UNIVERSIDADE D
COIMBRA

Joana da Silva Rosa Leiria

DATA ENTRY ERRORS DETECTION FOR QUALITY CONTROL VARIABLES USING DATA-DRIVEN MODELS

Dissertation in the scope of the Master's Degree in Engineering Physics, supervised by Professor Doctor Jérôme Amaro Pires Mendes and submitted to the Department of Physics of the Faculty of Sciences and Technology of the University of Coimbra.

February 2024

Faculty of Sciences and Technology
University of Coimbra

DATA ENTRY ERRORS DETECTION FOR QUALITY CONTROL VARIABLES USING DATA-DRIVEN MODELS

Joana da Silva Rosa Leiria

Dissertation in the scope of the Master's Degree in Engineering Physics, supervised by Professor Doctor Jérôme Amaro Pires Mendes, and submitted to the Department of Physics of the Faculty of Sciences and Technology of the University of Coimbra.

UNIVERSIDADE D
COIMBRA

Acknowledgements

First of all, I would like to thank my advisor, Professor Jérôme Mendes, and my colleague Rodrigo Salles for all the support and guidance during this dissertation. For the Professor in particular, for encouraging me to continue even when my availability was limited.

To my parents, my grandmother Alice and my grandfather António, and my little brother, for all the care, sometimes in the shape of home-made meals brought to me when I was overwhelmed with work, patience and understanding when I could not visit. You were always there for me, in the best and worst moments, and always trusted and encouraged my decisions.

To my friends, Mário and Rafaela, for all the help, writing sessions together, laughter and smiles in the middle of all the stress and all the good moments that helped me get through this last year. I also owe a big thank you to Disa, Beatriz, and Miguel who were always ready to receive daily updates of my dissertation, listen to complaints, and reassure me.

To my work colleagues and friends, especially Bárbara and Raquel, for always having my back and for cheering me up everyday of the week.

To μ.

Last but not least, to Zé, for all the small and big gestures that helped me keep going, for all the love and affection that charged me up and for being there everyday, placing a little stool so that I could reach and dream higher. Having you by my side made this journey not only easier by way more amusing.

This work was executed under the project "InGestAlgae - Plataforma Inteligente de Gestão da Produção de Microalgas", with reference CENTRO-01-0247-FEDER-046983, co-financed by Fundo Europeu de Desenvolvimento Regional (FEDER), through the Programa Operacional da Região Centro (CENTRO 2020).

Abstract

In today's industrial environments, quality control is essential for ensuring product reliability and operational effectiveness. Despite the widespread of automatic data collection and the use of data-driven models in quality processes, certain key variables still require manual entry due to laboratory analysis requirements, introducing errors stemming from human involvement. While research has addressed data entry errors in other domains, industrial contexts present distinct challenges that require data-driven solutions, in contrast with the manual methods presented in the literature.

This dissertation focuses on the development and application of two methodologies, one leveraging Soft Sensors (SS) and another based on Principal Component Analysis (PCA), to detect data entry errors in quality control variables. The developed work resulted in a framework for designing Soft Sensors, that stands out for the implementation of feature expansion to introduce non-linearity in the field variables and for the implementation and comparison of several variable selection methods and regression models.

Furthermore, the two methodologies to detect data entry errors were developed and tested in three different datasets with laboratory data from industrial facilities. Through a comprehensive characterization of entry data errors across various categories, such as blank spaces, doubles, measurement errors, order errors and extra number errors, this study provides valuable insights into the capabilities and limitations of the developed methodologies. The performance of the SS-based and the PCA-based methodologies was compared using classification metrics, such as precision, sensitivity, F1-score and specificity.

The performed tests revealed that the PCA-based methodology may not be adequate for all datasets as it performs poorly for cases with low variability within the target variable. On the other hand, the methodology leveraging Soft Sensors presented good overall results with exceptional performance for blank spaces and order errors. A common difficulty in detecting doubles was detected in both methodologies. This dissertation culminated in the recommendation of the SS-based approach for the implementation in real industrial scenarios, given its best overall performance and easy interpretability by the operator (relevant factor to guarantee the operator cooperation in real setups).

Keywords: data entry errors, error detection, PCA, Soft Sensors, industrial application, quality control.

Resumo

Nos contextos industriais atuais, o controlo de qualidade é essencial para garantir a confiabilidade dos produtos e a eficácia operacional. Apesar da disseminação da recolha automática de dados e do uso de modelos de inteligência computacional nos processos de qualidade, certas variáveis-chave ainda requerem entrada manual devido a requisitos de análise laboratorial, o que introduz erros decorrentes do envolvimento humano. Enquanto os erros de inserção de dados já foram estudados noutras áreas, os contextos industriais apresentam desafios distintos que requerem soluções baseadas em dados, em contraste com os métodos manuais apresentados na literatura.

Esta dissertação foca-se no desenvolvimento e aplicação de duas metodologias, uma aproveitando Sensores Virtuais e outra baseada na Análise de Componentes Principais, para detetar erros de inserção de dados em variáveis de controlo de qualidade. O trabalho desenvolvido resultou num *framework* para o design de Sensores Virtuais, que se destaca pela implementação de expansão de variáveis para introduzir não-linearidade nas variáveis de campo e pela implementação e comparação de vários métodos de seleção de variáveis e modelos de regressão.

Além disso, as duas metodologias usadas para detetar erros de inserção de dados foram desenvolvidas e testadas em três *datasets* diferentes com dados de laboratório de instalações industriais. Através de uma caracterização abrangente de erros de inserção de dados em várias categorias, como espaços em branco, duplos, erros de medição, erros de ordem e números extra, este estudo fornece perceções valiosas sobre as capacidades e limitações das metodologias desenvolvidas. O desempenho das metodologias foi comparado usando métricas de classificação, como precisão, sensibilidade, *F1-score* e especificidade.

Os testes realizados revelaram que a metodologia baseada na Análise de Componentes Principais pode não ser adequada para todos os *datasets*, pois apresenta baixo desempenho para casos com baixa variabilidade na variável alvo. Por outro lado, a metodologia que recorre a Sensores Virtuais apresentou bons resultados gerais com desempenho excepcional para espaços em branco e erros de ordem. Uma dificuldade comum na deteção de duplos foi detetada em ambas as metodologias. Esta dissertação culminou na recomendação da abordagem baseada em Sensores Virtuais para a implementação em cenários industriais reais, dada sua melhor performance geral e fácil interpretabilidade pelo operador (fator relevante para garantir a cooperação do operador em cenários reais).

Palavras-chave: erros de inserção de dados, deteção de erros, Análise de Componentes Principais, Sensores virtuais, aplicação industrial, controlo de qualidade.

Contents

List of Acronyms

API Application Programming Interface.

DL Deep Learning.

EN Elastic Net.

FE Feature Expansion.

FN False Negatives.

FP False Positives.

GMM Gaussain Mixture Models.

GMR Gaussian Mixture Regression.

LASSO Least Absolute Shrinkage and Selection Operator.

MI Mutual Information.

MLR Multiple Linear Regression.

MSE Mean Squared Error.

NRMSE Normalized Root Mean Squared Error.

PC Pearson's Correlation.

PCA Principal Components Analysis.

PCs Principal Components.

PMFD Process Monitoring and Fault Detection.

RBF Radial Basis Function.

RMSE Root Mean Squared Error.

RR Ridge Regression.

SPE Squared Prediction Error.

SPSS Statistical Package for the Social Sciences.

SS Soft Sensor.

SVD Singular Value Decomposition.

SVR Support Vector Regression.

TN True Negatives.

TP True Positives.

WTP Wastewater Treatment Plant.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem statement and motivation

The digital transformation within industrial sectors is now indispensable, offering solutions to numerous challenges towards more sustainable and greener operations. Among these challenges, quality control stands out as a critical pillar in industrial processes, ensuring product reliability, employee safety, and operational efficiency, while also contributing to energy saving and cost reduction. However, the evolving complexity of industrial operations has introduced new layers of difficulty to quality control processes. In response, technology and data have emerged as powerful tools to address this challenge. Industrial facilities today leverage data-driven models as indispensable techniques for quality control across their operations [1], integrating various sensors and amassing vast amounts of data to enable effective process monitoring and error detection. Yet, despite these technological advancements, certain crucial quality control variables need manual acquisition through laboratory analysis. In these instances, operators typically perform measurements and input them manually into systems such as spreadsheets or online platforms, thereby increasing the risk of errors due to human involvement.

Indeed, research has already addressed the impact of and strategies to prevent data entry errors in social science research environments [2] and clinical settings [3]. However, there is a notable gap in research concerning how to prevent such errors in industrial contexts. Moreover, existing techniques predominantly rely on manual methods, such as double entry strategies or visual checking techniques [4], with no exploration of data-driven models. These manual techniques have inherent limitations, primarily relying on human intervention and often overlooking certain types of errors, such as measurement errors arising from improper use of measurement machinery.

Although there already exist systems in industrial settings that would detect an entry data error as a fault in the middle of a process, preventing errors from being inserted into the system at the insertion stage is preferable as it avoids downstream repercussions and minimizes the potential for cascading effects throughout the production process. Upfront detection at the data entry stage reduces the likelihood of erroneous data propagating further, mitigating the need for costly corrective measures. Thus, it is crucial to identify these errors, particularly for quality control variables.

Acknowledging the research gap highlighted earlier and the criticality of detecting data entry errors, particularly for quality control variables, this work aims to deploy computational intelligence techniques for entry data error detection. To accomplish this goal, the approach involves leveraging established data-driven models commonly utilized

in industrial settings for process monitoring, namely Soft Sensor (SS) and Principal Components Analysis (PCA). Through the implementation of these techniques, the model should autonomously detect erroneous data entries as operators input measurements into a spreadsheet or a designated system. Upon identification of a potential error, the system promptly notifies the operator in real-time, allowing for immediate correction.

It is important to notice, it is not only the technical challenge that needs to be faced. It is equally imperative to consider the human factor in deploying these models. As this methodology is intended to assist operators in preventing erroneous data entries, it is essential for operators to trust and embrace the new system. In fact, Industry 5.0 envisions a collaborative relationship between humans and smart machines, where human expertise is augmented by technological capabilities [5]. For this vision to materialize, operators must have confidence in the efficacy of these systems. Simply providing new technologies is not sufficient, operators must comprehend how these innovations can enhance their work processes. Failure to foster this understanding may lead to resistance and reluctance to adopt the new tools, potentially hindering their effectiveness [6].

1.2 Objectives

The primary objective of this work is to investigate the implementation of two different approaches, one leveraging SS and and another based on PCA, for the detection of entry data errors related to quality control variables. To achieve this goal, specific objectives are outlined as follows:

- Characterize types of detectable errors: understand what types of data entry errors can be effectively detected by the implemented data-driven models. This involves discerning if the models are capable of detecting doubles, errors arising from erroneous measurements, or simply typing errors, such as inserting an extra number.

- Compare the performance of the two different approaches. The first is based on a data-driven regression model using the concept of SS. Other, not based on a regression model, is a PCA-based methodology. And conduct a comparative analysis to evaluate their respective performance in detecting data entry errors.

- Recommend for implementation: based on the findings and insights obtained, and having into account the interpretability of both approaches, define the best suitable methodology and parameters to be implemented in practice.

1.3 Main contributions and developed work

In order to achieve the defined goals, the developed work resulted in the following key contributions:

- Proposal of a framework to design a Soft Sensor, implementing several variable selection methods and regression models for Soft Sensor applications, culminating in an accepted conference paper titled “Soft Sensors for Industrial Applications: Comparison of Variable Selection Methods and Regression Models”, presented at the 2023 International Conference on Control, Automation and Diagnosis (ICCAD) [7] (see Appendix A). The proposed framework uses the following methods/steps:

  - Feature Expansion: to introduce nonlinearity in the models, the dataset was extended by adding the square, the inverse, and the root mean squared of each input variable, as well as the product between each two input variables.
  - Input variable selection: three variable selection methods were implemented. The well-known Pearson's correlation that measures the linear dependence between pairs of variables. The Mutual Information (MI) which measures the dependency between variables taking into account the probabilistic distribution of the variables. The fastTracker algorithm, a recent and efficient real-time algorithm that tracks the process behavior's changes by measuring sensitivity indices between variables.
  - Regression model: to implement the SS were used Multiple Linear Regression (MLR), Ridge Regression (RR), Least Absolute Shrinkage and Selection Operator (LASSO), Elastic Net (EN), Support Vector Regression (SVR), and Gaussian Mixture Regression (GMR) models.

  The main contributions of this work are the implementation of the Feature Expansion in the framework, and the implementation and comparison of several regression models and variable selection methods. In order to test the proposed framework, 3 datasets were used, two benchmark datasets and a third dataset provided by a cement industry. Each model for each dataset was learned/tested with 8 different sets of input variables (from the variable selection methods) for two datasets and 9 sets of input variables for another dataset. Each learning process was executed 30 times. Thus, 5250 tests were performed.

- Development and adaptation of two data-driven methodologies (SS and PCA) to detect data entry errors, in contrast with the traditional manual methods.

- Comprehensive characterization of entry data errors, offering valuable insights into the capabilities and limitations of the developed methodologies. Five error categories were studied: blank spaces, order errors, doubles, measurement errors and extra number errors.

- Comparative evaluation of the two methodologies across three laboratory datasets, assessing their performance using metrics such as precision, sensitivity, f1-score, and specificity.

  In order to test the two proposed methodologies, 3 datasets were used. For each dataset, 6 variations were created: one for each error category, containing exclusively that specific error type, and a general simulation comprising all error categories in equal proportion. To test the SS-based methodology, 4 thresholds (SS_{th}) were tested. Hence, resulting in a total of 90 tests performed. Furthermore, for specific error categories (measurement errors, order errors and extra number errors), a deeper analysis was conducted, to study how the subtypes of errors impacted the sensitivity of the model.

- Provision of recommendations on the most suitable method for implementation in real industrial scenarios, considering interpretability to ensure the operator's cooperation;

1.4 Project structure

The document is structured into seven chapters, each focusing on distinct aspects related to the detection of entry data errors and the utilization of SS and PCA methodologies. The organization of the remaining chapters is outlined as follows:

- Chapter 2 provides a comprehensive literature review on the impact of entry data errors and methodologies to prevent them, as well as an overview of SS application in quality control and, lastly, an introduction of PCA as a Process Monitoring and Fault Detection (PMFD) method.
- Chapter 3 delves into the steps to design a Soft Sensor. This includes data preprocessing, various variable selection methods, regression models, and a framework for designing and testing the Soft Sensor framework.
- Chapter 4 offers a thorough examination of Principal Components Analysis principles, followed by a description of a commonly used methodology that leverages PCA for PMFD.
- In Chapter 5, the process of simulating and inserting data entry errors into datasets is outlined, along with the metrics devised to evaluate SS performances and the ones chosen to assess how well the methodologies captured entry data errors. Additionally, the chapter presents the methodologies developed based on SS and PCA for detecting entry data errors.
- Chapter 6 focuses on the datasets utilized and the experimental tests conducted to assess the performance of the developed methodologies. It includes the tests results as well as a discussion on their implications.
- Finally, Chapter 7 offers concluding remarks, providing a comprehensive evaluation of the developed methodologies, recommendations for their practical implementation, and insights into potential limitations and future research.

Chapter 2

Literature review on data entry errors and concepts

This chapter presents the background for the work developed and the literature review, being organized in three sections. Section 2.1 delves into well-known methods for detecting errors in entry data and discusses their recognized impacts on data quality. Following this, Section 2.2 and Section 2.3 provide backgrounds on the specific methodologies, Soft Sensor and PCA, respectively. Both of these methodologies have been used in industrial contexts for quality control and fault detection and were chosen as foundational frameworks for the error detection methodologies developed in this work.

2.1 Data entry errors

Basic data entry errors wield significant impact, both in applied contexts and research domains. These inaccuracies can not only compromise reliability but also diminish statistical power [8]. Studies even suggest that such errors can make significant findings less probable [9], potentially revoking moderate correlations and even invalidating statistical analyses entirely. In industrial settings, data entry errors can incur substantial costs, while in medical scenarios, for example, they may lead to incorrect treatments, posing severe risks to patients. In research, for instance, inserting 55 instead of 5 can transform a statistically significant correlation into a near zero and non-significant correlation [2].

Given the potential devastating consequences of data entry errors, numerous efforts have emerged to mitigate their effects. One prevalent method involves data cleaning [10], wherein samples undergo a robust analysis post-entry. Various techniques, such as histogram analysis, boxplots, and z-score analysis, are employed to detect outliers. However, these methods are not flawless, and detecting errors post-entry may be too late, especially in quality control scenarios. Consequently, significant efforts have been dedicated to preventing errors outright, leading to innovations like alternative keyboard designs [11] and innovative data input methods such as voice recognition systems [12, 13], optical character recognition (OCR) [14], and barcodes [15].

Some of these technologies have demonstrated reduced error rates compared to manual data entry. For instance, a study by Smyth et al. [3] revealed that automated procedures, utilizing optical scanning, achieved an accuracy rate of 99.98% compared to 98.76% for manual entry methods. However, adopting such technologies often entails substantial initial investments. Costs include equipment procurement, software acquisition, operator training, and ongoing maintenance expenses.

Thus, several manual data entry methods have been studied due to their cost-effectiveness and easy application. Among the most commonly employed and researched methods are single entry with visual checking [2], reading aloud [4], and double entry [16]. It's worth noting that many of these methods were originally designed for scenarios where data is already recorded on paper (e.g. surveys) and needs to be transferred into a digital system or spreadsheet, often originating from social sciences research or clinical contexts [4, 3, 8]. In a laboratory setting, the paper record could be replaced with data directly obtained from the measurement equipment.

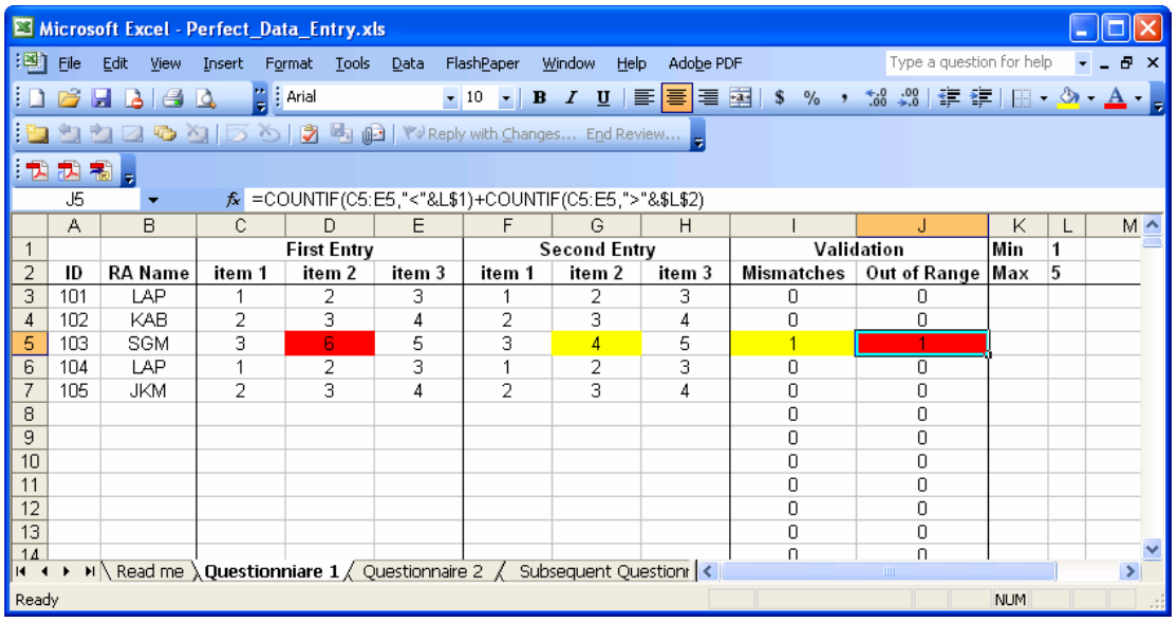
In the visual checking method, an operator has the data recorded on paper and then enters it into a digital format, typically a spreadsheet like Excel or a statistical program, like SPSS. After data entry, they compare their entries with the original paper records, correcting any discrepancies they find. However, this method is prone to errors due to the need for close attention and can be tedious. In fact, a study from K. Bachard and L. Pace [2] showed that visual checking was no more accurate than a single entry (just inserting the value with no verification of the paper sheet).

The read-aloud technique presents a variation of the previous method. In this approach, a second individual reads the entries aloud while the first person verifies by comparing the spoken value with the one recorded in the original paper sheet to ensure data accuracy [4]. Another variant involves the program itself reading out the entered data audibly. Both variations show promising performance compared to single entry. In a study by Kawado et al. [4], the read-aloud method with a second operator detected approximately 60% of errors occurring in a single data entry. It was also noted that the method performed marginally better with two operators compared to having the software read the data post-input.

Lastly, double entry stands as the recommended gold standard method for data entry, involving the insertion of data twice [17]. Typically, a spreadsheet is set up to receive two entries and compare them instantly: if a mismatch occurs, the cell highlights in red, drawing the operator's attention to verify and rectify the discrepancy. Additionally, this method can include range limits in cell formatting to warn the operator if an inserted value falls outside the allowable or predictive interval. Some researchers even employ a second operator in the process, where one person registers the value in one spreadsheet, and another person enters the same measurements in the second spreadsheet. All these alerts can be easily configured within software like Excel. However, other software solutions with these features tend to be costly [2].

Among the three methods discussed, double entry emerges as the most effective. In the study by Kawado et al. [4], double entry exhibited significantly better results compared to the reading aloud method. With two operators, double entry achieved an 88.3% detection error rate, whereas with a single operator, it achieved 69.0%. In contrast, the reading aloud method resulted in a detection error rate of 59.5% with different operators and 39.9% with a single operator. Similarly, in a study by Barchard et al. [8], double entry outperformed visual checking. In this study, 77.4% of participants using the double entry condition achieved perfect accuracy, whereas only 17.1% using visual checking did. Participants using double entry made an average of 0.34 errors across 1260 entries, while those using visual checking made an average of 10.39 errors. Moreover, participants in the single entry condition made an average of 12.03 errors. The study also found that participants using the double entry method were significantly more likely to obtain correct values for statistical analyses such as correlation and t-tests.

After extensive analysis, it becomes evident that the methods discussed previously



			First Entry			Second Entry			Validation		Min	1
ID	RA Name	item 1	item 2	item 3	item 1	item 2	item 3	Mismatches	Out of Range	Max	5
101	LAP	1	2	3	1	2	3	0	0		
102	KAB	2	3	4	2	3	4	0	0		
103	SGM	3	6	5	3	4	5	1	1		
104	LAP	1	2	3	1	2	3	0	0		
105	JKM	2	3	4	2	3	4	0	0		

Figure 2.1: Double-entry system in Excel (taken from [2]).

are either not applicable to laboratory or industrial contexts — for instance, barcodes are not suitable — or they are expensive and necessitate operator training, such as the OCR method. Even manual entry data methods, like double entry, while demonstrating effectiveness, still exhibit shortcomings. For instance, if an operator erroneously records a measurement and inputs it into the system, the error may go undetected. Additionally, valid value ranges may not be informative if a variable experiences significant fluctuations. For example, if a variable ranges from 2 to 80, an error inserting 15 instead of 51 in both entries would not be flagged. This underscores a research gap in understanding these errors within an industrial context and utilizing data-driven methods for error detection in entry data.

As a result, the subsequent sections introduce two computational intelligence methods commonly used in industrial settings, which could be leveraged to identify entry errors effectively.

2.2 Soft Sensors for quality control

In the context of digitized industries aiming for sustainability, predictive models have become crucial for inferring quality variables [18]. Soft Sensor, serving as inferential models, predict physical quantities online (eg. prediction of variables that cannot be automatically measured and are obtained using a laboratory analysis) in industrial processes based on field variables, typically obtained through sensors, and knowledge [19, 20]. SS find widespread application in quality control and in comprehending complex processes beyond the capabilities of models based solely on process knowledge [21]. Their significance has been recognized in various industries over the years [22, 18, 20, 23], contributing to advancements in quality monitoring and process understanding.

The first step of SS design involves selecting data from the plant's operating system (field variables) targeted for the model application. According to Kadlec [24], these are the most common challenges faced at this stage:

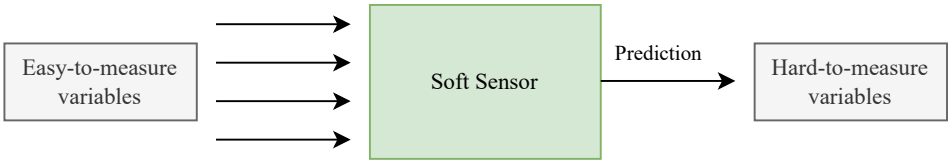


Figure 2.2: Illustration of how Soft Sensors work.

