{n}(w_{ij} - u_j)^2} \quad \textbf{(3.37)}$$

The function value can be described optimally as:

$$\boldsymbol{y_j = \frac{S_D}{S_S} = \frac{\sum_{i=1}^{n} h_i w_{ij}}{\sum_{i=1}^{n} h_i}} \quad \textbf{(3.38)}$$

$$\boldsymbol{y_j = \frac{\sum_{i=1}^{n} \exp\left(-\frac{D_i^2}{2\sigma^2}\right) w_{ij}}{\sum_{i=1}^{n} \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}} \quad \textbf{(3.39)}$$

$$\boldsymbol{D_i^2 = (x - u_i)^T (x - u_i)} \quad \textbf{(3.40)}$$

$$\boldsymbol{y_j = \frac{\sum_{i=1}^{n} \exp\left(-\frac{(x-u_i)^T (x-u_i)}{2\sigma^2}\right) w_{ij}}{\sum_{i=1}^{n} \exp\left(-\frac{(x-u_i)^T (x-u_i)}{2\sigma^2}\right)}} \quad \textbf{(3.41)}$$

Here, w_{ij} is the target weight or output corresponds to labeled training samples (x and y). All the neurons in the pattern layer were linked with each of the neurons in two summation layer which are represented as S and D. The output layer divides the output from S-summation layer to the output from D-summation layer to estimate the quality variable for the given input vector. The constant σ is known as smoothing function or spread parameter, helps to control the size of perceptive region. The successful design of GRNN depends upon the choice of selecting the distinct spread parameter (Specht, 1991; Kisi, 2006).

With aforementioned discussion, the procedure for implementing GRNN model is summarized as:

GRNN training procedure (Pani et al. 2012)

Step 1: Set – Input, Target and Spread values

Step 2: Design Feed forward neural network (Input: 1, Hidden: 1, Output: 1)

Step 3: Set –

No. of input layer nodes = No. of input variables;
No. of output layer nodes = No. of target variables;
No. of hidden layer nodes = No. of observations used for training the network

Step 4: Set –

Weight of input to hidden layer nodes (W_1) = Actual input values
Weight of hidden to output layer nodes (W_2) = Actual target values

Step 5: Net input = Euclidean distance (W_1 and Input values)

Step 6: Hidden layer output = Exponential of net input

Step 7: Final output = Hidden layer output * W_2

Selection of spread parameter or kernel width (σ) is crucial for successful design of GRNN model. Once the training data is available with labeled samples, GRNN maps arbitrary function by proper selection of spread parameter. The GRNN architecture has an inherent advantage of both statistical and machine learning scheme. It is based upon kernel regression and radial basis function neural network. The final regression part estimates the final output by minimizing the mean squared error between predicted and observed data. The estimation error tends to be zero, when large number of training data is taken for model development (Specht, 1991). GRNN has been used as a steady state quality monitoring soft sensor with reasonable estimation accuracy. Therefore, applying GRNN model with

continuous online model adaptation capability for adaptive soft sensor design is rarely investigated.

Selection of spread parameter for GRNN in adaptive soft sensor development

Selection of spread parameter or kernel width (σ) is crucial for successful design of GRNN model. The optimized kernel width (standard deviation) can either be found through cross-validation technique (grid search) by developing models through range of spread values (Pani & Mohanta, 2015), circuit training (Feng et al. 2017; 2019) or hold or leave one out method (Specht, 1991). In this work, we implemented grid search approach using fine grid mesh to search the optimum kernel width. The range of spread values for search operation is chosen as below:

Range of grid search (for JITL-D and JITL-A&D),

$$\boldsymbol{\sigma = [0.001, 0.01, 0.1, 1, 5, 10]} \qquad \textbf{(3.42)}$$

The methods of trial and error, leave one out, cross validation etc. are difficult to implement when the model parameter value is expected to change continuously during online use. Therefore, theoretical decision method proposed by Cherkassky and Ma (2004) is applied in this work to compute spread parameters for recursive and sliding window GRNN models.

According to this method, the analytical approach for computing σ is given as:

$$\boldsymbol{\sigma^d \sim (0.1 - 0.5)} \qquad \textbf{(3.43)}$$

Here, d is the number of input variables. For effective use of the above relation, σ^d was equated separately to 0.1 to 0.5 and from each equation, a kernel value was computed resulting in 5 kernel values for each dataset. Finally the values obtained were averaged to

obtain a unique σ value for a particular dataset. The results for σ computation of the three industrial datasets are shown in Table 3.1.

Table 3.1: GRNN spread parameter values computed (recursive and sliding window) using analytical method based on Cherkassky and Ma (2004)

⬇Process / σ^d ➡	0.1	0.2	0.3	0.4	0.5	Mean
Crude distillation unit (d = 6)	0.6813	0.7647	0.8182	0.8584	0.8909	0.8027
Debutanizer column (d = 7)	0.7197	0.7946	0.8420	0.8773	0.9057	0.8279
Sulfur recovery unit (d = 5)	0.6310	0.7248	0.7860	0.8326	0.8706	0.7690

Once the training data is available with labeled samples, GRNN maps arbitrary function by proper selection of spread parameter. The GRNN architecture has an inherent advantage of both statistical and machine learning scheme. It is based upon kernel regression and radial basis function neural network. The final regression part estimates the final output by minimizing the mean squared error between predicted and observed data. The estimation error tends to be zero, when large number of training data is taken for model development (Specht, 1991). GRNN has been used as a steady state quality monitoring soft sensor with reasonable estimation accuracy. Therefore, applying GRNN model for adaptive soft sensor design is yet to be explored in applications of process industries.

3.3 Integration of non-linear local models in the recursive Just-in-Time learning framework

In JITL technique, the database may be fixed or may be continuously updated. In most of the reported techniques the database is either fixed or continuously updated by including the query sample without any removal of data from database. In the former, the prediction accuracy may drop due to the neighborhood selection based on the past process states, while the current process states yet not updated in the database. However, in the later, this leads to an increase in database size with time and consequently, the computation time for each sample will increase with time (this is because, initially, distance of the query sample from each sample of the database has to be computed).

It is very important that the data having good information must be included and those with poor information must be excluded from the database (Kaneko & Funatsu, 2014). In this work, a database update technique is adopted where, the latest query data is included in the database and one sample already present in the database having the least similarity with query data, is removed. By following this recursive computation approach, the database size is fixed while ensuring continuous update of the database at each sampling instance. In the proposed integrated recursive approach, the entire database (training set) is taken for similarity calculation for each query objects which is slightly different from integrated moving window approach. However, the latter takes only the small window (decided by user) of database samples for similarity calculation. The database is updated only if the prediction error is less than or equal to the threshold limit, d which is considered here as 0.01.

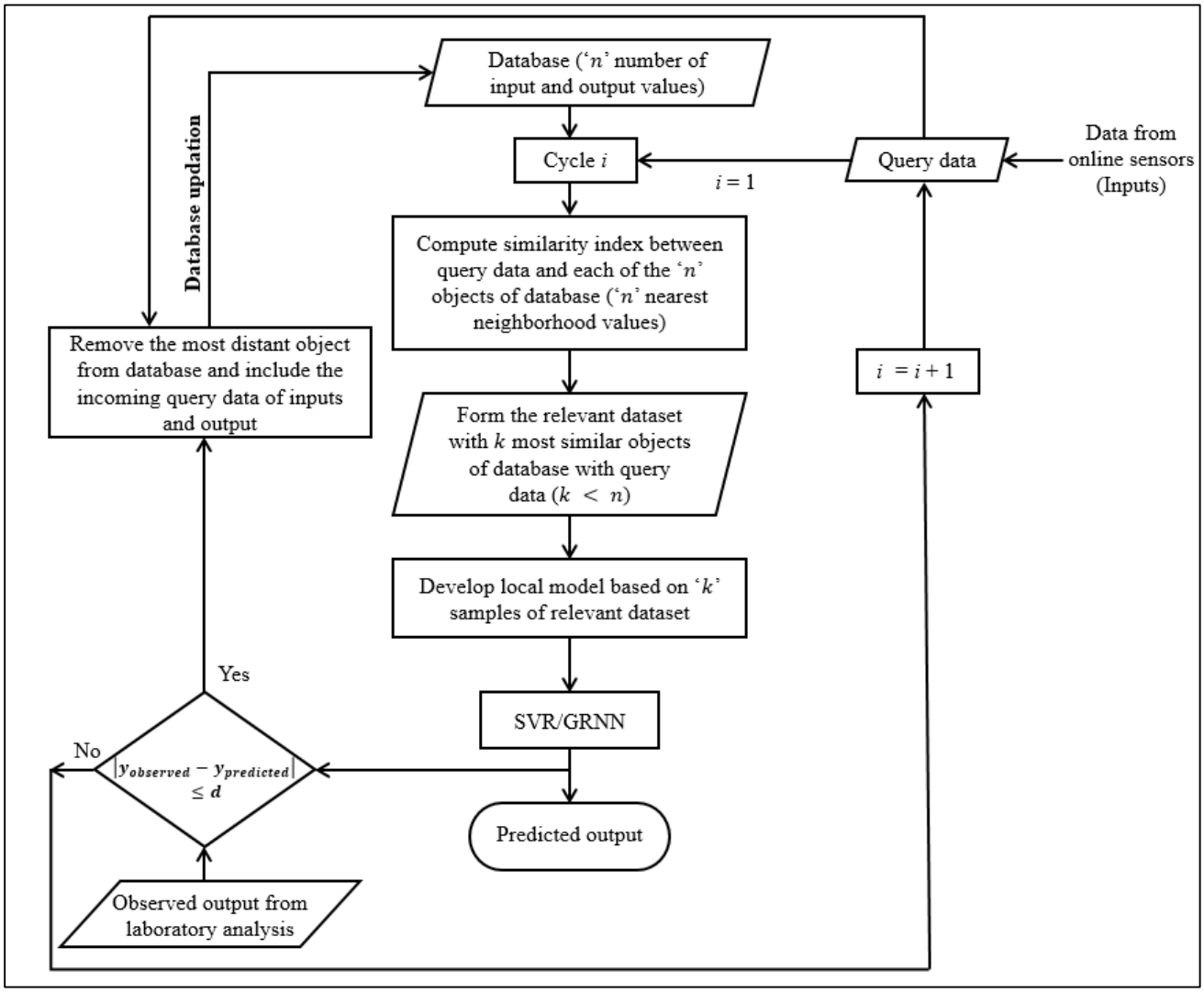


Figure 3.6: JITL flowchart for nonlinear models

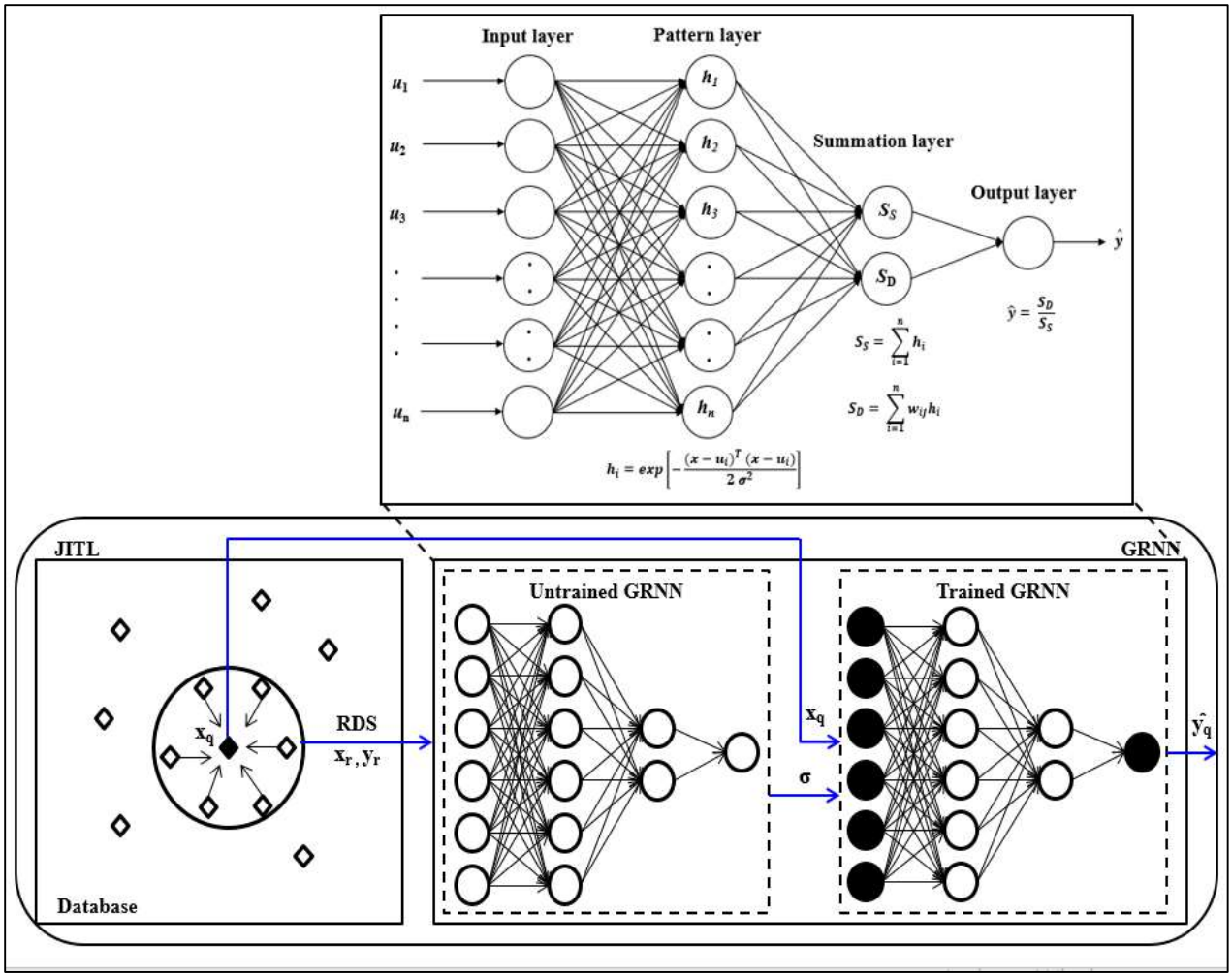


Figure 3.7: JITL-GRNN architecture

The mechanism of recursive JITL algorithm with non-linear local modeling strategy (SVR or GRNN) adopted in this work is presented in Figure 3.7. The SVR local model is optimized with SMO or ISDA algorithm. Hyper parameter of GRNN local model is determined by grid search method or analytical method. Pictorial representation of GRNN model embedded with just-in-time learning frameworks in revealed in Figure 3.8.

3.4 Performance improvement of adaptive models using bias update

In addition to use of non-linear local models and recursive database update, the discrepancy in the value of the primary variables obtained from laboratory and predicted through soft sensors during the model update process is also considered for better accuracy.

In the actual industrial scenario, where the output quality is monitored by offline laboratory analysis, there is substantial time interval between two consecutive quality data obtained from laboratory. Major causes for this time delay are analysis time taken by the offline or online analyzers, uncertainty in time of sampling, instrument failure and analyzers under repair/maintenance. New query samples will arrive for quality estimation in the subsequent mean time before the arrival of the feedback for the old sample which is already been processed. Due to this, model update also gets delayed. In order to address this practical industrial constraint, a bias update procedure is proposed in this work to get timely update of the current model using the predicted output. Error between predicted and actual values at a particular instance is called as bias at that instance.

The bias updating procedure encompasses addition of bias from the immediate previous instance (when the laboratory data was available) to the predicted output of the current instance. During the time interval of two consecutive laboratory information, the bias value is constant until, when next laboratory result arrives for update.

Let $\hat{y}(k)$ is the predicted output value of the adaptive model at k^{th} instant. Then according to bias update procedure, the final corrected output value at k^{th} instant is calculated as:

$$\boldsymbol{y_c(k) = \hat{y}(k) + bias(k)} \qquad \textbf{(3.44)}$$

An offset bias smoother is added for correcting the predicted output with respective to laboratory values. However, the initial value of bias is taken as zero. The weighting factor, ω is user adjusted parameter which depends on the individual process system (Mu et al., 2006; Ahmed et al., 2009; Jia et al. 2011).

$$\boldsymbol{bias(k) = \omega bias_0(k) + (1 - \omega) bias(k - i)} \quad \textbf{(3.45)}$$

$$\boldsymbol{bias_0(k) = y_0(k - i) - \hat{y}(k - i)} \quad \textbf{(3.46)}$$

Where, $bias_0(k)$ is the current offset bias; $bias\ (k - i)$ provides the difference at $(k - 2)$ instant; ω – bias weighting factor (range: 0.1-0.9); y_0 – Observed output from laboratory; $\hat{y}$ – Predicted output from developed model.

The final corrected output value, y_c is calculated as:

$$\boldsymbol{y_c(k) = \hat{y}(k) + bias(k)} \quad \textbf{(3.47)}$$

Usually, the range of bias weighting factor (ω) is chosen between 0.1 and 0.9. Lower the value of ω corresponds to small weight to current bias followed by higher weight to bias corresponds to previous instant and vice versa.

In this work, adaptive updating strategy for database is adopted to improve the model performance and reduce the computational load. Confidence limit, d is introduced in the algorithm to update the database adaptively only if it crosses the certain threshold value. If the squared prediction error between predicted and actual output higher than the confidence limit, then the database gets updated with actual output value obtained from laboratory analysis. Otherwise, the predicted output is subjected along with respective query for update of training database. The confidence limit, d is described as:

$$\boldsymbol{d = \frac{v}{2m} \chi^2} \quad \textbf{(3.48)}$$

Here, v - variance of output in training database (for sliding window model – samples taken in window); m – output mean; $\chi^2 = \frac{(\hat{y}-y_o)^2}{y_o}$

Figures 3.8 to 3.10 illustrate the mechanism of bias update in the recursive JITL, recursive and sliding window framework respectively.

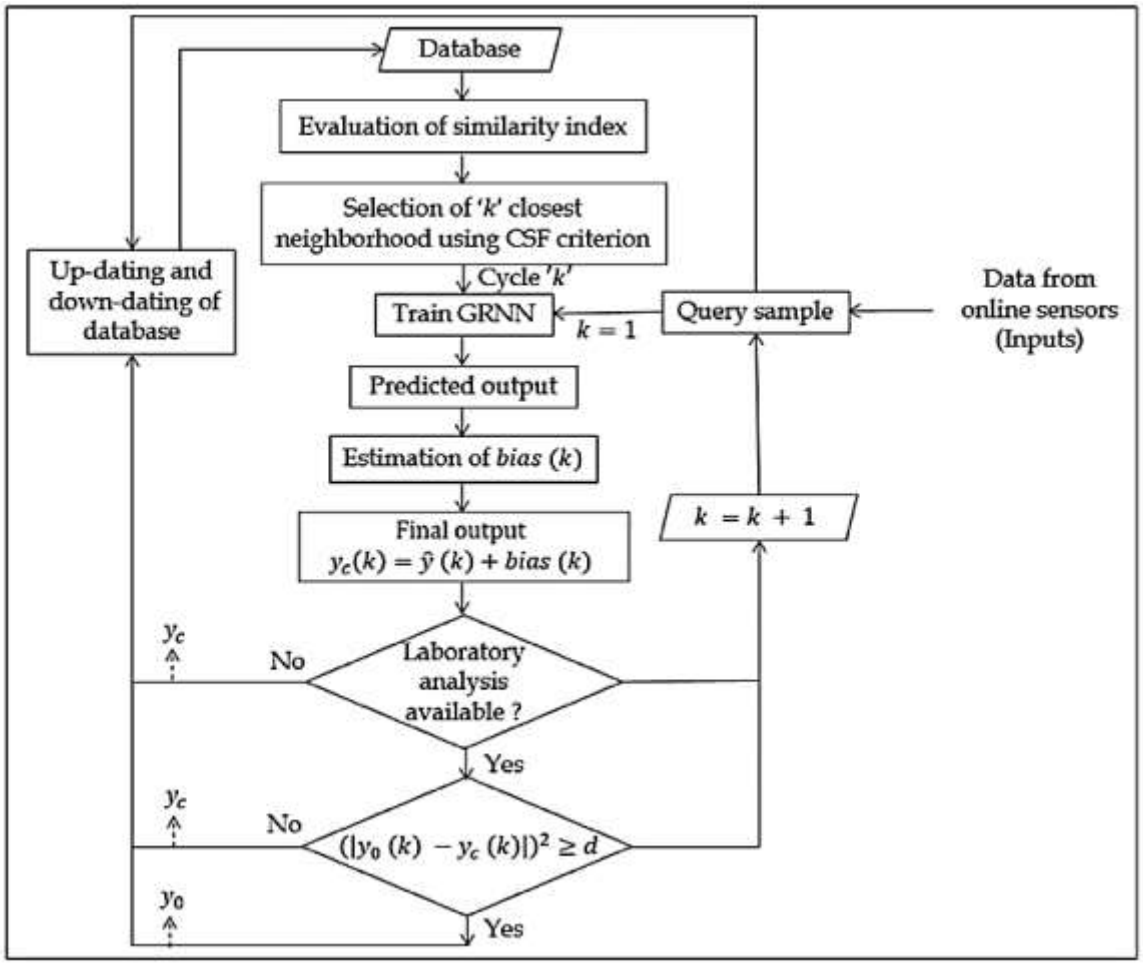


Figure 3.8: Recursive Just-in-Time Learning technique with bias update

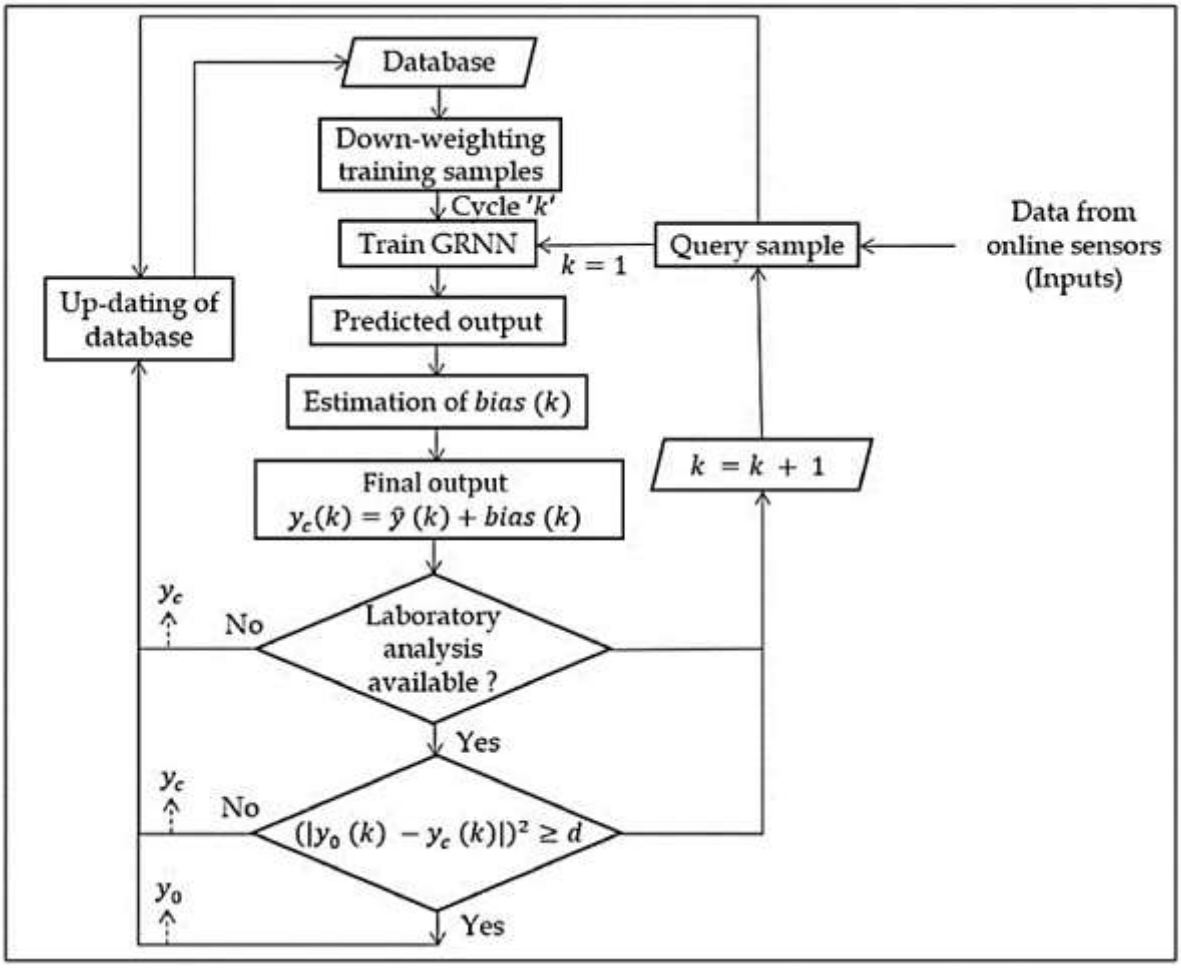


Figure 3.9: Recursive technique with bias update

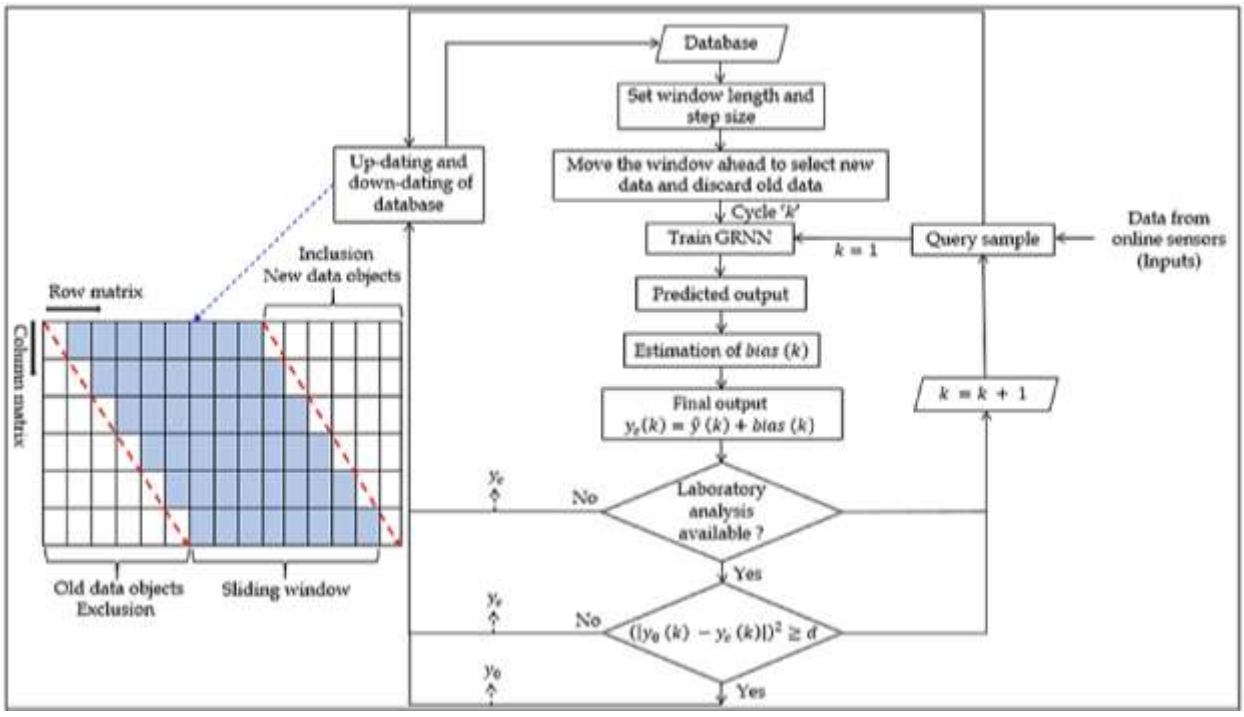


Figure 3.10: Sliding window technique with bias update

Chapter - 4

Case studies – Industrial Applications

Petroleum (also known as crude oil), a fossil fuel with complex hydrocarbons, forms a chief source of energy. The per-capita consumption of conventional petroleum and petroleum related products increases steadily with significant drop in exploration of oil wells simultaneously shows an alarming trend, which tends towards major energy crisis. It is indispensable for process industry as well as individual, to safeguard the energy resources for the future generation. Reduction in usage of energy resources forms the basis for the efficient and optimized control of industrial processes. The other agenda for process control is to avoid process variations and customer complaints of products. Models were used to design controllers, that are implemented to control the chemical processes.

Petroleum undergoes three different stages (upstream, midstream and downstream) of process before approaching the end usage of consumers. The upstream process includes dewatering, desalting and desulphurization units. The mid-stream stage comprises multi-component fractionation, hydro cracking, reforming and hydro treating. Finally, the downstream process includes petrochemical complexes such as formation of fertilizers, polymers, dyes and pigments. To meet the product specification and be customer compliant with reduced off-specification products and maximum profit, it is crucial to monitor and control the quality of products in petroleum refinery. Measurement of product quality through laboratory analysis is a time consuming process. Use of soft sensor will facilitate real time monitoring of product quality.

In this work, adaptive soft sensors are developed for monitoring quality parameters in three processes of refinery (as mentioned earlier in Chapter 1). The three processes along with the quality parameters to be monitored are presented in the following sections.

4.1 Naphtha splitter section

Raw naphtha is a mid-stream liquid distillate, fractionated from atmospheric distillation unit. It is a mixture of alkanes, cycloalkanes and aromatics with carbon atoms range from 5 to 11 and majorly used as a solvent for elastomers, diluents for paints and varnishes, hydrogen production and blended with gasoline to form high octane fuels (Duchene et al., 2020). For different crude asset (Paraffinic, Naphthenic or Aromatic), the boiling range of fractional cut varies based on composition.

4.1.1 Initial boiling point and end boiling point prediction of heavy naphtha

Initial boiling point (IBP) and End boiling point (EBP) are the two key indicators for naphtha quality which is obtained from laboratory analysis with significant time delay. Laboratory analysis based on EN ISO 3405 and ASTM D86 standards, provides the boiling range characteristics of different petroleum products under various conditions at atmospheric pressure. Moreover, online sensors e.g. gas chromatographs possess measurement delay. The sampling time varies from few minutes to several hours. In this case, the laboratory assays are conducted twice a day (Ujević et al., 2011). It is mandatory to maintain the quality of refinery products in which the properties of naphtha to be continuously monitored and controlled. Hence, there is a strong need for online monitoring of naphtha boiling point. In the absence of availability of any hardware sensor for the same, soft sensor can be a viable alternative.

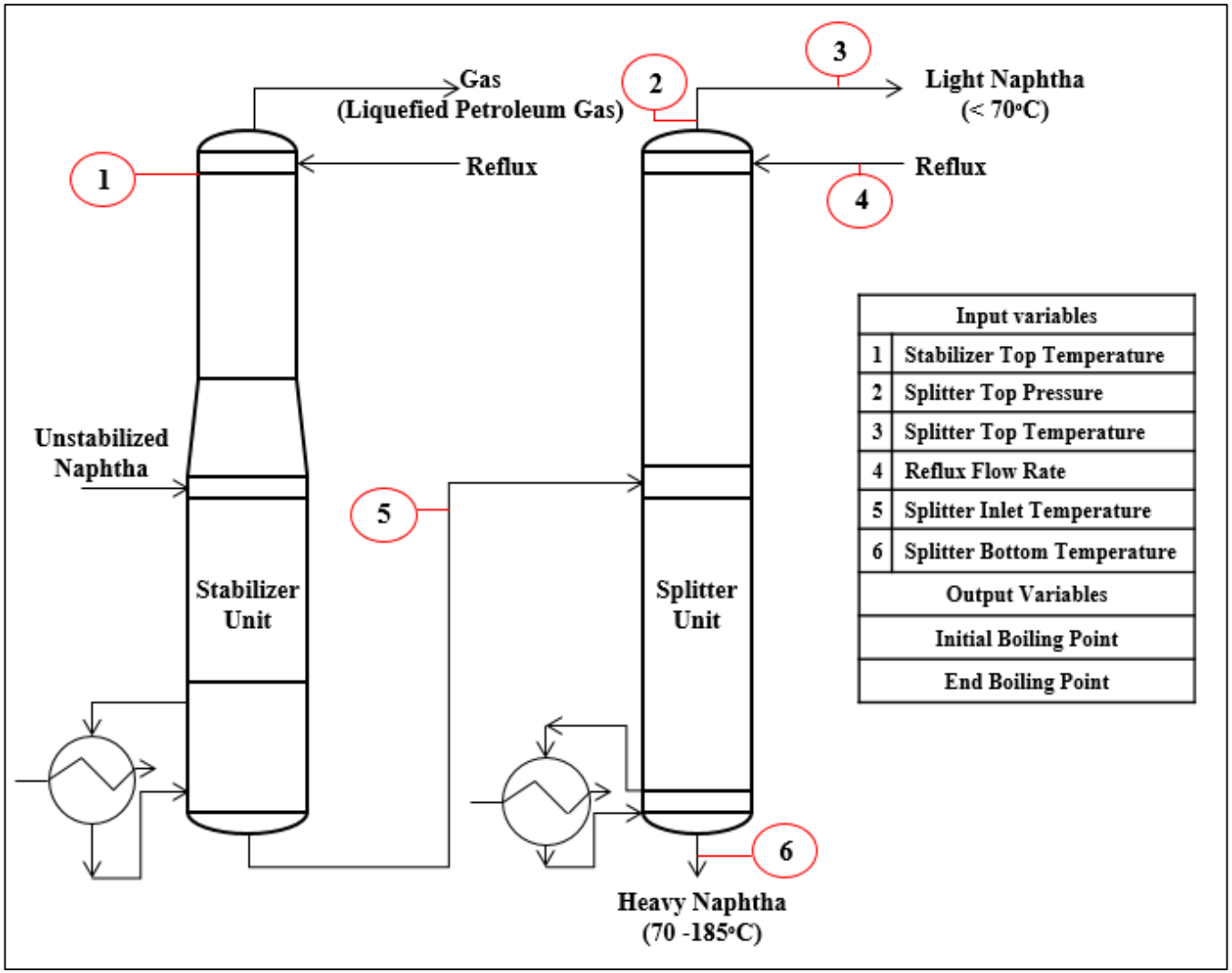


Figure 4.1: Process flow diagram - Naphtha splitter section

The naphtha splitter unit presented in Figure 4.1, forms a small portion of mid-stream crude fractionation process, which consists of atmospheric distillation column and stripping unit with a preheater at the bottom section and condensing system at the top overhead section. The un-stabilized naphtha (C_3-C_9) is subjected to a stabilizer unit, separating liquefied petroleum gas (C_3, C_4) in the top and stabilized naphtha at the bottom. The stabilized naphtha (C_4-C_9) is then fed to the stripping unit which then further separates light naphtha (C_5 < 85^0C) and heavy naphtha (C_5 > 85^0-190^0C). The temperature reading measured for the condensate first drop at the outlet of the condenser is initial boiling point and the upper temperature limit observed during the test is found as end boiling point.

It is noted that the end boiling point of the splitter bottom fraction should not exceed 204^0C because the rate of deactivation of platinum catalyst increases while processing through catalytic reforming units. Also, the initial boiling point of splitter boiling point of splitter bottom fraction should maintain between 75^0-100^0C, such that it prevents the formation of precursors for undesirable benzene above this range in catalytic reforming units (Ujević et al. 2011). Therefore, real time estimation of these parameters will be of great assistance in maintenance of naphtha quality. The variables of interest in this work are initial boiling point and end boiling point of naphtha.

The model output can be described as:

IBP: $$\boldsymbol{y_1(k) = f\,[x_1(k), x_2(k), x_3(k), x_4(k), x_5(k), x_6(k)]} \quad \textbf{(4.1)}$$

EBP: $$\boldsymbol{y_2(k) = g\,[x_1(k), x_2(k), x_3(k), x_4(k), x_5(k), x_6(k)]} \quad \textbf{(4.2)}$$

The input and output variables of naphtha splitter dataset used for model development is mentioned in Table 4.1:

Table 4.1: Description of input and output variables taken for model development in naphtha splitter unit

Process	Naphtha splitter unit
Inputs	x_1, Column top temperature (^{0}C) x_2, Splitter top pressure (kg/cm^2) x_3, Splitter top temperature (^{0}C) x_4, Reflux rate (m^3/h) x_5, Splitter inlet temperature (^{0}C) x_6, Splitter bottom temperature (^{0}C)
Outputs	Initial (y_1) & End (y_2) boiling point of heavy naphtha in splitter bottom (^{0}C)

Table 4.2: Literature review for soft sensing of naphtha fuel properties

Author (s)	Year	Output	Methods used
Dam & Saraf	2006	IBP and EBP of heavy naphtha	Genetic Algorithm-Artificial Neural Networks
Angelov & Zhou	2007	Naphtha 95% cut point	Extended evolving Takagi-Sugeno fuzzy model
Yan	2008	Naphtha 25% cut point	Modified nonlinear generalized ridge regression
Yan	2010	Naphtha dry point	Hybrid artificial neural networks
Ujević et al.	2011	IBP and EBP of heavy naphtha	Multiple linear regression, Multilayer perceptron and Radial basis function neural networks
Wang et al.	2013	Naphtha dry point	Backpropagation learning technique combining correlation pruning algorithm with multiple linear regression model
Shang et al.	2015	Naphtha 100% cut point	Dynamic partial least square regression
Torgashov et al.	2018	Desired cut 2 (mixture of naphtha and gasoline)	Static linear regression and dynamic finite impulse response model

Few literatures are available to predict the quality variables in naphtha splitter unit using conventional soft sensors: Estimation of initial and final boiling point of heavy naphtha using genetic algorithm in artificial neural networks (Dam & Saraf 2006). An online estimator based on extended evolving Takagi-Sugeno fuzzy model (Macias-Hernandez et al. 2007) was developed to estimate heavy naphtha 95% cut point. A modified nonlinear generalized ridge regression and differential evolution (Yan, 2008) based online virtual estimator for measuring naphtha 25% cut point at the top of atmospheric distillation unit. Hybrid artificial neural network (Yan, 2010) based soft sensing system was implemented for real-time prediction of naphtha dry point in preflash tower column of Crude Distillation Unit (CDU). Prediction of initial and final boiling point of heavy naphtha in the CDU splitter column using multiple linear regression, multilayer perceptron neural networks and radial basis function neural networks (Ujević et al. 2011).

Hybrid approach based on backpropagation learning technique combining correlation pruning algorithm with multiple linear regression model (Wang et al. 2013) for estimation of naphtha dry point in preflash tower of CDU. An improved dynamic partial least square regression (Shang et al. 2015a; Shang et al. 2015c) soft sensor for prediction of naphtha 100% cut point in the top product and 95% cut point of heavy diesel in CDU. The final boiling point temperature of desired cut 2 (mixture of naphtha and gasoline) in CDU was estimated using static linear regression and dynamic finite impulse response model (Torgashov et al. 2018) by applying constrained optimization approach.

4.2. Debutanizer column

Refining of petroleum to different cuts (products) is a complex process. The main processes in refinery are atmospheric distillation column, vacuum distillation column,

reforming, cracking, hydro-treating and coking. Refining process considered in this work comprises desulfurization and naphtha splitter plant.

4.2.1 Butane concentration prediction in the column bottom stream

Debutanizer column is a kind of multi-component distillation unit and considers as an integral part of the refinery, used to remove lighter fractions from atmospheric or vacuum distillation, cracking and coking units. In general, it separates liquefied petroleum gas (LPG) as overhead and stabilized naphtha as the bottom product from un-stabilized naphtha. Here, the purpose of debutanizer column is to proper fractionation of incoming feed to distillate and bottom ends, maximization of stabilized gasoline content in overheads, which is used as feed for LPG splitter and minimization of butane content in the bottom which is used as feed for naphtha splitter. The main equipment of debutanizer column is heat exchanger, overhead condenser, bottom reboiler, heat reflux pump, feed pump to the LPG splitter and reflux accumulator. To monitor the product quality, hardware sensors were implemented to measure the secondary variables in the plant. All the secondary variables measured are relevance to describe the product quality of described process. The variables which are highly relevant to debutanizer column, which is explained in Fortuna et al. (2005; 2007) includes: top pressure, reflux flow rate, flow to LPG splitter column, top temperature, 6th tray temperature, bottom temperature 1 and bottom temperature 2.

In this process, the butane content of debutanizer column is not detected on the bottom flow rather, the measurements through gas chromatography was done on the overheads of de-isopentanizer column (iC_5 flow to stock). The concentration of butane in iC_5 depends on the operating condition of debutanizer column. Assumptions is made that the concentration of butane detected in iC_5 flow is same as the butane concentration from bottom of

debutanizer column. Butane concentration is indirectly measured by gas chromatography in the bottom output flow of the debutanizer column. Measuring cycle of gas chromatograph is 15 minutes. The location of analyzer is far away such that the time delay of the range 30-75 minutes is needed to obtain the real time concentration values of butane. Real time estimation of butane concentration in C_5 is required for better control of debutanizer column. It is noted that input and output variables for the debutanizer column available in Fortuna et al. (2005) only to predict the butane concentration in the bottom flow.

To achieve better performance of the process, the main objective is to minimize the butane content (C_4) in the column bottom stream. Butane content, determined by either laboratory analysis or online gas chromatograph. The time for laboratory analysis varies from a few minutes to several hours (one per shift). In the process flowchart, the online gas chromatograph is located at the top of the de-isopentanizer column to quantify the butane content. Further, there is a significant time delay in assessing the butane content of the process stream through gas chromatograph. Hence, there is a need for a real time quality estimator in the bottom flow, which predicts the butane content instantaneously in the process stream by overcoming the time delay problem associated with hardware sensors.

Figure 4.2 represents the schematic process flow diagram of the debutanizer column. The data for debutanizer column is taken from Fortuna et al. (2007). The datasets comprise of 2394 samples (1197 as database samples and 1197 as query objects), of which 7 inputs and 1 output is considered for model development. For model development, a total of seven input variables and one output variable were considered (Table 4.2). Here, the bottom temperature 1 and 2 are the same, which are measured at either side of the debutanizer column.

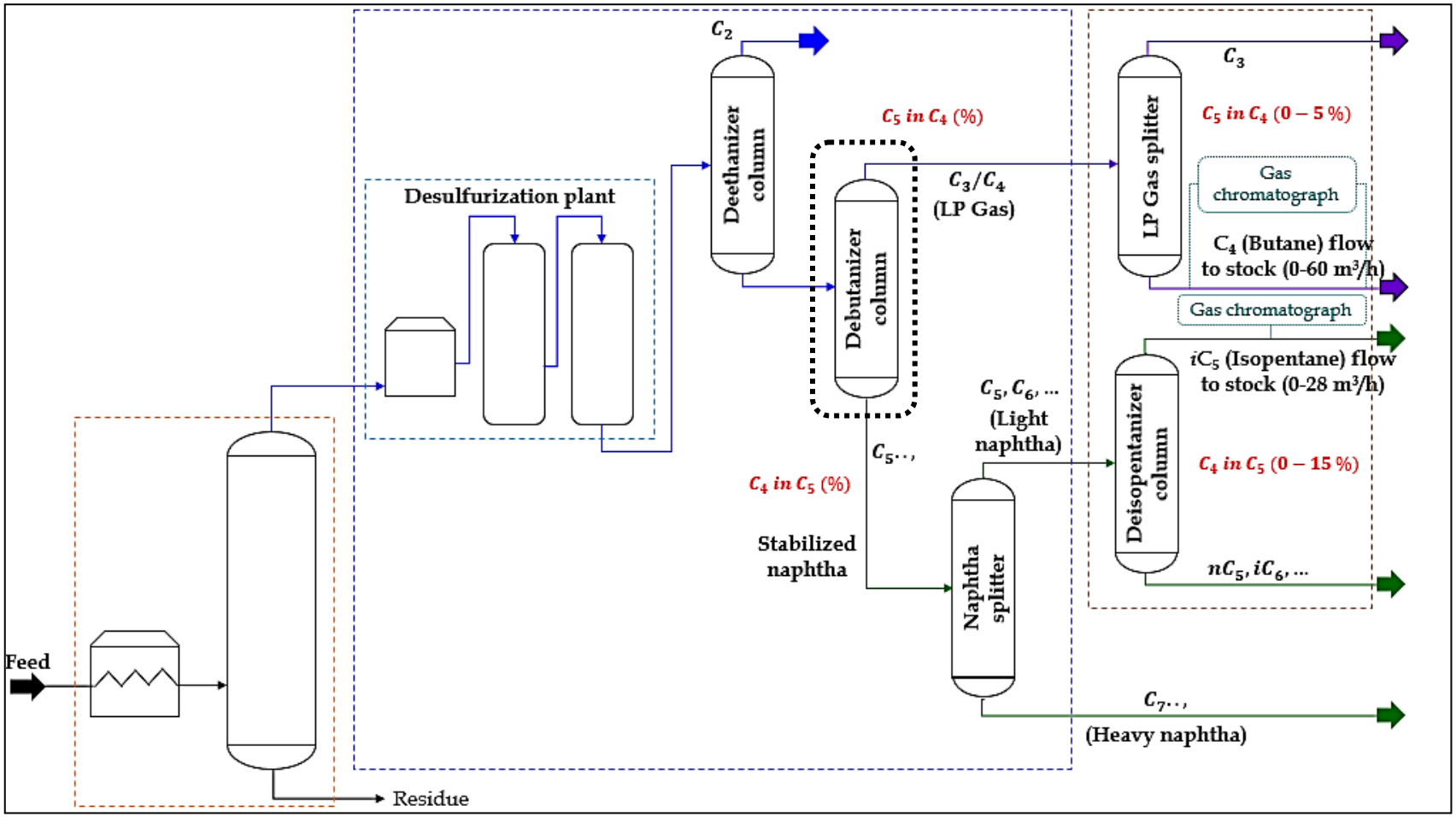


Figure 4.2: Process flow diagram - LPG splitter unit

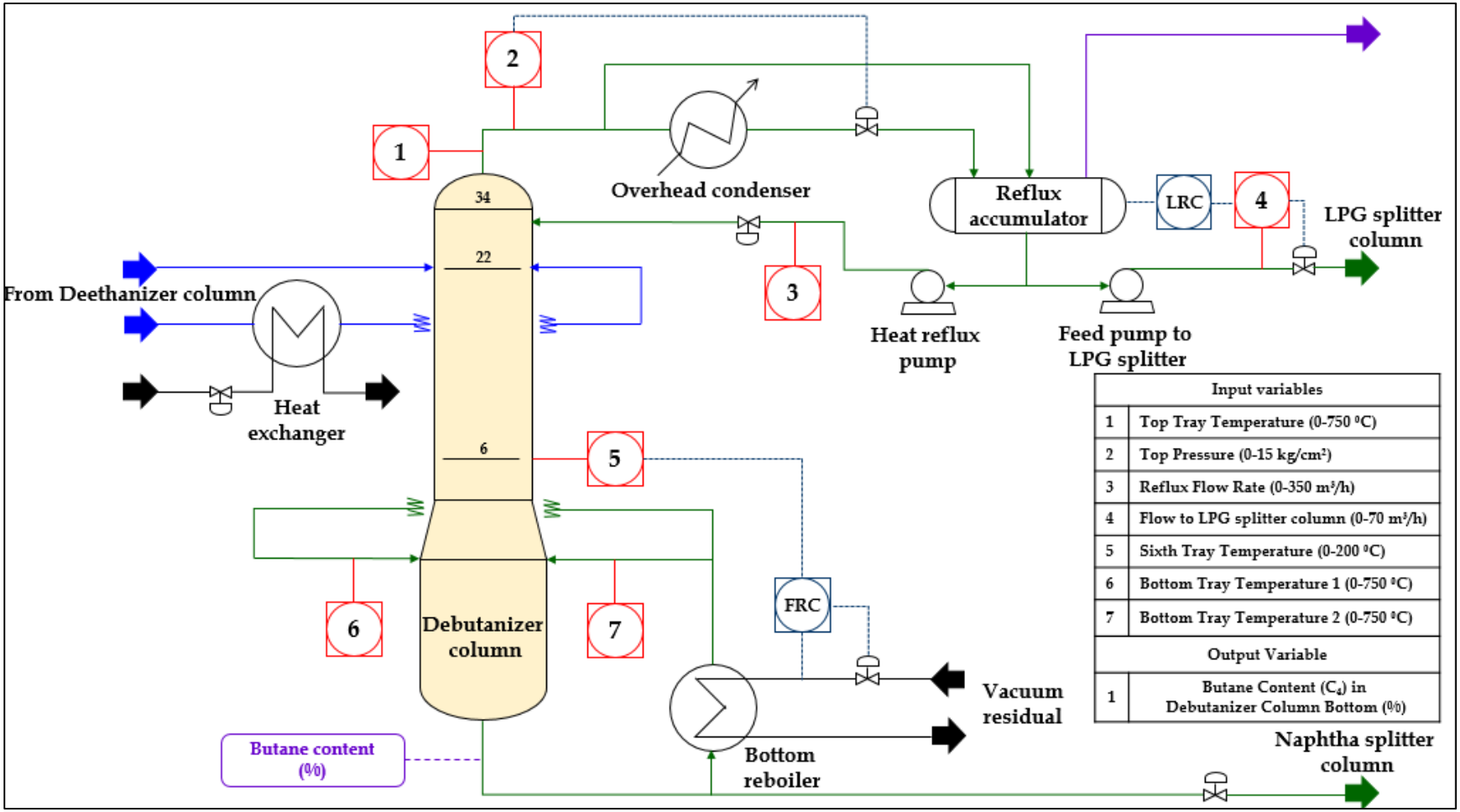


Figure 4.3: Process flow diagram - Debutanizer column

Table 4.3: Description of input and output variables taken for model development in debutanizer column

Process	Debutanizer column
Inputs	x_1, Top tray temperature (^0C) x_2, Top pressure (kg/cm^2) x_3, Reflux flow rate (m^3/h) x_4, Flow to next process (m^3/h) x_5, Sixth tray temperature (^0C) x_6, Bottom tray temperature 1 (^0C) x_7, Bottom tray temperature 2 (^0C)
Outputs	Butane content (C_4) in debutanizer column bottom (in %), y

The model output can be designed as:

Butane: $$\boldsymbol{y(k) = f\,[x_1(k), x_2(k), x_3(k), x_4(k), x_5(k), x_6(k), x_7(k)]} \quad \textbf{(4.3)}$$

To predict the butane content in debutanizer column using soft sensors, few literatures are available which are mentioned in Table 4.4.