- Missing values may manifest as ∞, zeros, or constant values unrelated to the measured quantity.
- Outliers are values outside typical ranges or violating physical and sensor limitations, such as velocities exceeding the speed of light.
- Drifting data can arise from environmental changes or internal processes like sensor abrasion.
- Data collinearity, stemming from redundant sensor arrangements, can create a "data-rich but information-poor" environment [25], increasing the model complexity.
- Sampling rates and measurement delays result from acquiring data at different time rates, forcing sensor synchronization.

Lastly, the prevalent lack of data in some industrial settings, often tied to the preceding challenges, significantly impacts model training and testing [26]. Data is collected and pre-processed with the objective of dealing with the issues above presented. To do that strategies like outlier detection, removing missing value samples and deleting non numeric values are applied.

After processing the data, the next step involves selecting the field variables to be used in the model. As previously mentioned, collinearity might be present in the collected data, and the model is specifically interested in variables that can elucidate the variability of the target variable [19]. While the selection could be informed by the understanding of the physical process, these processes are often too intricate for someone to accurately identify the most pertinent variables. Consequently, computational and mathematical methods are commonly employed to discern the most relevant variables.

Subsequently, the next step implicates selecting and training a model. Typically, a portion of the dataset is reserved for the validation step, where various model options are assessed. The models can be categorized into linear and nonlinear. It is often advisable to initially explore linear models such as MLR or RR due to their enhanced interpretability [19]. Despite the growing popularity of Deep Learning (DL) models, particularly with increased data availability and their high accuracy, their application in the Soft Sensor context is restricted by data limitations in industrial environments and the resulting loss of interpretability [26].

Next, the model undergoes validation using a distinct test dataset, employing metrics like the Root Mean Squared Error (RMSE) or the Determination Coefficient (R^2). If the metrics fail to align, it signals that the model is not well-suited to the dataset. This discrepancy may also prompt a reassessment of the pre-processing step, as poor results can result from outliers and other data-related issues.

Finally, the SS necessitates ongoing maintenance and adaptation, especially in the event of process changes or data drifting, as previously mentioned.

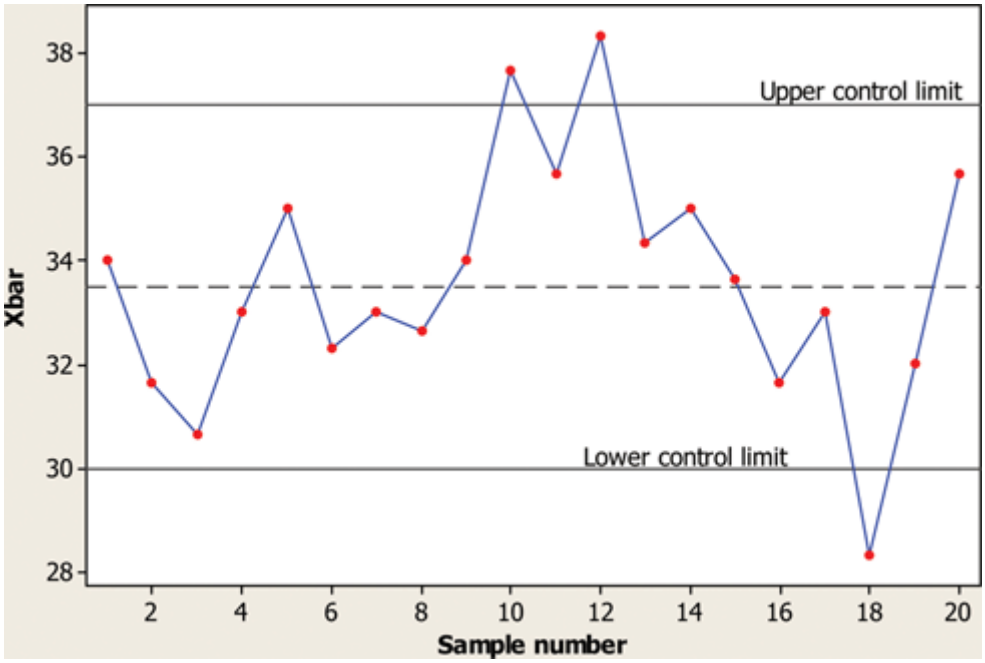


Figure 2.3: Example of a Shewhart monitoring chart (taken from [30]).

Chapter 3 gives more details on how SS are designed and presents several variable selection methods and regression models.

2.3 PCA for process monitoring and fault detection

Process Monitoring and Fault Detection plays an essential role in industrial settings, ensuring product quality, uninterrupted production, and operator and machinery safety [27]. Monitoring was defined by the SAFEPROCESS Technical Committee as "a continuous real-time task of determining the conditions of a physical system by recording information, recognizing, and indicating anomalies in behavior" [28]. Similarly, a fault is understood as "an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition". Therefore, process monitoring involves continuously observing and analyzing various variables and conditions in a production process to ensure they fulfill defined standards. Additionally, when a fault is detected, there needs to be performed a fault diagnosis which involves isolating and identifying the fault and its implications.

Process monitoring has its origins in the 1930s with the development of the Shewhart monitoring charts [29], as represented in Figure 2.3. Over time, manufacturing facilities and equipment have become increasingly complex, requiring advanced PMFD methods. This need has led to the development of high-end data acquisition systems capable of capturing numerous field variables at minute frequencies, resulting in vast multivariate real-time databases [31]. However, without proper processing, these databases often suffer from being "data-rich" but "information-poor". To address this challenge, various multivariate statistical approaches have been studied and applied, with a particular emphasis on Principal Components Analysis. PCA aims to reduce the dimensionality of the collected data by projecting it into a new subspace that only encapsulates the most relevant information.

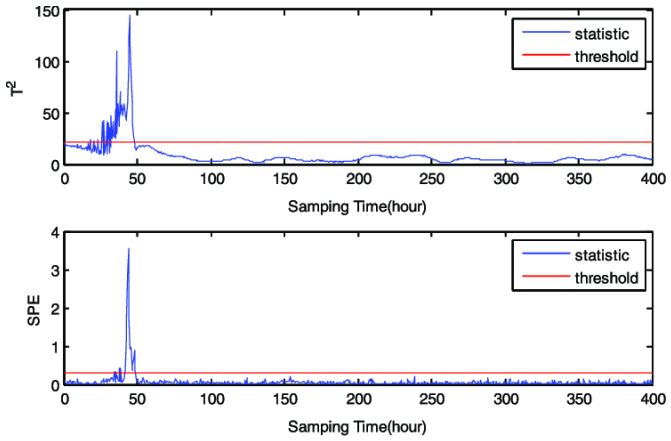


Figure 2.4: Example of a monitoring chart for PCA applied to the fed-batch penicillin fermentation process (taken from [40]).

In fact, PCA and its variations have been proven to perform well in several industries for PMFD [32, 33, 34, 35, 36, 37, 38]. Usually, PCA is applied by projecting data collected from industrial facilities into a reduced subspace defined by Principal Components (PCs). This new subspace encapsulates the most significant information and is constructed using historical plant data. Periodically, this subspace is updated through PCA or its variants, tailored to the specific characteristics of the data [31, 39]. This initial fault-free state serves as the basis for establishing reference metrics, such as the Squared Prediction Error (SPE) and Hotelling's T^2, which are utilized for sample classification. Upon the entry of a new sample into the system, it is projected onto the new subspace, and the reference metrics are computed. Samples that conform to the reference metrics are classified as fault-free, while those that deviate from the reference are flagged as faulty.

Figure 2.4 illustrates a monitoring chart utilizing PCA alongside SPE and T^2 to detect faults in a fed-batch penicillin fermentation process, a well-known benchmark process [40]. Faults are identified when both the statistical line of the SPE chart and the T^2 chart intersect the predetermined threshold.

Chapter 4 provides a comprehensive exploration of PCA and a structured methodology for fault detection.

Chapter 3

Soft Sensors for target variable prediction

SS are inferential models widely employed for estimating physical quantities in industrial settings. Their proven and extensively studied effectiveness in predicting laboratory variables [41] makes them a natural choice for the methodology used for data entry error detection described in Figure 3.1.

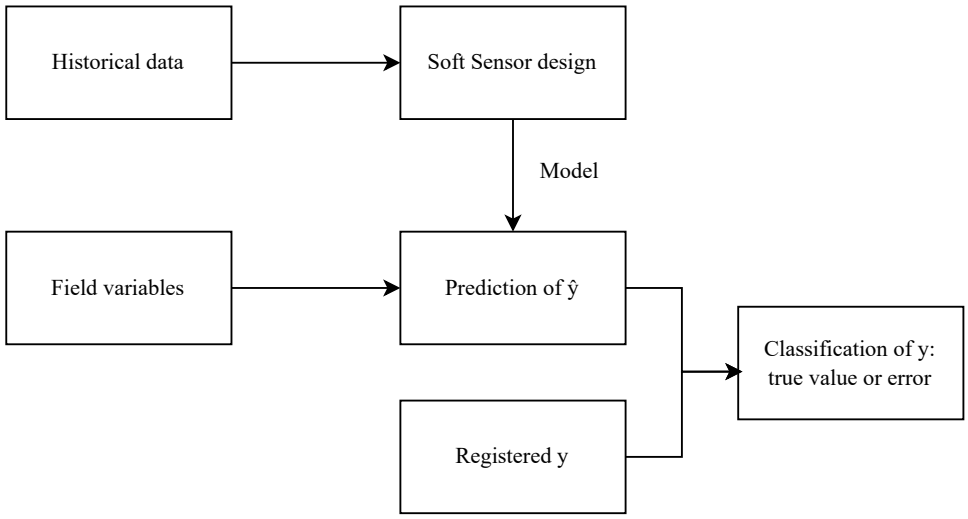


Figure 3.1: Methodology to detect data entry errors using SS.

SS uses historical data, encompassing information from the plant's operating system and laboratory records, to train and validate a model aiming to accurately predict the target variable $\hat{y}$. Then, when an operator makes a measurement of y, the model is employed with the corresponding field variables to predict $\hat{y}$. A comparison is then made between the predicted and the registered values, classifying the operator-inserted value as either an "error" or a "true value". Further details on this methodology are presented in Chapter 5.

This chapter explores the application of Soft Sensor as a predictive tool for physical quantities in industrial processes. The chapter unfolds in the following sections: Section 3.1 brings the adopted notation; Section 3.2 provides an overview of the steps to design of the SS; Section 3.3 introduces the data pre-processing methods applied; Section 3.4 presents the implemented variable selection methods on the proposed SS design; Section 3.5 presents the implemented regression models on the proposed SS design, and finally,

Section 3.6 introduces the framework used for SS design. The algorithm employed in this chapter for SS design, along with the variable selection methods and regression models presented herein, contributed to a comparative study resulting in a conference paper [7] [see Appendix A].

3.1 Notation

In this work the following notation is employed:

- Variables and their corresponding values are designated by capital and lowercase letters, respectively. For example, variable A is associated with the value a.
- Matrices and vectors are indicated by bold capital and lowercase letters, such as $\mathbf{A} = [a_{k,j}]_{K \times m}$ and $\mathbf{a} = [a_1, \ldots, a_m]$, respectively.
- $\hat{Y}$ denotes the prediction for the target variable Y, and $X = X_1, \ldots, X_m$ represents the input variables with values $x_{k,j} \in X_j$ where $k = 1, \ldots, K$ and $j = 1, \ldots, m$. Additionally, $\mathbf{x}_k = [x_{k,1}, \ldots, x_{k,m}]$, $\mathbf{X} = [x_{k,j}]_{K \times m}$, and $\mathbf{Y} = [y_k]_{K \times 1}$. The regression coefficients are denoted by β_0, and $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_m]^T$.

3.2 Soft Sensors design

As explained in Chapter 1, in industrial processes, certain critical quality control variables cannot be measured by conventional physical sensors. Instead, they are analyzed in a laboratory, which involves costs and human resources, as an operator is frequently required for the measurements. This results in data being obtained at a low frequency, specially when compared to physical sensors [22]. For instance, in the petrochemical industry, laboratory samples can take half an hour to one hour to be analyzed, with only one or two samples taken per day [42].

The main steps of the design of the Soft Sensor to predict a target variable are summarized using the diagram depicted in Figure 3.2.

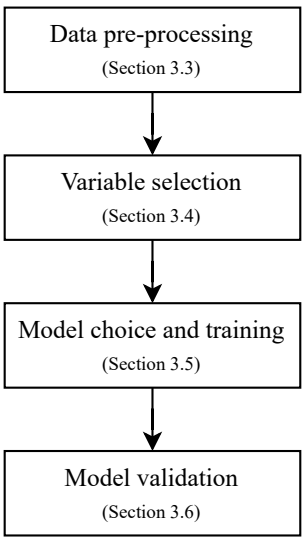


Figure 3.2: SS design.

3.3 Data pre-processing

During the data pre-processing stage, the goal is to ready the data for subsequent steps, enhancing its suitability for models and improving their effectiveness. This involves removing empty or partially filled columns, and eliminating non-numeric data as well as outlier samples. Additionally, variable names undergo a transformation. Field variables are renamed as $X_1, \ldots, X_m$, with m representing the total number of field variables, while the target variable is designated as Y.

In this work, Feature Expansion (FE) was incorporated into the Soft Sensor design as a means to introduce non-linearity to the model. FE is a technique employed to introduce non-linearity to models by transforming the original feature space into a new space. In this process, various operations are applied to each original input variable vector $\mathbf{x}_j$ and the dataset is expanded by adding the inverse, the root mean squared and the square of $\mathbf{x}_j$ as well as the product between each two input variables:

$$\mathbf{x}_j \rightarrow \mathbf{x}_j, \tfrac{1}{\mathbf{x}_j}, \sqrt{\mathbf{x}_i}, \mathbf{x}_j^2 \text{ and } \mathbf{x}_j \cdot \mathbf{x}_i$$

Where $\mathbf{x}_i$ is a input variable with $j = [1, \ldots, m]$ and $i = [1, \ldots, m]$, excluding j. The parameter m denotes the total number of input variables. To prevent data inaccuracies, the inverse transformation is not applied to variables $\mathbf{x}_j$ with values equal to zero, and the root mean squared transformation is also omitted for variables with negative values.

3.4 Variable selection methods

The effectiveness of a Soft Sensor heavily relies on the choice of variables used to infer the target variable. In order to test the performance of different variable selection methods, two well-known methods and a recent method were used for the selection of input variables:

- Pearson's correlation, a well-known method, gauges linearity between variables [43]. Described in Subsection 3.4.1.
- Mutual Information (MI) assesses the probabilistic distribution of variables to measure their dependency [44]. Described in Subsection 3.4.2.
- fastTracker [45] is a recent and efficient algorithm that identifies process variability using sensitivity indices between variables. Described in Subsection 3.4.3.

This section proceeds with the description of the methods employed for variable selection.

3.4.1 Pearson's correlation

Pearson's correlation, denoted by r, measures the linear dependency between variable pairs, providing insights into the strength of their relationship [43]. The values of r span from -1 to 1, with $r = 1$ signifying a perfect positive correlation and $r = -1$ indicating a perfect negative correlation. A value of $r = 0$ suggests that the analyzed variables lack correlation [46]. The Pearson's correlation between variables X_j and Y is mathematically expressed by the Equation (3.1):

$$r = \frac{\Sigma_{k=1}^{K}[(x_{k,j} - \mu_{X_j})(y_k - \mu_Y)]}{\sqrt{\Sigma_{k=1}^{K}(x_{k,j} - \mu_{X_j})^2}\sqrt{\Sigma_{k=1}^{K}(y_k - \mu_y)^2}}, \tag{3.1}$$

where μ_{X_j} and μ_Y are the arithmetic means of the variables X_j and Y, respectively, $j = [1, \ldots, m]$, with m representing the number of input variables, and K is the total number of observations.

3.4.2 Mutual Information

Mutual Information (MI) is a non-linear measure that considers the probability distribution of variables and utilizes entropy measurements to discern the dependency between variables. The MI between two discrete variables X and Y is expressed by Equation (3.2) [47].

$$I(Y, X) = H(Y) + H(X) - H(X; Y) \tag{3.2}$$

$$H(X) = \sum_{b=1}^{N} -log[P(x_b)]P(x_b) \tag{3.3}$$

$$H(X, Y) = \sum_{b=1}^{N}\sum_{k=1}^{N} -log[P(x_b, y_k)]P(x_b, y_k) \tag{3.4}$$

Where $H(X)$ and $H(X, Y)$ represent Shannon's Entropy, N denotes the number of bins, $P(x_b)$ is the probability density function, and $P(x_b, y_k)$ is the joint probability mass function of X and Y.

3.4.3 Fast tracker

fastTracker [45] is an algorithm for real-time causal-effect sensitivity analysis that tracks process variability. This objective is achieved by calculating sensitivity indices between input variables and the target variable. The primary steps of fastTracker are outlined in algorithm 1, where $X = [X_1, \ldots, X_m]$ represents a set of input variables, Y stands for the target variable, $k = 1, \ldots, K$ are the sample time, TT^j $(j = 1, \ldots, m)$ and ET are the trigger and event thresholds, respectively, and n signifies the number of batches per analysis span. The output of fastTracker consists of sensitivity indices, nSI^j $(j = 1, \ldots, m)$, corresponding to each input variable. More details are presented in [45].

3.5 Regression models

The choice of the model for the SS significantly influences the prediction results for the target variable. Consequently, various models were assessed for their suitability in fulfilling the SS function: Multiple Linear Regression (MLR), Ridge Regression (RR), Least Absolute Shrinkage and Selection Operator (LASSO), Elastic Net (EN), Support Vector Regression (SVR), and Gaussian Mixture Regression (GMR).

MLR, RR, LASSO, and EN were selected for their interpretability and success in capturing linear relations. Furthermore, RR, LASSO and EN are regularization models widely used to address multicollinearity and sparsity. Then, SVR is employed given its

Algorithm 1 fastTracker methodology [45].

Procedure:

1: **for** $k = 1, \ldots, K$ (for all data) **do**
2: **for each** input variable X_j **do**
3: Perform the trigger-event detection of two consecutive batches, i.e. determine if a variable represents a real change in the system state or not.
4: Determine the $XNOR$, i.e. verify the simultaneous existence or nonexistence of a change in each batch.
5: Obtain the sensitive index SI_k^j for instant k.
6: Obtain the normalized sensitivity index, nSI_k^j for instant k.
7: **end for**
8: **end for**

suitability when it comes to datasets with nonlinear relationships, providing flexibility for intricate patterns. And lastly, GMR is chosen for its effectiveness in handling complex patterns and mixed distribution types.

These models are presented respectively in the next subections.

3.5.1 Multiple Linear Regression

Multiple Linear Regression (MLR) is employed to identify a linear function that connects two or more independent variables with a single dependent variable. Due to its straightforward interpretation and implementation [48], MLR stands out as one of the most widely used statistical techniques and can be expressed mathematically using Equation (3.5).

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_j x_j + \cdots + \beta_m x_m + \varepsilon \tag{3.5}$$

Here, ε denotes the model's error term, and the regression coefficients β_0 and β_j are determined through the Least Squares method [49].

3.5.2 Ridge Regression

Ridge Regression (RR) stands as a regularization model, a variation of the conventional MLR, incorporating a penalty term into the linear least squares loss function. This penalty corresponds to the L2-norm of the coefficients, and the regularization strength is governed by the hyperparameter λ. RR, along with other regularization techniques, is primarily employed when there is high collinearity among independent variables, often leading to over-fitting [50]. The regularization serves to diminish the variance of the estimated parameters, mitigating the over-fitting effect [51]. In RR, this is achieved by minimizing the penalized residual sum of squares, as depicted in Equation (3.6).

$$\sum_{k=1}^{K} \left(y_k - \beta_0 - \sum_{j=1}^{m} x_{k,j}\beta_j\right)^2 + \lambda \sum_{j=1}^{m} \beta_j^2 \tag{3.6}$$

Where K represents the number of observations, m denotes the number of variables, β_0 and β_j are the coefficients and λ is the regularization term. Minimizing Equation (3.6)

and utilizing centered $x_{k,j}$ yields Equation (3.7).

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y} \tag{3.7}$$

Where $\boldsymbol{\beta}$ contains the regression coefficients and $\mathbf{I}$ represents the identity matrix.

3.5.3 Least Absolute Shrinkage and Selection Operator

The Least Absolute Shrinkage and Selection Operator (LASSO) is another widely utilized regulation method that distinguishes itself from RR by employing the L1-norm of the coefficients as the penalty term [52]. This property can lead less important variables to have null coefficients, making LASSO an effective feature selection method as well [53]. The regularization strength is controlled by the hyperparameter λ, as expressed in Equation (3.8) [53].

$$\boldsymbol{\beta} = \arg\min_{\boldsymbol{\beta}} \Big\{ \frac{1}{2} \sum_{k=1}^{K} \big(y(k) - \beta_0 - \sum_{j=1}^{m} x_j(k)\beta_j\big)^2 + + \lambda \sum_{j=1}^{m} |\beta_j| \Big\} \tag{3.8}$$

Where K represents the number of observations, m denotes the number of variables and β_0 and β_j are the regression coefficients, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_m]^T$.

3.5.4 Elastic Net

Elastic Net (EN) integrates both LASSO's L1-norm penalty and L2-norm penalty from RR to shrink prediction coefficients, offering a blend of LASSO's variable selection capabilities and RR's prediction performance [54]. The model is estimated by minimizing Equation (3.9).

$$\arg\min_{\beta_0, \boldsymbol{\beta}} \left\{ \frac{1}{K} \sum_{k=1}^{K} y_k \left(\beta_0 + \mathbf{x}_k\boldsymbol{\beta}\right) - \log\left(1 + e^{\beta_0 + \mathbf{x}_k\boldsymbol{\beta}}\right) + \lambda \left(\frac{(1-\alpha)||\boldsymbol{\beta}||^2}{2} + \alpha||\boldsymbol{\beta}|| \right) \right\} \tag{3.9}$$

Where α and λ are the hyperparameters responsible for tuning the model. The parameter α balances the effect of LASSO and RR penalties, ranging from 0 to 1, with 0 turning EN into RR and 1 into LASSO. Meanwhile, λ regulates the overall strength of regularization, influencing the trade-off between bias and variance in the estimated parameters. Again, K represents the number of observations and β_0 and $\boldsymbol{\beta}$ are the regression coefficients, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_m]^T$, with m denoting the number of input variables.

3.5.5 Support Vector Regression

SVR finds application in cases where the relationship between input variables and the target variable is nonlinear or displays intricate patterns, scenarios where conventional linear regression models tend to struggle. SVR employs a regression function that involves Lagrange multipliers, represented by α_k^* and α_k, and a kernel function K_e mapping the problem to different dimensions [55], as shown in Equation (3.10).

$$f(X) = \sum_{k,k'=1}^{K} (\alpha_k^* - \alpha_k) K_e(X_{k'}, X_k) + b \tag{3.10}$$

Where b is a constant. The Lagrange multipliers α_k^* and α_k are determined by maximizing the function $W(\alpha_k^*, \alpha_k)$, as expressed in Equation (3.11).

$$\begin{aligned} W(\alpha_k^*, \alpha_k) = -\frac{1}{2} \sum_{k,k'=1}^{K} (\alpha_k^* - \alpha_k)(\alpha_k^* - \alpha_k) K_e(X_k, X_{k'}) + \\ + \sum_{k=1}^{K} y_k(\alpha_k^* - \alpha_k) - \varepsilon \sum_{k=1}^{K} (\alpha_k + \alpha_k). \end{aligned} \tag{3.11}$$

Here, ε serves as a regularization parameter, K is the total number of observations, and the function adheres to the constraints: $\sum_k^K \alpha_k^* = \sum_k^K \alpha_k$ and $0 \leq \alpha_k^*, \alpha_k \leq C$, where C is a cost parameter determined through cross-validation [55].

3.5.6 Gaussian Mixture for Regression

GMM for regression is a probabilistic model that assumes the data is generated from a mixture of several Gaussian distributions and each Gaussian component represents a different local pattern or cluster in the data. This enables GMM to capture complex relationships and variations frequently found in real-world data [56]. The superposition is composed of probabilistic models and is obtained by Equation (3.12) [57].

$$p(X, Y) = \sum_{g=1}^{G} \pi^g \mathcal{N}_g(\mathbf{x}_k, y_k | \boldsymbol{\mu}_{XY}^g, \boldsymbol{\Sigma}_{XY}^g) \tag{3.12}$$

Where the joint probability distribution $p(X, Y)$ is obtained by Expectation-Maximization method [57], G is the number of Gaussian components, where the g-th component $\mathcal{N}^g(\cdot)$ is represented by the mean $\boldsymbol{\mu}_{XY}^g$, and variance $\boldsymbol{\Sigma}_{XY}^g$. π^g, with $\sum_{g=1}^K \pi^g = 1$, is the component weight.