Table 4.4: Literature review of soft sensing of butane concentration in debutanizer column

Author	Year	Model
Ge & Song	2011	Semi-supervised Principal Component Regression
Ge	2014	Active Learning Principal Component Regression
Yuan et al.	2014	Locally Weighted Kernel Principal Component Regression
Fan et al.	2014	Gaussian Mixture Model - Just-in-Time Learning
Ge et al.	2014	Mixture Semi-supervised Principal Component Regression
Matias et al.	2015	Online Sequential Extreme Learning Machine based on Recursive Partial Least Squares
Zhu et al.	2015	Mixture Robust Supervised Probabilistic Principal Component Analysis
Pani et al.	2016	Back Propagation Neural Network
Xiong et al.	2017	Local Time-delay Reconstruction based Moving Window Time Difference Gaussian Process Regression
Bidar et al.	2017	State Dependent Parameter Auto-regressive with

		Exogenous Variable Model
Yuan et al.	2017	Probabilistic Just-in-Time Learning
		Deterministic Just-in-Time Learning
Yuan et al.	2017	Locally Weighted Partial Least Squares Regression
		Weighted Gaussian Regression
Shao & Tian	2017	Semi-supervised Selective Ensemble Learning based on Distance to Model
Liu et al.	2018	Locally Weighted - Bayesian Network
		Bayesian Network with Time Difference
Kneale & Brown	2018	Random Forest - Partial Least Squares Regression Ensemble
Yuan et al.	2018	Variable-wise Weighted Stacked Auto-encoder
Siddharth et al.	2019	Adaptive Neuro-Fuzzy Inference System
Singh et al.	2019	Generalized Regression Neural Network
Shi & Xiong	2020	Ensemble Semi-supervised Learning Gaussian Process Regression
Yuan et al.	2020	Hybrid Variable-wise Weighted Stacked Auto-encoder
Urhan & Alakent	2020	Adaptive Moving Window and Just-in-Time learning Ensemble
Alakent	2020	Just-in-Time learning based on Online Weighted Euclidean Distance
Alakent	2020	Moving Window and Just-in-Time learning using transductive inference

4.3. Sulfur recovery unit

Sulfur and its associated compounds are inevitable (inorganic) constituents of petroleum found in earth's crust. Even if it presents in minor quantities, its toxic effluent on the environment and as a catalytic poison in downstream operations is found to be a major concern. Sulfur is removed in the form of elemental sulfur from its constituents as a by-product through desulphurization or gas sweetening process, a typical mid-stream unit.

4.3.1 SO_2 and H_2S concentration prediction in tail gas

Sulfur recovery unit is an integral part of refinery processes. Acid gases comprise two kinds of gases: Mono ethanol amine (MEA) gas and Sour water stripping (SWS) gas. MEA gas rich in H_2S, while SWS gas consists of H_2S and NH_3 coming from the plant. MEA gas

is fed in main combustion chamber and adequate air flow for combustion is supplied (Air_MEA). SWS gas is fed in secondary combustion chamber and the respective air flow is supplied (Air_SWS). Separate chamber is provided for combustion of SWS gas with excess air supply. This will prevent the formation of ammonium salts in the system, which will give rise to the formation of nitrogen and nitrogen oxides.

The following reactions that occur in this SRU process (Claus process):

H_2S combustion reaction in reaction furnace:

$$\boldsymbol{3H_2S + \frac{1}{2}O_2 \rightarrow SO_2 + 2H_2S + H_2O} \quad \textbf{(4.4)}$$

The products from the outlet of reaction furnace is further transformed in the downstream catalytic convertors:

$$\boldsymbol{SO_2 + 2H_2S \rightarrow S_x + 2H_2O} \quad \textbf{(4.5)}$$

Combustion of NH_3 in SWS gas occurs in reaction furnace:

$$\boldsymbol{4NH_3 + 3O_2 \rightarrow 2N_2 + 6H_2O} \quad \textbf{(4.6)}$$

H_2S is converted into pure elemental sulfur when acid gases are burnt in reactors by partial oxidation with air. Here, water vapor and elemental sulphur is formed by reaction in high temperature thermal convertors. Liquid sulfur (70%) is generated by cooling down the gaseous combustion products through water condenser coming from furnace and it is collected in catch basins. The fraction of gas comes out of the condenser is sent to first catalytic reactor and again cooled down through first condenser. The output of condenser is subjected to second catalytic reactor and second condenser to obtain 90% pure sulfur. Unconverted gas which is less than 5%, is fed to the maxi sulfur plant. Residual H_2S and SO_2 is coming out as tail gas, final gas stream from SRU.

Table 4.5: Description of input and output variables taken for model development in sulfur recovery unit

Process	Sulfur recovery unit
Inputs	x_1, MEA gas flow (m^3/h) x_2, Primary air flow (m^3/h) x_3 Secondary air flow (m^3/h) x_4, Gas flow in SWS zone (m^3/h) x_5, Air flow in SWS zone (m^3/h)
Outputs	H_2S (y_1) & SO_2 (y_2) composition in tail gas ($moles/m^3$)

In SRU, the typical process comprises acid gas and tail gas. Acid gas is the combination of two gasses from different process streams which are mono ethanol amine (MEA) gas from gas washing plants and SWS gas from sour water stripping (SWS) plant. MEA gas rich in hydrogen sulfide (H_2S) while, SWS gas rich in hydrogen sulfide (H_2S) and ammonia (NH_3). H_2S is converted into pure sulfur by burning acid gas along with air in the furnace (thermal convertor) through partial oxidation reaction. The liquid sulfur is generated by cooling down the exit gaseous products. Further, the liquid sulfur is subjected to high temperature catalytic converters loaded with aluminum or titanium as catalyst, to separate water vapor from sulfur. Finally, the unconverted acid gas is sent to the Maxi-sulfur plant for further conversion. The exit final gas stream from the Maxi-sulfur unit contains residual H_2S and SO_2, which is known as tail gas. Figure 4.3 represents the typical flow sheet of the SRU process.

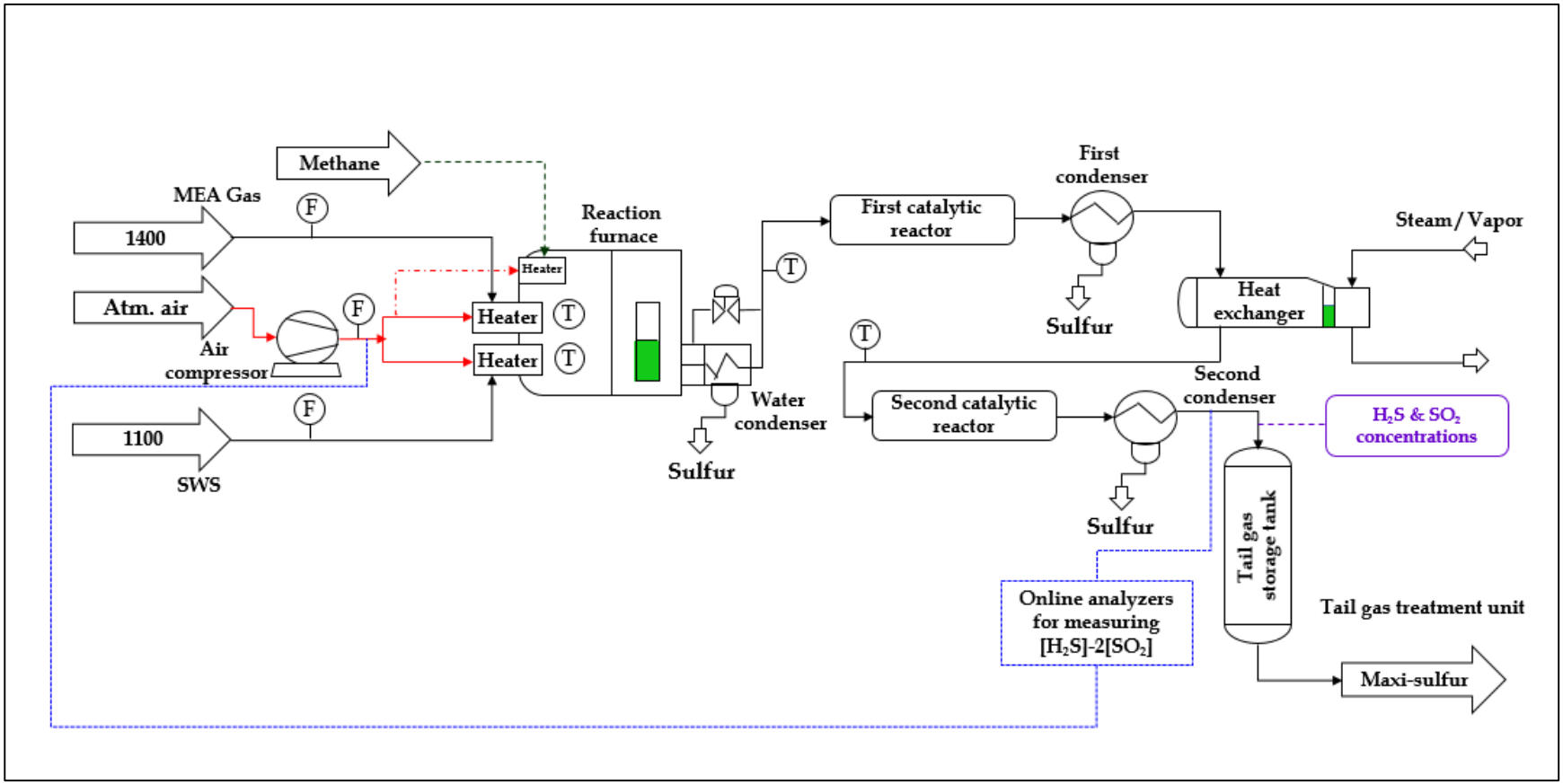


Figure 4.4: Process flow diagram - Sulfur recovery unit

If the input SWS gas flow to the secondary combustion chamber is low, MEA gas (MEA_Spilling) was added along with SWS gas to keep the flow constant. An adequate air flow is supplied for proper combustion. To ensure correct stoichiometric ratio in the tail gas composition, air flows combustion chambers are manually controlled by plant operators. In closed loop operation, control of air flow (Air_MEA_2) is taken based on the analysis of tail gas composition. However, when the analyzer is under maintenance purposes, the control loop becomes functionless which leads to decrease in performance of SRU. The quantity of air supply is the most critical parameter for the conversion of H_2S, which decides the tail gas composition. The conversion of SO_2 increases when excess air flow is supplied to the reaction chamber, in contrast low air flow tends to support opposite reaction.

To improve the sulfur recovery process by monitoring the performance of extraction process ($[H_2S]$-$2[SO_2]$) and controlling the air to feed ratio in the combustion chamber, an online analyzer is adopted to measure the concentration of H_2S and SO_2 in the tail gas of each sulfur line. Absence of pollutants in the tail gas or correct stoichiometric ratio of reactants in reaction (2) indicates that the $[H_2S]$-$2[SO_2]$ difference in tail gas is zero. This optimum condition indicates that sulfur compounds in the catalytic convertors should be removed completely. Moreover, hardware sensors are frequently taken off for maintenance due to the damage caused by the H_2S and SO_2 gases.

The entire dataset comprises 10081 inputs and output values. Five inputs and two output variables were considered for soft sensor development, which is mentioned in Table 4.3. Here, 5040 observations were taken as database samples, and remaining 5041 as query data samples.

The model output can be designed as:

$$H_2S: \quad y_1(k) = f[x_1(k), x_2(k), x_3(k), x_4(k), x_5(k)] \quad (4.7)$$

$$SO_2: \quad y_2(k) = f[x_1(k), x_2(k), x_3(k), x_4(k), x_5(k)] \quad (4.8)$$

Several literatures which are focused on estimation of tail gas composition in SRU are:

Table 4.6: Literature review of soft sensing of tail gas composition in SRU

Author	Year	Model
Fortuna et al.	2003	Non-linear Least Square Fitting
Ge & Song	2010	Relevant Vector Machine
Ge & Song	2011	Semi-supervised Principal component regression
Shao et al.	2015	Supervised Local and Non-local Structure Preserving Projections Locally Weighted Regression Just-in-Time learning
Shao & Tian	2015	Selective Ensemble of Local Partial Least Squares
Shang et al.	2015	Correlation based Slow Feature Regression
Graziani & Xibilia	2017	Multilayer Perceptron
		Deep Boltzmann Machine
Jain et al.	2017	Principal Component Regression
		Support Vector Regression
		Least Squares Support Vector Regression
Shao & Tian	2017	Semi-supervised Selective Ensemble Learning based on Distance to Model
Xiong et al.	2017	Local Time-delay Reconstruction based Moving Window Time Difference Gaussian Process Regression
Morey et al.	2018	Gaussian Process Regression with Marginal Log likelihood Maximization
		Adaptive Neuro Fuzzy Inference System
Shi & Xiong	2018	Active Learning Gaussian Process Regression
Kneale & Brown	2018	Random Forest - Partial Least Squares Regression Ensemble
Moghadam et al.	2018	Time Varying Parameter - Dynamic Auto-regressive with Exogenous Variable model
Zheng & Funatsu	2018	Partial Constrained Least Squares
Singh et al.	2019	Generalized Regression Neural Network
Yuan et al.	2020	Stacked Isomorphic Auto-encoder
Yuan et al.	2020	Stacked Enhanced Auto-encoder
Urhan & Alakent	2020	Adaptive Moving Window and Just-in-Time learning Ensemble
Alakent	2020	Just-in-Time learning based on Online Weighted Euclidean Distance
Alakent	2020	Moving Window and Just-in-Time learning using transductive inference
Patane & Xibilia	2021	Echo-state Networks – Intrinsic Plasticity

Chapter - 5

Results and Discussion

In this chapter, the details of five datasets used for adaptive soft sensor development are described. Then, the effect of various hyper parameters on model performance for the industrial processes are discussed in sub-sections. Finally, performance results of developed model using statistical performance index along with model computation time are mentioned in tabular form and the best performing model for the respective industrial cases are further validated using 4-plot analysis.

5.1 Details of datasets for adaptive soft sensor development

In the naphtha splitting unit, six input variables are considered for prediction of IBP and EBP. The input-output data are collected from Ujević et al. (2011). Input-output data for debutanizer column and sulphur recovery unit are available online (Fortuna, L., Graziani, S., Rizzo, A., & Xibilia, M. G. (2007). Springer Science & Business Media - Accessible online at https://storage.googleapis.com/sgw-extras/zip/2007/978-1-84628-480-9.zip). Datasets for each process is split into two sets: database and query dataset. The number of samples of each dataset and the dimension are presented in Table 5.1.

5.2 Naphtha splitter section: Prediction of Initial and end boiling point of heavy naphtha

In this section, the effect of RDS for linear models, effect of spread parameter for JITL-GRNN and the effect of loss function for JITL-SVR models are discussed for prediction of IBP and EBP of heavy naphtha.

Table 5.1: Details of five datasets used for adaptive model development

Process	Output variable	Input data	Output data	Total data	Database data	Query data
Naphtha splitting unit	IBP	210 × 6	210 × 1	210 × 7	151 × 7	59 × 6
	EBP	209 × 6	209 × 1	209 × 7	151 × 7	58 × 6
Debutanizer column	Butane content	2394 × 7	2394 × 1	2394 × 8	1197 × 8	1197 × 7
Sulfur recovery unit	H_2S	10081 × 5	10081 × 1	10081 × 6	5041 × 6	5040 × 5
	SO_2	10081 × 5	10081 × 1	10081 × 6	5041 × 6	5040 × 5

5.2.1 Effect of relevant dataset for linear models

An open issue in JITL based approach is to determine the optimum size of relevant dataset. In order to determine the optimum size, local linear models (MLR, LWR, PLS) were developed based on different number of relevant dataset size ranging from 10 to 150 and the prediction results for 59 objects (58 for EBP) of query data were determined. The dependency of model prediction accuracy on the relevant dataset size is shown in Figure 5.1.

In Figures 5.1 (a) & (b), PLS (3) and PLS (4) corresponds to the PLS local models with three and four latent variables respectively (It may be recalled that the number of actual input variables is six). A common perception is: the more the size of the data used for modeling, the better will be the model performance. This may be true in steady state soft sensor design where one model developed from a set of offline data is subsequently used for predicting outputs for all unknown inputs. However, this concept may not be always true in case of adaptive soft sensor design using local modeling concept where a model is built to predict output for just one set of input. A deep in performance around RDS 30 in Figure 5.1 (b) indicates existence of some optimum RDS size.

Both Figures 5.1 (a) & 5.1 (b) shows that there is negligible improvement in prediction accuracy beyond a dataset size of 50. Therefore, for linear models, the optimum relevant dataset size was decided to be 50 (k = 50; Figure 5.1). Further, for any dataset size, performance of JITL-LWR is found to be better than that of JITL-MLR, JITL-PLS (3) and JITL-PLS (4). Also, performance values for IBP prediction are better than that for EBP prediction.

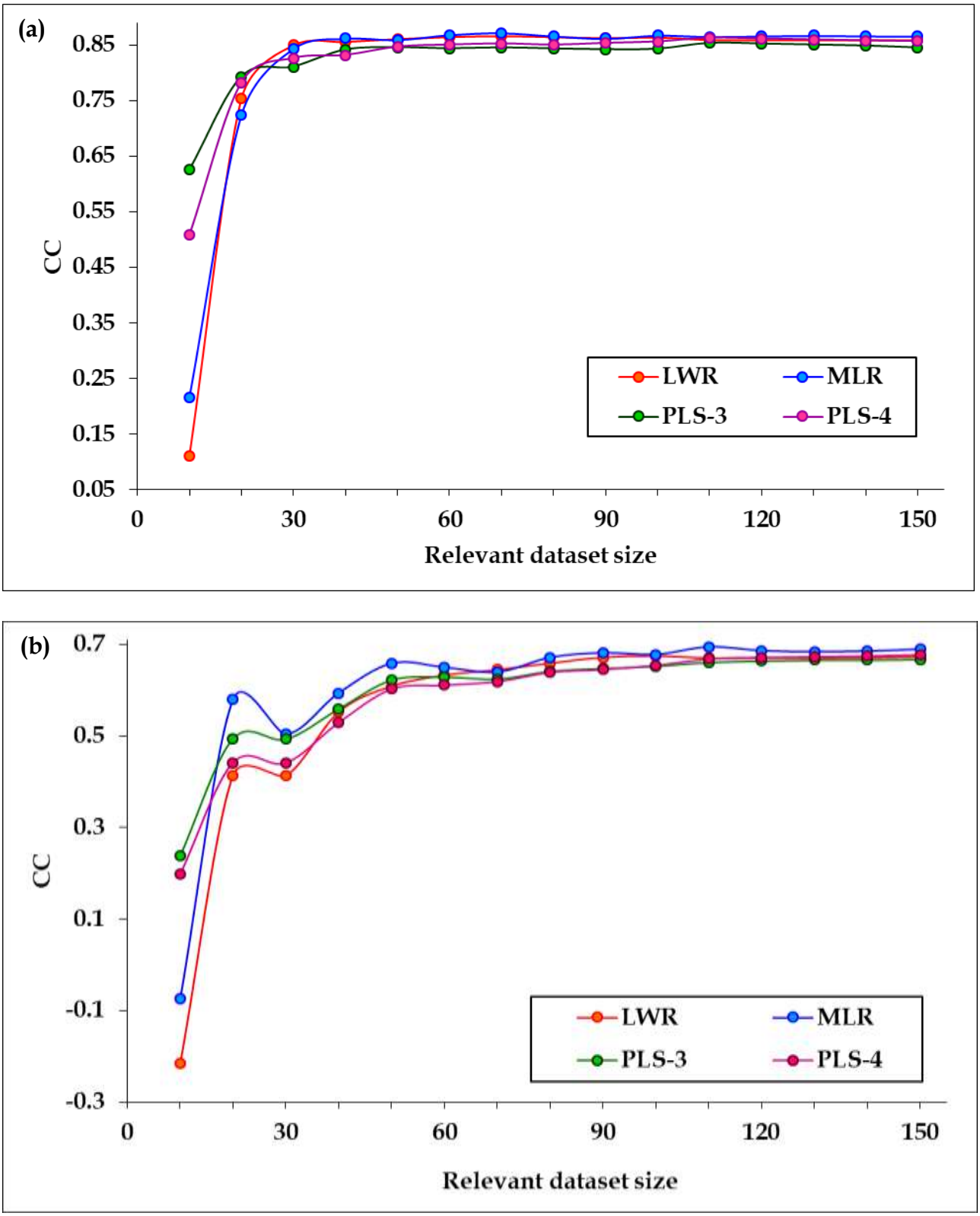


Figure 5.1: Effect of relevant dataset on prediction accuracy using linear local models (a) Initial boiling point; (b) End boiling point

In addition to the linear local models (MLR, LWR and PLS), three non-linear local model was also developed: generalized regression neural network (GRNN), support vector regression with sequential minimal optimization (SMO) and support vector regression with iterative single data algorithm optimization (ISDA) model. These adaptive soft sensors are mentioned as: JITL-GRNN, JITL-SVR: SMO and JITL-SVR: ISDA. It may be noted that though, the results for non-linear local model based soft sensor are only presented for RDS size of 50, models with different RDS size were developed and tested. Usually, model accuracy improves when the dataset size used for modeling increases as shown in Figure 5.1.

In this work, the local models are developed from the relevant dataset. Performance of linear local models in fact improves as the relevant dataset size increases and remained constant after some optimum size of 50 (as shown in Figure 5.1). Irrespective of linear or non-linear local model, it was observed that at low size of RDS, the accuracy improves with increase in size of RDS. However, at higher values of RDS size, the accuracy shows negligible improvement with increase in RDS size. Low RDS size results in under fitting (low accuracy) and higher RDS size leads to increase in computation load. For real applications, it will be difficult to select different relevant dataset size and/or constituent model parametric values along with computation complications. For better model generalization, an optimum relevant dataset size of 50 is taken at which all models have maximum or close to maximum accuracy.

5.2.2 Effect of RDS and spread parameter (σ) on performance of JITL-GRNN

In GRNN, the input – output relation is expressed as a probability density function determined from the relevant data set. Industrial modeling applications using GRNN can be found in Pani & Mohanta (2016).

A crucial issue in GRNN modeling is the optimum value of spread parameter to be used. Spread parameter is typically a variance or smoothing function helps to generalize the neural network architecture. The larger value of spread value (σ) tends to force estimated density population to follow smooth gaussian fit and capable of generalizing the system. Conversely, the smaller value of σ corresponds to less generalization and non-gaussian fit.

The optimum value was determined by developing GRNN models with different spread values at the optimum relevant dataset size of 50 and testing the prediction performance on the query data values. The effect of spread parameter on prediction accuracy is presented in Figures 5.2 (a) & 5.2 (b). Figures 5.2 (a) and 5.2 (b) shows some interesting results for GRNN. It can be noticed that for any size of RDS, the accuracy first increases, with increase in spread value, reaches a maximum at a spread value of 1 (minimum MAE and maximum CC) and then decreases. This trend resulted in determining the optimum spread value as 1.

Another notable observation is that the accuracy deteriorates as the relevant dataset size increases. This is contrary to the phenomenon observed in linear models where accuracy increases with increase in dataset size and after a certain size remains almost constant. For highest accuracy of GRNN model, the relevant dataset size was taken as 10.

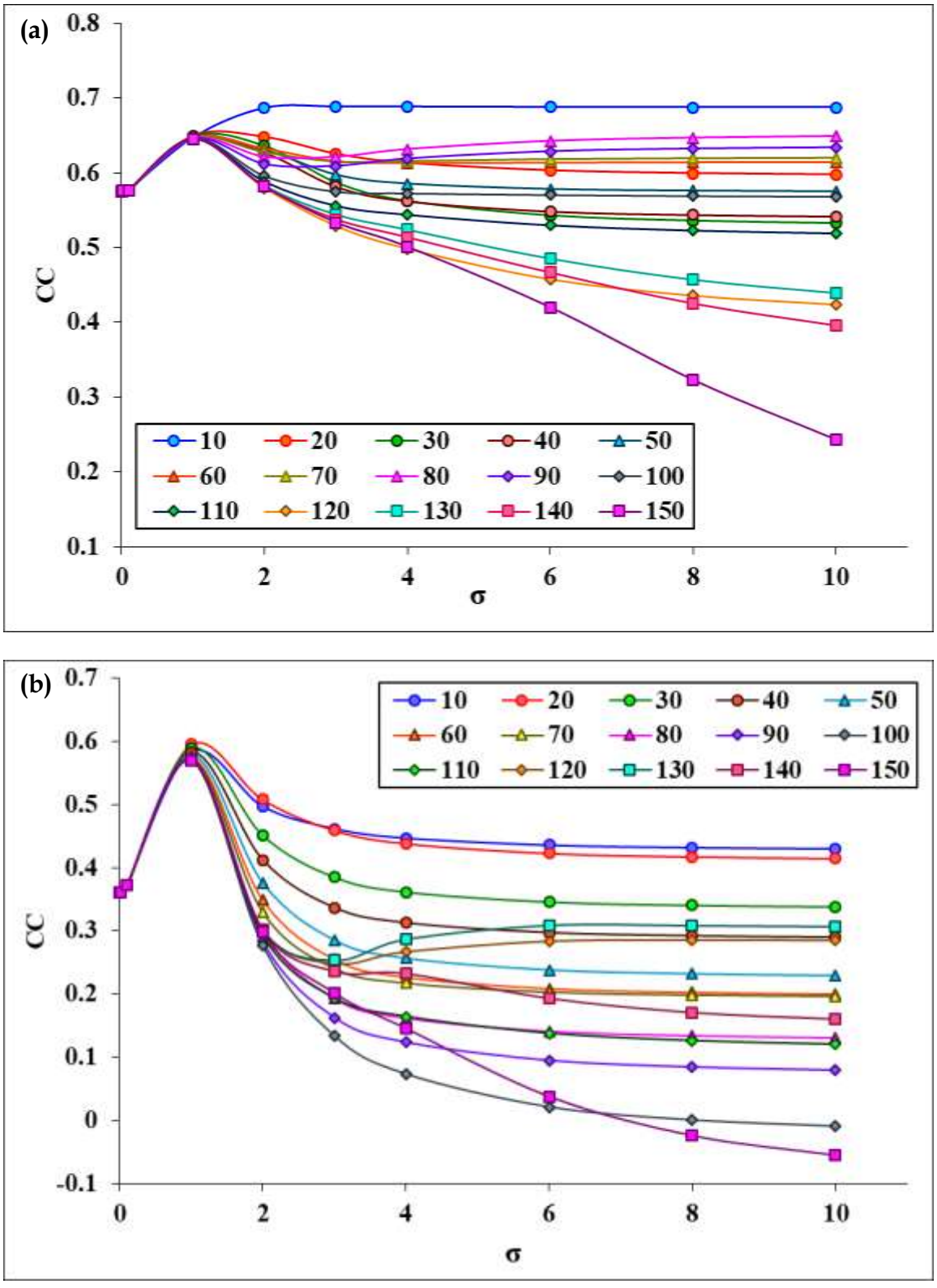


Figure 5.2: Prediction accuracy (CC) as a function of various relevant dataset size and spread value by JITL-GRNN (a) IBP; (b) EBP

5.2.3 Effect of RDS and loss function (ε) on performance of JITL-SVR

Selection of optimum value of loss function (ε) for modeling applications is very important to select the best performing models with reasonable prediction (Shokri et al., 2015; Yan et al., 2004). There is no unanimous formula for determination of hyper parameter values (loss function in this case). There is a range of methods starting from grid search to different evolutionary techniques (such as genetic algorithm, ant colony optimization, particle swarm optimization etc.) used by researchers to determine the hyper parameter values resulting in best prediction accuracy. One such interesting work was observed in Shokri et al. (2014), who proposed a soft optimization technique based on hybrid metaheuristic approach, which preset the hyper-parameter settings in advance before model development.

Grid search method is rigorous and cumbersome but yields reasonably better optimum values like other proposed techniques in literature because the model is tested over a wide possible range of values (Pani & Mohanta, 2014). Therefore, in this work the grid search method is adopted to determine the optimum loss function value. For both JITL-SVR: SMO and JITL-SVR: ISDA, the prediction performance was tested for ε value ranging from 0.001 to 1.5 for both IBP and EBP. The precise results are presented in Figure 5.3. Other results showing effects of ε and relevant dataset size on mean absolute error are provided subsequently. At 50 RDS, the mean error reached minimum value and started to climb higher by increasing the epsilon (ε) value. The CC value starts decreasing while increasing the loss function value after 0.05 for IBP prediction. For EBP prediction, model performances start to degrade beyond an ε value of 0.9. Therefore, the optimum ε value was found to be 0.05 for IBP and 0.9 for EBP in JITL (Refer Figure 5.3). Further, the best result

of JITL-SVR model was found to be better than other JITL (MLR, LWR, PLS and GRNN) models.

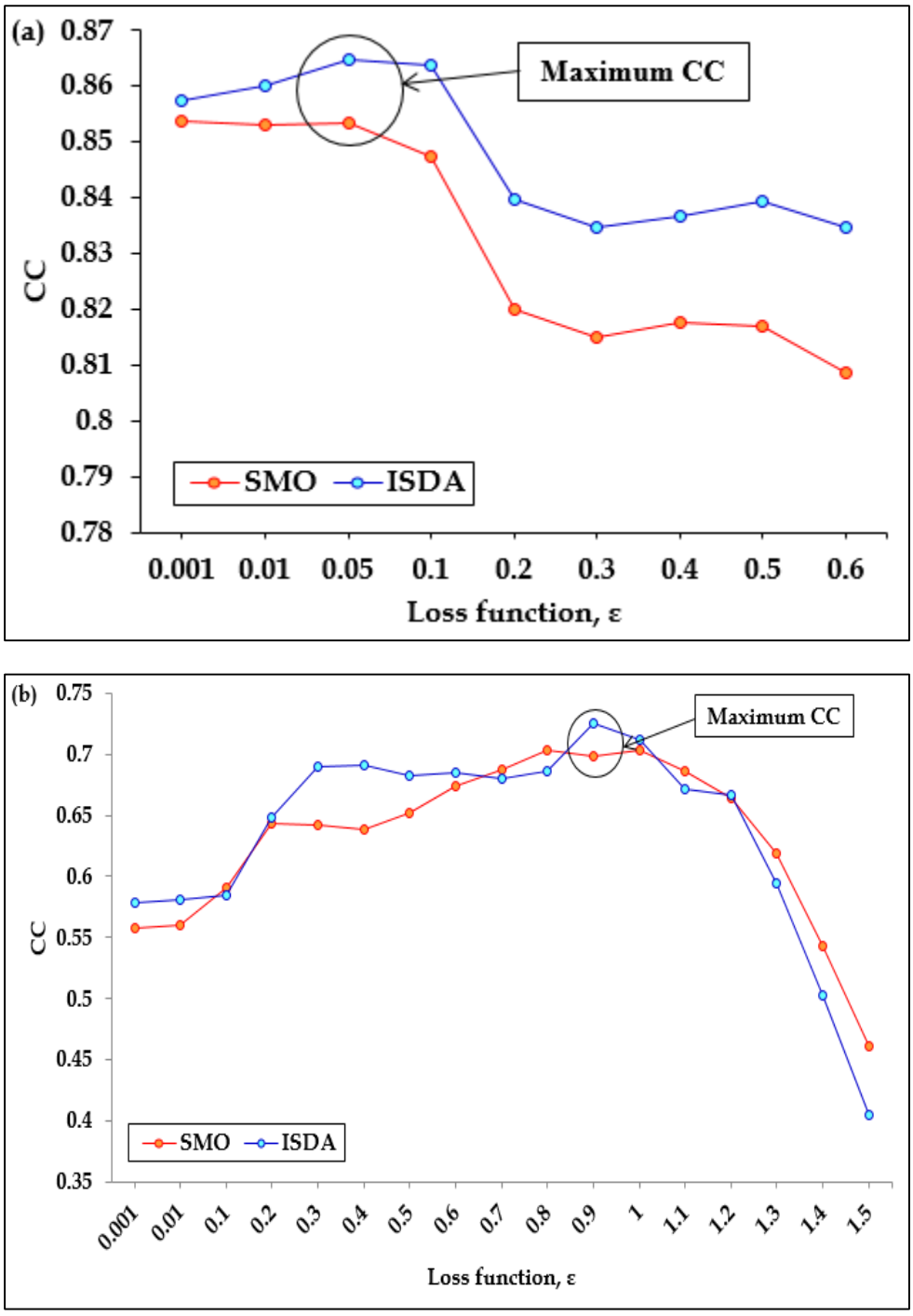


Figure 5.3: CC vs loss function (ε) at RDS-50 (Just-in-Time Learning support vector regression) for (a) IBP; (b) EBP

5.3 Debutanizer column

In adaptive just-in-time learning model, two approaches are followed to compute the similarity index: (1) Euclidean distance (2) Combined angle and Euclidean distance for debutanizer column. In case of combined angle and distance method, the weighting factor λ is a user specified parameter. Low value of λ gives more importance to angle only similarity index. On the other hand, high λ value indicates more importance to distance only method. For use of angle and distance method, a value of 0.5 was used for the weighting parameter which attributes equal weightage to both distance and angle.

5.3.1 Effect of RDS and spread parameter on the performance of JITL-GRNN

Figures 5.4 (a) & (b) represent the model accuracy variation of JITL-GRNN model for the two similarity index computation approaches. Each figure is a plot of model accuracy (CC) versus GRNN spread parameter σ for different sizes of RDS. It may be noted that, in Figure 5.4 (a), distance based similarity criterion is used and in Figure 5.4 (b) combined angle & distance similarity index is used.

To search for optimum kernel width of GRNN model, we implemented a grid search approach using fine grid mesh (Trial and Error approach). The optimum kernel width is decided as the one which results in either lowest RMSE value and/or highest CC value. The ranges of spread values for search operation are mentioned earlier. It can be observed in Figure 5.4 (a) (distance based similarity criterion) and Figure 5.4 (b) (Angle & Distance based similarity criterion), that the accuracy parameter CC reaches a maximum around σ = 0.01 irrespective of the relevant dataset size. Further, the performance of JITL-GRNN model at RDS 10 achieves better performance than other RDS values.

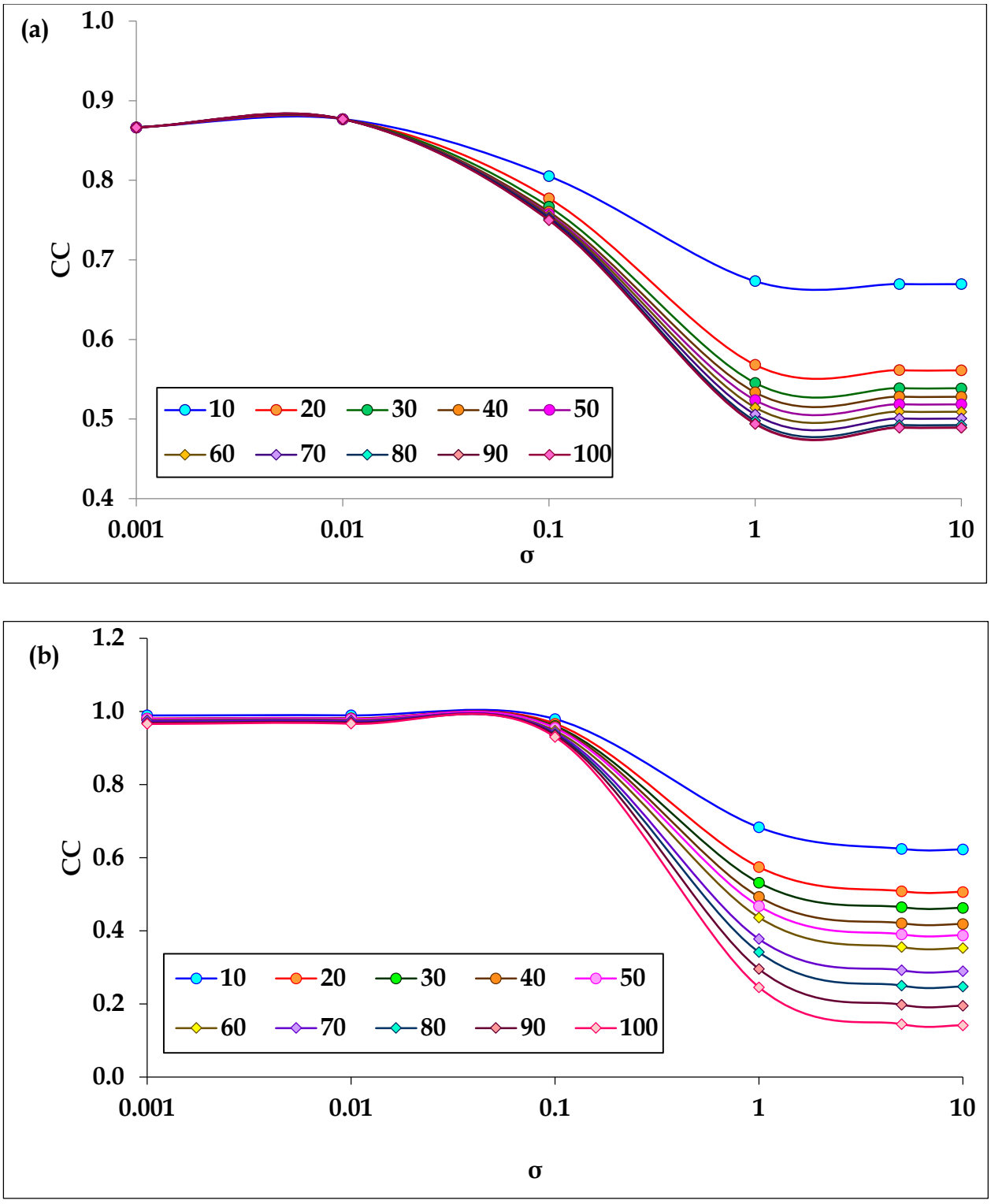


Figure 5.4: Model prediction accuracy of JITL-GRNN for two similarity indexes in debutanizer column (a) distance based (b) angle & distance based

One may wonder why the performance decreases when we are taking more values for modeling. This may be due to the reason that as we increase the size of RDS used for local model development, we are including more samples with decreasing relevance with the query sample. Therefore, based on these observations, the optimum value of σ and RDS are chosen as 0.01 and 10 respectively for debutanizer column.

5.4 Sulphur recovery unit

Two approaches (Distance and Combined angle & distance) are followed to compute the similarity index in adaptive just-in-time learning model for SRU. In combined angle and distance method, the value of $\lambda = 0.5$ was taken to accommodate equal weightage to both distance and angle.

5.4.1 Effect of RDS and spread parameter on the performance of JITL-GRNN

A plot of model accuracy (CC) versus GRNN spread parameter (σ) for different sizes of RDS is illustrated in Figure 5.5 & Figure 5.6. It may be noted that, in Figure 5.5 (a) & Figure 5.6 (a), distance based similarity criterion is used and in Figure 5.5 (b) & Figure 5.6 (b) combined angle & distance similarity index is used. Grid search approach was implemented using fine grid mesh to search for optimum kernel width.

The optimum kernel width is decided as the one which results in either lowest RMSE value and/or highest CC value. The ranges of spread values for search operation are mentioned earlier. It can be observed in Figure 5.5 (a) & Figure 5.6 (a) (distance based similarity criterion) and Figure 5.5 (b) & Figure 5.6 (b) (Angle & Distance based similarity criterion), that the accuracy parameter CC reaches a maximum around $\sigma = 0.01$ irrespective of the relevant dataset size. Further, the performance of JITL-GRNN model at RDS 10 achieves better performance than other RDS values.

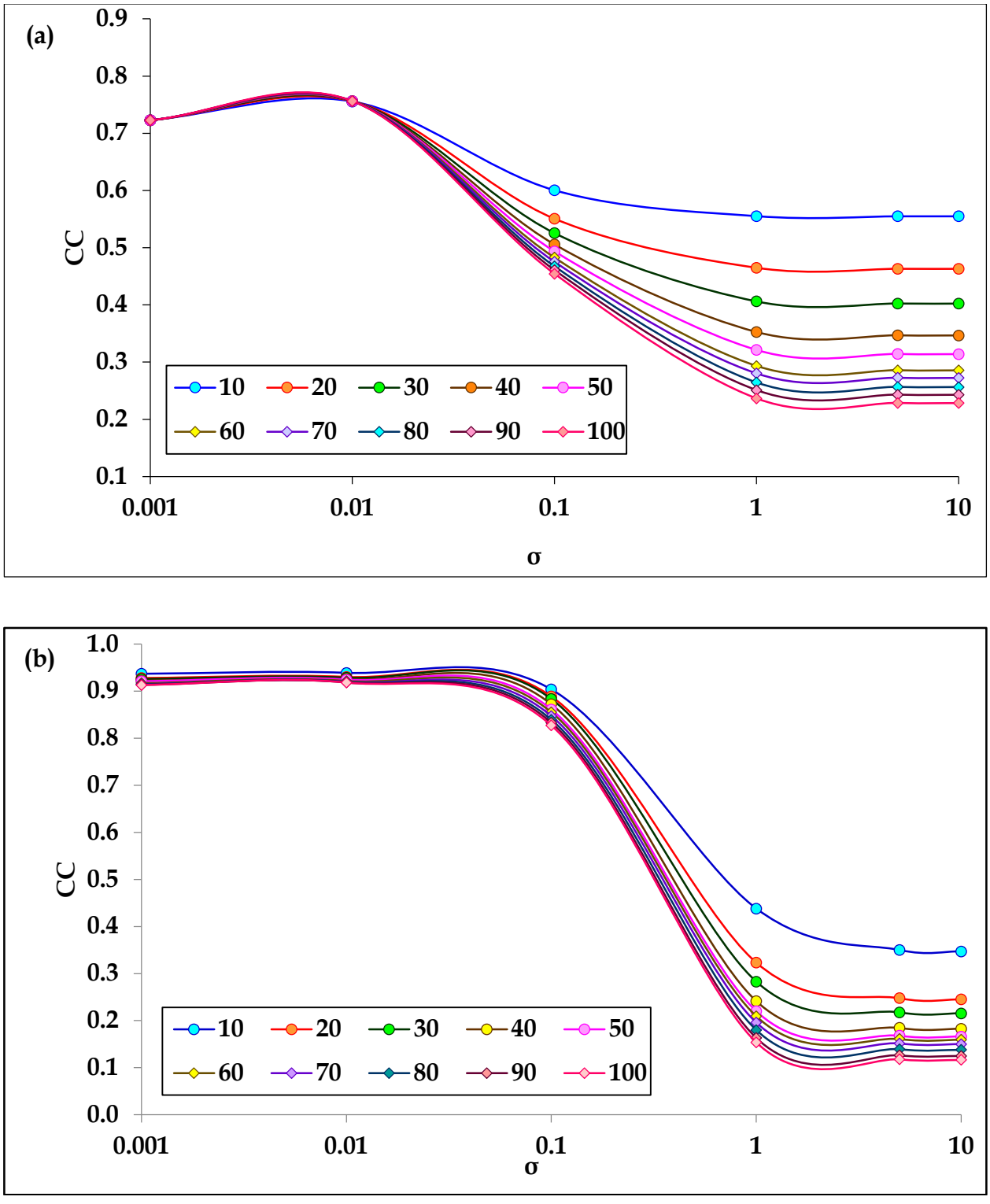
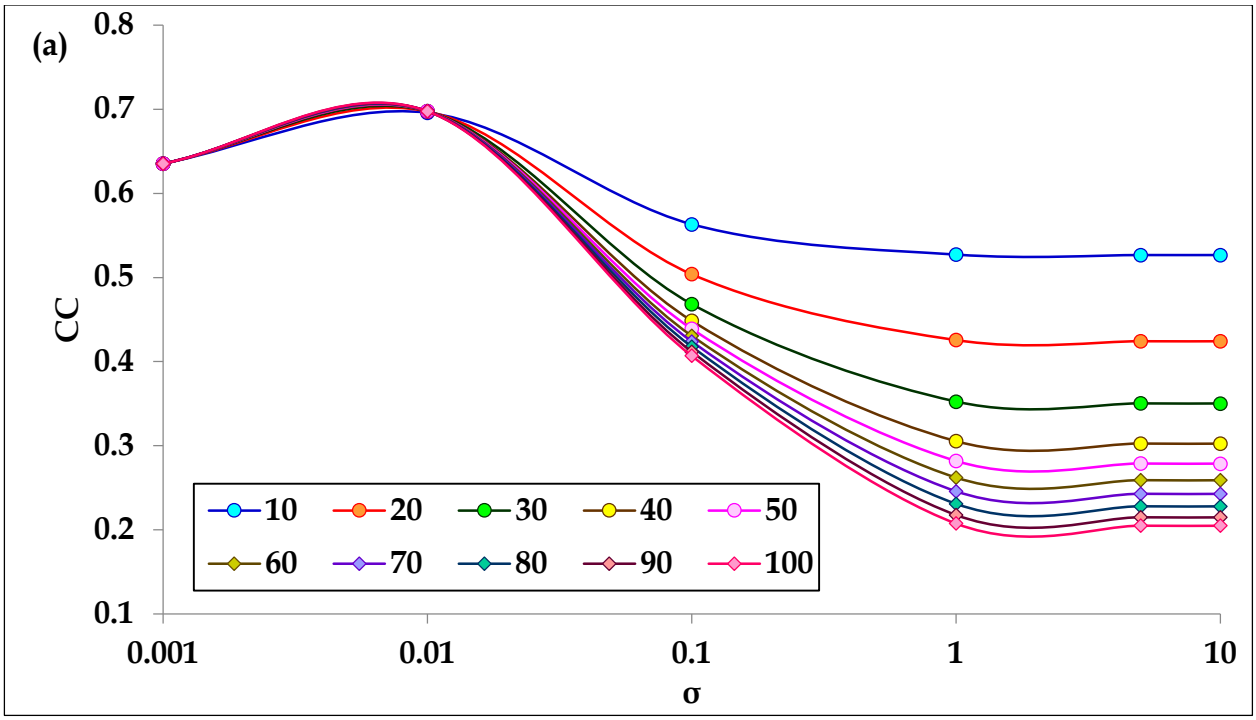


Figure 5.5: Model prediction accuracy of H_2S prediction using JITL-GRNN for two similarity indexes (a) distance based (b) angle & distance based



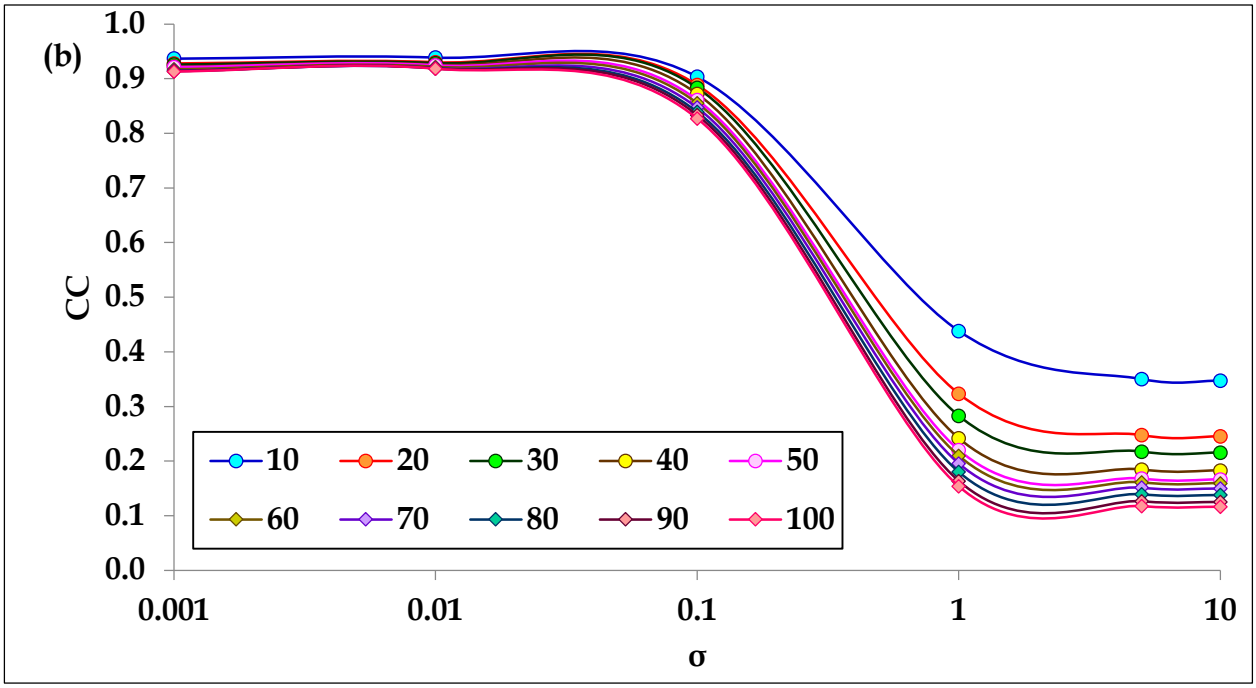


Figure 5.6: Model prediction accuracy of SO_2 prediction using JITL-GRNN for two similarity indexes (a) distance based (b) angle & distance based

Increasing the size of RDS contributes to detrimental effect in model performance. Therefore, based on these observations, the optimum value of σ and RDS are chosen as 0.01 and 10 respectively for both tail gas compositions of SRU.