For prediction, the Gaussian Mixture Regression (GMR) calculates the conditional distribution $p(Y|X)$ for a given sample using Equation (3.13) [57].

$$p(Y|X) = \sum_{g=1}^{G} \pi_{Y|X}^g \mathcal{N}_g(Y | \boldsymbol{\mu}_{Y|X}^g, \boldsymbol{\Sigma}_{Y|X}^g) \tag{3.13}$$

The component weight $\pi_{Y|X}^g$ is determined by Equation (3.14).

$$\pi_{Y|X}^g = \frac{\mathcal{N}_g(X | \boldsymbol{\mu}_{\mathbf{x}}^g, \boldsymbol{\Sigma}_X^g)}{\sum_{l=1}^{G} \mathcal{N}_l(\mathbf{x}_k | \boldsymbol{\mu}_X^l, \boldsymbol{\Sigma}_X^l)} \tag{3.14}$$

3.6 Framework to design a Soft Sensor

In this section, the framework formulated for designing a SS is presented, aligning with the key steps introduced in Section 3.2. The main stages of the framework are outlined in Algorithm 2, and can be split into the following phases: data pre-processing (Step 1 and 2), variable selection (Step 3), model selection (Steps 4 and 5) and subsequent model training and testing (Step 6 to 15).

Algorithm 2 Soft Sensor Methodology.

Input: dataset.
Procedure:

1: Pre-process the dataset (delete outliers, non numeric variables, and empty variables).
2: Perform the feature expansion.
3: Select the variable selection method (Section 3.4): Pearson's correlation, MI, or fastTracker.
4: Define the model to be used (Section 3.5): MLR, RR, LASSO, EN, SVR, or GMR.
5: Define hyperparameters for the selected model.
6: **for** $it = 1, \ldots, it_{max}$: **do**
7: Split randomly the dataset in train (70%) and test (30 %).
8: Tune hyperparameters using Grid Search and k-fold Cross-Validation, with k = 10.
9: Fit the model using the best hyperparameters (from step 7) to the training dataset.
10: Predict the target variable Y for the test dataset.
11: Calculate the error metrics.
12: Save the model and hyperparameters values.
13: **end for**
14: Obtain the average of the error metrics.
15: Select the model with the best error metrics.

Steps 1 and 2:

The initial step involves pre-processing the dataset, which includes removing non-numeric columns, as well as empty or partially empty columns. Outliers are identified using the z-score, a statistical measure indicating how many standard deviations a data point deviates from the population mean. The z-score for each features's sample is calculated using Equation (3.15).

$$z = \frac{x - \mu}{\sigma} \tag{3.15}$$

Where μ represents the mean and σ is the standard deviation. In this context, a sample is assumed as being an outlier if $z > 3$. Subsequently, outliers are scrutinized and removed, along with samples containing empty values.

Subsequently, the dataset is split into input variables (X) and output variable (Y). The input variables undergo feature expansion, generating the square, inverse, and root mean squared values of each variable X_j, along with the products of each pair of variables. To mitigate potential data inaccuracies, two exceptions are implemented: the root mean squared is omitted if an original variable contains negative values, and the inverse is excluded in the presence of zero values.

Step 3:

Following this phase, one of the variable selection methods outlined in Section 3.4 is applied — either Pearson's correlation, Mutual Information, or fastTracker.

Steps 4 and 5:

Following variable selection, a model is selected from the options detailed in Section 3.5: MLR, RR, LASSO, EN, SVR, or GMR. After that, based on the chosen model, appropriate values for hyperparameters are defined.

Steps 6 to 15:

To mitigate potential biases stemming from dataset division and ensure robust model description, the training and testing process is repeated for a total of it_{max} iterations. During each iteration, the dataset is randomly split into a training set (70%) and a test set (30%).

Within each iteration, hyperparameters are fine-tuned using a Grid Search with K-folds Cross-Validation (where $K = 10$) to prevent overfitting. GridSearch is a widely employed technique for hyperparameter optimization. For every hyperparameter combination, the algorithm undergoes training and validation using K-Fold Cross-Validation. The training dataset is partitioned into K equally-sized subsets (folds), with K-1 folds used for model training and the remaining subset for validation in each iteration. This process yields K performance scores, and their mean is calculated for comparison across all hyperparameter combinations. The combination with the lowest Mean Squared Error (MSE) is chosen and used to fit the model to the training dataset.

Once the model parameters are determined, predictions for the target variable Y are made using the input variables from the test dataset. The predicted values are then compared with the actual values of Y, and error metrics, as well as the model itself and its hyperparameters, are computed and stored.

Upon completion of all iterations (it_{max}), the average of the error metrics is calculated, and the model with the best metrics is selected as the final Soft Sensor.

Chapter 4

Principal Component Analysis for process monitoring and fault detection

Principal Components Analysis (PCA) stands out as a prominent linear technique for dimension reduction, dividing data information into its most significant patterns [58, 32]. Due to its versatility, PCA finds widespread applications in data analysis, such as image processing, feature extraction, and pattern recognition [59]. Among its various applications, fault detection stands out as a particularly significant and well-studied area [60, 61, 62, 63]. Consequently, PCA has been selected for the purpose of data entry error detection.

The upcoming chapter is organized into two sections. Section 4.1 explores the mathematical foundations of PCA, providing a comprehensive understanding of its principles. Subsequently, Section 4.2 outlines a methodology leveraging PCA for Process Monitoring and Fault Detection that will be applied to data entry error detection.

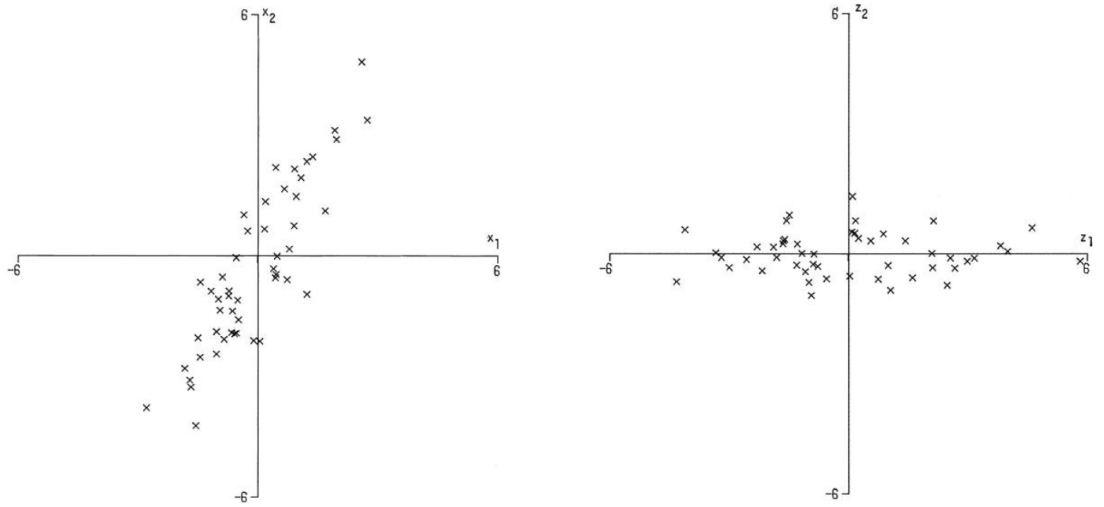
4.1 Introduction to PCA

The fundamental idea behind PCA is to reduce the dimensionality of a dataset while preserving the most relevant information [64]. This is achieved by projecting the data into a new subspace defined by Principal Components (PCs). These components are uncorrelated and prioritize capturing the majority of the variability within the original data. Figure 4.1 provides a visual representation of the transformation process, showcasing how 50 data points are mapped from their original two-dimensional space to a new subspace delineated by the principal components z_1 and z_2, where the principal component z_1 captures the majority of the data variability.

The initial step of PCA involves an observation matrix $\mathbf{X} \in \mathbb{R}^{K \times m}$, where each column expresses a variable, and each row represents a sample. Matrix $\mathbf{X}$ is then normalized to zero mean and unit variance, creating a normalized observation matrix $\mathbf{X} = [x_{k,j}]_{K \times m}$, with each $\mathbf{x}_k = [x_{k,1}, \ldots, x_{k,m}]$ with $k = 1, \ldots, K$ denoting the k-th normalized observation vector.

Next, the covariance matrix ($\mathbf{\Sigma} \in \mathbb{R}^{m \times m}$) is computed using Equation (4.1).

$$\mathbf{\Sigma} = \frac{\mathbf{X}^T \mathbf{X}}{K - 1} \tag{4.1}$$



(a) Plot of 50 observations on two variables x_1 and x_2.

(b) Plot of the 50 observations with respect to their PCs z_1 and z_2.

Figure 4.1: PCA in practice with 2 variables [64].

Where K is the total number of observations. Covariance matrix is a square matrix where diagonal elements contain the variances [65] of individual variables, and the off-diagonal elements represent the covariance between pairs of variables, given by:

$$\mathbf{\Sigma} = \begin{bmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \dots & \mathrm{Cov}(X_1, X_m) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \dots & \mathrm{Cov}(X_2, X_m) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_m, X_1) & \mathrm{Cov}(X_m, X_2) & \dots & \mathrm{Var}(X_m) \end{bmatrix},$$

where $\mathrm{Var}(\cdot)$ is the variance, representing the measure of the dispersion of a single random variable around its mean, and $\mathrm{Cov}(\cdot, \cdot)$ is covariance, indicating the measure of the extent to which two random variables change together, reflecting their joint variability and relationship. For a comprehensive understanding of how to calculate variance and covariance, refer to Navidi's "Statistics for Engineers and Scientists" [66].

Subsequently, the eigenvectors and eigenvalues [67] of $\mathbf{\Sigma}$ need to be calculated. Eigenvalues and eigenvectors are the scalar and vector quantities respectively associated with matrices used for linear transformations. Eigenvectors remain in the same direction after a linear transformation is applied to it and eigenvalues are scalars attached to them. Thus, $\mathbf{\Sigma}$ is decomposed using Singular Value Decomposition (SVD) [68], as presented in Equation (4.2).

$$\mathbf{\Sigma} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T \tag{4.2}$$

Where $\mathbf{\Lambda} \in \mathbb{R}^{m \times m}$ is a diagonal matrix that contains the eigenvalues (λ_j) of $\mathbf{\Sigma}$, i.e., $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, ..., \lambda_m)$ with $\lambda_1 \geq ... > \lambda_m \geq 0$, and $\mathbf{P} \in \mathbb{R}^{m \times m}$ is known as the loadings matrix and comprises the eigenvectors of $\mathbf{\Sigma}$ corresponding to the eigenvalues in $\mathbf{\Lambda}$. Each column of $\mathbf{P}$ represents a new coordinate axis, the Principal Components, and rows represent the original variables. Consequently, the values in the loading matrix $\mathbf{P}$ serve as coefficients (or weights), indicating the contribution of each original variable to the PCs. The following matrix represents the loadings matrix, with $p_{k,j}$ being the weight of the original variable X_j on the j^{th} Principal Component,

	PC_1	PC_2	$\dots$	PC_m
X_1	$p_{1,1}$	$p_{1,2}$	$\cdots$	$p_{1,m}$
X_2	$p_{2,1}$	$p_{2,2}$	$\cdots$	$p_{2,m}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
X_m	$p_{m,1}$	$p_{m,2}$	$\cdots$	$p_{m,m}$

.

The eigenvalues within matrix $\mathbf{\Lambda}$ quantify the variability along the PCs.

The following step involves selecting the number of PCs, l, to consider and use in the data transformation. A common criteria is based on the cumulative explained variance [69]. The explained variance refers to the proportion of the total variance in the original data that is captured by a Principal Component and it quantifies how much of the variability in the dataset explained by that specific PC. Explained variance is calculated using the obtained eigenvalues. For each eigenvalue, the explained variance is computed by dividing it by the total sum of eigenvalues. Mathematically, this can be expressed as shown in Equation (4.3).

$$\text{Explained Variance}(\lambda_j) = \frac{\lambda_j}{\sum_{j=1}^{m} \lambda_j} \tag{4.3}$$

Where λ_j is the j^{th} eigenvalue of $\mathbf{\Lambda}$, corresponding to the j^{th} PC, and m is the total number of original variables.

Subsequently, the cumulative explained variance represents the total amount of variance in the original data that is explained by a given number of PCs. It is obtained by calculating the cumulative sum of the explained variances, providing a measure of the total variance explained as the number of PCs increases. This process is represented by the Equation (4.4).

$$\text{Cumulative Explained Variance} = \sum_{j=1}^{l} \text{Explained Variance}(\lambda_j) \tag{4.4}$$

Where l is the ideal number of PCs. The selection of l is determined based on the condition that the cumulative explained variance is equal to or higher than a specified threshold, e.g. 90%, as Equation (4.5) shows.

$$\arg\min_{l} \left\{ \sum_{j=1}^{l} \text{Explained Variance}(\lambda_j) \geq 0,9 \right\} \tag{4.5}$$

With the number of Principal Components defined, matrix $\mathbf{P}$ is divided according to the selected l, $P = [\mathbf{P}_{\text{pc}}\mathbf{P}_{\text{res}}]$. Here, $\mathbf{P}_{\text{pc}} \in \mathbb{R}^{m \times l}$ is the part of the loading matrix that contains the PCs that capture the most significant variations in the data. $\mathbf{P}_{\text{pc}}$ is then used to project the data into the new coordinate system where the axes are aligned with the selected PCs. Conversely, $\mathbf{P}_{\text{res}} \in \mathbb{R}^{m \times (m-l)}$ includes the eigenvectors that map the residual subspace, capturing directions with less variability in the data. Data projected into $\mathbf{P}_{\text{res}}$ is often treated as noise or outliers.

Matrix $\mathbf{\Lambda}$ is also decomposed into two matrices:

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_{\text{pc}} & 0 \\ 0 & \mathbf{\Lambda}_{\text{res}} \end{bmatrix}.$$

Here, $\mathbf{\Lambda}_{\text{pc}}$ is the diagonal matrix containing the eigenvalues up to the selected number of components, denoted as $\mathbf{\Lambda}_{\text{pc}} = \text{diag}(\lambda_1, ..., \lambda_l)$, while $\mathbf{\Lambda}_{\text{res}}$ includes the eigenvalues corresponding to the residual subspace, expressed as $\mathbf{\Lambda}_{\text{res}} = \text{diag}(\lambda_{l+1}, ..., \lambda_m)$.

Algorithm 3 PCA methodology for fault detection [59].

Input: Dataset $\mathbf{X}$ (train data), significant level α
Offline Stage:
1: Normalize dataset $\mathbf{X}$ to zero mean and unit variance.
2: Determine the covariance matrix $\mathbf{\Sigma}$ (Equation (4.1)).
3: Determine the loadings matrix $\mathbf{P}$ and the eigenvalues $\mathbf{\Lambda}$ using SVD (Equation (4.2)).
4: Select the number of Principal Components (l) that explain 90% of the variance (Equation (4.5)).
5: Determine the SPE threshold, δ_α^2 (Equation (4.10)).
6: Determine the T^2 threshold for the selected significant level α, T_α^2 (Equation (4.14)).
Online Stage:
7: For each new sample, determine SPE and T^2 and classify it:
8: **if** $SPE \leq \delta_\alpha^2$ and $T^2 \leq T_\alpha^2$ **then**
9: sample is fault-free.
10: **else**
11: sample is faulty.
12: **end if**

Finally, the dataset is projected onto the new subspace. This operation is achieved by multiplying matrix $\mathbf{X}$ by the transpose of $\mathbf{P}_{\text{pc}}$, as illustrated in Equation (4.6).

$$\mathbf{Z} = \mathbf{X}\mathbf{P}_{\text{pc}}^T \tag{4.6}$$

Where $\mathbf{Z}$ contains the scores along the selected Principal Components for each sample, resulting in a reduced-dimension representation that effectively captures the majority of the variance in $\mathbf{X}$ while minimizing the impact of noise.

4.2 PCA methodology for process monitoring and fault detection

This section explores one of the commonly employed algorithms, leveraging PCA as a classification method to identify faults [59].

The main steps of the methodology are outlined in Algorithm 3. The procedure can be categorized into the selection and normalization of K samples (Step 1), the application of Singular Value Decomposition to the covariance matrix (Steps 2 and 3), the determination of the number of Principal Components, l (Step 4), the definition of thresholds (Steps 5 and 6), and classification of each new sample using Squared Prediction Error (SPE) and Hotelling's T^2 (Steps 7 to 12).

Step 1:

Initially, the samples are gathered and arranged in an observation matrix $\mathbf{X} \in \mathbb{R}^{K \times m}$, where K denotes the number of samples, and m represents the number of variables. This matrix is then normalized to zero mean and unit variance.

Steps 2 and 3:

Subsequently, the standard PCA procedure is followed, involving the determination of the covariance matrix $\mathbf{\Sigma}$ through Equation (4.1). The SVD is then applied to $\mathbf{\Sigma}$ using Equation (4.2), resulting in the extraction of the loadings matrix $\mathbf{P}$ and the diagonal matrix $\mathbf{\Lambda}$. As elaborated in Section 4.1, the columns of $\mathbf{P}$ represent Principal Components, while the diagonal values of $\mathbf{\Lambda}$ (eigenvalues of $\mathbf{\Sigma}$) quantify the variability along these PCs.

Step 4:

The number of PCs, l, is determined based on the eigenvalues, aiming to identify the minimum number of components that ensures a specific Cumulative Explained Variance, as depicted in Equation (4.5). In this work, l components had to explain 90% of the variance.

With l defined, matrix $\mathbf{\Lambda}$ is divided (as shown in Matrix 4.1) into $\mathbf{\Lambda}_{\text{pc}} = \text{diag}(\lambda_1, ..., \lambda_l)$, containing the top-$l$ eigenvalues, and $\mathbf{\Lambda}_{\text{res}} = \text{diag}(\lambda_{l+1}, ..., \lambda_m)$, encompassing the eigenvalues linked to the residual subspace (as explained in Section 4.1).

The loadings matrix $\mathbf{P}$ is also partitioned based on the selected l, resulting in $P = [\mathbf{P}_{\text{pc}}\mathbf{P}_{\text{res}}]$, where $\mathbf{P}_{\text{pc}} \in \mathbb{R}^{m \times l}$ includes the PCs utilized to project the data into the new coordinate axes. Additionally, $\mathbf{P}_{\text{res}} \in \mathbb{R}^{m \times (m-l)}$ comprises the eigenvectors that map the residual subspace, generating noise and outliers when used for data projection. Matrix $\mathbf{P}_{\text{pc}}$ is subsequently employed to project new samples into the new subspace.

Steps 5-6:

To classify the samples as "fault-free" or "fault", the literature suggests using SPE and Hotelling's T^2 [59, 70, 71]. SPE measures the difference between the original observation matrix $\mathbf{X}$ and its approximation using the PCs (projection using $\mathbf{P}_{\text{pc}}$ and Equation (4.6)). This index can be computed, at the sample k, by projecting the sample row $\mathbf{x}_k \in \mathbb{R}^m$ ($\mathbf{x}_k = [x_{k,1}, \ldots, x_{k,m}]$) onto the residual subspace (utilizing $\mathbf{P}_{\text{res}}$) and subsequently calculating its squared Euclidean norm. Equation (4.7) illustrates the projection of a normalized vector sample $\mathbf{x}_k$ into the residual subspace, while Equation (4.8) outlines the computation of the SPE.

$$\mathbf{z}_k = \mathbf{x}_k \cdot \mathbf{P}_{\text{res}} \tag{4.7}$$

$$SPE = ||\mathbf{z}_k||^2 \tag{4.8}$$

Here, $||\cdot||$ denotes the Euclidean norm, and $\mathbf{z}_k \in \mathbb{R}^{m-l}$, where m is the number of variables and l is the number of PCs, represents the sample projected into the residual subspace. Equation (4.8) is equivalent to Equation (4.9).

$$SPE = \mathbf{x}_k^T \mathbf{P}_{\text{res}} \mathbf{P}_{\text{res}}^T \mathbf{x}_k \tag{4.9}$$

Where $\mathbf{x}_k$ is the normalized row with the sample measurements and $\mathbf{P}_{\text{res}}$ contains the PCs that map the residual subspace.

SPE provides a measure of how well a sample can be represented in the new subspace defined by the Principal Components. A lower SPE value is desirable, indicating that the sample aligns well with the detected patterns and is well represented by the model. Conversely, a higher SPE value suggests that the new sample ($\mathbf{x}_k$) behaves like an outlier and deviates significantly from the patterns captured by the PCs. To assess whether the SPE of a new sample is within a normal range, a threshold must be defined, denoted as

δ_α^2. Jackson and Mudholkar [72] developed an expression for this threshold that can be seen in Equation (4.10).

$$\delta_\alpha^2 = \theta_1 \left(\frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right)^{\frac{1}{h_0}} \tag{4.10}$$

Where c_α represents the confidence interval corresponding to the $1 - \alpha$ percentile of the normal distribution, where α is the significance level. The coefficients θ_i are computed using the eigenvalues (λ_j) associated with the residual subspace, as expressed in Equation (4.11). The parameter h_0 is determined by combining θ_1, θ_2, and θ_3 as outlined in Equation (4.12).

$$\theta_i = \sum_{j=l+1}^{m} \lambda_j^i, i = 1, 2, 3 \tag{4.11}$$

$$h_0 = 1 - \frac{2\theta_1 \theta_3}{3\theta_2^2} \tag{4.12}$$

A normal SPE value is lower than or equal to δ_α^2. Detection of an anomaly ($SPE > \delta_\alpha^2$) indicates that one or more variables are no longer varying as predicted by the Principal Components.

Since faults can manifest in various ways, a more robust detection system incorporates Hotelling's T^2 as a metric for identifying potential errors. While SPE is sensitive to deviations orthogonal to the PCs (in the residual subspace), T^2 provides information about deviations along the PC directions. T^2 evaluates how far a sample deviates from the mean of the observation matrix in the principal subspace and can be calculated using Equation (4.13) [70].

$$T^2 = \mathbf{x}_k^T \mathbf{P}_{\text{pc}} \mathbf{\Lambda}_{\text{pc}}^{-1} \mathbf{P}_{\text{pc}}^T \mathbf{x}_k \tag{4.13}$$

Where $\mathbf{x}_k$ contains the measurements for the k^{th} sample, $\mathbf{P}_{\text{pc}}$ is the matrix containing the Principal Components and $\mathbf{\Lambda}_{\text{pc}}^{-1}$ is the inverse of the diagonal matrix containing the eigenvalues that correspond to the selected PCs.

Assuming that the data follows a multivariable normal distribution, the T^2 statistic is related to an F distribution [73]. Leveraging this correlation and assuming the mean is known, an upper control limit for T^2, T_α^2, can be defined for a given significance level α using Equation (4.14) [74].

$$T_\alpha^2 = \frac{l(K-1)}{K-l} F_{l,K-l;\alpha} \tag{4.14}$$

Where $F_{l,K-l;\alpha}$ is an F distribution with l and $K - l$ degrees of freedom, K is the number of samples collected, and l is the selected number of PCs.

Steps 7 to 12:

While SPE captures deviations orthogonal to the PCs axes, T^2 provides insights about possible faults along the PCs. Consequently, considering both metrics offers a more robust approach to identify faults that might be overlooked if only one of them is considered. With the metrics and thresholds defined, for every new sample, the SPE and T^2 are computed using Equations (4.9) and (4.13), respectively. Based on these measures, the sample is classified using the following rule:

$$\begin{gathered} \text{if } SPE \leq \delta_\alpha^2 \text{ and } T^2 \leq T_\alpha^2 \rightarrow \text{fault free} \\ \text{otherwise} \rightarrow \text{faulty} \end{gathered}$$

Chapter 5

Methodology to detect data entry errors

In this chapter, it is proposed a methodology designed to identify errors within entry data. The chapter unfolds as follows: Section 5.1 delineates the artificial insertion of errors into the dataset; Section 5.2 expounds on the metrics employed to assess the model's performance, and Section 5.3 concludes by providing an description of the defined models for error detection using a Soft Sensor methodology and a PCA methodology.

5.1 Artificial errors

To test the methodology proposed in Section 5.3, the datasets were manipulated in order to replicate artificially data entry errors. Since there are no available ready-to-use datasets with this characteristic, the errors were artificially inserted into the selected datasets. There were considered five distinct error categories: blank spaces, order errors, doubles, measurement errors and extra numbers.

Blank spaces are a prevalent error type that can be encountered in various datasets, including those without manually entered data. In an industrial context, these errors may occur when a laboratory operator unintentionally skips a sample, resulting in a gap. To simulate this scenario, the target variable is replaced by a zero.

Order errors manifest when an operator mistakenly swaps the positions of two digits while recording a value. For instance, instead of inputting 195, one might mistakenly record 159. The impact of these errors on results depends on the specific digits involved. For example, confusing 195 with 159 may have a less significant impact than erroneously entering 915. Therefore, order errors can be analyzed in terms of the change in the order of magnitude that they cause. For instance, replacing a value of 1.35 with 13.5 induces a change in the order of magnitude of +1.

Doubles represent another common error, occurring when a operator records the same value twice. Simulating this scenario within a dataset involves duplicating the target variable value of the sample immediately preceding the selected one. It's important to note that, to prevent the introduction of duplicated errors, the chosen dataset samples were intentionally non-successive.

Measurement errors constitute well-explored errors, particularly in the context of laboratory instruments. This issue arises not only from the improper use of instruments but also holds significance in the areas of metrology and instrument construction. In this

study, the assumption is that the error stems from the misapplication of the instrument by the user only. Thereby, three subcategories were created:

- measurement error: 5 to 10 %
- measurement error: 10 to 25 %
- measurement error: 25 to 50 %

For the selected sample within each subcategory, an error value is randomly generated. For instance, in the case of measurement errors ranging from 5 to 10 %, a value is chosen between 0.9 and 0.95 (representing an error of -5 to -10 %) and 1.05 and 1.1 (indicating an error of 5 to 10 %). Subsequently, the target variable value is multiplied by the error value.

Lastly, extra number errors may manifest during data entry when an operator repeats a digit. For instance, the value 324 could be mistakenly entered as 3224, 3244, or 3324. In this simulation, for each selected value, a digit is randomly chosen and duplicated. In cases involving decimal numbers, the impact of the error may vary. For instance, if the original number is 15.9 and the introduced error is 15.99, the difference is nearly imperceptible. However, when the duplicated digit is integral, such as the 5 (becoming 155.9), the disparity becomes substantial. Recognizing the significance of this difference in results analysis, the possible change in the order of magnitude is registered. It's worth noting that decimal points were intentionally excluded from duplication to avoid having non numeric values (such as 15..9).

5.2 Metrics

To properly evaluate the performance of the proposed algorithms, it is crucial to establish robust metrics. In this work, two categories of metrics are essential: regression metrics, measuring the accuracy of Soft Sensor in predicting the target variable, and classification metrics, assessing the algorithms' ability to predict whether a sample represents an error or a true value.

5.2.1 Regression metrics

Three metrics were used: Root Mean Squared Error (RMSE), Normalized Root Mean Squared Error (NRMSE), and the Determination Coefficient (denoted as R^2).

RMSE is a regression metric frequently applied to quantify the accuracy of model predictions by providing a measure of the average magnitude of the errors between predicted and actual values. It is calculated by taking the square root of the mean of the squared differences between predicted ($\hat{Y}$) and actual (Y) values across all samples:

$$\text{RMSE} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(\hat{y}_k - y_k)^2}. \tag{5.1}$$

Here, K represents the total number of samples.

A lower RMSE value indicates better predictive performance, since it reflects smaller errors. Thereby, it is considered a valuable metric for assessing how well a model aligns with the observed values.

The NRMSE is the normalized version of the Root Mean Squared Error. By normalizing the error values it provides a relative measure of the predictive performance. Being so, it is particularly useful for comparing models with different scale of target variables. NRMSE is calculated by dividing the RMSE by the range of the actual values, and its values fall between 0 and 1:

$$\text{NRMSE} = \left(\frac{\text{RMSE}}{\max(\mathbf{y}) - \min(\mathbf{y})} \right). \tag{5.2}$$

Where $\mathbf{y}$ is the vector of the target variable.

Just as with the RMSE, a reduced NRMSE indicates better model performance, reflecting a smaller percentage of error in relation to the overall range of actual values.

Finally, R^2 is a statistical measure that assesses how much of the variability in the target variable can be explained by the independent variables. In other terms, it quantifies the extent to which the model's predictions capture the fluctuations observed in the target variable. It can be determined as follows:

$$R^2 = 1 - \frac{\sum_{k=1}^{K} (\hat{y}_k - y_k)^2}{\sum_{k=1}^{K} (y_k - \bar{y})^2} \tag{5.3}$$

Where K stands for the total number of samples and $\bar{y}$ represents the mean of $\mathbf{y}$.

R^2 ranges from 0 to 1, where 1 indicates a perfect fit. A higher value implies that a greater proportion of the target variable variance is elucidated by the model's predictions, indicating a better fit of the Soft Sensor to the data. It is important to keep in mind that a negative value for R^2 can exist. That result indicates that a flat curve describes better the data than the tested model.

The metrics described above can be used to understand how well a model describes the data. To pick the best Soft Sensor, a mix of the R^2 and the NRMSE was used. The best model will be the one with the lowest NRMSE and the highest R^2, and consequently obtains the lowest value in Equation 5.4.

$$\frac{(1 - R^2) + \text{NRMSE}}{2} \tag{5.4}$$

5.2.2 Classification metrics

Unlike regression metrics, that intend to assess how well model's predictions align with observed values, the classification metrics goal is to measure the model's ability to correctly classify instances into their respective categories. In this case, into the categories of "errors" and "true values".

In a binary classification case, metrics frequently rely on the concept of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). Since the goal of the applied algorithm is to identify errors, the above terms are defined as follows:

- TP: samples that are correctly classified as "error";
- TN: samples that are correctly classified as "true value";
- FP: samples that are incorrectly classified as "error" when they are actually "true value";
- FN: samples that are incorrectly classified as "true value" when they are actually "error".

Table 5.1: Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	TP	FN
Actual Negative	FP	TN

These terms can show the number of correct and incorrect predictions for each class and are organized in a confusion matrix presented in Table 5.1.

Having the confusion matrix, four metrics were selected: Precision, Sensitivity, F1-score, and Specificity.

Precision, also referred to as Positive Predictive Value, measures the accuracy of the positive predictions made by the model. It's the proportion of true positive predictions among all positive predictions and it ranges between 0 and 1, given by:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{5.5}$$

In this scenario, precision indicates the proportion of correctly predicted "error" samples relative to all the "error" predictions (both correct and incorrect). High precision (values closer to 1) implies that the model's positive predictions are highly accurate and there is a low rate of False Positives (FP).

Sensitivity, also known as True Positive Rate or Recall, gauges the proportion of positive instances accurately identified by the model. It offers insights into how well the model captures positive instances within the dataset and its numeric values span from 0 to 1, given by:

$$\text{Sensitivity} = \frac{TP}{TP + FN}. \tag{5.6}$$

In this context, sensitivity reflects the fraction of accurately predicted "error" samples in relation to all "error" instances, including samples identified as "true value" that were indeed "error". Sensitivity closer to 1 means that the model is effective at detecting most of the positive samples (even if it results in more false positives)

To balance precision and sensitivity, the F1-score considers both false positives and false negatives. It can be determined using Equation (5.7).

$$\text{F1 score} = \frac{2 \times \text{Sensitivity} \times \text{Precision}}{\text{Sensitivity} + \text{Precision}} \tag{5.7}$$

F1 score ranges from 0 to 1, where 0 indicates poor performance or complete misclassification and 1 represents perfect precision and sensitivity. A higher score is desirable as it signifies a better trade-off between precision and recall and it indicates a more balanced classification performance.

Lastly, specificity, commonly referred to as True Negative Rate, is a metric similar to the precision but focused on the negative values, since it evaluates the ability of a classification model to correctly identify negative instances. Its values are confined to the range of 0 to 1 and can be calculated through Equation (5.8).

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{5.8}$$

Within this framework, specificity measures the proportion of actual "true value" instances that are correctly identified by the model. A higher specificity value indicates that

the model is effective at identifying negative instances, minimizing the occurrence of false positives, and it complements precision and sensitivity when evaluating a model's performance.

5.3 Setup of the methodologies for data entry error detection

Having outlined the procedure for artificially introducing errors to simulate human data entry mistakes, and with the defined metrics in place, two methodologies were created to detect entry data errors. The first methodology takes advantage of the Soft Sensor approach mentioned in Chapter 3, while the second uses the PCA algorithm described in Chapter 4.

5.3.1 Soft Sensors

In accordance with the concepts detailed in Chapter 3, Soft Sensor serve as inferential models employed to predict physical quantities, proving to be a robust tool for estimating laboratory variables in industrial settings. Consequently, they have been integrated into a methodology designed to identify data entry errors.

The key steps of this methodology are outlined in Algorithm 4 and can be divided into the following main phases: selection of the Soft Sensor (Step 1), data pre-processing (Step 2), error artificial insertion (Step 3), variable selection (Step 4 and 5), model training and testing (Step 6 to 11), and samples classification (Step 12 to 23).

Step 1:

The initial step is to determine the SS model that best characterizes the chosen dataset. This entails the selection of a model formed by combining a variable selection method and a regression model. To identify the optimal combination, every variable selection method and regression model described in Chapter 3 is systematically tested following Algorithm 2. Besides, tests are also performed without any variable selection method and applying only Feature Expansion. From these tests, the average NRMSE and R^2 are extracted. The subsequent selection of the best combination considers a metric that incorporates both NRMSE and R^2, as defined by Equation 5.4. This measure aligns with the objective outlined in Section 5.2, where the aim is to find a model with a lowest NRMSE (closer to zero) and an R^2 as close to one as possible. Models with a negative R^2 value are preemptively excluded due to their implication of a poor fit to the dataset, which could potentially impact the results. Additionally, the R^2 and NRMSE are independently verified to confirm the selection of the soft sensor model.

Step 2:

Following the selection of the SS model, and before proceeding to the subsequent steps, it's important to highlight that certain procedures, such as pre-processing, variable selection, and model fitting, carried out in Step 1, will need to be re-executed as part of the ensuing stages.

With the SS selected, the dataset is pre-processed. Here, columns and samples with non numeric data are eliminated as well as the ones that are empty or semi-empty and

Algorithm 4 Soft Sensor for error detection methodology.

Input: Dataset, error proportion, error categories to be studied, Soft Sensors models, the set of values of SS_{th} to be studied.

Procedure:

1: Select the Soft Sensor model that better fits the dataset using Algorithm 2.
2: Pre-process dataset (remove outliers, non-numeric variables, empty variables, rename columns, reset indexes).
3: Insert artificial errors according to error categories and error proportion.
4: Variable division: create the input variables $\mathbf{X}$, $\mathbf{Y}$ for regression ($\mathbf{Y}_{\text{reg}}$), and $\mathbf{Y}$ for classification ($\mathbf{Y}_{\text{clas}}$).
5: Select variables using the elected variable selection method (from Step 1).
6: Split the dataset into train (70%) and test (30%), selecting all errors data for the test dataset.
7: Tune hyperparameters using Grid Search and k-fold Cross-Validation in the training dataset.
8: Fit the model (chosen in Step 1) using the best hyperparameters (from Step 5) on the training dataset.
9: Predict the target variable $\mathbf{Y}_{\text{reg}}$ for the test dataset.
10: Calculate regression metrics (NRMSE and R^2).
11: Save the model and metrics.
12: **for** each test sample $k = 1, \ldots, K$ **do**
13: **for** each possible SS_{th} **do**
14: Classify k sample for the test dataset:
15: **if** $y_{reg}(k) \in \hat{y}_{pred}(k) \times [1 - SS_{th}, 1 + SS_{th}]$ **then**
16: $y_{clas}(k)$ is "true value"
17: **else**
18: $y_{clas}(k)$ is "error"
19: **end if**
20: Calculate the classification metrics.
21: **end for**
22: Save classification metrics for each SS_{th}.
23: **end for**

outlier samples are removed. Furthermore, the indexes are reset, and the columns are systematically renamed (designated as X_1, X_2, ..., X_m and Y). Finally, two new columns are introduced: "classification", that labels each samples as "true value" or "error", and "error category", which encapsulates the type of error associated with each error sample.

Step 3:

Moving forward, the artificial errors are inserted into the prepared dataset. The number of errors to insert is determined by the input variable "error proportion". For instance, if that value is 0.15, then 15% of the samples will be converted into errors. The selection of those samples is done randomly but repeatable. The number of indexes to select is rounded up to an integer and the samples selected are never adjacent (to ensure that in doubles the repeated value is not an error). In case the error categories include more than one error type, the number of samples per error category is the total number of errors divided by the number of categories (rounded down to an integer and the rest of

the division is split into the categories). For example, if all five categories are selected and there are 26 samples to be converted into errors, four categories will have five error samples and one category will have six. As described in Section 5.1, the error categories in study are: blank spaces, order errors, doubles, measurement errors and extra number errors. Each of these samples is classified as an "error" (in the "classification" column) and the categories are registered in the "error category" column.

Steps 4 and 5:

Then, the dataset is separated into $\mathbf{X}$ and $\mathbf{Y}$. $\mathbf{Y}$ is also split into two categories: $\mathbf{Y}$ for regression ($\mathbf{Y}_{\text{reg}}$) that includes the y values and $\mathbf{Y}$ for classification ($\mathbf{Y}_{\text{clas}}$) that contains the "classification" column previously created. This is also the stage where the variable selection method, previously selected in Step 1, is implemented to $\mathbf{X}$.

Step 6:

Subsequently, the dataset is split into train (70%) and test (30%), with the training dataset exclusively incorporating "true value" samples. All errors are included in the test dataset, making it crucial to maintain an error proportion of approximately 15%. Otherwise, the test dataset will be unbalanced and have either an excess of "error" samples or "true values".

Step 7:

Then, if the selected Soft Sensor model (from Step 1) involves hyperparameters (applicable to all models except MLR), these parameters undergo fine-tuning through Grid Search and k-fold Cross-Validation, as detailed in Chapter 3. This optimization process is only applied to the training dataset.

Steps 8 to 11:

Upon selecting the optimal hyperparameters, the model is fitted to the training dataset and subsequently employed to predict Y for the test dataset. Following this prediction, R^2, RMSE and NRMSE are computed using Equations (5.3), (5.1) and (5.2). They compare the predicted values, $\hat{y}_{\text{test}}$, with the original values y_{test} before error insertion. This approach ensures the confirmation of the model's performance (if the errors were considered in this step, the metrics would not assess the real dataset). Lastly, both the metrics and the model itself are saved.

Steps 12 to 23:

After this step, for each potential SS_{th} under consideration, the test samples undergo classification. In this process, the predicted y ($\hat{y}_{\text{pred}}(k)$) is treated as a reliable approximation of the true $y(k)$ value. Utilizing SS_{th}, upper and lower limits of the interval are determined, classifying the sample as a "true value" within this range. The interval is defined as a span around the predicted value $\hat{y}_{\text{pred}}(k)$, stretching from $(1 - SS_{th}) \times \hat{y}_{\text{pred}}(k)$ to $(1 + SS_{th}) \times y_{\text{pred}}(k)$. For each sample in the test dataset, if the $y_{reg}(k)$ falls within this interval, the sample is classified as a "true value". Conversely, if the $y_{reg}(k)$ lies outside the designated interval, the sample is classified as an "error".

Algorithm 5 PCA for data entry error detection methodology.

Input: Dataset, error proportion, error categories, significance level α.
Procedure:

1: Pre-process dataset (remove outliers, non-numeric variables, empty variables, rename columns, reset indexes).
2: Insert artificial errors according to error categories and error proportion.
3: Variable division: create $\mathbf{X}$ and $\mathbf{Y}$ for classification ($\mathbf{Y}_{\text{clas}}$).
4: Split the dataset into train (70%) and test (30%), selecting all errors for the test dataset.
5: Determine the number of PCs, l, that explain 90% of the cumulative explained variance (employing Equation (4.5)).
6: Determine SPE threshold, δ_α^2 (Equation (4.10)), and T^2 threshold, T_α^2 (Equation (4.14)), using the training dataset.
7: **for** each test sample $k = 1, \ldots, K$ **do**
8: Determine SPE (using Equation (4.9)) and T^2 (through Equation (4.13)) for each sample of the test dataset and classify $y_{clas}(k)$:
9: **if** $SPE <= \delta_\alpha^2$ and $T^2 <= T_\alpha^2$ **then**
10: $y_{clas}(k)$ is "true value"
11: **else**
12: $y_{clas}(k)$ is "error"
13: **end if**
14: Calculate the classification metrics.
15: **end for**
16: Save classification metrics.

Subsequently, using the actual label of the samples and the predicted classification, the confusion matrix is computed, and precision (Equation (5.5)), sensitivity (Equation (5.6)), f1-score (Equation (5.7)), and specificity (Equation (5.8)) are determined as outlined in Section 5.2. These results are then preserved for further analysis.

In cases where the error categories under investigation are specifically measurement errors, order errors or extra numbers, a more intricate analysis needs to be taken due to the existence of subcategories (as detailed in Section 5.1). In addition to the aforementioned metrics, the error records are isolated and further categorized into existing subcategories. For instance, in the case of measurement errors, they are split into errors from 5% to 10%, 10% to 25% and 25% to 50%. For each subcategory, the sensitivity is determined. Given that only positive samples ("errors") are analyzed in this scenario, sensitivity is the sole meaningful metric among the four used.

5.3.2 PCA

As detailed in Chapter 4, PCA is a valuable tool for fault detection. The main steps of this tailored approach are elucidated in Algorithm 5, which is based on Algorithm 3.

Steps 1 and 2:

The initial steps involve pre-processing the dataset and introducing artificial errors. These phases are integral components of the methodology employed for detecting data

entry errors with Soft Sensor, detailed thoroughly in Section 5.3.1. Two new columns are created: "classification" and "error category".

Step 3 and 4:

Upon artificially incorporating errors, the variables undergo division. The data is divided into $\mathbf{X}$, that contains the values of the field variables (X_j) and the target variable (Y), and $\mathbf{Y}$ for classification, denoted as $\mathbf{Y}_{\text{clas}}$, that contains the "classification" column only. Following this division, the samples are further segregated into training (70%) and test (30%) subsets. Once again, errors are entirely integrated into the test dataset, ensuring that the training phase exclusively utilizes "true value" samples.

Step 5

Subsequently, using solely the training dataset, Algorithm 5 is executed. In essence, the number of PCs, l, is determined based on the count required to explain 90% of the dataset variability as presented in Equation 4.5.

Step 6

The Principal Components are identified, and thresholds δ_α^2 and T_α^2 are calculated using Equation (4.10) and Equation (4.14), respectively.

Steps 7 to 15

Moving forward, for each sample in the test dataset, SPE and T^2 are computed using Equation (4.9) and Equation (4.13), respectively. If these values fall below their predetermined thresholds, the sample is classified as a "true value"; otherwise, it is deemed an "error".

Step 16

Following this classification for the entire test dataset, the confusion matrix and classification metrics, outlined in Section 5.2.2, are computed, encompassing precision, sensitivity, F1-score, and specificity.

To conclude, subcategories of measurement errors, order errors and extra numbers are scrutinized separately, employing the same approach elucidated in Section 5.3.1. For these categories, only "error" samples are considered, and sensitivity is the only metric calculated.

Chapter 6

Results

In this chapter, the results of the proposed methodologies in Algorithms 4 and 5 are presented and discussed. The chapter is organized into four sections. Section 6.1 provides a description of the considered datasets. Section 6.2 clarifies the experimental procedures and the input values used to evaluate the two methodologies. Subsequently, Section 6.3 details the results obtained for each methodology, and Section 6.4 analyzes in detail the outcomes for each dataset and algorithm.

6.1 Datasets

Three datasets were used to test the methodologies: one publicly available dataset (Concrete), another dataset from a real urban Wastewater Treatment Plant (WTP), and a third dataset provided by a cement industry (Cement).

Concrete dataset is a publicly available online benchmark dataset provided by the UCI Machine Learning Repository [75]. The dataset contains 1030 samples obtained from civil engineering practices [23] where the target variable, Y, the compressive strength of concrete, is measured in a laboratory and it's considered a function of eight field variables related to the concrete components and its age. The variables are described in Table 6.1.

Table 6.1: Variables of concrete dataset.

Variable	Description
X_1	Cement;
X_2	Blast furnace slag;
X_3	Fly ash;
X_4	Water;
X_5	Superplasticizer;
X_6	Coarse aggregate;
X_7	Fine aggregate;
X_8	Age of testing;
Y	Concrete compressive strength.

WTP dataset comes from a real-world urban wastewater treatment plant. It has 11 field variable (X_1, ..., X_{11}) measured by physical sensors and a target variable, Y, that is measured in a laboratory. Field variables correspond to physical attributes, such as pH, turbidity, and color of the water, and they are measured online by the plant sensors with a sampling time of 2 hours. The target variable is the fluoride concentration in the effluent

and it is determined by laboratory analysis once a day. All the variables are described in Table 6.2.

Table 6.2: Variables of water treatment plant dataset.

Variable	Description
X_1	Chlorine in the raw water;
X_2	Chlorine in the effluent;
X_3	Turbidity in the raw water;
X_4	Turbidity in the coagulated water;
X_5	Turbidity in the effluent;
X_6	pH in the raw water;
X_7	pH in the coagulated water;
X_8	pH in the effluent;
X_9	Color in the raw water;
X_{10}	Color in the coagulated water;
X_{11}	Color in the effluent;
Y	Fluoride in the effluent.

The wastewater treatment is a long process that can take up to 24 hours since the incoming water (called raw water) goes from the influent until it reaches the effluent, where Y is measured [76]. Consequently, variables assessed at the influent point (X_1, X_3, X_4, X_6, X_7, X_9 and X_{10}) must account for potential time lags within the range of 18-26 hours. Variables measured at the effluent (X_2, X_5, X_8 and X_{11}) may exhibit time lags ranging from 0-8 hours. Thus, the permissible time lags for variables are denoted by $n_{X_j} = \{9, 10, 11, 12, 13\}$ for the variables $X_j = 1, 3, 4, 6, 7, 9, 10$ and $n_{X_j} = \{0, 1, 2, 3, 4\}$ for the variables $X_j = 2, 5, 8, 11$, where n_{X_j} is the maximum time-lag considered for variable X_j [77]. Given these transformations, the number of input variables increases to $|\mathbf{X}| = 55$.

Lastly, the Cement dataset was provided by a cement factory and it comprises 263 samples and 23 field variables. The target variable (Y) is the compressive strength measured in a laboratory for 2 days. The compressive strength test is crucial in the cement industry for quality control and it is performed every few days. The field variables were assessed by physical sensors and encompassed the cement properties.

Table 6.3 provides a summary of the main characteristics of the dataset, where K denotes the number of samples, $|\mathbf{X}|$ represents the count of input variables, $\overline{Y}$ stands for the mean of variable Y and $min(Y)$ and $max(Y)$ present the minimum and maximum values of Y, respectively.

Table 6.3: Main characteristics of the datasets.

Dataset	K	$\lvert\mathbf{X}\rvert$	$\overline{Y}$	$min(Y)$	$max(Y)$
Concrete	1030	8	35,82	2,33	82,60
WTP	352	55	0,19	0,11	0,30
Cement	263	23	31,02	27,50	34,60

Table 6.4: Hyperparameters for the regression models to be chosen by the Grid Search procedure.

Models	**Hyperparameters**	
MLR	-	-
RR	λ	1e-10, 1e-4, 1e-3, 1e-2, 1, 2, 5, 10, 20
LASSO	λ	1e-10, 1e-4, 1e-3, 1e-2, 1, 2, 5, 10, 20
EN	λ	1e-10, 1e-4, 1e-3, 1e-2, 1, 2, 5, 10, 20
	α	0:1 (0.1)
SVR (linear)	C	1e-2, 1e-1, 1, 10, 100, 1000
	ϵ	1e-3, 1e-2, 1e-1
SVR (rbf)	C	1e-2, 1e-1, 1, 10, 100, 1000
	ϵ	1e-3, 1e-2, 1e-1
	γ	1e-4, 1e-3, 1e-2, 1e-1, 1, 10
GMM	G	5, 10, 20, 50

6.2 Experimental tests

Having established the methodologies and presented the datasets, the next step involves the model's validation and performance assessment in detecting data entry errors. This section delineates the experimental setup devised for testing both methodologies. Subsection 6.2.1 elucidates the experiments conducted with the SS-based methodology, encompassing the framework designed for SS in Chapter 3, while Subsection 6.2.2 outlines the tests conducted using the PCA-based methodology for data entry error detection.

6.2.1 Experimental setup for the Soft Sensors methodology

To evaluate the performance of the SS-based methodology in detecting data entry errors, Algorithm 4 must be executed. Since the first step consists in selecting the SS model that better fits the dataset in analysis, before elaborating on the experimental structure, it's important to outline the SS model selection process.

For each dataset detailed in Section 6.1, the optimal SS model is determined through an exhaustive exploration of variable selection methods and regression models introduced in Chapter 3. Algorithm 2 guides these tests. Each dataset undergoes the data pre-processing steps and feature expansion is applied. Tests are performed with 1) no variable selection (using only the original input variables), 2) original variables plus expanded ones without selection (only applied to Concrete dataset, given the WTP and Cement datasets extensive number of variables), and 3) selection methods detailed in Section 3.4 — Pearson's correlation, MI, and fastTracker — applied to the expanded datasets. Subsequently, the possible hyperparameters for each model are defined, as presented in Table 6.4, to be fine-tuned using Grid Search and k-Fold Cross Validation, with $k = 10$.

Training encompasses every regression model from Section 3.5: MLR, RR, LASSO, EN, SVR with linear kernel, SVR with Radial Basis Function (RBF) kernel, and GMR. The training and testing are performed 30 times ($it_{max} = 30$) for each variable selection method and regression model. In each iteration, the dataset is split randomly into 70% for training and 30% for testing, and the average metrics (NRMSE and R^2 as described in Section 5.2) are calculated across all iterations to ensure reliable results independent of dataset partitioning.

The SS model with the optimal combination of both mean NRMSE and mean R^2 is identified by applying Equation (5.4) and selecting the combination with the lowest value.