5.5 Analysis of model accuracy and determination of model computation time

In this section, the performances of various developed adaptive models for the three industrial case studies are analyzed using three statistical performance index and model computation time. Best models for the respective industrial case study is further validated using 4-plot analysis. Comparison of developed models with existing models in literature are mentioned in Table 5.7 & Table 5.8. The program for recursive, sliding window, just-in-time (JIT) architecture and simulation of all the models were developed on MATLAB® software (R2018b).

Statistical performance index and model computation time

The predictive performance of adaptive models for three industrial processes were evaluated by computation of three statistical parameters: correlation coefficient (CC), root mean square error (RMSE) and mean absolute error (MAE). Correlation coefficient provides information about the linear dependency of actual and predicted value. It ranges between -1 to +1 with negative and positive values indicating negative and positive correlation respectively. Zero (0) signifies no correlation and 1 implies perfect positive correlation between predicted and actual values. $RMSE$ and MAE explain the deviation of predicted output from actual output.

The following formulas are used to compute the three parameters.

$$\boldsymbol{CC = \frac{\sum(y-\bar{y})\,(\hat{y}-\bar{\hat{y}})}{\sqrt{\sum(y-\bar{y})^2\ \sum(\hat{y}-\bar{\hat{y}})^2}}} \quad \textbf{(5.1)}$$

$$\boldsymbol{RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y-\hat{y})^2}{n}}} \quad \textbf{(5.2)}$$

$$\boldsymbol{MAE = \frac{1}{n}\sum_{i=1}^{n}|y-\hat{y}|} \quad \textbf{(5.3)}$$

y - Actual observed value; $\hat{y}$ - Model predicted value; $\bar{y}$ - Observed mean; $\bar{\hat{y}}$ - Predicted mean; n - Total number of observations taken for output estimation.

Significance of model computation time

In static soft sensors, the model parameters are determined during offline model development and the parameters remain same during their online use. However, in adaptive soft sensors, the model parameters are computed for each query data object. Therefore, for a static soft sensor, the model computation time involves the time for simulation of the already developed model when an input data set is supplied to the soft sensor. In adaptive soft sensors (such as recursive and sliding window), computation time includes time required for model parameter computation and model simulation for each incoming input vector. Further, in case of JITL based adaptive soft sensor, the computation time involves computation of similarity of the query sample with each sample of the dataset, preparation of the relevant dataset, development of the local model (model identification) and finally simulation of the developed model with the query sample. Unless properly designed, a model possessing good accuracy may not be useful due to computational complexity. Therefore, an important issue for adaptive soft sensors is the model computation time. The computation time was determined using MATLAB® '*timeit*' function. Here, the computation time for every

model was found in triplicates and the average of those triplicates was used for comparisons to minimize the computational error.

The computer configuration used in this research work is mentioned below:

Specification for software and hardware used in this study are:

Operating system (OS): Windows 10 (64 bit), CPU: 3.40 GHz Intel Core TM-i7-4770 CPU processor, 32.0 Gigabyte (GB) of Random Access Memory (RAM) and 932GB solid-state drive (SSD) storage.

5.5.1 Prediction accuracy for Initial and end boiling point of heavy naphtha

The models for IBP and EBP are developed separately. 59 objects (58 objects for EBP) were extracted from the available 210 objects (209 objects for EBP soft sensor) and used for query data. The remaining 151 objects are used as a database.

Recursive and sliding window model were developed using GRNN model with fixed spread parameter (σ = 0.8027) for both IBP and EBP. For recursive and sliding window approach, spread parameter is decided based on Cherkassky & Ma (2004) method of selection. The results for the respective approaches were mentioned in Table 5.2 & Table 5.3. The variations in computation time are between 0.008 and 0.0087 s. If we consider three places after decimal, then the effect of dataset size on computation time is insignificant. The effect is in the order of 10^{-4} s. This computation time is acceptable for online implementation of the algorithm because the online sensors monitoring the secondary variables are expected to have sampling time higher than this computation time. Hence it can be concluded that a relevant dataset size of 50 can be used in subsequent nonlinear model development.

Average computation time for the JITL soft sensors based on non-linear local models (JITL-SVR: SMO and JITL-SVR: ISDA) are determined for a relevant dataset size of 50. The average computation time required for the JITL-SVR: SMO model varies from 0.0125 to 0.0148 s and 0.0124 to 0.0153 s in JITL-SVR: ISDA model. The average computation time required to predict the required output for non-linear local model based JITL soft sensor is higher than that based on linear local models (This may be due to nonlinear relationships between model parameters and predictor variables involved in case of non-linear models).

The average computation time in case of any model for both IBP and EBP are very close to each other. Further, computation time for non-linear models is higher than that for linear local models. This may be due to the fact that higher degree of complexity is involved in development of the non-linear local model at each sampling instance. Though, the computation time was slightly higher, it is still well below the hardware sensor sampling rate which makes it suitable for online implementation. Therefore, the choice of whether to go for linear or non-linear local model is left to the user to decide. Both models are computationally acceptable. Both models almost give the same performance for IBP as well. However, in addition to IBP, if we also want reasonable accuracy for EBP prediction, then nonlinear local model is to be preferred. Considering model accuracy and model computation time (Refer Table 5.2 & Table 5.3) as the criteria for model selection it can be concluded that JITL-SVR: ISDA has the best prediction accuracy for both naphtha IBP and EBP prediction. Figures 5.7 & 5.8 represent the performance of all developed models for IBP and EBP estimation.

Table 5.2: Performance results using adaptive model for naphtha IBP prediction

Adaptive approach	Local model	CC	MAE, ^{0}C	RMSE, ^{0}C	Average computation time, s
Recursive§	GRNN	0.2093	6.66	8.03	0.043
Sliding window§		0.4143	5.38	7.06	0.043
Just-in-Time learning	LWR	0.8610	2.45	3.32	0.008
	MLR	0.8594	2.44	3.35	0.008
	PLS (3)	0.8453	2.70	3.46	0.008
	PLS (4)	0.8465	2.55	3.45	0.008
	GRNN (D)	0.6471	3.91	4.94	0.031
	GRNN (A&D)	0.6841	3.87	4.84	0.029
	GRNN (MD)§	0.1493	6.44	7.97	0.085
	SVR: SMO	0.8535	2.68	3.37	0.012
	SVR: ISDA	0.8646	2.58	3.27	0.012

§ - With bias update

Table 5.3: Performance results using adaptive model for naphtha EBP prediction

Adaptive approach	Local model	CC	MAE, ^{0}C	RMSE, ^{0}C	Average computation time, s
Recursive§	GRNN	0.0336	4.80	6.66	0.043
Sliding window§		0.2049	6.27	7.83	0.041
Just-in-Time learning	LWR	0.6089	3.44	4.30	0.008
	MLR	0.6585	3.38	4.19	0.008
	PLS (3)	0.6228	3.47	4.24	0.008
	PLS (4)	0.6027	3.45	4.35	0.008
	GRNN (D)	0.5781	3.39	4.21	0.031
	GRNN (A&D)	0.6529	2.96	3.82	0.029
	GRNN (MD)§	0.0542	4.71	6.58	0.088
	SVR: SMO	0.6986	3.15	3.81	0.013
	SVR: ISDA	0.7256	2.87	3.59	0.012

§ - With bias update

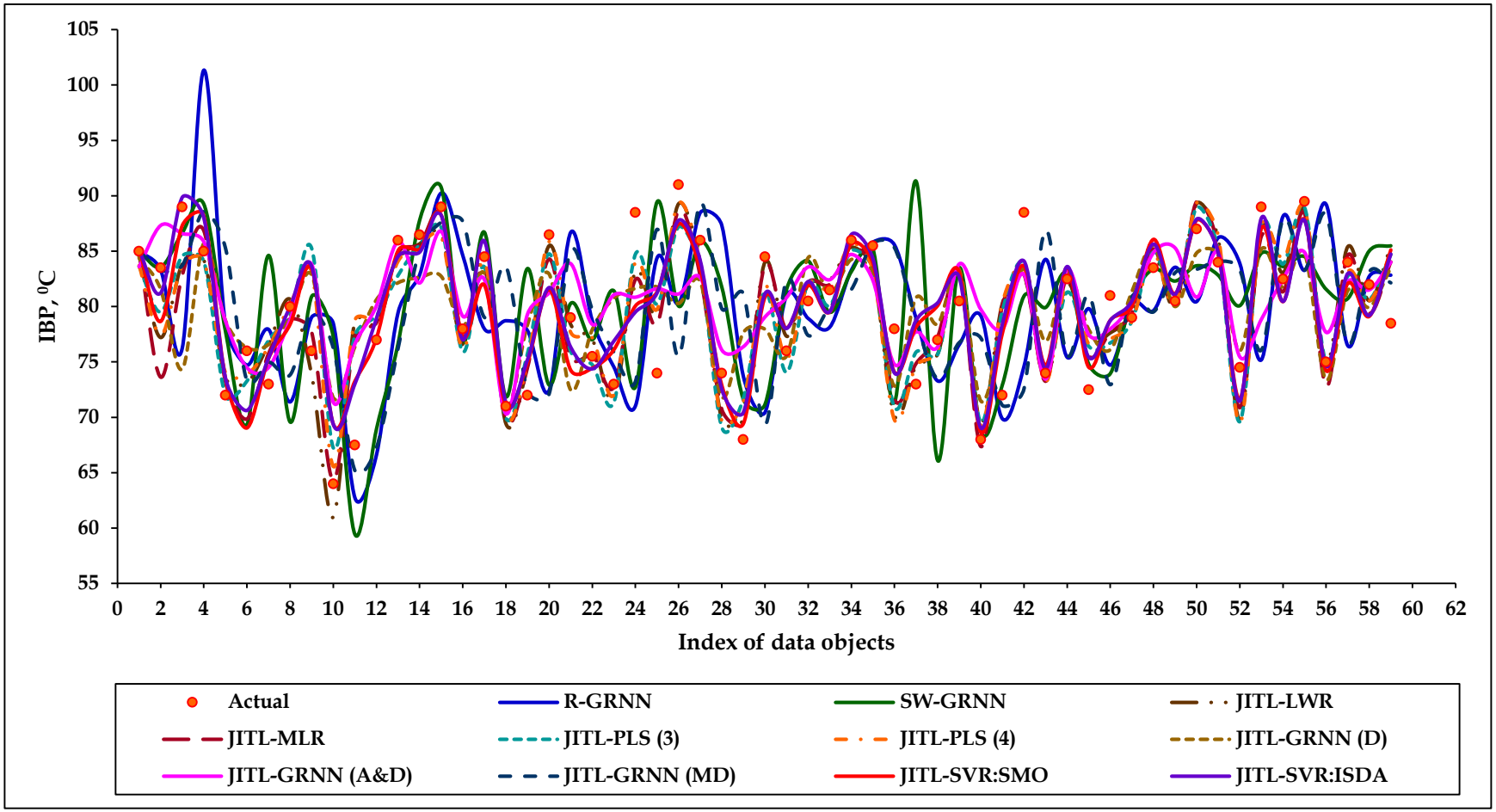


Figure 5.7: Actual Vs predicted values of naphtha IBP by various adaptive soft sensors

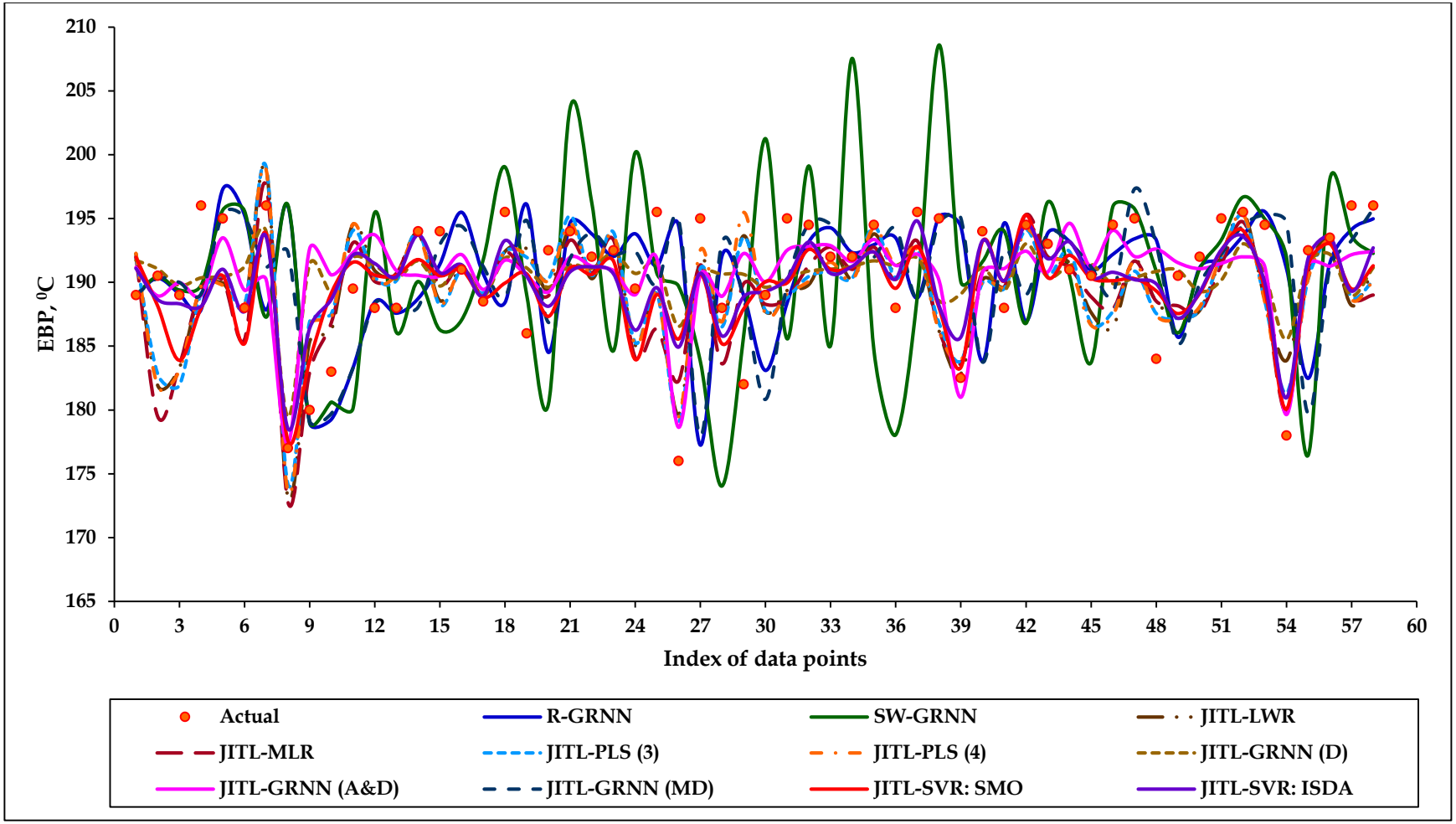


Figure 5.8: Actual Vs predicted values of naphtha EBP by various adaptive soft sensors

5.5.2 Prediction accuracy for Butane concentration prediction in the bottom stream

In case of the debutanizer column, the butane content is estimated by developing three adaptive models (using recursive, sliding window and just-in-time learning) with datasets (Refer Table 5.4). In adaptive just-in-time learning model two approaches are followed to compute the similarity index: (1) Euclidean distance (2) Combined angle and Euclidean distance. Based on two similarity index, two distinct soft sensors were developed based on JITL and performance results were mentioned in Table 5.4.

Table 5.4 provides the results achieved through three adaptive approaches. Recursive and sliding window approach were developed using GRNN model by keeping the fixed spread parameter value (σ = 0.8279).

Figure 5.9 presents the prediction performance of various proposed model against actual values for debutanizer column. The respective figure validates the results shown in Tables 5.4 and show closeness of estimation through various adaptive models.

Next, with σ and RDS values fixed at 0.01 and 10, we present the comparison of two similarity index calculation techniques: Combined angle and distance method Vs distance method in Table 5.4. It is clearly observed in Table 5.4 that, combined angle and distance similarity index method performed better than the distance method in debutanizer column. Model performance index CC, increased by more than 12% (for debutanizer column) while using angle + distance criterion over distance only criterion for similarity measurement. Also, performance of recursive and sliding window approaches are presented along with just-in-time learning approach. In adaptive soft sensors, computation time includes time required for training the model and simulation of model for each incoming input vector. For JITL approach, this involves computation of similarity of the query sample with each

sample of the dataset, preparation of the relevant dataset, development of the trained network (model identification) and finally simulation of the developed model with the query sample. However, just-in-time learning with Mahalanobis distance approach provides better prediction accuracy than other just-in-time learning approaches. Further, the average computation time for just-in-time learning approach is also mentioned along with other two adaptive frameworks.

Table 5.4: Performance results using adaptive model for debutanizer column

Adaptive approach	Local model	CC	MAE	RMSE	Average computation time, s
Recursive§	GRNN	0.9950	0.0117	0.0175	0.05
Sliding window§		0.9982	0.0085	0.0120	0.04
Just-in-Time learning	GRNN (D)	0.8768	0.0383	0.0873	0.08
	GRNN (A & D)	0.9894	0.0139	0.0256	0.08
	GRNN (MD)§	0.9947	0.0121	0.0180	0.23

§ - With bias update

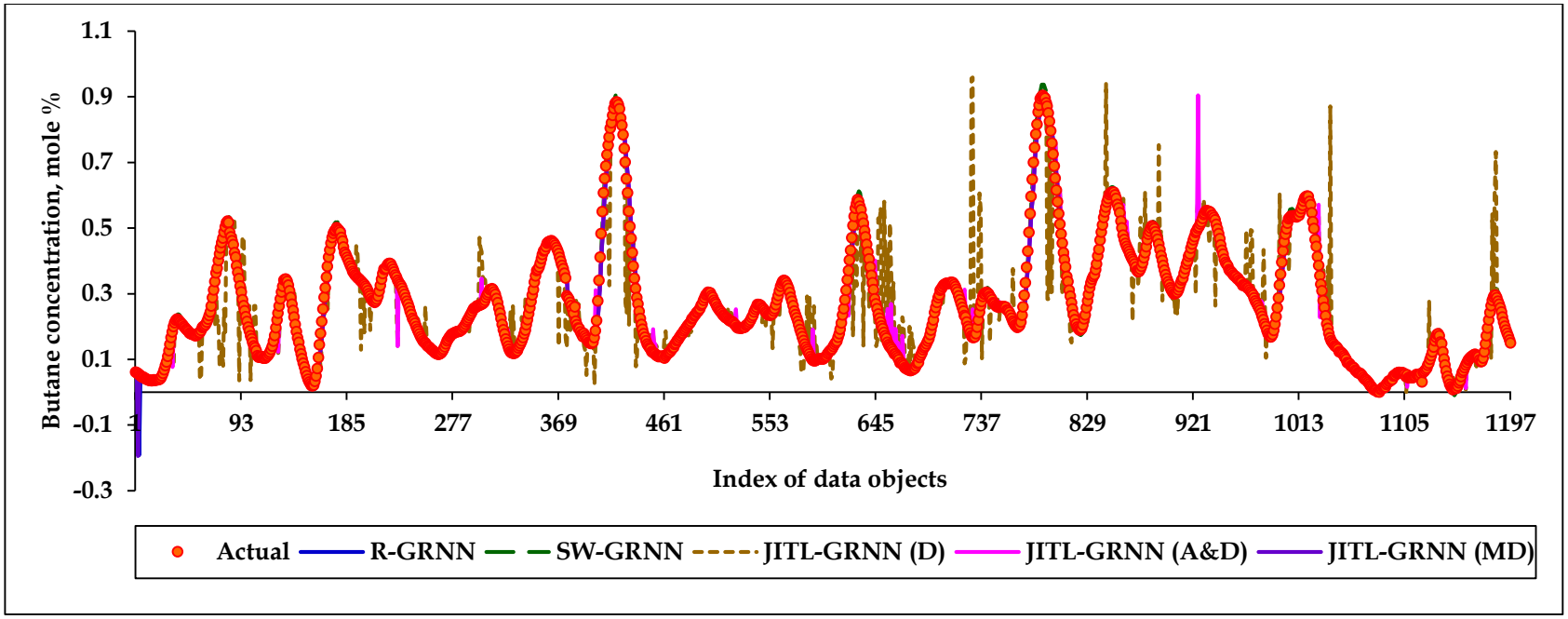


Figure 5.9: Prediction of butane concentration using various adaptive models developed

5.5.3 Prediction accuracy for H_2S and SO_2 concentration in tail gas

In SRU, two outputs (H_2S and SO_2) are estimated by the three adaptive models developed in this work: recursive, sliding window and just-in-time learning approaches based on generalized regression neural network model. For recursive and sliding window approach fixed spread parameter (σ = 0.7690) value is taken during execution.

In Table 5.5 & Table 5.6, we present the comparison of all adaptive models developed for prediction of tail gas composition in SRU. Figures 5.10 & 5.11 presents the prediction performance of various proposed model against actual values for prediction of H_2S and SO_2 in tail gas of SRU. The figures 5.10 & 5.11 validates the results shown in Table 5.5 & Table 5.6 and reveal closeness of estimation with actual values through various adaptive models.