After establishing the SS model, rigorous testing is conducted to evaluate its effectiveness in detecting various data entry errors using Algorithm 4. This evaluation covers all defined error categories: blank spaces, doubles, extra number errors, measurement errors, and order errors, as detailed in Section 5.1. These categories are simulated by introducing errors into the selected datasets, replacing a portion of the original samples. Thus, six variations were created for each dataset outlined in Section 6.1: one for each error category, containing exclusively that specific error type, and a general simulation comprising all error categories in equal proportion.

With an error proportion set at 15%, the dataset is divided into training and testing sets, with a 70% and 30% split, respectively. Hence, half of the test dataset exclusively comprised error samples, as all error samples were included in the testing set. This approach ensures a balanced representation of categories (true values and errors) for the subsequent classification process. Figure 6.1 illustrates the test datasets with inserted errors encompassing all categories, depicted as red crosses, against their original values shown as green crosses, allowing for a direct comparison. True values are additionally marked as black dots for comprehensive visualization of the entire test dataset.

The model is trained again using the training datasets and applied to predict Y for the test dataset, using NRMSE and R^2 as regression metrics. In the classification phase, four potential values for the threshold SS_{th} are considered: 0.05, 0.1, 0.15 and 0.2. These threshold values were selected based on the NRMSE of the selected models (exposed in Subsection 6.3.1). At last, after the classification step is performed, the precision, sensitivity, f1-score and specificity are calculated. For tests with only measurement errors, extra number errors and order errors inserted, a more detailed analysis is conducted, and sensitivity is calculated for the existent error's subcategories.

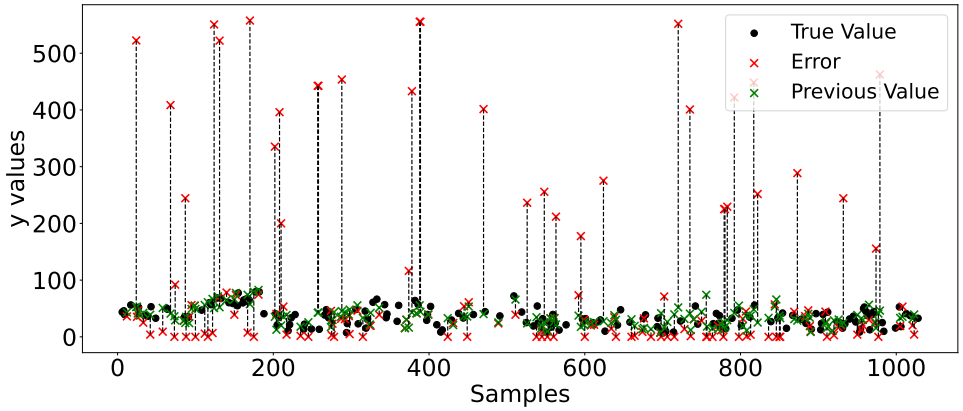
6.2.2 Experimental setup for the PCA methodology

To evaluate the effectiveness of the PCA methodology detailed in Section 5.3.2, experiments were conducted in accordance with Algorithm 5. Similar to the procedure for the SS methodology, six variants of each dataset described in Section 6.1 were generated and tested, five representing each error category exclusively and a general simulation with all error categories equally represented.

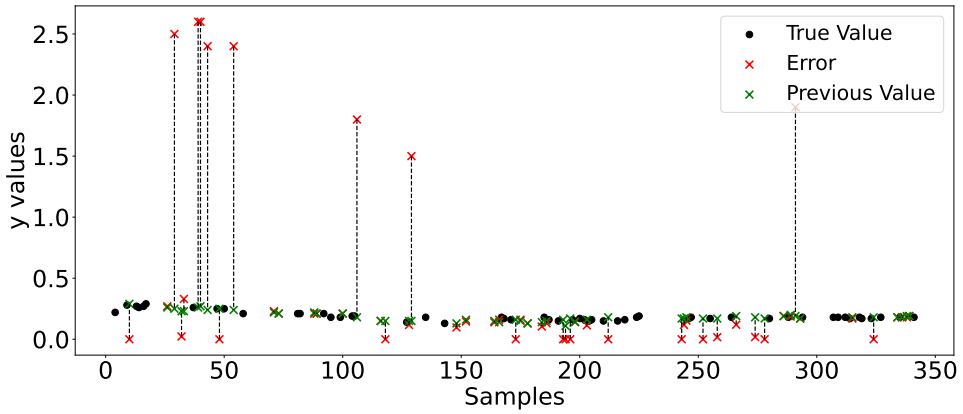
A significance level $\alpha = 0.05$ was considered when determining the SPE threshold, δ^2_α, ad the T^2 threshold, T^2_α. The evaluation metrics mirrored those chosen for assessing the SS methodology, namely precision, sensitivity, f1-score, and specificity, facilitating a direct comparison between the methodologies. Furthermore, a more detailed analysis was performed for measurement error, extra number errors and order errors using the sensitivity metric.

6.3 Experimental results

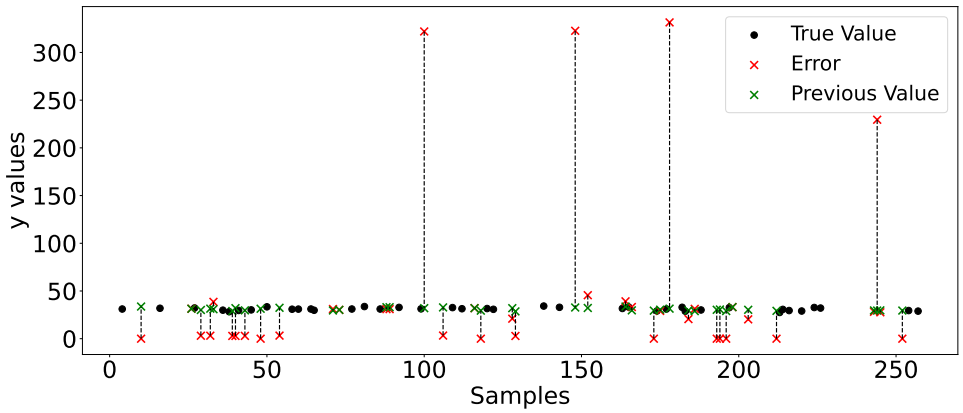
This section provides the results of the experimental tests outlined in Section 6.2, which are divided into two parts. Subsection 6.3.1 presents the outcomes of experiments conducted with the Soft Sensor methodology, subdivided into regression results (Sub-subsection 6.3.1.1) and classification results for data entry detection (Sub-subsection



(a) Insertion of errors in the Concrete dataset.



(b) Insertion of errors in the WTP dataset.



(c) Insertion of errors in the Cement dataset.

Figure 6.1: Insertion of errors in the selected datasets.

6.3.1.2). Lastly, Section 6.3.2 presents the results obtained using the PCA methodology for data entry error detection.

6.3.1 Soft Sensors results

6.3.1.1 Regression results

Tables 6.5 and 6.6 display the outcomes of regression tests conducted to determine the optimal Soft Sensor model for each dataset. Table 6.5 showcases the R^2 metric, while

Table 6.5: R^2 results for each dataset.

Dataset	Variable Selection		|X|	MLR	RR	LASSO	EN	SVR (linear)	SVR (rbf)	GMR
Concrete	without VS		8	0,604	0,604	0,604	0,604	0,581	0,865	0,716
	FE	-	57	0,869	0,869	0,859	0,858	0,852	**0,908**	-48,641
		PC = 0,2	32	0,842	0,842	0,837	0,837	0,824	0,886	-4,916
		PC = 0,3	26	0,835	0,834	0,834	0,834	0,826	0,887	0,619
		PC = 0,4	11	0,726	0,726	0,726	0,726	0,713	0,829	0,777
		MI = 1,0	26	0,406	0,407	0,410	0,410	0,396	0,420	-0,481
		MI = 1,3	14	0,589	0,589	0,579	0,579	0,542	0,790	0,634
		fT = 0,5	26	0,832	0,832	0,820	0,821	0,807	0,883	-10,762
		fT = 0,6	13	0,572	0,571	0,573	0,573	0,534	0,810	-1,862
WTP	without VS		55	0,616	0,690	0,695	0,693	0,643	**0,758**	0,553
	FE	PC = 0,7	52	0,522	0,621	0,598	0,592	0,553	0,689	-6,101
		PC = 0,72	26	0,546	0,567	0,553	0,569	0,532	0,674	-1,089
		PC = 0,73	11	0,554	0,565	0,557	0,567	0,543	0,679	0,398
		MI = 0,85	51	0,626	0,695	0,680	0,680	0,676	0,712	-0,326
		MI = 0,87	22	0,668	0,683	0,676	0,682	0,662	0,701	0,042
		fT = 0,725	46	0,619	0,667	0,663	0,665	0,665	0,661	-2,060
		fT = 0,73	21	0,578	0,575	0,576	0,572	0,571	0,598	0,098
Cement	without VS		22	0,378	0,387	**0,395**	0,394	0,329	0,336	0,200
	FE	PC = 0,3	60	0,076	0,359	0,369	0,350	0,213	0,318	-1,378
		PC = 0,4	22	0,343	0,351	0,349	0,354	0,310	0,345	-2,445
		PC = 0,45	13	0,305	0,299	0,308	0,309	0,251	0,259	-0,130
		MI = 2,55	37	0,252	0,362	0,345	0,353	0,246	0,337	-10,946
		MI = 2,62	14	0,304	0,313	0,298	0,310	0,237	0,277	-0,044
		fT = 0,93	31	0,302	0,362	0,337	0,358	0,267	0,353	-3,215
		fT = 0,935	10	0,329	0,324	0,326	0,325	0,306	0,326	0,128

Table 6.6 presents the NRMSE results. In the '**Variable Selection**" column of both tables, "without VS" denotes the use of original input variables, while "FE" indicates the use of expanded variables. The labels "PC", "MI", and "fT" with their respective values represent thresholds chosen for Pearson's correlation, Mutual Information, and fastTracker, respectively. The third column ($|\mathbf{X}|$) indicates the number of selected variables. The remaining columns list the tested regression models. Bold values highlight the best metrics for each dataset, indicating the optimal SS model based on the table's metric. Thus, the highest R^2 values imply superior results in Table 6.5, while in Table 6.6 bold values represent the lowest NRMSE metrics. Further discussion on these findings is provided in Section 6.4.

Figure 6.2 contains the results of the predictions made with the best model (from the thirty iterations) found for each regression model for 80 random samples of the test dataset.

Based on the results provided by Tables 6.5 and 6.6, the optimal Soft Sensor model was determined using Equation (5.4), selecting the combination with the lowest value. The successful combinations for each dataset were:

- Concrete: SVR with RBF kernel with FE;
- WTP: SVR with RBF kernel without variable selection;
- Cement: LASSO without variable selection.

Table 6.6: NRMSE results for each dataset.

Dataset	Variable Selection		|X|	MLR	RR	LASSO	EN	SVR (linear)	SVR (rbf)	GMR
Concrete	without VS		8	0,130	0,130	0,130	0,130	0,134	0,076	0,110
	FE	-	57	0,075	0,075	0,078	0,078	0,079	**0,063**	0,929
		PC = 0,2	32	0,082	0,082	0,083	0,083	0,087	0,070	0,264
		PC = 0,3	26	0,084	0,084	0,084	0,084	0,086	0,069	0,111
		PC = 0,4	11	0,108	0,108	0,108	0,108	0,111	0,085	0,097
		MI = 1,0	26	0,133	0,133	0,134	0,134	0,140	0,095	0,295
		MI = 1,3	14	0,159	0,159	0,159	0,159	0,161	0,158	0,242
		fT = 0,5	26	0,085	0,085	0,087	0,087	0,091	0,070	0,171
		fT = 0,6	13	0,135	0,135	0,135	0,135	0,141	0,090	0,124
WTP	without VS		55	0,126	0,111	0,111	0,111	0,121	**0,100**	0,132
	FE	PC = 0,7	52	0,137	0,121	0,126	0,126	0,132	0,111	0,474
		PC = 0,72	26	0,137	0,132	0,132	0,132	0,137	0,116	0,263
		PC = 0,73	11	0,132	0,132	0,132	0,132	0,137	0,116	0,153
		MI = 0,85	51	0,121	0,111	0,116	0,116	0,111	0,111	0,216
		MI = 0,87	22	0,116	0,111	0,111	0,111	0,116	0,111	0,179
		fT = 0,725	46	0,121	0,116	0,116	0,116	0,116	0,116	0,332
		fT = 0,73	21	0,132	0,132	0,132	0,132	0,132	0,126	0,179
Cement	without VS		22	0,159	**0,156**	**0,156**	**0,156**	0,157	0,157	0,157
	FE	PC = 0,3	60	0,197	0,158	0,155	0,161	0,162	0,161	0,256
		PC = 0,4	22	0,162	0,156	0,155	0,155	0,154	0,157	0,175
		PC = 0,45	13	0,158	0,156	0,156	0,156	0,159	0,154	0,172
		MI = 2,55	37	0,186	0,164	0,161	0,162	0,171	0,165	0,244
		MI = 2,62	14	0,172	0,172	0,169	0,169	0,170	0,169	0,191
		fT = 0,93	31	0,166	0,163	0,164	0,163	0,166	0,163	0,206
		fT = 0,935	10	0,171	0,167	0,171	0,169	0,171	0,170	0,178

Table 6.7: Regression results for each dataset.

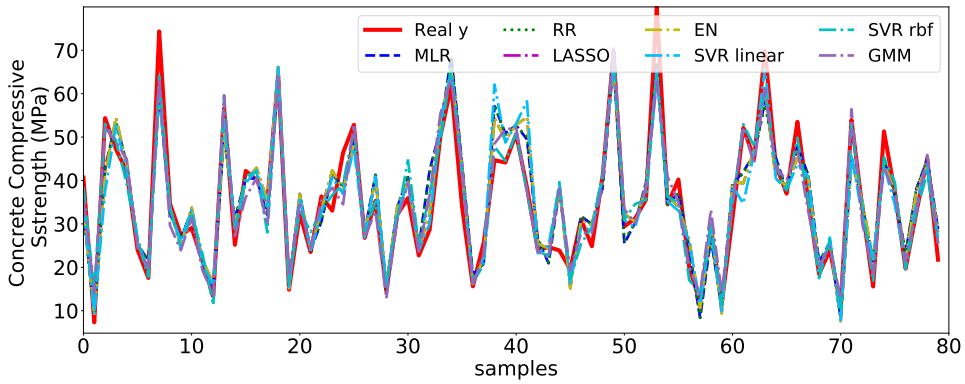
Dataset	RMSE	NRMSE	R^2
Concrete	4,85	0,06	0,91
WTP	0,016	0,09	0,83
Cement	1,20	0,18	0,44

6.3.1.2 SS results for data entry errors detection

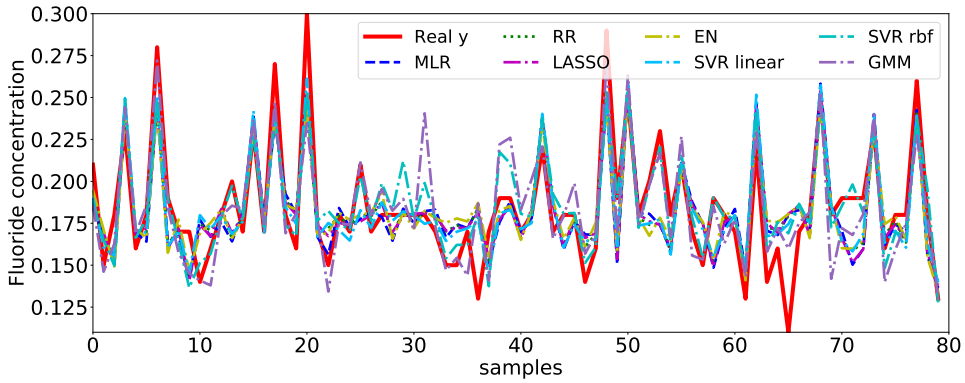
The selected SS models were trained and tested again when applying Algorithm 4. The regression metrics RMSE, NRMSE, and R^2 were saved for each dataset, as presented in Table 6.7.

Table 6.8 and Table 6.9 depict the classification outcomes from the implementation of the SS-based methodology across different dataset variations. While Table 6.8 provides insights into precision, sensitivity, and f1-score, Table 6.9 specifically focuses on specificity. Metrics were separated in distinct tables given that specificity is influenced solely by actual true values, making it inappropriate to analyze alongside the "**Error categories**" column that denotes the type of error introduced in Table 6.8. For both tables, the "$\mathbf{SS_{th}}$" column denotes the applied threshold for the classification process, as stipulated in Algorithm 4, with subsequent columns showcasing the employed classification metrics.

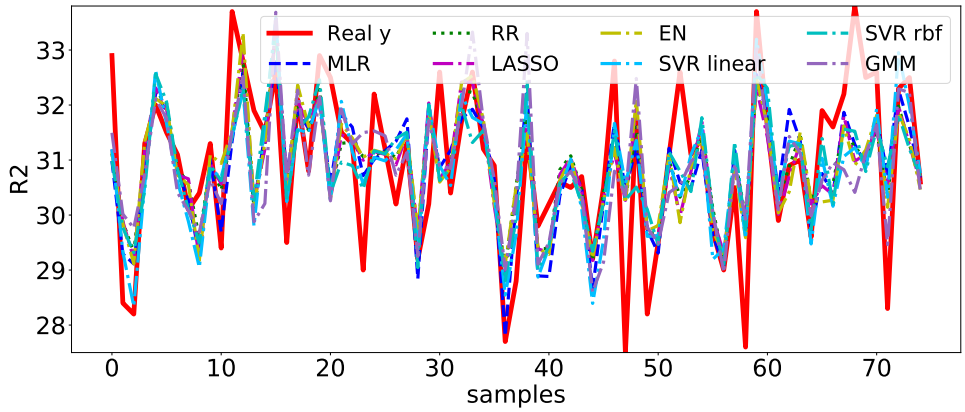
To complement the interpretation of the results presented in Table 6.8, a series of



(a) SS predictions for Concrete dataset.



(b) SS predictions for WTP dataset.



(c) SS predictions for Cement dataset.

Figure 6.2: Predictions made with the best model found for each regression model.

plots were generated. Figures 6.3, 6.4 and 6.5 specifically focus on actual errors. Each plot illustrates the original value of the sample, accompanied by the corresponding SS predicted value. The margins for classifying a sample as a "true value" are also depicted based on the defined SS_{th}. Additionally, the inserted error values are represented as crosses, distinguishing between values correctly classified as errors (with the color red) and false negatives (with the color black). These plots offer valuable insights into the model's behavior, and provide a visual aid in understanding the predictive capabilities of the SS. Figures 6.3, 6.4, and 6.5 depict the analysis of datasets containing all error categories, and to enhance visibility, the y-axis has been scaled. This adjustment was necessary due to the presence of samples with significantly higher values. For plots with

Table 6.8: Classification results for data entry error detection using the Soft Sensor methodology.

Dataset	Error categories	SS_{th}	Precision	Sensitivity	F1-score
Concrete	All categories	0,05	0,62	0,93	0,74
		0,1	0,71	0,88	0,78
		0,15	0,76	0,85	0,8
		0,2	0,84	0,79	0,81
	Blank spaces	0,05	0,63	1	0,78
		0,1	0,73	1	0,85
		0,15	0,79	1	0,88
		0,2	0,87	1	0,93
	Doubles	0,05	0,6	0,86	0,7
		0,1	0,67	0,74	0,71
		0,15	0,71	0,65	0,68
		0,2	0,79	0,57	0,66
	Extra number	0,05	0,59	0,84	0,69
		0,1	0,67	0,74	0,71
		0,15	0,72	0,67	0,69
		0,2	0,8	0,63	0,7
	Measurement error	0,05	0,59	0,84	0,69
		0,1	0,67	0,75	0,71
		0,15	0,69	0,6	0,64
		0,2	0,75	0,46	0,57
	Order error	0,05	0,63	0,99	0,77
		0,1	0,73	0,99	0,84
		0,15	0,79	0,99	0,88
		0,2	0,86	0,98	0,92
WTP	All categories	0,05	0,61	0,79	0,69
		0,1	0,72	0,72	0,72
		0,15	0,92	0,62	0,74
		0,2	0,97	0,57	0,71
	Blank spaces	0,05	0,66	1	0,8
		0,1	0,78	1	0,88
		0,15	0,95	1	0,97
		0,2	0,98	1	0,99
	Doubles	0,05	0,49	0,49	0,49
		0,1	0,42	0,21	0,28
		0,15	0,5	0,06	0,1
		0,2	0,67	0,04	0,07
	Extra number	0,05	0,57	0,68	0,62
		0,1	0,59	0,42	0,49
		0,15	0,84	0,3	0,44
		0,2	0,93	0,26	0,41
	Measurement error	0,05	0,63	0,87	0,73
		0,1	0,72	0,74	0,73
		0,15	0,91	0,6	0,73
		0,2	0,97	0,57	0,71
	Order error	0,05	0,66	1	0,8
		0,1	0,78	1	0,88
		0,15	0,95	1	0,97
		0,2	0,98	1	0,99
Cement	All categories	0,05	0,82	0,79	0,81
		0,1	1	0,72	0,84
		0,15	1	0,67	0,8
		0,2	1	0,67	0,8
	Blank spaces	0,05	0,85	1	0,92
		0,1	1	1	1
		0,15	1	1	1
		0,2	1	1	1
	Doubles	0,05	0,61	0,28	0,39
		0,1	1	0,08	0,14
		0,15	0	0	0
		0,2	0	0	0
	Extra number	0,05	0,82	0,79	0,81
		0,1	1	0,74	0,85
		0,15	1	0,74	0,85
		0,2	1	0,74	0,85
	Measurement error	0,05	0,84	0,95	0,89
		0,1	1	0,74	0,85
		0,15	1	0,56	0,72
		0,2	1	0,49	0,66
	Order error	0,05	0,85	1	0,92
		0,1	1	1	1
		0,15	1	1	1
		0,2	1	1	1

predictions for datasets with a single error category refer to Appendix B.1.

Additionally, the classification outcomes for datasets with all error categories are illustrated in Figures 6.6, 6.7 and 6.8. In these figures, symbols denote the actual category of the samples, with crosses representing errors and dots representing true values. The color of each symbol corresponds to its classification by the model, where green indicates a correct classification, and red denotes an incorrect one. As clarified in Section 5.2, TP refers to True Positives, signifying errors correctly classified as such, TN corresponds to

Table 6.9: Specificity for data entry error detection using the Soft Sensor methodology.

Dataset	$\mathbf{SS_{th}}$	Specificity
Concrete	0,05	0,42
	0,1	0,64
	0,15	0,74
	0.2	0,85
WTP	0,05	0,49
	0,1	0,72
	0,15	0,94
	0,2	0,98
Cement	0,05	0,82
	0,1	1
	0,15	1
	0,2	1

True Negatives, representing true values correctly classified, FP stands for False Positives, indicating samples wrongly classified as errors, and FN represents False Negatives, which are errors misclassified as true values. Furthermore, predictions for datasets with individual error categories can be found in Appendix B.2.

Subsequently, a more comprehensive examination was carried out, focusing on datasets containing only measurement errors, extra number errors and order errors. In Table 6.10, the sensitivity is presented for datasets exclusively filled with measurement errors. The results are further categorized based on the subtypes of errors, specifically errors between 5 to 10%, 10 to 25%, and 25 to 50%, and column "$\mathbf{SS_{th}}$" presents the threshold applied. Additionally, the table provides the count of errors inserted by subcategory (column "$\mathbf{N}_{\text{error}}$").

Lastly, Tables 6.11 and 6.12 present the sensitivity of the model for each dataset, considering the error subtypes resulting from extra number errors and order errors, respectively. In "**Error subtype**" column, "o.m." stands for order of magnitude. Thus, if the error subtype is "-1 o.m.", it indicates a change of -1 in the order of magnitude (for example, the original value was 100 and the inserted error is 10). Additionally, the tables provide the count of errors inserted by subcategory ("$\mathbf{N}_{\text{error}}$").

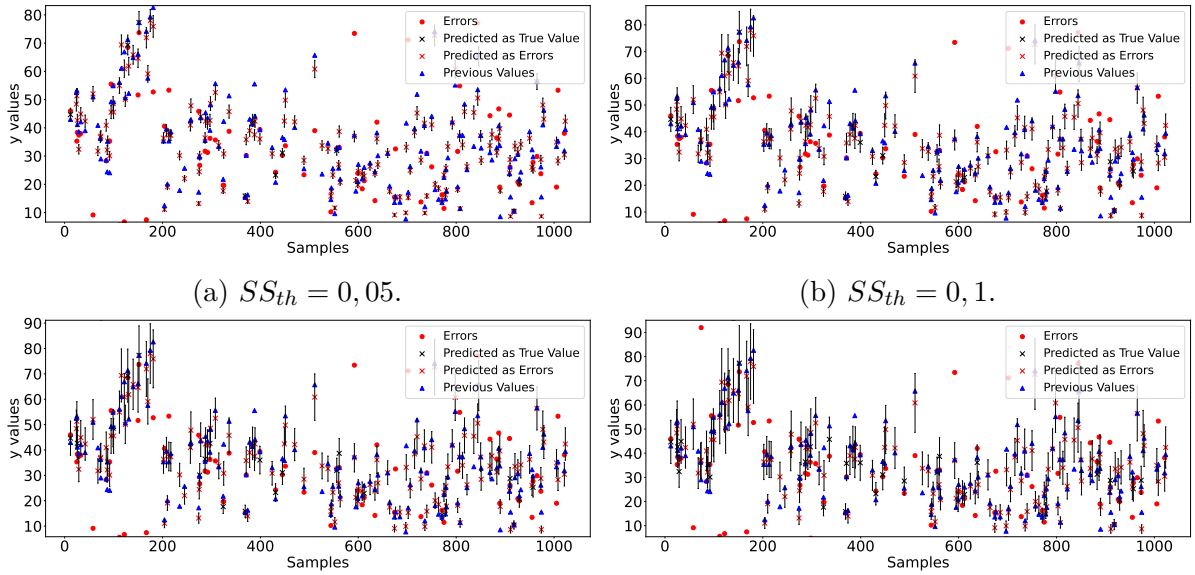
6.3.2 PCA results for data entry errors detection

Through the cumulative explained variance calculated using Equation (4.5), the number of PCs chosen for each dataset was the following:

- Concrete: $l = 6$;
- WTP: $l = 14$;
- Cement: $l = 12$.

Figure 6.9 presents the cumulative explained variance by the number of PCs for each dataset.

The composition of Principal Components holds crucial information for further conclusions and explaining results in the future. Therefore, Tables 6.13, 6.14 and 6.15 present



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

Figure 6.3: SS predictions and classification for error samples in Concrete dataset with all error categories (scaled).

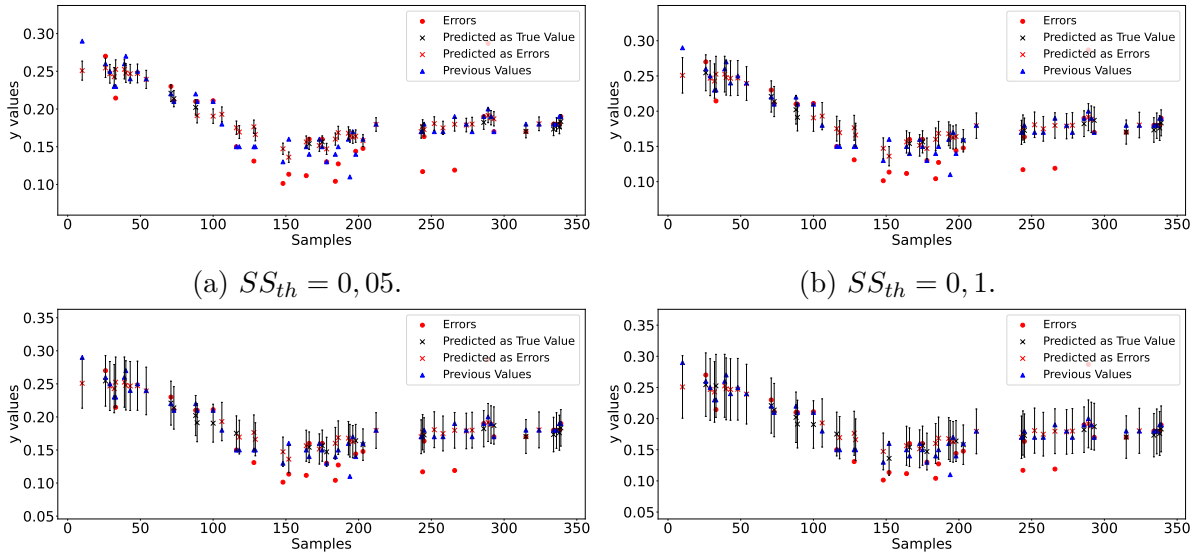
the five most influential variables for each selected PC along with their corresponding weights. The weight serves as a measure of variance, indicating the extent to which a variable contributes to the PC. A larger absolute value signifies greater relevance, and a positive value suggests a positive correlation, while a negative value implies a negative relationship.

The results of the experiments conducted with the PCA methodology to detect data entry errors are presented in Tables 6.16 and 6.17. Table 6.16 presents precision, sensitivity and f1-score and Table 6.17 shows the obtained specificity.

Moreover, to enrich the analysis of Tables 6.16 and 6.17, classification results can be visualized in various plots. Figures 6.10a, 6.10b and 6.10c depict the classification outcomes for datasets with all error categories included, and Appendix B.3 encompasses the plots illustrating the classification results for data entry errors detection for datasets with individual error categories using the PCA methodology. Once again, TP, TN, FP, and FN adhere to the definitions outlined in Section 5.2, symbols represent the actual category of a sample (crosses for errors and dots for true values), and color is green if the model classification is correct and red if it is incorrect.

A comprehensive examination of the subcategories within measurement errors, extra number errors, and order errors was conducted, replicating the analysis performed in Section 6.3.1.2. Thus, Table 6.18 presents the sensitivity for a dataset exclusively comprising measurement errors. The results are further differentiated based on the subtypes of errors and the table includes the count of errors inserted by subcategory (column "$\mathbf{N}_{\text{error}}$") as well.

Lastly, Tables 6.19 and 6.20 present the performance details of the PCA methodology in detecting subtypes of errors from the extra number error and order error categories, respectively. The analysis takes into account the resulting changes in the order of magnitude of the Y values. Both tables provide the sensitivity of the model for each dataset, considering the error subtypes (changes in the order of magnitude, "o.m."), along with the count of errors inserted by each subcategory (column "$\mathbf{N}_{\text{error}}$").



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

Figure 6.4: SS predictions and classification for error samples in WTP dataset with all error categories (scaled).

6.4 Discussion

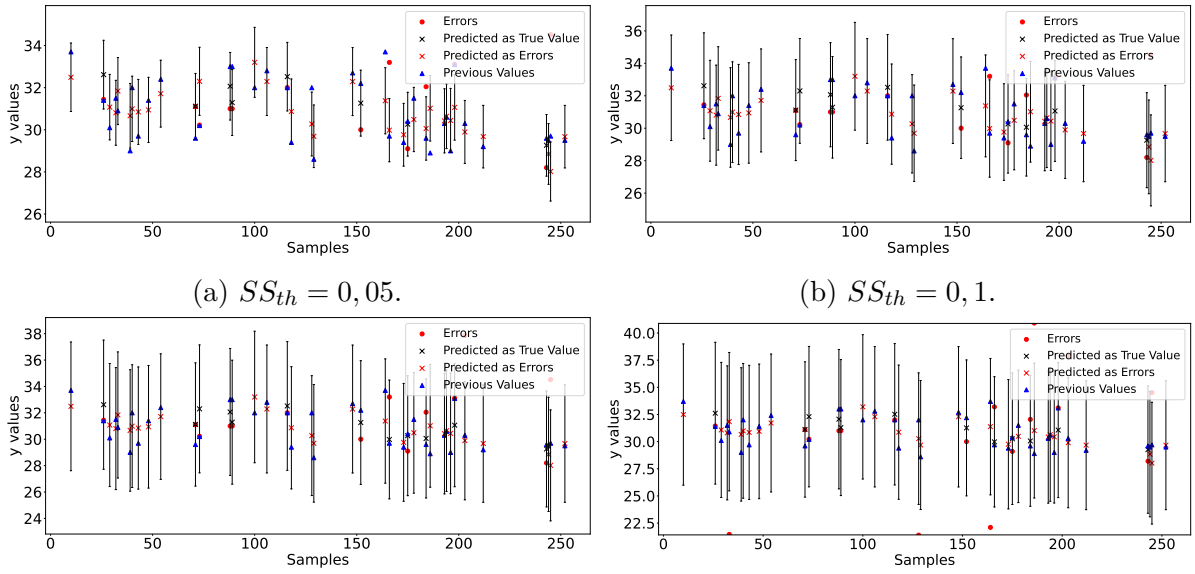
This section delves into the analysis and interpretation of the results presented in Section 6.3 and it is divided into three subsections. Subsection 6.4.1 analyzes the results achieved using the Soft Sensor-based methodology, Subsection 6.4.2 focuses on the outcomes obtained through the PCA-based methodology, and the last one, Subsection 6.4.3, undertakes a comparative analysis of the two methodologies, followed by an overall discussion about their efficacy. Moreover, the initial subsection is split into two segments. The first segment (Subsubsection 6.4.1.1) delves into the analysis of the developed Soft Sensor framework and the comparison tests conducted. The second segment (Subsubsection 6.4.1.2) focuses on evaluating the efficacy of the Soft Sensor-based methodology in detecting data entry errors.

6.4.1 Analysis of Soft Sensor results

6.4.1.1 Analysis of Soft Sensors regression results

Table 6.6 provides insights into the performance of different regression models across various datasets. Particularly, for the Concrete and WTP datasets, the SVR model with RBF kernel stands out, showcasing the lowest NRMSE values across all variable selection methods. Notably, this model achieved remarkable scores of 6.3% for the Concrete dataset with expanded variables and 10% for the WTP dataset without any variable selection. Although the performance on the Cement dataset was not as robust as the other two, the linear models with penalizations (RR, LASSO and EN) demonstrated the lowest NRMSE values at 15.6%.

Table 6.5 highlights the variability in the R^2 values across different datasets. For the Concrete dataset, the SVR model with a RBF method stands out with an excellent value of 0.908, showcasing high performance. The WTP dataset presents more consistent but generally mediocre results, where the SVR model with a RBF kernel again leads with a



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

Figure 6.5: SS predictions and classification for error samples in Cement dataset with all error categories (scaled).

value of 0.758. However, the R^2 values for the Cement dataset are notably low, indicating that the models may not adequately explain the variability within the target variable. This could be due to the models not being a good fit for the data or the field variables not being strongly correlated with the response variable.

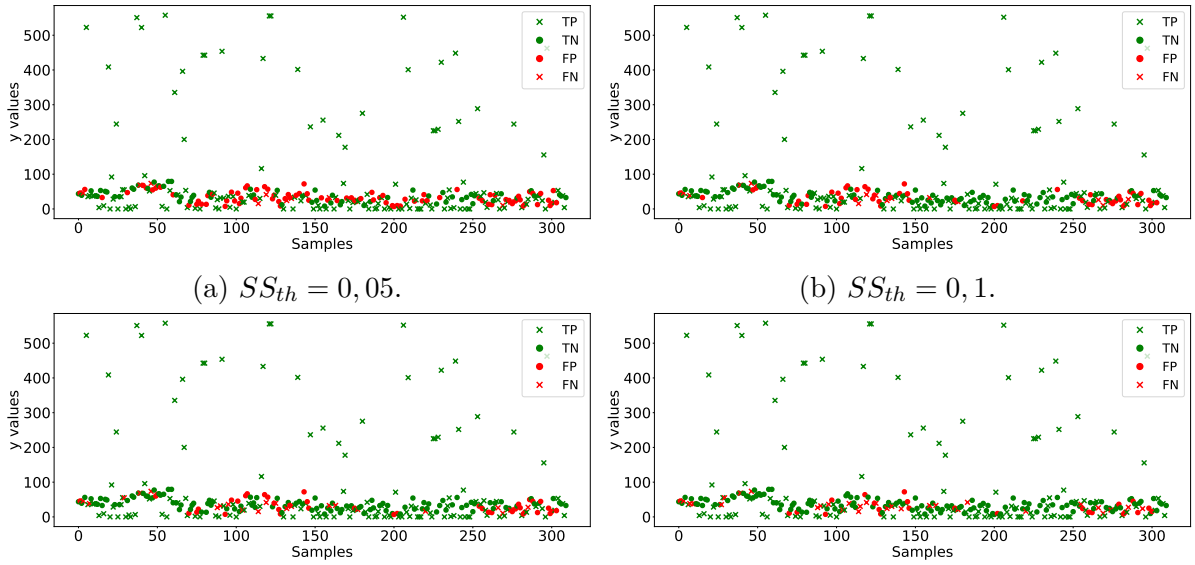
From the analysis of the results, it is evident that SVR with an RBF kernel consistently outperforms the other models, whereas GMR generally underperforms and does not align well with the use case at hand.

Regarding variable selection, for the Concrete dataset, the best outcomes were achieved using expanded variables, while for WTP and Cement datasets, no variable selection method yielded the best results. However, it is challenging to definitively rank the variable selection methods as the efficacy varies across datasets. Notably, the fastTracker method stands out for its speed. Additionally, certain regression models are more reliant on input variables than others; for instance, the performance of GMR significantly varies with the number of input variables, usually exhibiting worse predictions with higher $|X|$. In the case of the Concrete dataset, the variable expansion notably improved model performance.

In conclusion, the selected SS models demonstrate excellent results for the Concrete and WTP datasets, indicating a strong fit with the data. However, for the Cement dataset, the model displayed a reasonable NRMSE but a low R^2, hinting at potentially inferior error detection capabilities for this dataset.

6.4.1.2 Analysis of results using the Soft Sensors-based methodology

In this section, the performance of the Soft Sensor methodology in detecting data entry errors is analyzed, based on the results presented in Subsection 6.3.1. To facilitate comprehension, the analysis follows a structured approach: each metric is assessed across the three datasets and the different error categories, culminating in a concise summary stating the most relevant patterns and insights.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

Figure 6.6: Classification results for Concrete dataset with all error categories, using Soft Sensors.

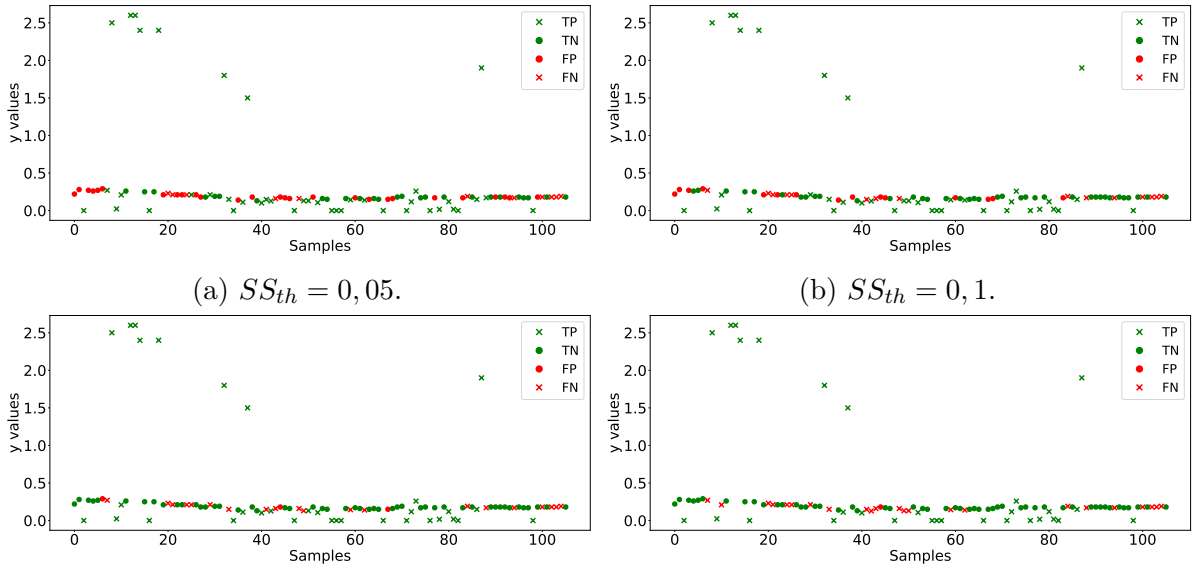
Precision

According to Table 6.8, precision exhibits elevated values, displaying slight variations across different datasets. Since precision has into account TP and FP (Equation (5.5)), this high scores indicate the methodology is good at not classifying actual true values as errors. In the Concrete dataset, precision presents moderate to good results, ranging between 0.6 and 0.8. For the WTP dataset precision presents better overall scores with high variability (ranging nearly from 0,6 to 0,98) for each error category, except for the double category, which displays lower scores.

Notably, in these two datasets (except for doubles in WTP case) an upward trend in precision is observed with increasing classification threshold (SS_{th}). This aligns with expectations, as a larger threshold reduces false positives (actual true values misclassified as errors). In fact, false positives depend solely on the threshold, since actual true values are unaffected by the error category inserted. Figure 6.11 visually depicts this trend, showcasing an actual true value, a Soft Sensor prediction, and intervals defined by two SS_{th} values. In this scenario, a threshold of 0.05 misses the true value classification, while 0.15 successfully captures it. Thus, the variability in precision across error categories reflects the variability within True Positives and hence, the model's capability to accurately identify various error types.. For Concrete dataset, the model appears slightly more proficient in handling all error categories, exclusively blank spaces, and order errors, although the distinction is not substantial compared to other error categories. For WTP, the results are consistent between error categories, except for doubles, as previously mentioned.

The doubles exception in WTP dataset can be attributed to the limited variability in Y values, making the difference between errors and preceding values imperceptible, as depicted in Figure B.5. With higher thresholds, more errors are misclassified, consistent which the observed decrease in sensitivity (higher number of false negatives).

In the Cement dataset, precision consistently reaches one (indicating no FP detected) for all applied errors except doubles and for all applied thresholds except $SS_{th} = 0,05$.



(a) $SS_{th} = 0,05$.

(b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$.

(d) $SS_{th} = 0,2$.

Figure 6.7: Classification results for WTP dataset with all error categories, using Soft Sensors.

Doubles stand out due to interesting results. For $SS_{th} = 0.05$, it presents a reasonable value, indicating the detection of some true positives along with some false positives. For $SS_{th} = 0.1$, precision is maximum with no false positives, despite low sensitivity. For higher thresholds, both precision and sensitivity are zero, indicating no error is well classified. This indicates that although higher thresholds lead to less FP, they can also lead to less TP and higher FN. It's essential to note the small dataset size with only 39 actual error samples, introducing higher uncertainty in metrics like precision and sensitivity.

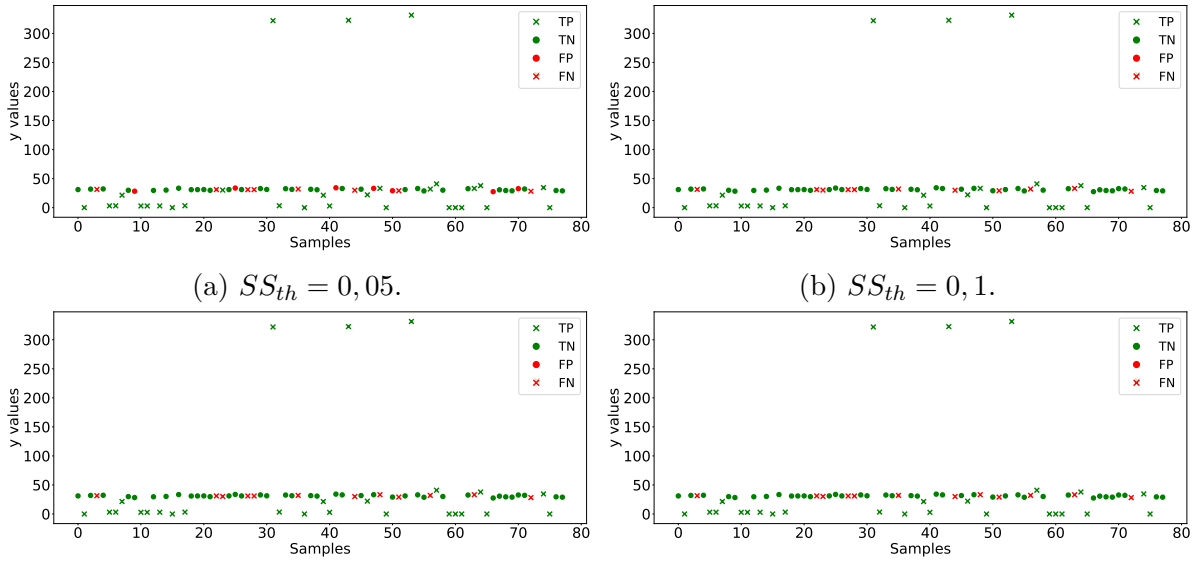
Sensitivity

Shifting the focus to sensitivity, Tables 6.8, 6.10, 6.12 and 6.11 present generally favorable results, despite with variations between datasets. A consistent trend observed across all datasets in Table 6.8 is that, for variants with blank spaces and order errors, sensitivity consistently registers one or nearly one, suggesting that in such cases, no actual error was mistakenly classified as a true value (avoiding false negatives). This highlights the model's ability to recognize a these two error categories.

Another pattern captured in Table 6.8 is that sensitivity has a decline with increasing SS_{th} across all categories, except for blank spaces and order errors. This implies a rise in false negatives (actual errors misclassified of true values) with wider classification intervals for true values (caused by larger thresholds). This can be visualized when comparing Figure B.10a to Figure B.10d, for example.

Overall, the three datasets demonstrate great performance in tests covering all error categories, with Concrete achieving the best results (ranging from 0.79 to 0.93), and good results were also achieved for the tests with lower thresholds with the remaining error categories. However, exceptions to this pattern are evident, with doubles in WTP and Cement datasets, and tests involving extra number errors in WTP, resulting in poor performance, especially in cases of doubles.

Upon closer examination of these discrepancies, the doubles can again be attributed



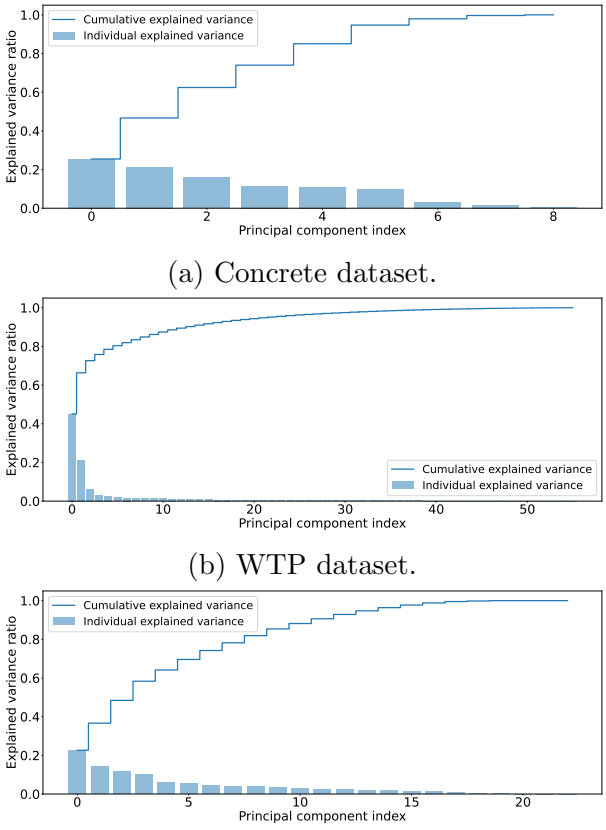
(a) $SS_{th} = 0,05$.

(b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$.

(d) $SS_{th} = 0,2$.

Figure 6.8: Classification results for Cement dataset with all error categories, using Soft Sensors.



(a) Concrete dataset.

(b) WTP dataset.

(c) Cement dataset.

Figure 6.9: Cumulative explained variance by the number of PCs for each dataset.

to the limited variability in Y values for WTP and Cement, leading to increased false negatives as more samples fall within the "true value" classification with higher thresholds. For the Cement dataset, Figure B.6 shows that almost all error samples are misclassified

Table 6.10: Measurement error analysis using the Soft Sensor methodology.

Dataset	Error subtype	N_{error}	SS_{th}	Sensitivity
Concrete	5 to 10 %	51	0,05	0,69
			0,1	0,47
			0,15	0,27
			0,2	0,16
	10 to 25 %	51	0,05	0,84
			0,1	0,78
			0,15	0,57
			0,2	0,33
	25 to 50 %	53	0,05	0,98
			0,1	0,98
			0,15	0,94
			0,2	0,87
WTP	5 to 10 %	17	0,05	0,65
			0,1	0,29
			0,15	0,06
			0,2	0,06
	10 to 25 %	17	0,05	0,94
			0,1	0,88
			0,15	0,71
			0,2	0,59
	25 to 50 %	19	0,05	1
			0,1	1
			0,15	1
			0,2	1
Cement	5 to 10 %	13	0,05	0,85
			0,1	0,23
			0,15	0
			0,2	0
	10 to 25 %	13	0,05	1
			0,1	1
			0,15	0,69
			0,2	0,46
	25 to 50 %	13	0,05	1
			0,1	1
			0,15	1
			0,2	1

as "true value", particularly for higher values of SS_{th}.

Measurement errors and extra number errors can also be further analyzed using Tables 6.10 and 6.11, respectively. As depicted in Table 6.10, the subcategory with higher magnitude errors (25 to 50 %) demonstrates significantly better results than the others. This subcategory consistently achieves a maximum sensitivity score (sensitivity = 1) for datasets WTP and Cement. In contrast, the sensitivity for errors between 5 to 10 % and 10 to 25 % is highly influenced by the applied threshold, with a noticeable decrease in sensitivity as the thresholds increase.