It is clearly observed in Table that, just-in-time learning with Mahalanobis distance approach performed better than the other just-in-time learning approaches in SRU. Model performance index CC, increased by more than 36% (for SRU) while using Mahalanobis distance criterion over distance only criterion for similarity measurement. Moreover, sliding window approach performs better than other two approaches (recursive and just-in-time learning) for prediction of H_2S and SO_2 in the tail gas of SRU as shown in Table 5.5 & Table 5.6. Moreover, the applicability domain for sliding window is less (it takes samples within the particular window as specified by the user) as compared to recursive or just-in-time learning approaches.

Average computation time per query data (incoming sample) for the all approaches is mentioned along with the prediction results in the respective case studies. The computation time of SRU is higher than CDU and debutanizer column. This can be understood by referring to Table 5.2 - Table 5.6. The database size for CDU is 151 and debutanizer column

is 1197 whereas for SRU, this is 5040. Therefore, for SRU, the computation of vector will require more time because, this involves calculation of similarity of the query data with each sample of the database (particularly JITL) and simulation of models developed.

Table 5.5: Performance results using adaptive model for prediction of H_2S

Adaptive approach	Local model	CC	MAE	RMSE	Average computation time, s
Recursive§	GRNN	0.9445	0.0059	0.0170	0.14
Sliding window§		0.9449	0.0065	0.0175	0.04
Just-in-Time learning	GRNN (D)	0.7555	0.0176	0.0364	0.22
	GRNN (A & D)	0.9388	0.0064	0.0180	0.26
	GRNN (MD)§	0.9438	0.0062	0.0171	1.12

§ - With bias update

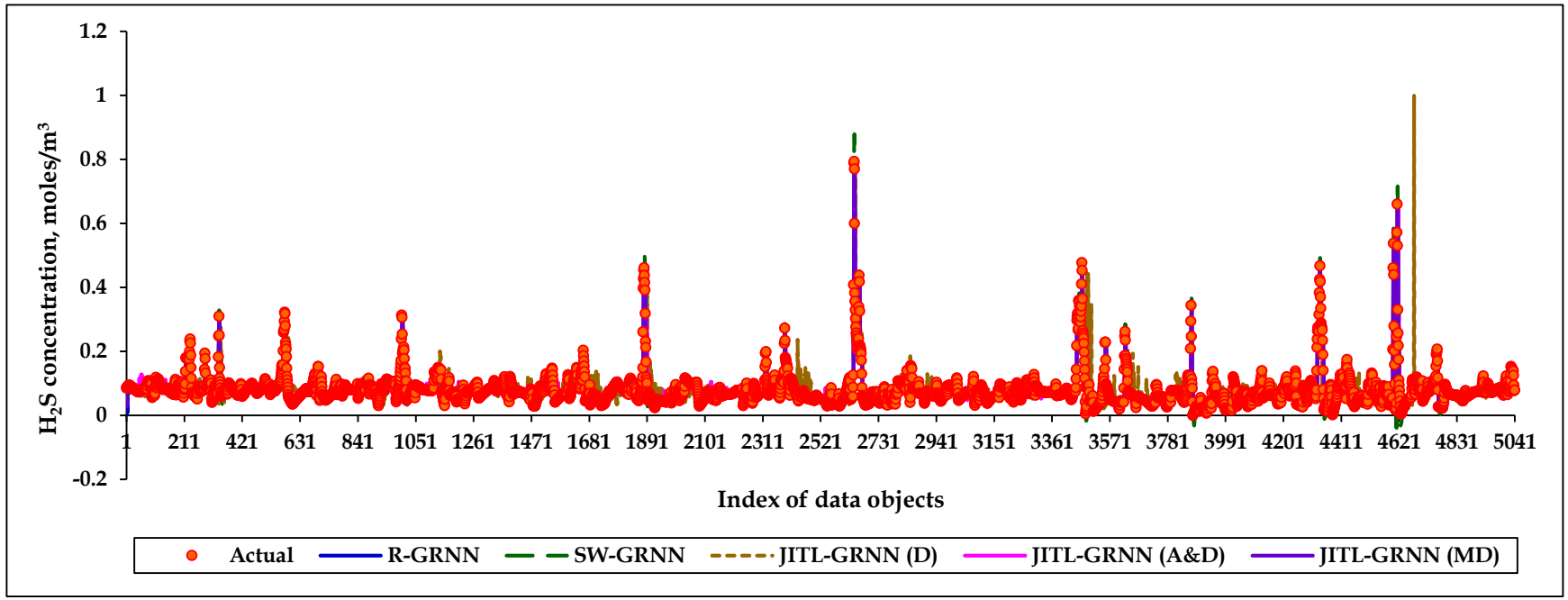


Figure 5.10: Prediction of H_2S concentration in tail gas of SRU using various adaptive models developed

Table 5.6: Performance results using adaptive model for prediction of SO_2

Adaptive approach	Local model	CC	MAE	RMSE	Average computation time, s
Recursive§	GRNN	0.9600	0.0070	0.0147	0.13
Sliding window§		0.9613	0.0077	0.0150	0.04
Just-in-Time learning	GRNN (D)	0.6959	0.0255	0.0393	0.22
	GRNN (A & D)	0.9495	0.0080	0.0165	0.26
	GRNN (MD)§	0.9584	0.0078	0.0150	1.13

§ - With bias update

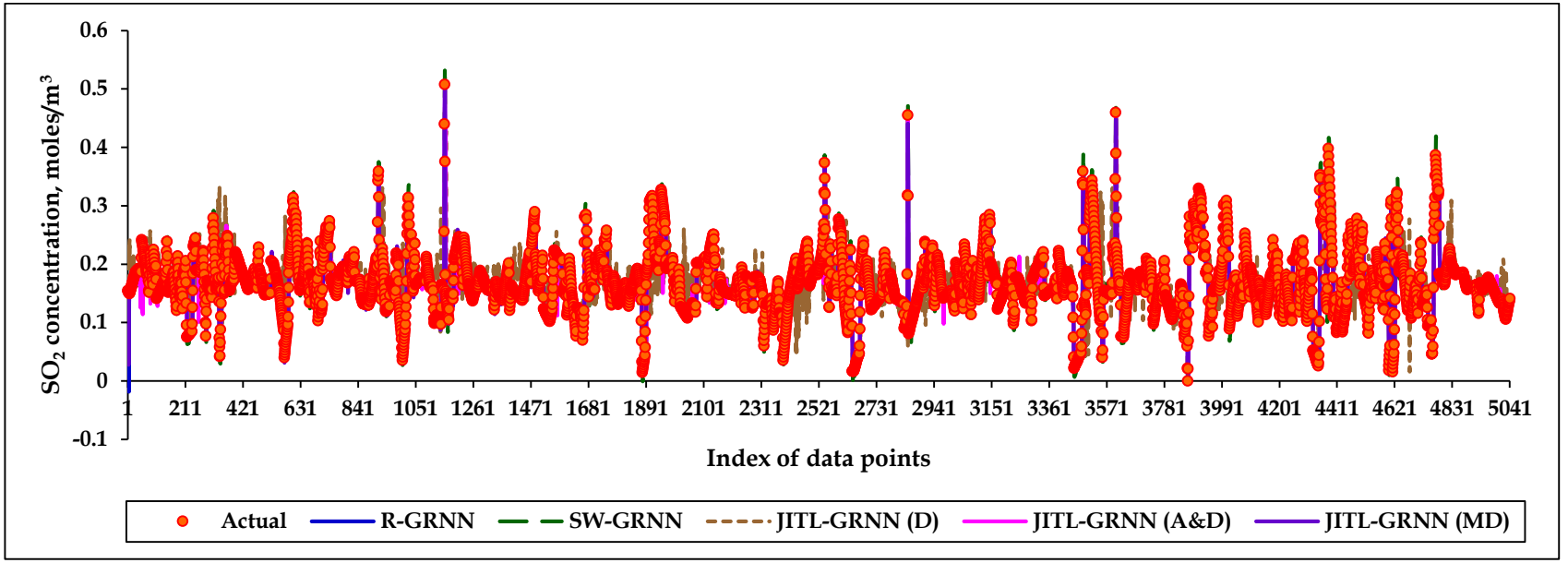


Figure 5.11: Prediction of SO_2 concentration in tail gas of SRU using various adaptive models developed

However, for online implementation purposes, computation times of the proposed algorithm shown in performance results (which are fractions of a second) are acceptable because the online sensors monitoring the secondary variables are expected to have sampling time higher than this computation time (usually in the order of seconds). Moreover, for online implementation of adaptive JITL based soft sensor, one may choose to store less number of samples in the database which will result in less model computation time. As analogous to JITL, recursive algorithm holds similar patterns for all industrial processes. Rather, sliding window model requires less average model computation time, which is almost fraction of seconds for all the cases. This is due to the few number of samples within the window is taken for model development and estimation.

5.5.4 4-plot analysis for the best performing model

Four plots are used for residual analysis to identify the distribution of penalty for predicted variables over the observed output. It includes run sequence plot, lag plot, histogram and normal probability plot. Run sequence plot provides the information about residual trend around the mean while propagating through the time ordered observations. The trend may be increasing, decreasing or constant variance trend. The ideal trend observed to be constant variance trend, as the error variance seems no process drift as moving though the large number of time varying observations. Lag plot shows the graph of sequence of observations against the one time lagged sequence of observations. This provides the information about the spread of error variance around the mean. It is also observed to be random in nature and therefore not dependent on time. Histogram represents, how the error variance is normally distributed around the mean. The assumption of normal distribution is found to be true, when the histogram shape is symmetric and bell shaped. The normal

probability plot shows the distribution of error variance in sequential dispositions against the probability percentile, which is normally distributed. The linear graph indicates that the residual is normally distributed.

The residual analysis of JITL-SVR: ISDA soft sensor for crude distillation unit, SW-GRNN soft sensor for debutanizer column and SRU is carried out by four-plot analysis technique (Fortuna et al., 2007; Pani & Mohanta, 2016).

The results of JITL-SVR: ISDA for CDU are shown in the following Figures 5.12 & 5.13.

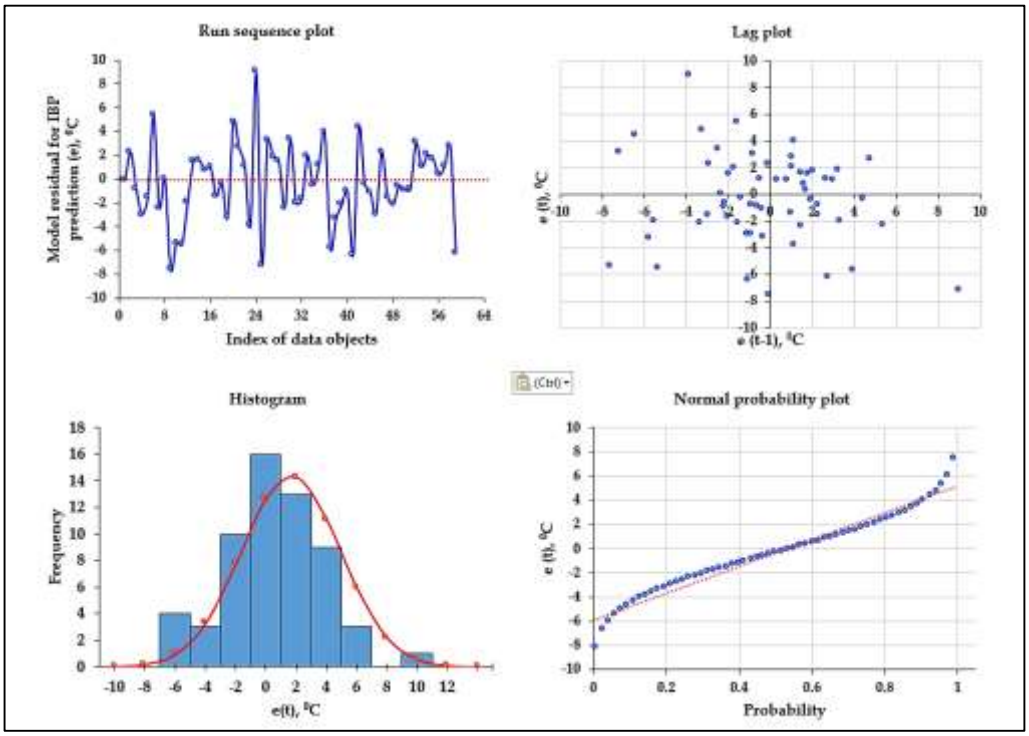


Figure 5.12: JITL-SVR: ISDA model validation for IBP prediction

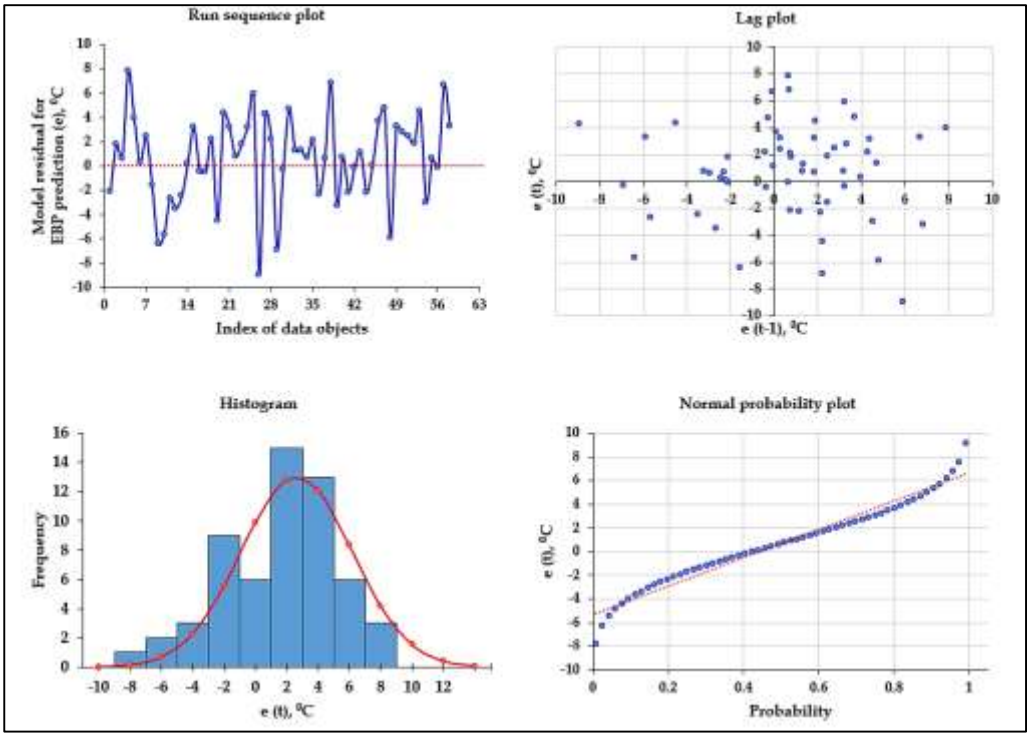


Figure 5.13: JITL-SVR: ISDA model validation for EBP prediction

Figure 5.14 provides four plot analysis of JITL-GRNN (MD) for debutanizer column.

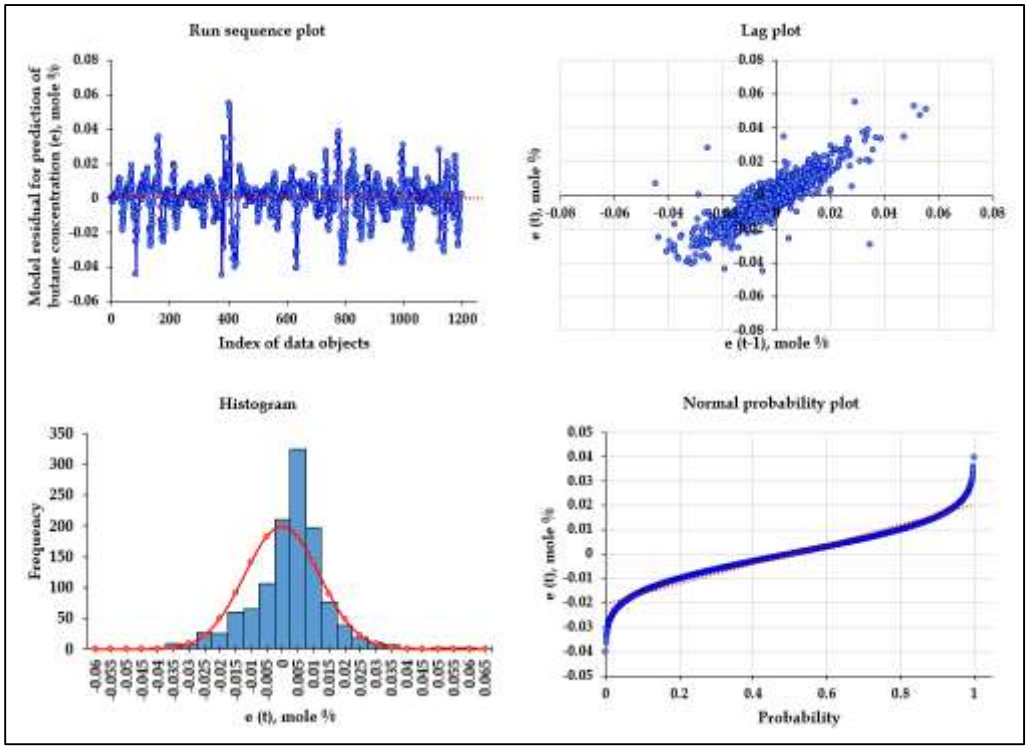


Figure 5.14: JITL-GRNN model validation for prediction of butane concentration

Figure 5.15 & Figure 5.16 provide four plot analysis of JITL-GRNN (MD) for prediction of H_2S and SO_2 in the tail gas of SRU.

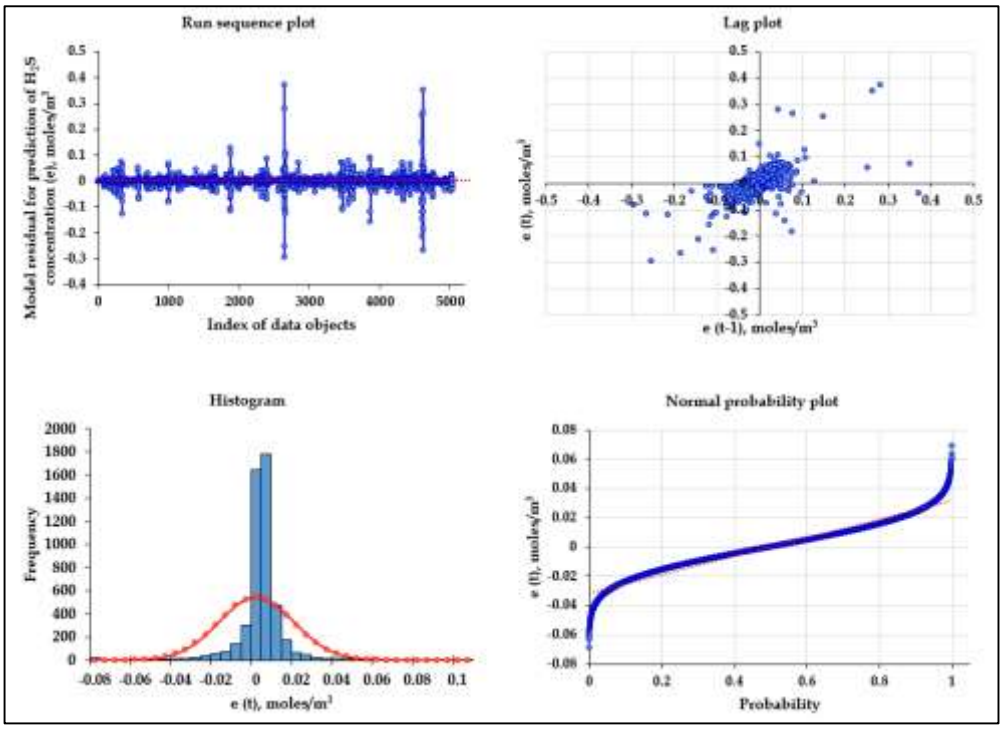


Figure 5.15: JITL-GRNN model validation for prediction of H_2S concentration

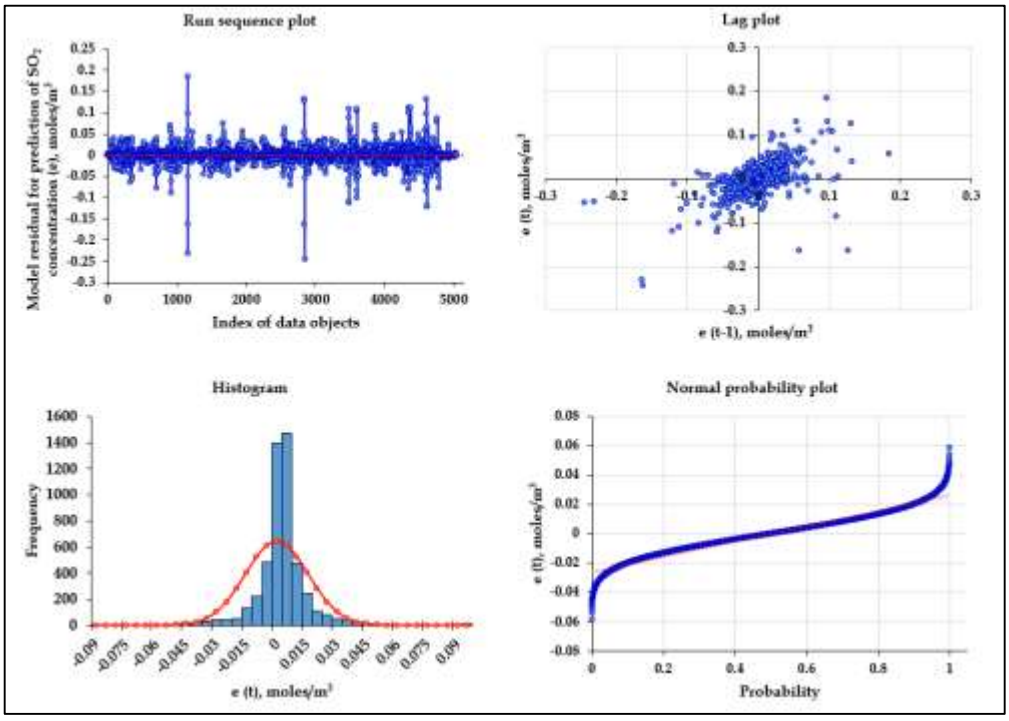


Figure 5.16: JITL-GRNN model validation for prediction of SO_2 concentration

5.5.5 Comparison of predictive performance with other models

There is a scope for comparison of different modeling techniques reported in literature by various researchers on same benchmark datasets. A number of different modeling approaches are reported in the literature (Table 5.7 & Table 5.8) for the two benchmark datasets of debutanizer column and SRU.