Regarding the analysis of extra number errors, Table 6.11 demonstrates that the model performs optimally when the inserted error causes an increase of one order of magnitude. However, when there is no change in the order of magnitude, the results vary between

Table 6.11: Extra number error analysis in terms of change of order of magnitude (o.m.) using the Soft Sensor methodology.

Dataset	Error subtype	N_{error}	SS_{th}	Sensitivity
Concrete	1 o.m.	86	0,05	1
			0,1	1
			0,15	1
			0,2	1
	0 o.m.	69	0,05	0,64
			0,1	0,42
			0,15	0,26
			0,2	0,16
WTP	0 o.m.	53	0,05	0,68
			0,1	0,42
			0,15	0,3
			0,2	0,26
Cement	1 o.m.	29	0,05	1
			0,1	1
			0,15	1
			0,2	1
	0 o.m.	10	0,05	0,2
			0,1	0
			0,15	0
			0,2	0

datasets. For Concrete and WTP, sensitivity presents similar results, with satisfactory scores for a low threshold, but poorer results as SS_{th} increases, as can be observed in Figures B.10 and B.11. Conversely, for the Cement dataset, extra numbers that caused no change in the order of magnitude are not detected by the model.

F1-score

Following the precision and sensitivity analysis, a perceptible pattern emerges: in the majority of conducted tests, when precision increased, sensitivity decreased. This observed tendency is both common and expected. In fact, if the model classified all samples as errors, sensitivity would be 1, but precision would be poor since the number of false positives would very high. Conversely, a model could detect no false positives but classify numerous samples as false negatives. To provide a balanced evaluation of the model's performance, it is crucial to analyze the F1-score.

Table 6.8 demonstrates overall strong results for F1-score with variations across datasets. For tests encompassing all error categories, Concrete exhibits the highest value at $SS_{th} = 0.2$ with a score of 0.81. A noteworthy trend is observed in which the F1-score increases as the threshold increases, indicating that for this dataset, precision increases more rapidly than sensitivity declines. This pattern, however, is not replicated in WTP, which attains its best value (0.74) at $SS_{th} = 0.15$, nor for the Concrete dataset, which reaches its peak (0.84) for $SS_{th} = 0,1$.

Order errors and blank spaces yield the highest scores, with F1-score rising as SS_{th} increases. This trend aligns with expectations, as sensitivity for these tests consistently approaches or reaches 1, while precision increases with the threshold.

On the opposite side, doubles stand out given its poor results in WTP and Cement

Table 6.12: Order error analysis in terms of change of order of magnitude (o.m.) using the Soft Sensor methodology.

Dataset	Error subtype	N_{error}	SS_{th}	Sensitivity
Concrete	-1 o.m.	62	0,05	1
			0,1	1
			0,15	1
			0,2	1
	0 o.m.	48	0,05	0,98
			0,1	0,98
			0,15	0,96
			0,2	0,94
	1 o.m.	45	0,05	1
			0,1	1
			0,15	1
			0,2	1
WTP	-1 o.m.	24	0,05	1
			0,1	1
			0,15	1
			0,2	1
	1 o.m.	29	0,05	1
			0,1	1
			0,15	1
			0,2	1
Cement	-1 o.m.	26	0,05	1
			0,1	1
			0,15	1
			0,2	1
	0 o.m.	13	0,05	1
			0,1	1
			0,15	1
			0,2	1

and the trend to decrease with the increase of the threshold. For the Cement dataset, with $SS_{th} = 0,15$ and $SS_{th} = 0,2$, tests performed on doubles presented null F1-scores, indicating the model's inability to detect these types of errors.

Specificity

The final metric to assess is specificity, which evaluates the model's ability to identify actual true values. This analysis is independent of the errors inserted and results can be seen in Table 6.9. For Concrete and WTP datasets, specificity increases with the threshold elevation. Cement exhibits the highest specificity values, scoring 1 for all thresholds except the first ($SS_{th} = 0.05$), where it achieves 0.82. In contrast, Concrete results range from 0.42 to 0.85, and WTP ranges from 0.49 to 0.98. This indicates that lower thresholds capture more false positives, in alignment with the results seen for precision.

In summary, for the results from Soft Sensor methodology, these were the most relevant patterns observed for the three datasets:

- Precision consistently improves with higher classification thresholds (SS_{th}) across datasets;

Table 6.13: Principal Components (PCs) composition for Concrete dataset.

PC	X	Weight	PC	X	Weight
	X_4	0,57		X_3	0,58
	X_5	-0,55		X_7	-0,48
PC 1	X_3	-0,37	PC 4	X_8	0,42
	X_7	-0,33		X_1	-0,35
	X_8	0,22		X_6	0,26
	Y	**0,62**		X_8	0,6
	X_1	0,54		X_7	0,48
PC 2	X_7	-0,28	PC 5	X_6	-0,47
	X_8	0,28		X_2	-0,32
	X_3	-0,26		X_4	0,23
	X_2	-0,68		X_6	0,41
	X_6	0,56		X_3	-0,4
PC 3	X_1	0,36	PC6	X_8	0,38
	X_5	-0,24		X_4	-0,37
	X_8	0,16		X_7	0,37

- Sensitivity values are generally favorable across datasets, with best results often observed at $SS_{th} = 0.05$, indicating lower thresholds reduce false negatives;

- While F1-score suggests overall strong model performance, a distinct pattern is not evident in the results. Notably, for tests covering all error categories, the Concrete dataset peaks at $SS_{th} = 0.2$, WTP at $SS_{th} = 0.15$, and Cement at $SS_{th} = 0.15$.

- Specificity, evaluating the model's ability to identify true values, increases with the elevation of the threshold across datasets.

- Blank spaces and order errors are the categories that are best captured by the model and the best results were obtained using $SS_{th} = 0,2$.

- Doubles are the most difficult error category to capture in datasets with a low variability in Y.

- Lower thresholds are optimal for measurement errors in datasets with low Y variability. Measurement errors between 25 and 50 % are easily detected by the model.

6.4.2 Analysis of results using the PCA-based methodology

Moving on to assess the effectiveness of the PCA-based methodology in identifying data entry errors, it's crucial to consider the composition of principal components as outlined in Tables 6.13, 6.14, and 6.15. These tables exclusively highlight the five most impactful variables, identified by their higher absolute weights. The magnitude of these weights reflects the contribution of each original variable to the PCs, with the sign indicating the direction of the correlation (positive or negative). Upon inspection, an expectation emerges that Cement would yield superior results, given its representation of Y in three principal components (2nd, 3rd, and 9th). In contrast, Concrete features Y prominently in its 2nd PC with substantial weight, while WTP incorporates Y only in the 5th PC.

The insights from Table 6.16 show that the model's overall performance is lower than the expected.

Table 6.14: Principal Components (PCs) composition for WTP dataset.

PC	X	Weight	PC	X	Weight
	X_3	-0,18		X_{43}	0,55
	X_{21}	-0,18		X_{40}	-0,49
PC 1	X_{22}	-0,18	PC 8	X_{41}	0,28
	X_{23}	-0,18		X_5	0,22
	X_{24}	-0,18		X_{46}	-0,19
	X_{11}	0,28		X_{42}	0,52
	X_{10}	0,27		X_2	0,42
PC 2	X_{12}	0,27	PC 9	X_{47}	-0,39
	X_{13}	0,27		X_{41}	-0,26
	X_{14}	0,27		X_{44}	0,25
	X_5	0,27		X_{40}	0,58
	X_7	0,27		X_1	-0,54
PC 3	X_8	0,27	PC 10	X_{42}	-0,34
	X_6	0,26		X_{41}	0,23
	X_1	-0,25		X_2	0,18
	X_6	0,34		X_{41}	-0,61
	X_7	0,33		X_{43}	0,41
PC 4	X_8	0,33	PC 11	X_{40}	0,33
	X_5	0,3		X_{42}	-0,25
	X_9	0,25		X_1	0,24
	X_{43}	-0,29		X_{42}	-0,36
	X_{41}	-0,27		X_{40}	-0,29
PC 5	X_{36}	0,25	PC 12	X_{29}	-0,27
	X_{35}	0,24		X_5	-0,25
	Y	**0,22**		X_{41}	-0,24
	X_{39}	-0,27		X_{55}	0,36
	X_{54}	-0,25		X_{45}	-0,35
PC 6	X_{38}	-0,24	PC 13	X_{44}	-0,25
	X_{53}	-0,23		X_{43}	-0,24
	X_{26}	-0,22		X_{46}	-0,24
	X_1	-0,53		X_5	0,3
	X_{42}	0,43		X_{39}	0,29
PC 7	X_2	-0,32	PC 14	X_1	-0,27
	X_{43}	0,32		X_{38}	0,25
	X_{41}	-0,23		X_{44}	-0,24

Precision

When considering precision, it is important to acknowledge that false positives remain consistent within a dataset (as these samples represent actual true values unaffected by error insertion). Precision scores range between 0.51 and 0.75, reflecting varying capabilities in detecting true positives (as false positives are constant). In tests covering all error categories, WTP leads in precision (0.68), followed by Cement (0.63), and Concrete (0.61). Notably, blank spaces and order errors present the highest scores for WTP and Cement, with similarly high scores within the Concrete dataset. Conversely, doubles error consistently display low precision across all datasets, indicating challenges in identifying duplicates as errors.

Sensitivity

In terms of sensitivity, Concrete demonstrates poor results, while Cement achieves excellent performance across all tests except for doubles, and WTP shows mixed results,

Table 6.15: Principal Components (PCs) composition for Cement dataset.

PC	X	Weight	PC	X	Weight
	X_5	0,39		X_2	0,53
	X_4	-0,35		X_1	-0,44
PC 1	X_{10}	-0,34	PC 7	X_9	0,35
	X_7	-0,29		X_{17}	-0,28
	X_{18}	-0,29		X_6	-0,22
	X_{11}	-0,4		X_{17}	0,53
	X_{14}	0,37		X_1	-0,49
PC 2	X_{12}	0,36	PC 8	X_3	-0,37
	X_{15}	0,3		X_8	0,32
	Y	**-0,29**		X_9	-0,26
	X_{20}	-0,46		X_3	0,58
	X_{19}	0,37		X_{17}	0,42
PC 3	X_{12}	-0,3	PC 9	**Y**	**-0,36**
	X_{21}	0,29		X_2	0,33
	Y	**-0,29**		X_9	-0,28
	X_6	-0,53		X_{22}	0,43
	X_7	0,37		X_1	0,41
PC 4	X_{10}	0,32	PC 10	X_9	0,41
	X_1	0,28		X_{17}	0,32
	X_{19}	-0,25		X_7	-0,3
	X_{22}	-0,62		X_8	0,75
	X_9	0,4		X_{17}	-0,35
PC 5	X_{17}	0,32	PC 11	X_7	-0,24
	X_3	0,21		X_9	-0,22
	Y	**0,21**		X_{16}	0,2
	X_{18}	0,45		X_{13}	0,51
	X_{16}	0,43		X_{21}	-0,46
PC 6	X_{20}	-0,36	PC 12	X_2	0,43
	X_9	-0,31		X_3	-0,23
	X_1	0,24		X_{15}	-0,23

performing excellently for half of the tests and poorly for the remaining half. Analyzing the error categories, doubles consistently display low sensitivity across all datasets and measurement errors present low scores for Concrete and WTP but remarkably high performance for Cement (0.92). Cement's success in this category is evident in Table 6.18, where errors with larger magnitudes achieve the best result (1), and those between 5 to 10% have a good score (0.77). In contrast, Concrete and WTP struggle with errors ranging from 25 to 50%, with scores of 0.45 and 0.47, respectively. This might result from the lower prevalence of Y in the PCs composition and the distribution of Y in these datasets.

When looking at sensitivity with respect to extra numbers, the tests conducted with the WTP dataset yield a low score of 0.4, whereas the Concrete and Cement datasets record higher scores of 0.72 and 0.82, respectively. Table 6.19 consistently shows that errors in an order of magnitude higher than the original values are well identified by the model (sensitivity = 1). However, when there is no change in the order of magnitude, the model struggles to detect errors, as it happens for the WTP dataset. The seemingly high sensitivity in Cement may result from significantly fewer instances with no change in

Table 6.16: Classification results for data entry errors detection using PCA methodology

Dataset	Error categories	Precision	Sensitivity	F1-score
Concrete	All categories	0,61	0,62	0,62
	Blank space	0,61	0,61	0,61
	Double	0,51	0,41	0,46
	Extra number	0,65	0,72	0,68
	Measurement error	0,52	0,43	0,47
	Order error	0,64	0,7	0,67
WTP	All categories	0,68	0,72	0,7
	Blank space	0,75	1	0,85
	Double	0,53	0,38	0,44
	Extra number	0,54	0,4	0,46
	Measurement error	0,56	0,43	0,49
	Order error	0,75	1	0,85
Cement	All categories	0,63	0,82	0,71
	Blank space	0,67	1	0,8
	Double	0,51	0,51	0,51
	Extra number	0,63	0,82	0,71
	Measurement error	0,65	0,92	0,77
	Order error	0,67	1	0,8

Table 6.17: Specificity for data entry error detection using the PCA methodology.

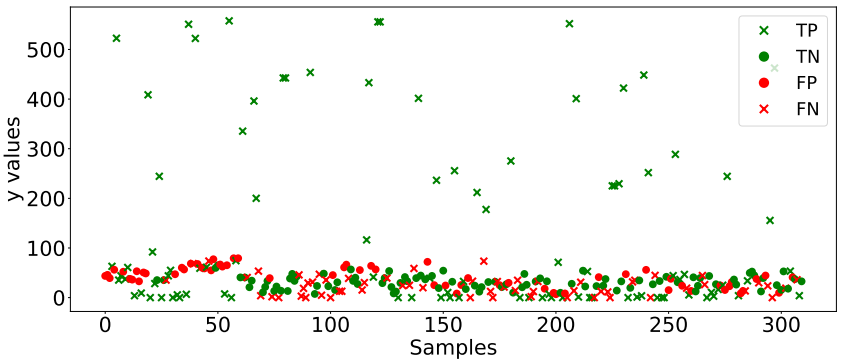
Dataset	Specificity
Concrete	0,61
WTP	0,66
Cement	0,51

Table 6.18: Measurement error analysis using the PCA methodology

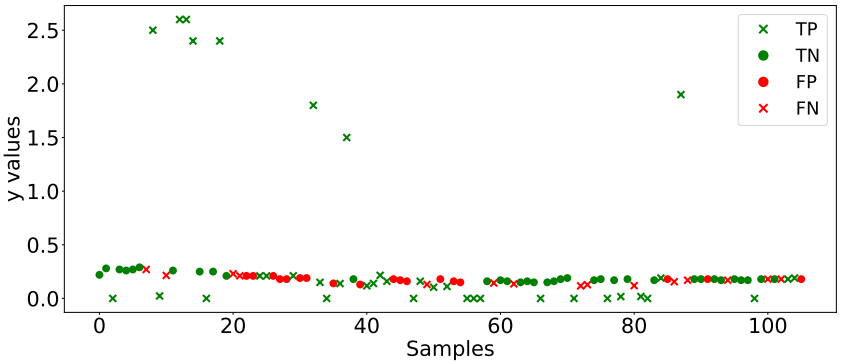
Dataset	N_{error}	Subtype Error	Sensitivity
Concrete	51	5 to 10 %	0,45
	51	10 to 25 %	0,37
	53	25 to 50 %	0,45
WTP	17	5 to 10 %	0,18
	17	10 to 25 %	0,65
	19	25 to 50 %	0,47
Cement	13	5 to 10 %	0,77
	13	10 to 25 %	1
	13	25 to 50 %	1

order of magnitude compared to those with a change.

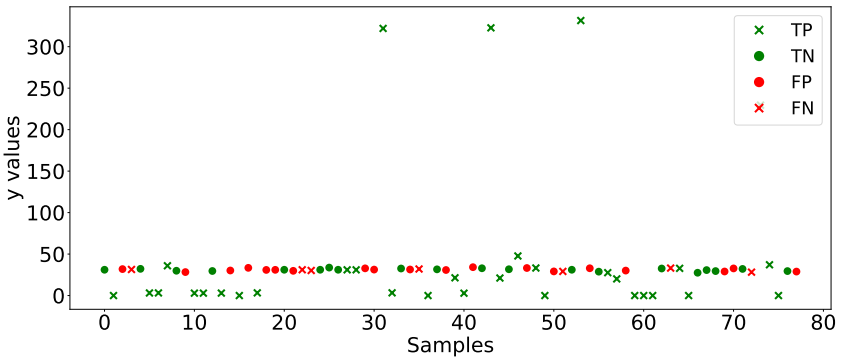
Order errors and blank spaces exhibit sensitivity levels of 1 in the Cement and WTP datasets. Further insights into the order error results can be found in Table 6.20. Cement notably excels in identifying all errors, even when the order of magnitude remains unchanged. The Concrete dataset demonstrates a sensitivity score of 1 for errors that cause an increase in the order of magnitude, but lower scores for cases where the order of magnitude remains unchanged or decreases. Figures B.35 provide visual context for this behavior in the Concrete dataset, illustrating that a change of -1 is often indistinguishable.



(a) PCA classification results for Concrete dataset with all error categories.



(b) PCA classification results for WTP dataset with all error categories.



(c) PCA classification results for Cement dataset with all error categories.

Figure 6.10: Classification results for the selected datasets with all error categories, using PCA.

F1-score

Moving forward, F1-score provides an overall evaluation of the model's ability to correctly label samples as errors. Cement and Wastewater Treatment Plant datasets exhibit comparable performances, with generally strong F1-score values, except for doubles tests and WTP also present poor results in extra number errors and measurement errors. Notably, Concrete yields inferior results, particularly notable in doubles and measurement error tests, which stand out for their particularly poor performance.

Table 6.19: Extra number error analysis using the PCA methodology

Dataset	N_{error}	Subtype Error	Sensitivity
Concrete	69	0 o.m.	0,36
	86	1 o.m.	1
WTP	53	0 o.m.	0,4
Cement	10	0 o.m.	0,3
	29	1 o.m.	1

Table 6.20: Order error analysis using the PCA methodology

Dataset	N_{error}	Subtype Error	Sensitivity
Concrete	62	-1 o.m.	0,53
	48	0 o.m.	0,62
	45	1 o.m.	1
WTP	24	-1 o.m.	1
	29	1 o.m.	1
Cement	13	-1 o.m.	1
	26	0 o.m.	1

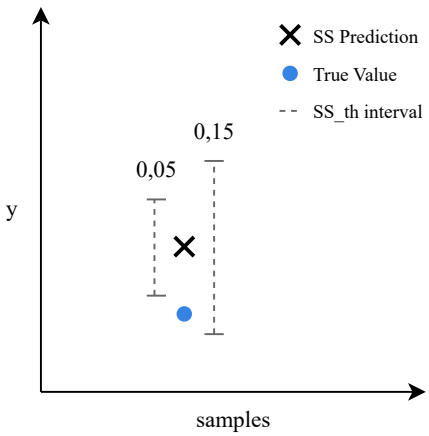


Figure 6.11: Illustration of how increasing SS_{th} can lead to reducing false positives

Specificity

Lastly, specificity analysis indicates how well the model identifies true negatives (true values) and its values are presented in table 6.17. Cement presents the worst value (0.51), while WTP exhibits the best (0.66). These results suggest the model's low performance in identifying true values across datasets.

To sum up the discussion around PCA-based methodology results, these were the most notorious conclusions:

- Precision yields moderate to poor results across all datasets;
- Sensitivity and F1-scores demonstrate the methodology's efficiency in identifying errors for the Cement dataset, mixed results for WTP (heavily dependent on the error category), and poor results for Concrete.
- Specificity values are low, showing a limited ability to identify true values;

- In terms of Principal Components, Cement dataset stands out, featuring Y in three PCs, while Concrete incorporates Y in the 2nd PC and WTP in the 5th. This created the expectation that the Cement would present scores, followed by Concrete and WTP. Although Cement results overcame the others, WTP dataset presented better overall metrics than the Concrete, which invalidates the set expectation;

- Overall, results are substantially inferior to those obtained using the Soft Sensor methodology.

6.4.3 Comparative analysis and overall discussion

After thorough analysis, it becomes evident that the Soft Sensors-based methodology consistently delivers superior results across all evaluated metrics.

In contrast, the methodology employing PCA not only yielded unsatisfactory metrics but also lacked transparency and a clear understanding of what caused the variability within results, making it challenging to interpret. Furthermore, the representation of Y in the PCs is not controllable, making it easy for the methodology to fail (if Y has low weights on the PCs, it is clear the methodology won't deliver good results).

On the other hand, the SS-based methodology demonstrated significantly stronger performance, exhibiting clear patterns. It also has adjustable parameters such as the SS model and the threshold (SS_{th}), which enhance its adaptability to new datasets or use cases. Precision, sensitivity, and F1-score emerged as key metrics yielding optimal results. The most adequate SS_{th} can be indicated for almost all error categories:

- Blank spaces and order errors were the categories best captured by the model, especially at $SS_{th} = 0.2$.

- Doubles were the most challenging error category, particularly in datasets with low variability in Y. $SS_{th} = 0.05$ proved to be the most suitable threshold, although its results are low as well.

- Lower thresholds were optimal for measurement errors and errors between 25 and 50% were easily detected by the model. Across the three datasets, $SS_{th} = 0.05$ yielded the best results.

- For extra number errors and all error categories together, there is no clear threshold to attribute, since the results vary a lot across datasets.

Chapter 7

Conclusion and future work

7.1 Conclusions

Throughout this work, two methodologies were designed to detect data entry errors within an industrial setting, utilizing data-driven models: Soft Sensor and Principal Components Analysis. Upon analysis, the SS-based methodology demonstrated superior overall performance across various datasets and metrics, suggesting its efficacy in error detection.

It is important to note, that for the specific use case considered, where operators enter new measurements into spreadsheets or system programs, the most critical metric to consider is sensitivity. Prioritizing sensitivity ensures that false negatives (i.e., actual errors incorrectly classified as true values) are minimized, a crucial aspect in preventing erroneous data from entering the system. Certainly, achieving a balance among the various metrics is also crucial for ensuring the reliability of the system and maintaining the operator's responsiveness to system alerts. Thus, managing false positives is also imperative to prevent any decrease of the system's credibility.

Given these considerations, the SS-based methodology outlined in Subsection 5.3.1 emerges as the preferred approach for detecting entry data errors in quality control variables. Its superior sensitivity results, coupled with enhanced interpretability, render it well-suited for operator comprehension and cooperation. Recommendations for parameter settings are as follows:

- Error proportion should be set at 15% for initial performance evaluation, with a dataset split of 70/30 %, to guarantee a balance of the labels;
- Among the threshold values (SS_{th}) examined, the suggested predefined value is 0.05 to optimize sensitivity. Ideally, however, the threshold should be determined following a thorough analysis of the dataset, considering the Soft Sensor's capability to characterize the data.

While the PCA-based methodology exhibited a promising performance for one of the datasets in sensitivity (excluding the test performed only with doubles), these outcomes appear to rely on the weighting and representation of the target variable in the Principal Components, which cannot be controlled beforehand. Additionally, PCA is challenging to elucidate to operators, and discerning clear patterns from the conducted tests to justify the obtained results is difficult. Hence, the utilization of this approach is not advisable.

It is essential to acknowledge the inherent limitations of this study. Notably, the transformation of regression models into classification models poses significant challenges,

particularly concerning threshold determination and output interpretation. These issues, as highlighted in prior research [78], underscore the need for further methodological refinement and exploration in future investigations.

In conclusion, the objectives set for this study (defined in Section 1.2) have been satisfactorily accomplished. The primary aim of developing two methodologies to detect data entry errors in quality control variables using data-driven models has been successfully achieved. Furthermore, the specific objectives outlined, including the characterization of detectable errors and the comparison of the performance of Soft Sensor and PCA-based methodologies, have also been met. Through comprehensive analysis and experimentation, insights into the capabilities and limitations of each approach have been gained, facilitating the recommendation of the most suitable methodology and parameters for practical implementation in industrial settings.

7.2 Future work

In terms of future work, it would be beneficial to compare the proposed data-driven methodology with the manual methods outlined in Section 2.1. This comparison should encompass not only the performance of each method in detecting errors but also the preferences of operators regarding which method or combination of methods they find most effective.

Furthermore, enhancements to the methodology itself could be explored, such as incorporating simple techniques already utilized in manual methods, such as detecting duplicate entries automatically, since doubles were the error category that the developed methodologies struggled the most with. Moreover, optimizing and automating threshold selection could be achieved using techniques like a ROC (Receiver Operating Characteristic) curve, which evaluates model performance across various threshold values by plotting sensitivity against specificity [79].

Additionally, given the challenges associated with converting the Soft Sensors regression model into a classification model, incorporating a statistical component into classification results could be advantageous. This might involve providing a probabilistic output, such as a confidence score, although careful consideration would be needed to balance this with interpretability and practical usability from the operator's perspective.