Table 5.7: Performance comparison of JITL-GRNN (MD) model with reported models in literature for debutanizer column

Debutanizer column			Butane content		
Author (s)	**Year**	**Model**	**CC**	**RMSE**	**MAE**
Fortuna et al.	2007	Neural network model (Multi-layer perceptron)	0.9409		
Ge & Song	2010	Least Square Support Vector Regression – Just-in-Time Learning	0.9132	0.1418[a]	
Fan et al.	2014	Gaussian Mixture Model - Just-in-Time Learning		0.125[b]	
Matias et al.	2015	Online Sequential Extreme Learning Machine based on Recursive Partial Least Squares		0.4898[b]	
Pani et al.	2016	Back Propagation Neural Network	0.856	0.076	0.055
Bidar et al.	2017	State Dependent Parameter Auto-regressive with Exogenous Variable Model	0.9975	0.0101	0.0109
Yuan et al.	2016	Probabilistic Just-in-Time Learning		0.0624	
		Deterministic Just-in-Time Learning		0.0944	
Yuan et al.	2017	Locally Weighted Partial Least Squares Regression	0.9205[c]	0.0621	
		Weighted Gaussian Regression	0.9071[c]	0.0669	
Siddharth et al.	2019	Adaptive Neuro-Fuzzy Inference System	0.8829	0.0672	0.05
Singh et al.	2019	Generalized Regression Neural Network	0.834	0.082	0.059
Urhan & Alakent	2020	Adaptive Moving Window and Just-in-Time learning Ensemble	0.9940[c]	0.0234	0.0132
Alakent	2020	Just-in-Time Learning based on Online Weighted Euclidean Distance	0.9737[c]	0.0409	
Alakent	2020	Moving Window and Just-in-Time Learning using transductive inference	0.9970[c]	0.0149	
This work	2022	Recursive JITL-GRNN (A&D)			
		Recursive JITL-GRNN (MD)	0.9947	0.0180	0.0121

[a] Data available in graphical form (Approximate); [b] Data given in MSE (Converted to RMSE); [c] Data given in R^2 (Converted to R)

Table 5.8: Performance comparison of JITL-GRNN (MD) model with reported models in literature for sulfur recovery unit

Sulfur recovery unit			H_2S			SO_2		
Author (s)	**Year**	**Model**	**CC**	**RMSE**	**MAE**	**CC**	**RMSE**	**MAE**
Fortuna et al.	2003	Non-linear Least Square Fitting	0.848	0.0283[a]		0.905	0.02[a]	
Shao et al.	2015	Supervised Local and Non-local Structure Preserving Projections Locally Weighted Regression Just-in-Time learning		0.0093	0.091		0.0101	0.089
Shang et al.	2015b	Correlation based Slow Feature Regression	0.6561	0.03	0.017	0.8721	0.0258	0.0206
Graziani & Xibilia	2017	Multilayer Perceptron		1.33				
		Deep Boltzmann Machine		1.20				
Jain et al.	2017	Principal Component Regression		0.017	0.019		0.0247	0.026
		Support Vector Regression		0.021	0.015		0.035	0.025
		Least Squares Support Vector Regression		0.056	0.045		0.12	0.095
Moghadam et al.	2018	Time Varying Parameter - Dynamic Auto-regressive with Exogenous Variable model	0.9507	0.0158	0.0024	0.9648	0.0138	0.0018
Morey et al.	2018	Gaussian Process Regression with Marginal Log likelihood Maximization	0.718	0.035	0.017	0.66	0.043	0.025
		Adaptive Neuro-Fuzzy Inference System	0.271	0.027	0.019	0.314	0.036	0.027
Singh et al.	2019	Generalized Regression Neural Network[b]	0.631	0.024	0.018	0.631	0.024	0.018
This work	2022	Recursive JITL-GRNN (A&D)						
		Recursive JITL-GRNN (MD)	0.9438	0.0171	0.0062	0.9584	0.0150	0.0078

[a] Data given in MSE (Converted to RMSE); [b] Data given in mean of statistical indexes between two outputs

Chapter - 6

Conclusions

Prediction of primary quality variables in real time with adaptation capability for varying process conditions is a critical task in process industries. Due to inherent complexities associated with various chemical processes, developing first principle model for prediction of primary quality variables is highly challenging task. Empirical models are developed from actual input-output data collected from various industries. Moreover, there have been few applications of adaptive soft sensing techniques in chemical processes such as refineries etc., Therefore, the prime focus of our research work is to develop adaptive soft sensing techniques for continuous monitoring in various processes of refinery.

This section summarizes the work carried out along with its significant findings during the course of research. Finally, future scope of this research work is also mentioned.

6.1 Summary of proposed implementation

Refineries is a highly complex process which involves fractionation of petroleum (from higher end to lighter end products). Estimation of primary quality variables requires time consuming laboratory analysis and is a tedious one. Real time prediction of primary variables using predictive model is mandatory for tight control of product quality. Moreover, the available soft sensors (predictive model) are conventional in nature. During time varying conditions (such as process state changes, catalyst deterioration etc.,) conventional soft sensor fails to predict the quality variables. In order to cope up with time varying conditions, adaptive soft sensors were placed in operation. Adaptive models developed in this work include recursive, sliding window and just-in-time learning approaches. The developed adaptive

models were tested using three industrial datasets: Naphtha splitter unit, Debutanizer column and Sulfur recovery unit.

Naphtha splitter unit

For naphtha splitter unit, the dataset for prediction of IBP and EBP of heavy naphtha are collected from Ujević et al. (2011). Totally 210 data objects for IBP and 209 data objects for EBP with six inputs and two outputs are taken. For IBP, 151 for database and 59 for query data objects are taken and for EBP, 151 for database and 58 for query objects are taken. Three types of adaptive models were developed using this datasets: Recursive (GRNN), Sliding window (GRNN), Just-in-Time Learning-D (LWR, MLR, PLS-3, PLS-4, GRNN, SVR: SMO and SVR: ISDA), Just-in-time Learning-A&D (GRNN) and Just-in-time Learning-MD (GRNN).

Debutanizer column

For debutanizer column, the benchmark datasets are taken from Fortuna et al. (2007). Totally 2394 data objects were collected with seven inputs and one output. For estimation of butane concentration, 1197 objects for database and 1197 for query objects are taken. Three types of adaptive models were developed: Recursive (GRNN), Sliding window (GRNN), Just-in-Time Learning (GRNN-D, GRNN-A&D, GRNN-MD).

Sulfur recovery unit

Likewise, for sulfur recovery unit, the benchmark datasets are taken from Fortuna et al. (2007). Totally 10081 data objects were collected with five inputs and two outputs. For estimation of H_2S and SO_2 concentration, 5040 for database and 5041 for query objects are

taken. Three types of adaptive models were developed: Recursive (GRNN), Sliding window (GRNN), Just-in-Time Learning (GRNN-D, GRNN-A&D, GRNN-MD).

Performance analysis of all developed models was carried out by estimating the model outputs of various industrial processes considered. The performance analysis was carried out using three distinct statistical performance indices: Correlation coefficient (CC or R), Mean absolute error (MAE) and Root mean square error (RMSE). Along with this, average model computation time was determined for single query objects of every model developed by considering implementation purpose. The best desired model is selected based on the prediction accuracy in terms of performance indices values for all the three industrial processes. Four plot analysis is carried out for the best performing model in each industrial case study.

6.2 Significant observations and findings

In this section, the significant findings of this research work is presented. This comprises of four sub-sections, which are general observations followed by respective observations and findings related to Naphtha splitter section, DC and SRU.

6.2.1 General observations

1. For all three industrial case studies, three types of adaptive models were developed based on recursive, sliding window and just-in-time learning approaches. Size of the datasets was different for various industrial cases.
2. For all case studies, recursive, sliding window model and just-in-time learning (Mahalanobis distance) are developed based on GRNN (with fixed spread parameter) alone. Also, just-in-time learning based on GRNN model (spread parameter

calculated using grid search technique) was developed using two types of similarity index (distance and combined angle & distance).

3. Model performance was analyzed using three performance indices: correlation coefficient, mean absolute error and root mean square error. Along with this, average model computation time per query object was calculated using the same software function.
4. During model evaluation, the correlation coefficient index was given higher preference than mean absolute error and root mean square error for selection of the best model.
5. Residual analysis is carried out for best performance model using four plot analysis.

6.2.2 Soft sensing of naphtha splitter section

1. Initial and end boiling points of heavy naphtha are important process parameters which need to be maintained at specified values for ensuring better process performance in subsequent stages of petroleum refineries. Hence, adaptive soft sensors have huge scope of relevance to ensure quality of naphtha in fractionation process.
2. In this case study, recursive, sliding window and just-in-time learning concept is applied to develop adaptive soft sensors for online monitoring of heavy naphtha IBP and EBP. During the course of model development various aspects of JITL modeling are investigated. These include effect of various hyper parameters on model performance, effect of relevant dataset size on model performance and model computation time.

3. Using recursive and sliding window approaches, the correlation coefficient is obtained as 0.2093 and 0.4143 with model computation time of 1.4841 s and 1.3038 s respectively.
4. In just-in-time learning approach, both linear and non-linear local models are designed. The best correlation coefficient values by any linear model is 0.86 for IBP and 0.66 for EBP (JITL-MLR) with a model computation time of 0.0083 s. The correlation coefficient values for JITL-SVR: ISDA are 0.86 for IBP and 0.73 for EBP with a model computation time of 0.012 s. The improvement in IBP prediction was insignificant. However, there is an improvement of more than 10% for EBP prediction in case of the proposed nonlinear JITL model.
5. After performance analysis of various models using just-in-time learning approach, it was observed that support vector regression model produced better prediction accuracy than multiple linear regression, partial least squares regression and locally weighted regression models.
6. Further, just-in-time learning based SVR model provides better accuracy than recursive and sliding window based GRNN models.
7. The performance was further validated using residual errors by four plot analysis. The criteria chosen for good modeling are, better generalization capability and low computational time. An error margin of about $\pm 3^0$C temperature is achieved by the JITL-SVR: ISDA model for prediction of IBP and EBP.
8. Reasonably low computation time of model simulation indicates that the proposed JITL-SVR model can be implemented online as adaptive soft sensor for continuous estimation of naphtha quality.

6.2.3 Soft sensing of DC

1. Prediction of butane content in the bottom of debutanizer column helps to operate the column in an economically profitable way. Hence, adaptive soft sensors have huge scope of relevance to ensure product quality such that the butane should not escape from the bottom of the debutanizer column.
2. Adaptive models such as recursive, sliding window, just-in-time learning approaches based on GRNN were developed for online monitoring of butane content in debutanizer column. In this research work, investigation of effect of similarity index and relevant dataset size on model performance and model computation time are included.
3. The best correlation coefficient value by just-in-time learning model based on Mahalanobis distance method is 0.9947 for butane content prediction with a model computation time of 0.23 s. The improvement in prediction of butane content based on Mahalanobis distance method is significant as compared to Euclidean distance (>12%) method.
4. Further, there is an improvement in the predictive performance while using recursive and sliding window approaches in which the correlation coefficients of 0.9950 and 0.9982 are achieved with average model computation time of 0.05 s and 0.04 s respectively.
5. After performance analysis, it was observed that JITL-GRNN-MD has better prediction accuracy than JITL (distance only, combined angle and distance method) approaches. The performance is further compared with other existing models

proposed in literature (Refer Table 5.7 in Section 5.5.3), which shows that the proposed model is having better prediction accuracy like other models.

6. Reasonably low computation time of model simulation indicates that the JITL-GRNN-MD model can be implemented online as adaptive soft sensor for estimation of butane content in debutanizer column.

6.2.4 Soft sensing of SRU

1. Estimation of H_2S and SO_2 concentrations in the tail gas of sulfur recovery unit is mandatory as the sulfur acts as major cause for environmental pollution (i.e., acid rain) and is found to be of major concern. Sulfur is removed in the form of elemental sulfur from its constituents as a by-product through desulfurization or gas sweetening process.
2. Adaptive models such as recursive, sliding window, just-in-time learning approaches based on GRNN were developed for online monitoring of H_2S and SO_2 concentrations in the tail gas of sulfur recovery unit. The present research includes investigation of effect of similarity index and relevant dataset size on model performance and model computation time.
3. The best correlation coefficient values by just-in-time learning model Mahalanobis distance method for sulfur recovery unit are 0.9438 for H_2S and 0.9584 for SO_2 with a model computation time of 1.12 s for H_2S and 1.13 s for SO_2. The improvement in prediction of tail gas composition (>36%) based on Mahalanobis distance method is significant as compared to Euclidean distance method.
4. Further, there is a slight improvement in the predictive performance while using recursive and sliding window approaches in which the correlation coefficients of

0.9445 and 0.9449 are achieved with average model computation time of 0.14 s and 0.04 s for H_2S respectively. Likewise, the correlation coefficient achieved by recursive and sliding window approaches for SO_2 are 0.9600 and 0.9613 with average model computation time of 0.13 s and 0.04 s respectively.

5. After performance analysis, it was observed that JITL-GRNN-MD has better prediction accuracy than other JITL (distance only, combined angle and distance method) approaches. The performance is further compared with other existing models proposed in literature (Refer Table 5.8 in Section 5.5.3), which shows that the proposed model is having better prediction accuracy like other models.
6. Reasonably low computation time of model simulation indicates that the JITL-GRNN-MD model can be implemented online as adaptive soft sensor for estimation of H_2S and SO_2 concentrations in tail gas of SRU.

6.3 Major contributions

1. As of now, for naphtha boiling point estimation, only steady state soft sensors are reported in the literature. In this work, adaptive soft sensors are developed for prediction of initial and end boiling point of heavy naphtha.
2. There are some adaptive models reported in DC and SRU process. However, use of GRNN in the adaptation framework is a novel contribution of this work. Also, new adaptive soft sensor based on GRNN as local modeling strategy for estimation of quality variables in all three case studies (NS, DC and SRU)
3. In addition to above mentioned algorithm, recursive and sliding window approaches (based on Generalized regression neural networks) are other newly developed

algorithms associated with this research work and has the potential to implemented in nonlinear continuous processes for quality estimation.

4. Recursive Just-in-Time Learning algorithm is a unified approach by combining recursive and just-in-time learning frameworks based on nonlinear models (SVR and GRNN) for quality estimation is not available in the literature. However, unified approach based on linear model (PLS) for estimation of final boiling point of diesel oil by Chen et al. (2014) is so far reported.
5. In JITL approach, extensive investigation on effect of different similarity index, computation time on model prediction accuracy is not reported in the literature. In this work, three similarity measures (D, A&D and MD) incorporated in the JITL frameworks are compared and reported.
6. The algorithm for all three adaptive soft sensors are incorporated with bias-update procedure to enhance the predictive accuracy.

6.4 Future scope of research

In this section, we discuss the future scope of this research work in industrial processes.

1. There is a scope of developing new adaptive model, that has the capability to include time delayed values (dynamic model) into the model architecture, which has the capability to improve the predictive performance, as well as more robust to handle process disturbances.
2. Also, the model can be tested with various process shifts (drifting phenomena) occurring in industrial processes and test its predictive performance.

3. There is a huge scope in exploration of hybrid modeling technique by integration of multiple global models (as ensembles) in adaptive frameworks in various chemical processes.
4. Most of the non-linear models require different hyper parameters (loss function in SVR, spread parameter in GRNN etc.) to be optimally determined. In this work, rigorous grid search and analytical formula were used for determination of hyper parameter values. Application of various evolutionary optimization techniques in the adaptation framework can be explored for optimum hyper parameter computation.
5. Applications of combining adaptive approach with deep learning neural network (reinforcement learning) model in prediction of quality variables are yet to be explored in chemical processes.

References

Abdi, H. (2003). Partial least square regression (PLS regression). *Encyclopedia for research methods for the social sciences*, *6*(4), 792-795.

Abdolkarimi, V., Sari, A., & Shokri, S. (2022). Robust prediction and optimization of gasoline quality using data-driven adaptive modeling for a light naphtha isomerization reactor. *Fuel*, *328*, 125304.

Abusnina, A., &Kudenko, D. (2013, February). Adaptive soft sensor based on moving gaussian process window. In *2013 IEEE international conference on Industrial Technology (ICIT)* (pp. 1051-1056). IEEE.

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine learning*, *6*(1), 37-66.

Ahmed, F., Nazir, S., & Yeo, Y. K. (2009). A recursive PLS-based soft sensor for prediction of the melt index during grade change operations in HDPE plant. *Korean Journal of Chemical Engineering*, *26*(1), 14-20.

Alakent, B. (2020a). Online tuning of predictor weights for relevant data selection in just-in-time-learning. *Chemometrics and Intelligent Laboratory Systems*, *203*, 104043.

Alakent, B. (2020b). Soft sensor design using transductive moving window learner. *Computers & Chemical Engineering*, *140*, 106941.

Alakent, B. (2021a). Employing Adaptive Just-In-Time-Learning in a Transfer Learning Frame for Soft-Sensor Design. In *Computer Aided Chemical Engineering* (Vol. 50, pp. 913-918). Elsevier.

Alakent, B. (2021b). Soft-sensor design via task transferred just-in-time-learning coupled transductive moving window learner. *Journal of Process Control*, *101*, 52-67.

Atkeson, C. G., Moore, A. W., &Schaal, S. (1997). Locally weighted learning. *Lazy learning*, 11-73.

Bhartiya, S., &Whiteley, J. R. (2001). Development of inferential measurements using neural networks. *ISA transactions*, *40*(4), 307-323.

Bidar, B., Sadeghi, J., Shahraki, F., &Khalilipour, M. M. (2017). Data-driven soft sensor approach for online quality prediction using state dependent parameter models. *Chemometrics and Intelligent Laboratory Systems*, *162*, 130-141.

Birattari, M., Bontempi, G., &Bersini, H. (1998). Lazy learning meets the recursive least squares algorithm. *Advances in neural information processing systems*, *11*.

Bosca, S., &Fissore, D. (2011). Design and validation of an innovative soft-sensor for pharmaceutical freeze-drying monitoring. *Chemical Engineering Science*, *66*(21), 5127-5136.

Bowden, G. J., Nixon, J. B., Dandy, G. C., Maier, H. R., & Holmes, M. (2006). Forecasting chlorine residuals in a water distribution system using a general regression neural network. *Mathematical and computer modeling*, *44*(5-6), 469-484.

Braun, M. W., Rivera, D. E., &Stenman, A. (2001). A 'Model-on-Demand' identification methodology for nonlinear process systems. *International Journal of Control*, *74*(18), 1708-1717.

Chen, K., Ji, J., Wang, H., Liu, Y., & Song, Z. (2011). Adaptive local kernel-based learning for soft sensor modeling of nonlinear processes. *Chemical Engineering Research and Design*, *89*(10), 2117-2124.

Chen, M., Khare, S., & Huang, B. (2014). A unified recursive just-in-time approach with industrial near infrared spectroscopy application. *Chemometrics and Intelligent Laboratory Systems*, *135*, 133-140.

Chen, X., Chen, X., She, J., & Wu, M. (2017). A hybrid just-in-time soft sensor for carbon efficiency of iron ore sintering process based on feature extraction of cross-sectional frames at discharge end. *Journal of Process Control*, *54*, 14-24.

Chen, X., Mao, Z., Jia, R., & Zhang, S. (2019). Ensemble regularized local finite impulse response models and soft sensor application in nonlinear dynamic industrial processes. *Applied Soft Computing*, *85*, 105806.

Cheng, C., & Chiu, M. S. (2004). A new data-based methodology for nonlinear process modeling. *Chemical engineering science*, *59*(13), 2801-2810.

Cheng, C., & Chiu, M. S. (2005). Nonlinear process monitoring using JITL-PCA. *Chemometrics and Intelligent Laboratory Systems*, *76*(1), 1-13.

Cherkassky, V., & Ma, Y. (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural networks*, *17*(1), 113-126.

Ciochina, S., Paleologu, C., Benesty, J., &Enescu, A. A. (2009, July). On the influence of the forgetting factor of the RLS adaptive filter in system identification. In *2009 International Symposium on Signals, Circuits and Systems* (pp. 1-4). IEEE.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297.

Cregan, V., Lee, W. T., &Clune, L. (2017). A soft sensor for the Bayer process. *Journal of Mathematics in Industry*, *7*(1), 1-6.

Cybenko, G. (1996). Just-in-time learning and estimation. Identification, Adaptation, Learning: The Science of Learning Models from Data, edited by Sergio Bittanti, Giorgio Picci (*Nato ASI Series F Computer and Systems Sciences), 153*, 423-434.

Dam, M., &Saraf, D. N. (2006). Design of neural networks using genetic algorithm for on-line property estimation of crude fractionator products. *Computers & chemical engineering*, *30*(4), 722-729.

Damour, C., Benne, M., Grondin-Perez, B., &Chabriat, J. P. (2010). Soft-sensor for industrial sugar crystallization: On-line mass of crystals, concentration and purity measurement. *Control Engineering Practice*, *18*(8), 839-844.

Desai, K., Badhe, Y., Tambe, S. S., & Kulkarni, B. D. (2006). Soft-sensor development for fed-batch bioreactors using support vector regression. *Biochemical Engineering Journal*, *27*(3), 225-239.

Duchêne, P., Mencarelli, L., &Pagot, A. (2020). Optimization approaches to the integrated system of catalytic reforming and isomerization processes in petroleum refinery. *Computers & Chemical Engineering*, *141*, 107009.

Facco, P., Doplicher, F., Bezzo, F., & Barolo, M. (2009). Moving average PLS soft sensor for online product quality estimation in an industrial batch polymerization process. *Journal of Process Control*, *19*(3), 520-529.

Fan, M., Ge, Z., & Song, Z. (2014). Adaptive Gaussian mixture model-based relevant sample selection for JITL soft sensor development. *Industrial & Engineering Chemistry Research*, *53*(51), 19979-19986.

Feng, Y., Cui, N., Gong, D., Zhang, Q., & Zhao, L. (2017). Evaluation of random forests and generalized regression neural networks for daily reference evapotranspiration modelling. *Agricultural Water Management*, *193*, 163-173.

Feng, Y., Cui, N., Hao, W., Gao, L., & Gong, D. (2019). Estimation of soil temperature from meteorological data using different machine learning models. *Geoderma*, *338*, 67-77.

Fortuna, L., Graziani, S., & Xibilia, M. G. (2005). Soft sensors for product quality monitoring in debutanizer distillation columns. *Control Engineering Practice*, *13*(4), 499-508.

Fortuna, L., Graziani, S., Rizzo, A., & Xibilia, M. G. (2007). *Soft sensors for monitoring and control of industrial processes* (Vol. 22). London, UK: Springer.

Fortuna, L., Rizzo, A., Sinatra, M., & Xibilia, M. G. (2003). Soft analyzers for a sulfur recovery unit. *Control Engineering Practice*, *11*(12), 1491-1500.

Fujiwara, K., Kano, M., Hasebe, S., &Takinami, A. (2009). Soft-sensor development using correlation-based just-in-time modeling. *AIChE Journal*, *55*(7), 1754-1765.

Galicia, H. J., He, Q. P., & Wang, J. (2012). Comparison of the performance of a reduced-order dynamic PLS soft sensor with different updating schemes for digester control. *Control Engineering Practice*, *20*(8), 747-760.

Ge, Z., & Song, Z. (2010). A comparative study of just-in-time-learning based methods for online soft sensor modeling. *Chemometrics and Intelligent Laboratory Systems*, *104*(2), 306-317.

Godoy, J. L., Minari, R. J., Vega, J. R., &Marchetti, J. L. (2011). Multivariate statistical monitoring of an industrial SBR process. Soft-sensor for production and rubber quality. *Chemometrics and Intelligent Laboratory Systems*, *107*(2), 258-268.

Gonzaga, J. C. B., Meleiro, L. A. C., Kiang, C., &MacielFilho, R. (2009). ANN-based soft-sensor for real-time process monitoring and control of an industrial polymerization process. *Computers & chemical engineering*, *33*(1), 43-49.

Graziani, S., & Xibilia, M. G. (2017, May). A deep learning based soft sensor for a sour water stripping plant. In *2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)* (pp. 1-6). IEEE.

Heddam, S., Bermad, A., &Dechemi, N. (2011). Applications of radial-basis function and generalized regression neural networks for modeling of coagulant dosage in a drinking water-treatment plant: comparative study. *Journal of Environmental Engineering*, *137*(12), 1209-1214.

Herceg, S., Andrijić, Ž. U., &Bolf, N. (2019). Development of soft sensors for isomerization process based on support vector machine regression and dynamic polynomial models. *Chemical Engineering Research and Design*, *149*, 95-103.

Inapakurthi, R. K., Miriyala, S. S., &Mitra, K. (2020). Recurrent neural networks based modelling of industrial grinding operation. *Chemical engineering science*, *219*, 115585.

Jaffel, I., Taouali, O., Harkat, M. F., &Messaoud, H. (2016). Moving window KPCA with reduced complexity for nonlinear dynamic process monitoring. *ISA transactions*, *64*, 184-192.

Jain, V., Kishore, P., Kumar, R. A., &Pani, A. K. (2017, January). Inferential sensing of output quality in petroleum refinery using principal component regression and support vector regression. In *2017 IEEE 7th International Advance Computing Conference (IACC)* (pp. 461-465). IEEE.

Jalanko, M., Sanchez, Y., Mahalec, V., &Mhaskar, P. (2021). Adaptive system identification of industrial ethylene splitter: A comparison of subspace identification and artificial neural networks. *Computers & Chemical Engineering*, *147*, 107240.

Jia, R., Mao, Z. Z., Chang, Y. Q., & Zhao, L. P. (2011). Soft-sensor for copper extraction process in cobalt hydrometallurgy based on adaptive hybrid model. *Chemical Engineering Research and Design*, *89*(6), 722-728.

Jiang, Y., Yin, S., Dong, J., &Kaynak, O. (2020). A review on soft sensors for monitoring, control, and optimization of industrial processes. *IEEE Sensors Journal*, *21*(11), 12868-12881.

Jin, H., Chen, X., Yang, J., & Wu, L. (2014). Adaptive soft sensor modeling framework based on just-in-time learning and kernel partial least squares regression for nonlinear multiphase batch processes. *Computers & Chemical Engineering*, *71*, 77-93.

Kadlec, P., &Gabryś, B. (2011). Local learning-based adaptive soft sensor for catalyst activation prediction. *AIChE Journal*, *57*(5), 1288-1301.

Kadlec, P., Gabrys, B., &Strandt, S. (2009). Data-driven soft sensors in the process industry. *Computers & chemical engineering*, *33*(4), 795-814.

Kadlec, P., Grbić, R., &Gabryś, B. (2011). Review of adaptation mechanisms for data-driven soft sensors. *Computers & chemical engineering*, 35(1), 1-24.

Kaneko, H., & Funatsu, K. (2011a). Development of soft sensor models based on time difference of process variables with accounting for nonlinear relationship. *Industrial & engineering chemistry research*, *50*(18), 10643-10651.

Kaneko, H., & Funatsu, K. (2011b). Maintenance-free soft sensor models with time difference of process variables. *Chemometrics and Intelligent Laboratory Systems*, *107*(2), 312-317.

Kaneko, H., & Funatsu, K. (2013). Classification of the degradation of soft sensor models and discussion on adaptive models. *AIChE Journal*, *59*(7), 2339-2347.

Kaneko, H., & Funatsu, K. (2014). Database monitoring index for adaptive soft sensors and the application to industrial process. *AIChE Journal*, *60*(1), 160-169.

Kaneko, H., & Funatsu, K. (2015). Moving window and just-in-time soft sensor model based on time differences considering a small number of measurements. *Industrial & Engineering Chemistry Research*, *54*(2), 700-704.

Kaneko, H., & Funatsu, K. (2016). Ensemble locally weighted partial least squares as a just-in-time modeling method. *AIChE Journal*, *62*(3), 717-725.

Kanno, Y., & Kaneko, H. (2020). Ensemble just-in-time model based on Gaussian process dynamical models for nonlinear and dynamic processes. *Chemometrics and Intelligent Laboratory Systems*, *203*, 104061.

Kano, M., & Ogawa, M. (2010). The state of the art in chemical process control in Japan: Good practice and questionnaire survey. *Journal of Process Control*, *20*(9), 969-982.

Kano, M., & Nakagawa, Y. (2008). Data-based process monitoring, process control, and quality improvement: Recent developments and applications in steel industry. *Computers & Chemical Engineering*, *32*(1-2), 12-24.

Kecman, V., Huang, T. M., & Vogt, M. (2005). Iterative single data algorithm for training kernel machines from huge data sets: Theory and performance. In *Support vector machines: Theory and Applications* (pp. 255-274). Springer, Berlin, Heidelberg.

Khosrozade, A., & Mehranbod, N. (2020). Comparison of support vector regression-and neural network-based soft sensors for cement plant exhaust gas composition. *International Journal of Environmental Science and Technology*, *17*(5), 2865-2874.

Kisi, Ö. (2006). Generalized regression neural networks for evapotranspiration modeling. *Hydrological Sciences Journal*, *51*(6), 1092-1105.

Kneale, C., & Brown, S. D. (2018). Small moving window calibration models for soft sensing processes with limited history. *Chemometrics and Intelligent Laboratory Systems*, *183*, 36-46.

Ko, Y. D., & Shang, H. (2011). A neural network-based soft sensor for particle size distribution using image analysis. *Powder Technology*, *212*(2), 359-366.

Kulkarni, S. G., Chaudhary, A. K., Nandi, S., Tambe, S. S., & Kulkarni, B. D. (2004). Modeling and monitoring of batch processes using principal component analysis (PCA) assisted generalized regression neural networks (GRNN). *Biochemical Engineering Journal*, *18*(3), 193-210.

Li, Y. G., Gui, W. H., Yang, C. H., &Xie, Y. F. (2013). Soft sensor and expert control for blending and digestion process in alumina metallurgical industry. *Journal of Process Control*, *23*(7), 1012-1021.

Li, Z., Lee, Y. S., Chen, J., & Qian, Y. (2021). Developing variable moving window PLS models: Using case of NOx emission prediction of coal-fired power plants. *Fuel*, *296*, 120441.

Liu, Y., & Chen, J. (2013). Integrated soft sensor using just-in-time support vector regression and probabilistic analysis for quality prediction of multi-grade processes. *Journal of Process control*, *23*(6), 793-804.

Liu, J., Chen, D. S., & Shen, J. F. (2010). Development of self-validating soft sensors using fast moving window partial least squares. *Industrial & Engineering Chemistry Research*, *49*(22), 11530-11546.

Liu, Y. (2017). Adaptive just-in-time and relevant vector machine based soft-sensors with adaptive differential evolution algorithms for parameter optimization. *Chemical Engineering Science*, *172*, 571-584.

Liu, Y., Gao, Z., Li, P., & Wang, H. (2012). Just-in-time kernel learning with adaptive parameter selection for soft sensor modeling of batch processes. *Industrial & Engineering Chemistry Research*, *51*(11), 4313-4327.

Liu, Y., Huang, D., & Li, Y. (2012). Development of interval soft sensors using enhanced just-in-time learning and inductive confidence predictor. *Industrial & Engineering Chemistry Research*, *51*(8), 3356-3367.

Liu, Z., Ge, Z., Chen, G., & Song, Z. (2018). Adaptive soft sensors for quality prediction under the framework of Bayesian network. *Control Engineering Practice*, *72*, 19-28.

Macias-Hernandez, J. J., Angelov, P., & Zhou, X. (2007, October). Soft sensor for predicting crude oil distillation side streams using evolving takagi-sugeno fuzzy models. In *2007 IEEE International Conference on Systems, Man and Cybernetics* (pp. 3305-3310). IEEE.

Marengo, E., Bobba, M., Robotti, E., &Liparota, M. C. (2006). Modeling of the polluting emissions from a cement production plant by partial least-squares, principal component regression, and artificial neural networks. *Environmental Science & Technology*, *40*(1), 272-280.

Matias, T., Souza, F., Araújo, R., Gonçalves, N., &Barreto, J. P. (2015). On-line sequential extreme learning machine based on recursive partial least squares. *Journal of Process Control*, *27*, 15-21.

Mattera, C. G., Quevedo, J., Escobet, T., Shaker, H. R., &Jradi, M. (2018). A method for fault detection and diagnostics in ventilation units using virtual sensors. *Sensors*, *18*(11), 3931.

Min, H., & Luo, X. (2016). Calibration of soft sensor by using Just-in-time modeling and AdaBoost learning method. *Chinese journal of chemical engineering*, *24*(8), 1038-1046.

Mitra, K., &Ghivari, M. (2006). Modeling of an industrial wet grinding operation using data-driven techniques. *Computers & chemical engineering*, *30*(3), 508-520.

Montgomery, D. C. (2017). *Design and analysis of experiments*. John Wiley & sons.

Morey, A., Pradhan, S., Kumar, R. A., Pani, A. K., Vijayan S, V., Jain, V., & Gupta, A. (2019). Pollutant monitoring in tail gas of sulfur recovery unit with statistical and soft computing models. *Chemical Engineering Communications*, *206*(1), 69-85.

Mu, S., Zeng, Y., Liu, R., Wu, P., Su, H., & Chu, J. (2006). Online dual updating with recursive PLS model and its application in predicting crystal size of purified terephthalic acid (PTA) process. *Journal of Process Control*, *16*(6), 557-566.

Napier, L. F., & Aldrich, C. (2017). An IsaMill™ Soft Sensor based on random forests and principal component analysis. *IFAC-Papers Online*, *50*(1), 1175-1180.

Napoli, G., & Xibilia, M. G. (2011). Soft Sensor design for a Topping process in the case of small datasets. *Computers & chemical engineering*, *35*(11), 2447-2456.

Ni, W., Tan, S. K., & Ng, W. J. (2011). Recursive GPR for nonlinear dynamic process modeling. *Chemical engineering journal*, *173*(2), 636-643.

Ni, W., Tan, S. K., Ng, W. J., & Brown, S. D. (2012a). Moving-window GPR for nonlinear dynamic system modeling with dual updating and dual preprocessing. *Industrial & engineering chemistry research*, *51*(18), 6416-6428.

Ni, W., Tan, S. K., Ng, W. J., & Brown, S. D. (2012b). Localized, adaptive recursive partial least squares regression for dynamic system modeling. *Industrial & engineering chemistry research*, *51*(23), 8025-8039.

Ni, W., Brown, S. D., & Man, R. (2014). A localized adaptive soft sensor for dynamic system modeling. *Chemical Engineering Science*, *111*, 350-363.

Nogueira, I., Fontes, C., Sartori, I., Pontes, K., &Embiruçu, M. (2017). A model-based approach to quality monitoring of a polymerization process without online measurement of product specifications. *Computers & Industrial Engineering*, *106*, 123-136.

Pani, A. K. (2021). Non-linear process monitoring using kernel principal component analysis: A review of the basic and modified techniques with industrial applications. *Brazilian Journal of Chemical Engineering*, 1-18.

Pani, A. K., &Mohanta, H. K. (2011). A survey of data treatment techniques for soft sensor design. *Chemical Product and Process Modeling*, *6*(1).

Pani, A. K., &Mohanta, H. K. (2014). Soft sensing of particle size in a grinding process: Application of support vector regression, fuzzy inference and adaptive neuro fuzzy inference techniques for online monitoring of cement fineness. *Powder technology*, *264*, 484-497.

Pani, A. K., &Mohanta, H. K. (2015). Online monitoring and control of particle size in the grinding process using least square support vector regression and resilient back propagation neural network. *ISA transactions*, *56*, 206-221.

Pani, A. K., &Mohanta, H. K. (2016). Online monitoring of cement clinker quality using multivariate statistics and Takagi-Sugeno fuzzy-inference technique. *Control Engineering Practice*, *57*, 1-17.

Pani, A. K., Amin, K. G., &Mohanta, H. K. (2012, July). Data driven soft sensor of a cement mill using generalized regression neural network. In *2012 International Conference on Data Science & Engineering (ICDSE)* (pp. 98-102). IEEE.

Pani, A. K., Amin, K. G., &Mohanta, H. K. (2016). Soft sensing of product quality in the debutanizer column with principal component analysis and feed-forward artificial neural network. *Alexandria Engineering Journal*, *55*(2), 1667-1674.

Pani, A. K., Vadlamudi, V. K., &Mohanta, H. K. (2013). Development and comparison of neural network based soft sensors for online estimation of cement clinker quality. *ISA transactions*, *52*(1), 19-29.

Park, S., & Han, C. (2000). A nonlinear soft sensor based on multivariate smoothing procedure for quality estimation in distillation columns. *Computers & Chemical Engineering*, *24*(2-7), 871-877.

Parvizi Moghadam, R., Shahraki, F., &Sadeghi, J. (2018). Online monitoring for industrial processes quality control using time varying parameter model. *International Journal of Engineering*, *31*(4), 524-532.

Patanè, L., & Xibilia, M. G. (2021). Echo-state networks for soft sensor design in an SRU process. *Information Sciences*, *566*, 195-214.

Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. Advances in kernal methods: support vector learning. https://dl.acm.org/doi/10.5555/299094 .299105

Poerio, D. V., & Brown, S. D. (2018). Highly-overlapped, recursive partial least squares soft sensor with state partitioning via local variable selection. *Chemometrics and Intelligent Laboratory Systems*, *175*, 104-115.

Qiao, J., & Chai, T. (2012). Soft measurement model and its application in raw meal calcination process. *Journal of Process Control*, *22*(1), 344-351.

Qin, S. J. (1998). Recursive PLS algorithms for adaptive data modeling. *Computers & Chemical Engineering*, *22*(4-5), 503-514.

Radhakrishnan, V. R., & Mohamed, A. R. (2000). Neural networks for the identification and control of blast furnace hot metal quality. *Journal of process control*, *10*(6), 509-524.

Ramli, N. M., Hussain, M. A., Jan, B. M., & Abdullah, B. (2014). Composition prediction of a debutanizer column using equation based artificial neural network model. *Neurocomputing*, *131*, 59-76.

Rogina, A., Šiško, I., Mohler, I., Ujević, Ž., &Bolf, N. (2011). Soft sensor for continuous product quality estimation (in crude distillation unit). *Chemical Engineering Research and Design*, *89*(10), 2070-2077.

Salah, B., Zoheir, M., Slimane, Z., &Jurgen, B. (2015). Inferential sensor-based adaptive principal components analysis of mould bath level for breakout defect detection and evaluation in continuous casting. *Applied Soft Computing*, *34*, 120-128.

Seo, K. W., Seo, J., Kim, K., Lim, S. J., & Chung, J. (2021). Prediction of biogas production rate from dry anaerobic digestion of food waste: Process-based approach vs. recurrent neural network black-box model. *Bioresource Technology*, *341*, 125829.

Shang, C., Gao, X., Yang, F., Lyu, W., & Huang, D. (2015a). A comparative study on improved DPLS soft sensor models applied to a crude distillation unit. *IFAC-Papers Online*, *48*(8), 234-239.

Shang, C., Huang, B., Yang, F., & Huang, D. (2015b). Probabilistic slow feature analysis-based representation learning from massive process data for soft sensor modeling. *AIChE Journal*, *61*(12), 4126-4139.

Shang, C., Huang, X., Suykens, J. A., & Huang, D. (2015c). Enhancing dynamic soft sensors based on DPLS: A temporal smoothness regularization approach. *Journal of Process Control*, *28*, 17-26.

Shao, W., & Tian, X. (2015). Adaptive soft sensor for quality prediction of chemical processes based on selective ensemble of local partial least squares models. *Chemical Engineering Research and Design*, *95*, 113-132.

Shao, W., & Tian, X. (2017). Semi-supervised selective ensemble learning based on distance to model for nonlinear soft sensor development. *Neurocomputing*, *222*, 91-104.

Shao, W., Tian, X., & Wang, P. (2015). Supervised local and non-local structure preserving projections with application to just-in-time learning for adaptive soft sensor. *Chinese Journal of Chemical Engineering*, *23*(12), 1925-1934.

Shi, X., &Xiong, W. (2018). Approximate linear dependence criteria with active learning for smart soft sensor design. *Chemometrics and Intelligent Laboratory Systems*, *180*, 88-95.

Shi, X., &Xiong, W. (2020). Adaptive ensemble learning strategy for semi-supervised soft sensing. *Journal of the Franklin Institute*, *357*(6), 3753-3770.

Shokri, S., Marvast, M. A., Sadeghi, M. T., &Narasimhan, S. (2016). Combination of data rectification techniques and soft sensor model for robust prediction of sulfur content in HDS process. *Journal of the Taiwan Institute of Chemical Engineers*, *58*, 117-126.

Shokri, S., Sadeghi, M. T., &Marvast, M. A. (2014). High reliability estimation of product quality using support vector regression and hybrid meta-heuristic algorithms. *Journal of the Taiwan Institute of Chemical Engineers*, *45*(5), 2225-2232.

Shokri, S., Sadeghi, M. T., Marvast, M. A., &Narasimhan, S. (2015). Improvement of the prediction performance of a soft sensor model based on support vector regression for production of ultra-low sulfur diesel. *Petroleum Science*, *12*(1), 177-188.

Siddharth, K., Pathak, A., &Pani, A. K. (2019). Real-time quality monitoring in debutanizer column with regression tree and ANFIS. *Journal of Industrial Engineering International*, *15*(1), 41-51.

Singh, H., Pani, A. K., &Mohanta, H. K. (2019). Quality monitoring in petroleum refinery with regression neural network: Improving prediction accuracy with appropriate design of training set. *Measurement*, *134*, 698-709.

Smola, A. J., &Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, *14*(3), 199-222.

Specht, D. F. (1991). A general regression neural network. *IEEE transactions on neural networks*, *2*(6), 568-576.

Torgashov, A., Goncharov, A., &Zhuravlev, E. (2018). Evaluation of steady-state and dynamic soft sensors for industrial crude distillation unit under parametric constraints. *IFAC-Papers Online*, *51*(18),566-571. https://doi.org/10.1016/j.ifacol.2018.09.364

Ujević, Ž., Mohler, I., &Bolf, N. (2011). Soft sensors for splitter product property estimation in CDU. *Chemical Engineering Communications*, *198*(12), 1566-1578.

Urhan, A., &Alakent, B. (2020). Integrating adaptive moving window and just-in-time learning paradigms for soft-sensor design. *Neurocomputing*, *392*, 23-37.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, *10*(5), 988-999.

Vapnik, V., Golowich, S., &Smola, A. (1996). Support vector method for function approximation, regression estimation and signal processing. *Advances in neural information processing systems*, *9*.

Wang, J., Yu, L. C., Lai, K. R., & Zhang, X. (2016). Locally weighted linear regression for cross-lingual valence-arousal prediction of affective words. *Neurocomputing*, *194*, 271-278.

Wang, Y., Chen, C., & Yan, X. (2013). Structure and weight optimization of neural network based on CPA-MLR and its application in naphtha dry point soft sensor. *Neural Computing and Applications*, *22*(1), 75-82.

Wu, F., & Chai, T. (2010). Soft sensing method for magnetic tube recovery ratio via fuzzy systems and neural networks. *Neurocomputing*, *73*(13-15), 2489-2497.

Xianghua, C., Ouguan, X., &Hongbo, Z. (2009, June). Recursive PLS soft sensor with moving window for online PX concentration estimation in an industrial isomerization unit. In *2009 Chinese Control and Decision Conference* (pp. 5853-5857). IEEE.

Xie, L., Zeng, J., & Gao, C. (2013). Novel just-in-time learning-based soft sensor utilizing non-Gaussian information. *IEEE Transactions on Control SystemsTechnology*, *22*(1), 360-368.

Xiong, W., Li, Y., Zhao, Y., & Huang, B. (2017). Adaptive soft sensor based on time difference Gaussian process regression with local time-delay reconstruction. *Chemical Engineering Research and Design*, *117*, 670-680.

Yamada, N., & Kaneko, H. (2021). Adaptive soft sensor ensemble for selecting both process variables and dynamics for multiple process states. *Chemometrics and Intelligent Laboratory Systems*, *219*, 104443.

Yamakage, S., & Kaneko, H. (2022). Design of adaptive soft sensor based on Bayesian optimization. *Case Studies in Chemical and Environmental Engineering*, *6*, 100237.

Yan, W., Shao, H., & Wang, X. (2004). Soft sensing modeling based on support vector machine and Bayesian model selection. *Computers & chemical engineering*, *28*(8), 1489-1498.

Yan, X. (2008). Modified nonlinear generalized ridge regression and its application to develop naphtha cut point soft sensor. *Computers & Chemical Engineering*, *32*(3), 608-621.

Yan, X. (2010). Hybrid artificial neural network based on BP-PLSR and its application in development of soft sensors. *Chemometrics and Intelligent Laboratory Systems*, *103*(2), 152-159.

Yao, L., & Ge, Z. (2017a). Online updating soft sensor modeling and industrial application based on selectively integrated moving window approach. *IEEE Transactions on Instrumentation and Measurement*, *66*(8), 1985-1993.

Yao, L., & Ge, Z. (2017b). Moving window adaptive soft sensor for state shifting process based on weighted supervised latent factor analysis. *Control Engineering Practice*, *61*, 72-80.

Yeo, W. S., Saptoro, A., & Kumar, P. (2019). Adaptive soft sensor development for non-Gaussian and nonlinear processes. *Industrial & Engineering Chemistry Research*, *58*(45), 20680-20691.

Yuan, X., Ge, Z., & Song, Z. (2014). Locally weighted kernel principal component regression model for soft sensing of nonlinear time-variant processes. *Industrial & Engineering Chemistry Research*, *53*(35), 13736-13749.

Yuan, X., Ge, Z., Huang, B., & Song, Z. (2016). A probabilistic just-in-time learning framework for soft sensor development with missing data. *IEEE Transactions on Control Systems Technology*, *25*(3), 1124-1132.

Yuan, X., Wang, Y., Yang, C., Gui, W., & Ye, L. (2017). Probabilistic density-based regression model for soft sensing of nonlinear industrial processes. *Journal of Process Control*, *57*, 15-25.

Yuge, N., Tanaka, K., Kaneko, H., & Funatsu, K. (2018). Selective use of adaptive models considering the prediction efficiencies. *Industrial & Engineering Chemistry Research*, *57*(42), 14286-14296.

Zhang, X., Deng, X., & Wang, P. (2020). Double-level locally weighted extreme learning machine for soft sensor modeling of complex nonlinear industrial processes. *IEEE Sensors Journal*, *21*(2), 1897-1905.

Zhou, P., Chai, T., & Sun, J. (2012). Intelligence-based supervisory control for optimal operation of a DCS-controlled grinding system. *IEEE Transactions on Control Systems Technology*, *21*(1), 162-175.

Zhu, S., Han, H., Guo, M., &Qiao, J. (2017). A data-derived soft-sensor method for monitoring effluent total phosphorus. *Chinese journal of chemical engineering*, *25*(12), 1791-1797.

List of publications

Published

1. Morey, A., Pradhan, S., Kumar, R. A., Pani, A. K., **Vijayan S, V.**, Jain, V., & Gupta, A. (2019). Pollutant monitoring in tail gas of sulfur recovery unit with statistical and soft computing models. ***Chemical Engineering Communications***, *206*(1), 69-85.
2. **Vijayan, S. V.**, Mohanta, H. K., & Pani, A. K. (2021). Support vector regression modeling in recursive just-in-time learning framework for adaptive soft sensing of naphtha boiling point in crude distillation unit. ***Petroleum Science***, *18*(4), 1230-1239.
3. **Vijayan, S. V.**, Mohanta, H. K., & Pani, A. K. (2022). Adaptive non-linear soft sensor for quality monitoring in refineries using Just-in-Time Learning—Generalized regression neural network approach. ***Applied Soft Computing***, *119*, 108546.
4. **Vijayan, S. V.**, Mohanta, H. K., Rout, B. K., & Pani, A. K. (2023). Adaptive soft sensor design using a regression neural network and bias update strategy for non-linear industrial processes. ***Measurement Science and Technology***, 34(8), 085012.

Workshop attended

GIAN course on 'Bigdata Applications in Process Operations: Modeling, Monitoring & Control', Chemical Engineering Department, National Institute of Technology, Rourkela, Odissa, India. Date: 25.06.2018 – 30.06.2018.

Biography of Student

S.Venkatavijayanis joined as PhD student in chemical engineering department, BITS-Pilani in the year 2016, under Dr. Ajaya Kr. Pani and Prof. Hare Krishna Mohanta. He completed his M.E. (Chemical Engineering) from BITS-Pilani in 2015 and B. Tech (Petrochemical Technology) from BIT campus in 2008, Bharathidasan University, Tiruchirapalli. He has over 3 years of industrial experience and 1 year of research experience.

Biographies of Supervisor and Co-supervisor

Dr. Ajaya Kumar Pani is Assistant Professor in the Department of Chemical Engineering in BITS-Pilani (Pilani Campus), Rajasthan. He has done his graduation (Chemical Engineering) from NIT, Rourkela, M. Tech from Institute of Technology, Banaras Hindu University (Presently IIT-BHU), Varanasi and PhD., from BITS Pilani in 2015. His research interests include data-driven process modeling using artificial intelligence, multivariate statistics and machine learning techniques for soft sensing of industrial process outputs and process fault detection. He has several publications in international/national journals and reviewer of several international journals such as Journal of Process Control, Industrial & Engineering Chemistry Research, Computers and Chemical Engineering, Powder Technology, Chemical Engineering Communications. Chemical & Biochemical Engineering Quarterly etc.,

Prof. Hare Krishna Mohantais Associate Professor in the Department of Chemical Engineering in BITS-Pilani (Pilani Campus), Rajasthan. He has over 24 years of teaching experience and 1-year industrial experience. He has obtained his B.E. (Chemical

Engineering) degree in 1995 from NIT Rourkela, M.Tech. (Chemical Engineering) in 1998 from IIT Kanpur, and PhD., in 2006 from BITS Pilani. He is a Life Associate Member of Indian Institute of Chemical Engineers (IIChE) and a Member of Institution of Engineers (India). He has several publications in international and national journals. His areas of research include wavelet-based process identification & control, design of soft sensors for process monitoring & control, reactive distillation design & control, nonlinear control & state estimation, catalytic pyrolysis of hydrocarbons besides modeling & simulation. He is a reviewer of journals such as International Journal of Automation and Control and Journal of Engineering and Computer Innovations.