Finally, while this work has focused on establishing the theoretical foundations and conducting experiments to assess the feasibility of the developed methodologies, future efforts should aim to integrate the selected methodology into laboratory systems used in real industrial settings and find strategies to update them regularly. This may involve understanding the data registration programs employed by companies. In case they used simple Excel spreadsheets, the integration is simple and straightforward as Excel supports integrating Python models [80]. For other data collection programs, creating an Application Programming Interface (API) may be necessary, although this process is typically facilitated by the ease of integration offered by Python and the availability of API development tools.

Bibliography

[1] F. Souza, T. Offermans, R. Barendse, G. Postma, J. Jansen, Contextual mixture of experts: Integrating knowledge into predictive modeling, IEEE Transactions on Industrial Informatics (2022) 1–12doi:10.1109/TII.2022.3224973.

[2] K. A. Barchard, L. A. Pace, Preventing human error: The impact of data entry methods on data accuracy and statistical results, Computers in Human Behavior 27 (5) (2011) 1834–1839. doi:10.1016/j.chb.2011.04.004.

[3] E. T. Smyth, G. McIlvenny, J. G. Barr, L. M. Dickson, I. M. Thompson, Automated entry of hospital infection surveillance data, Infection Control & Hospital Epidemiology 18 (7) (1997) 486–491.

[4] M. Kawado, S. Hinotsu, Y. Matsuyama, T. Yamaguchi, S. Hashimoto, Y. Ohashi, A comparison of error detection rates between the reading aloud method and the double data entry method, Controlled Clinical Trials 24 (5) (2003) 560–569. doi:10.1016/S0197-2456(03)00089-8.

[5] S. Nahavandi, Industry 5.0—a human-centric solution, Sustainability 11 (16) (2019). doi:10.3390/su11164371.

[6] P. Panday, G. Kaur, Talent management and employee outlook on industry 5.0, in: Handbook of Research on Education Institutions, Skills, and Jobs in the Digital Era, IGI Global, 2023, pp. 299–306.

[7] J. Leiria, R. Salles, J. Mendes, P. Sousa, Soft sensors for industrial applications: Comparison of variables selection methods and regression models, in: 2023 International Conference on Control, Automation and Diagnosis (ICCAD), 2023, pp. 1–6. doi:10.1109/ICCAD57653.2023.10152323.

[8] K. A. Barchard, L. A. Pace, Meeting the challenge of high quality data entry: A free double-entry system, International Journal of Services and Standards 4 (4) (2008) 359–376. doi:10.1504/IJSS.2008.020053.

[9] R. R. Wilcox, How many discoveries have been lost by ignoring modern statistical methods?, American Psychologist 53 (3) (1998) 300–314. doi:10.1037//0003-066x.53.3.300.

[10] W. E. Winkler, Methods for evaluating and creating data quality, Information Systems 29 (7) (2004) 531–550. doi:10.1016/j.is.2003.12.003.

[11] J. Noyes, The QWERTY keyboard: a review, International Journal of Man-Machine Studies 18 (3) (1983) 265–281. doi:10.1016/S0020-7373(83)80010-8.

[12] D. J. White, A. P. King, S. D. Duncan, Voice recognition technology as a tool for behavioral research, Behavior Research Methods, Instruments, and Computers 34 (1) (2002) 1–5. doi:10.3758/BF03195418.

[13] R. Damper, M. Tranchant, S. Lewis, Speech versus keying in command and control: effect of concurrent tasking, International Journal of Human-Computer Studies 45 (3) (1996) 337–348. doi:10.1006/ijhc.1996.0055.

[14] G. Nagy, T. A. Nartker, S. V. Rice, Optical character recognition: an illustrated guide to the frontier, in: D. P. Lopresti, J. Zhou (Eds.), Document Recognition and Retrieval VII, Vol. 3967, International Society for Optics and Photonics, SPIE, 1999, pp. 58 – 69. doi:10.1117/12.373511.

[15] C. D. Emery, The use of portable barcode scanners in collections inventory, Collection Management 13 (4) (1991) 1–17. doi:10.1300/J105v13n04_01.

[16] R. A. Reynolds-Haertle, R. McBride, Single vs. Double data entry in CAST, Controlled Clinical Trials 13 (6) (1992) 487–494. doi:10.1016/0197-2456(92)90205-E.

[17] J. Cummings, J. Masten, Customized dual data entry for computerized data analysis, Quality assurance (San Diego, Calif.) 3 (3) (1994) 300—303.

[18] M. Rao, J. Corbin, Q. Wang, Soft sensors for quality prediction in batch chemical pulping processes, in: Proceedings of 8th IEEE International Symposium on Intelligent Control, 1993, pp. 150–155. doi:10.1109/ISIC.1993.397722.

[19] F. A. A. Souza, R. Araújo, J. Mendes, Review of soft sensors methods for regression applic1ations, Chemometrics and Intelligent Laboratory Systems 152 (2016) 69–79. doi:10.1016/j.chemolab.2015.12.011.

[20] L. Fortuna, S. Graziani, A. Rizzo, M. G. Xibilia, et al., Soft sensors for monitoring and control of industrial processes, Vol. 22, Springer, 2007.

[21] A. Raich, A. Çinar, Statistical Process Monitoring and Disturbance Diagnosis in Multivariable Continuous Processes, AIChE Journal 42 (4) (1996) 995–1009. doi:10.1002/aic.690420412.

[22] J. L. Godoy, J. L. Marchetti, J. R. Vega, An integral approach to inferential quality control with self-validating soft-sensors, Journal of Process Control 50 (2017) 56–65. doi:10.1016/j.jprocont.2016.12.001.

[23] I. C. Yeh, Modeling of strength of high-performance concrete using artificial neural networks, Cement and Concrete Research 28 (12) (1998) 1797–1808. doi:S0008-8846(98)00165-3.

[24] P. Kadlec, B. Gabrys, S. Strandt, Data-driven Soft Sensors in the process industry, Computers and Chemical Engineering 33 (4) (2009) 795–814. doi:10.1016/j.compchemeng.2008.12.012.

[25] D. Dong, T. J. Mcavoy, Nonlinear principal component analysis - Based on principal curves and neural networks, Computers and Chemical Engineering 20 (1) (1996) 65–78. doi:10.1016/0098-1354(95)00003-K.

[26] Q. Sun, Z. Ge, A survey on deep learning for data-driven soft sensors, IEEE Transactions on Industrial Informatics 17 (9) (2021) 5853–5866. doi:10.1109/TII.2021.3053128.

[27] A. Das, J. Maiti, R. N. Banerjee, Process monitoring and fault detection strategies: A review, International Journal of Quality & Reliability Management 29 (7) (2012) 720–752. doi:10.1108/02656711211258508.

[28] R. Isermann, P. Ballé, Trends in the application of model-based fault detection and diagnosis of technical processes, Control Engineering Practice 5 (5) (1997) 709–719. doi:10.1016/S0967-0661(97)00053-1.

[29] D. Montgomery, Introduction to statistical quality control, 3rd Edition, Wiley, New York, NY [u.a.], 1997.

[30] H. Nazir, M. Schoonhoven, M. Riaz, R. Does, Quality quandaries: A stepwise approach for setting up a robust shewhart location control chart, Quality Engineering 26 (2014) 246–252. doi:10.1080/08982112.2013.874562.

[31] D. X. Tien, K. W. Lim, L. Jun, Comparative study of PCA approaches in process monitoring and fault detection, IECON Proceedings (Industrial Electronics Conference) 3 (2004) 2594–2599. doi:10.1109/IECON.2004.1432212.

[32] S. W. Choi, C. Lee, J. M. Lee, J. H. Park, I. B. Lee, Fault detection and identification of nonlinear processes based on kernel PCA, Chemometrics and Intelligent Laboratory Systems 75 (1) (2005) 55–67. doi:10.1016/j.chemolab.2004.05.001.

[33] J. V. Kresta, J. F. Macgregor, T. E. Marlin, Multivariate statistical monitoring of process operating performance, The Canadian journal of chemical engineering 69 (1) (1991) 35–47.

[34] N. Kettaneh, A. Berglund, S. Wold, Pca and pls with very large data sets, Computational Statistics & Data Analysis 48 (1) (2005) 69–85.

[35] J. F. MacGregor, C. Jaeckle, C. Kiparissides, M. Koutoudi, Process monitoring and diagnosis by multiblock pls methods, AIChE Journal 40 (5) (1994) 826–838.

[36] R. Dunia, S. J. Qin, T. F. Edgar, T. J. McAvoy, Identification of faulty sensors using principal component analysis, AIChE Journal 42 (10) (1996) 2797–2812.

[37] T. J. Rato, J. Blue, J. Pinaton, M. S. Reis, Translation-invariant multiscale energy-based pca for monitoring batch processes in semiconductor manufacturing, IEEE Transactions on Automation Science and Engineering 14 (2) (2017) 894–904. doi:10.1109/TASE.2016.2545744.

[38] T. J. Rato, M. S. Reis, Fault detection in the tennessee eastman benchmark process using dynamic principal components analysis based on decorrelated residuals (dpca-dr), Chemometrics and Intelligent Laboratory Systems 125 (2013) 101–108. doi:10.1016/j.chemolab.2013.04.002.

[39] V. H. Nguyen, J. C. Golinval, Fault detection based on Kernel Principal Component Analysis, Engineering Structures 32 (11) (2010) 3683–3691. doi:10.1016/j.engstruct.2010.08.012.

[40] Z. Lv, X. Yan, Q. Jiang, Batch process monitoring based on multiple-phase online sorting principal component analysis, ISA Transactions 64 (May 2016) (2016) 342–352. doi:10.1016/j.isatra.2016.04.022.

[41] Y. Jiang, S. Yin, J. Dong, O. Kaynak, A review on soft sensors for monitoring, control, and optimization of industrial processes, IEEE Sensors Journal 21 (11) (2021) 12868–12881. doi:10.1109/JSEN.2020.3033153.

[42] D. Wang, J. Liu, R. Srinivasan, Data-driven soft sensor approach for quality prediction in a refining process, IEEE Transactions on Industrial Informatics 6 (1) (2010) 11–17. doi:10.1109/TII.2009.2025124.

[43] M.-T. Puth, M. Neuhäuser, G. D. Ruxton, Effective use of pearson's product–moment correlation coefficient, Animal behaviour 93 (2014) 183–189.

[44] T. M. Cover, J. A. Thomas, Entropy, Relative Entropy and Mutual Information, John Wiley & Sons, Ltd, 1991, Ch. 2, pp. 12–49.

[45] M. Gonçalves, P. Sousa, J. Mendes, M. Danishvar, A. Mousavi, Real-time event-driven learning in highly volatile systems: A case for embedded machine learning for scada systems, IEEE Access 10 (2022) 50794–50806. doi:10.1109/ACCESS.2022.3173376.

[46] S. Raschka, V. Mirjalili, Python machine learning: Machine learning and deep learning with python, Scikit-Learn, and TensorFlow. Second edition ed 3 (2017).

[47] F. Souza, R. Araújo, S. Soares, J. Mendes, Variable selection based on mutual information for soft sensors applications, in: Proc. 9th Portuguese Conference on Automatic Control (CONTROLO 2010), Coimbra, Portugal, 2010, pp. 1–6.

[48] J. D. Jobson, Multiple Linear Regression, Springer New York, New York, NY, 1991, pp. 219–398.

[49] J. Mendes, F. Souza, R. Araújo, N. Gonçalves, Genetic fuzzy system for data-driven soft sensors design, Applied Soft Computing 12 (10) (2012) 3237–3245. doi:10.1016/j.asoc.2012.05.009.

[50] Q. Hu, C. Xu, Y. Dong, Dam deformation analysis based on ridge regression, in: 2009 International Conference on Information Engineering and Computer Science, IEEE, 2009, pp. 1–4.

[51] D. Schreiber-Gregory, Ridge regression and multicollinearity: An in-depth review, Model Assisted Statistics and Applications 13 (09 2018). doi:10.3233/MAS-180446.

[52] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society. Series B (Methodological) 58 (1) (1996) 267–288.

[53] T. Hastie, R. Tibshirani, J. H. Friedman, The elements of statistical learning: data mining, inference, and prediction, 2nd Edition, Springer, New York, NY, 2009.

[54] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, Journal of the Royal Statistical Society. Series B (Statistical Methodology) 67 (2) (2005) 301–320.

[55] C.-H. Wu, J.-M. Ho, D.-T. Lee, Travel-time prediction with support vector regression, IEEE transactions on intelligent transportation systems 5 (4) (2004) 276–281.

[56] A. Fabisch, gmr: Gaussian mixture regression, Journal of Open Source Software 6 (62) (2021) 3054.

[57] R. Salles, J. Mendes, R. Araújo, C. Melo, P. Moura, Prediction of key variables in wastewater treatment plants using machine learning models, in: Proc. 2022 IEEE International Joint Conference on Neural Networks (IJCNN 2022), at the 2022 World Congress on Computational Intelligence (WCCI 2022), Padova, Italy, 2022, pp. 1–9. doi:10.1109/IJCNN55064.2022.9892661.

[58] P. P. Lid, S. Planning, PRINCIPAL COMPONENTS ANALYSIS (PCA), Vol. 19, 1993.

[59] S. Yin, S. X. Ding, A. Haghani, H. Hao, P. Zhang, A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process, Journal of Process Control 22 (9) (2012) 1567–1581. doi:10.1016/j.jprocont.2012.06.009.

[60] R. Luo, M. Misra, D. M. Himmelblau, Sensor fault detection via multiscale analysis and dynamic PCA, Industrial and Engineering Chemistry Research 38 (4) (1999) 1489–1495. doi:10.1021/ie980557b.

[61] J. V. Kresta, J. F. Macgregor, T. E. Marlin, Multivariate statistical monitoring of process operating performance, The Canadian Journal of Chemical Engineering 69 (1) (1991) 35–47. doi:10.1002/cjce.5450690105.

[62] S. Ding, P. Zhang, E. Ding, S. Yin, A. Naik, P. Deng, W. Gui, On the application of pca technique to fault diagnosis, Tsinghua Science & Technology 15 (2) (2010) 138–144. doi:10.1016/S1007-0214(10)70043-2.

[63] S. J. Qin, Data-driven fault detection and diagnosis for complex industrial processes, IFAC Proceedings Volumes 42 (8) (2009) 1115–1125, 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes. doi:10.3182/20090630-4-ES-2003.00184.

[64] I. T. Jolliffe, Principal Component Analysis, Springer-Verlag, Berlin; New York, 1986.

[65] S. L. Crump, The present status of variance component analysis, Biometrics 7 (1) (1951) 1–16.

[66] W. C. Navidi, Statistics for engineers and scientists, Vol. 2, McGraw-Hill New York, 2006.

[67] S. Stewart, M. Thomas, Eigenvalues and eigenvectors: Formal, symbolic, and embodied thinking, in: The 10th Conference of the Special Interest Group of the Mathematical Association of America on Research in Undergraduate Mathematics Education, San Diego California, 2007, pp. 275–296.

[68] M. E. Wall, A. Rechtsteiner, L. M. Rocha, Singular value decomposition and principal component analysis (2003). arXiv:physics/0208101.

[69] F. Kherif, A. Latypova, Chapter 12 - principal component analysis, in: A. Mechelli, S. Vieira (Eds.), Machine Learning, Academic Press, 2020, pp. 209–225. doi:10.1016/B978-0-12-815739-8.00012-2.

[70] S. J. Qin, Statistical process monitoring: Basics and beyond, Journal of Chemometrics 17 (8-9) (2003) 480–502. doi:10.1002/cem.800.

[71] L. Peng, G. Han, A. Landjobo Pagou, J. Shu, Electric submersible pump broken shaft fault diagnosis based on principal component analysis, Journal of Petroleum Science and Engineering 191 (2020) 107154. doi:10.1016/j.petrol.2020.107154.

[72] J. E. Jackson, G. S. Mudholkar, Control procedures for residuals associated with principal component analysis, Technometrics 21 (3) (1979) 341–349. doi:10.1080/00401706.1979.10489779.

[73] A. Pini, A. Stamm, S. Vantini, Hotelling's t2 in separable hilbert spaces, Journal of Multivariate Analysis 167 (2018) 284–305. doi:10.1016/j.jmva.2018.05.007.

[74] J. E. Jackson, A user's guide to principal components, John Wiley & Sons, 2005.

[75] I.-C. Yeh, Concrete Compressive Strength, UCI Machine Learning Repository, DOI: https://doi.org/10.24432/C5PK67 (2007).

[76] B. de Matos, R. Salles, J. Mendes, J. R. Gouveia, A. J. Baptista, P. Moura, A review of energy and sustainability kpi-based monitoring and control methodologies on wwtps, Mathematics 11 (1) (2023). doi:10.3390/math11010173.

[77] J. Mendes, R. Araújo, F. Souza, Adaptive fuzzy identification and predictive control for industrial processes, Expert Systems with Applications 40 (17) (2013) 6964–6975. doi:10.1016/j.eswa.2013.06.057.

[78] L. C. M. Félix, S. O. Rezende, M. C. Monard, C. W. Caulkins, Transforming a regression problem into a classification problem using hybrid discretization, Computación y Sistemas 4 (1) (2000) 44–52.

[79] Z. H. Hoo, J. Candlish, D. Teare, What is an roc curve?, Emergency Medicine Journal (2017). doi:10.1136/emermed-2017-206735.

[80] Introduction to Python in Excel, https://support.microsoft.com/en-us/office/introduction-to-python-in-excel-50f8dfc7-0f93-4e30-8ba7-cb7232f87127, accessed: January 2024.

Appendix A

Paper ICCAD: Soft Sensors for Industrial Applications: Comparison of Variables Selection Methods and Regression Models

Soft Sensors for Industrial Applications: Comparison of Variables Selection Methods and Regression Models

Joana Leiria*, Rodrigo Salles*, Jérôme Mendes*, and Pedro Sousa†
* University of Coimbra, Institute of Systems and Robotics,
Department of Electrical and Computer Engineering, Pólo II, 3030-290 Coimbra, Portugal
Email: joana.leiria@isr.uc.pt, rodrigo.salles@isr.uc.pt, jerome.mendes@uc.pt
† Oncontrol Technologies, Rua Cidade Poitiers, nº 155 – 1º Andar 3000-108 Coimbra, Portugal
Email: pedro.sousa@oncontrol-tech.com

***Abstract*—The digitalization of industrial environments has enabled the development of tools that make the production process more efficient and safer. In this sense, the Soft Sensor (SS) plays a fundamental role. Through historical data and indirect measurements, it is possible to estimate the value of important variables that are difficult to measure. This paper presents the SS development process: data collection and pre-processing, variable selection, model selection for SS implementation, model training and testing, and performance evaluation. The selection of variables was made with the help of Pearson Correlation, Mutual Information, and fastTracker algorithm techniques. For the implementation of SS have been tested several models: Multiple Linear Regression, Ridge Regression, Least Absolute Shrinkage and Selection Operator, Elastic Net, Support Vector Regression and Gaussian Mixture Models. Four datasets were used to test the development of the SS.**

***Index Terms*—Soft Sensors, regression models, variable selection, industrial application.**

I. Introduction

With the increasing demand for industrial digitization toward a more sustainable and greener industrial future, inferential models have been used for the online prediction of quality variables (eg. variables that cannot be automatically measured and are obtained by means of a laboratory analysis) [1]. In general terms, Soft Sensor (SS) refers to inferential models that are used to estimate certain physical quantities or product quality in the industrial processes based on the available measurements (field variables) and knowledge [2], [3]. There are different approaches to designing a SS, but in general terms, the main steps are [2]: selection of data from the plant's operating history that will be used by the SS, treatment of data such as filtering and detection of outliers, selection of the structure of the model to be used, estimations made with the selected model, and model validation.

The elaboration of a SS depends on obtaining data from the operating history of the plant where the SS is intended to be used, and this is a limiting factor for its use in industrial environments. The performance of many models used for the implementation of the SS is associated with the existence of sufficient data for training and testing [4], and in the industrial environment the lack of data is common, or the existence of poor quality data, which compromises the confidence of a SS. Deep Learning (DL), a subset of Machine learning, are complex models with high levels of accuracy and has become increasingly popular due to the availability of data, however, DL requires a huge amount of data and its interpretability by the human expert operator is lost as presented in [5], and this limits its use as SS in an industrial environment, as for example, to estimate laboratory variables.

The present work presents a framework for the soft sensor design and does a comparison study between several regression models and variable selection methods, introducing variable expansion in the framework. The proposed framework uses the following methods/steps:

- Variable expansion: to introduce nonlinearity in the models, the dataset was extended by adding the square, the inverse, and the root mean squared of each input variable, as well as the product between each two input variables.
- Input variable selection: three variable selection methods were implemented. The well-known Pearson's correlation that measures the linear dependence between pairs of variables [6]. The Mutual Information (MI) which measures the dependency between variables taking into account the probabilistic distribution of the variables [7]. The fastTracker algorithm [8], a recent and efficient real-time algorithm that tracks the process behavior's changes by measuring sensitivity indices between variables.
- Model: to implement the SS were used Multiple Linear Regression (MLR), Ridge Regression (RR), Least Absolute Shrinkage and Selection Operator (LASSO), Elastic Net (EN), Support Vector Regression (SVR), and Gaussian Mixture Models for Regression (GMM) models.

The main contributions of this work are the implementation of the variable expansion in the framework, and the implementation and comparison of several regression models and

This research was co-financed by the European Regional Development Fund, through Centro Regional Operational Program 2014/2020 (Centro2020), of Portugal 2020. Project InGestAlgae (CENTRO-01-0247-FEDER-046983) and iProMo (CENTRO-01-0247-FEDER-069730).

variable selection methods. DL models were excluded given the focus on the model interpretability and the small datasets scenarios. In order to test the proposed framework, 4 benchmark datasets were used, where each model for each dataset was learned/tested with 9 different sets of input variables (from the variable selection methods) and each learning process was executed 30 times. Thus, 7350 tests were performed.

The paper is organized as follows. Section II brings the notations adopted in the work, Section III deals with the methods for selecting the variables, Section IV describes the regression models, Section V presents the methodology for SS design, Section VI presents the experimental results, and the conclusions are presented in Section VII.

II. Notation

The following notation is used in this paper. Variables and their values are defined by capital and lowercase letters, respectively, e.g. variable A and corresponding value a. Matrices and vectors are defined by bold capital letters, e.g. $\mathbf{A} = [a_{k,j}]_{K\times m}$ and bold lowercase letters, e.g. $\mathbf{a} = [a_1, \ldots, a_m]$, respectively. $\hat{Y}$ represents the prediction for target variable Y, $X = \{X_1, \ldots, X_m\}$ are the input variables with the values $x_{k,j} \in X_j$ ($k = 1, \ldots, K$ and $j = 1, \ldots, m$), $\mathbf{x}_k = [x_{k,1}, \ldots, x_{k,m}]$, $\mathbf{X} = [x_{k,j}]_{K\times m}$ and $\mathbf{Y} = [y_k]_{K\times 1}$. The regression coefficients are represented by β_0 and $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_m]^T$.

III. Variable Selection

The performance of a soft sensor depends largely on the selection of the independent variables used to predict the target variable. This section describes the methods used to select the variables used in this work.

A. Pearson Correlation

Pearson's correlation is used to determine how strong the relationship between two or more variables is [6]. It is the measure of the linear dependence between pairs of variables. Pearson's correlation, represented by r, has values ranging from -1 to 1, where two variables have a perfect positive correlation if $r = 1$, and if $r = -1$ the correlation is perfect negative, while if the correlation is zero ($r = 0$), the analyzed variables are not correlated [9]. Pearson's correlation between variables X_j and Y is given by the equation (1).

$$r = \frac{\Sigma_{k=1}^{K}[(x_{k,j} - \mu_{X_j})(y_k - \mu_Y)]}{\sqrt{\Sigma_{k=1}^{K}(x_{k,j} - \mu_{X_j})^2}\sqrt{\Sigma_{k=1}^{K}(y_k - \mu_y)^2}}, \quad (1)$$

where μ_{X_j} and μ_Y are the arithmetic means of the variables X_j and Y, respectively.

B. Mutual Information

Mutual Information (MI) is a non-linear measure of dependency between variables that take into account the probability distribution of the variables and is obtained through entropy measurements. The MI between two discrete variables X and Y is given by equation (2) [10], where $H(X)$ and $H(X, Y)$ represent Shannon's Entropy, and N the number of bins.

$$I(Y, X) = H(Y) + H(X) - H(X; Y) \quad (2)$$

$$H(X) = \sum_{b=1}^{N} -log[P(x_b)]P(x_b) \quad (3)$$

$$H(X, Y) = \sum_{b=1}^{N}\sum_{k=1}^{N} -log[P(x_b, y_k)]P(x_b, y_k) \quad (4)$$

Where $P(x_b)$ is the probability density function, and $P(x_b, y_k)$ is the joint probability mass function of X and Y.

C. fastTracker Algorithm

fastTracker algorithm [8] is an efficient real-time causal-effect sensitivity analysis algorithm that tracks the process behavior's changes by measuring the sensitivity indices between input variables and the target variable. Algorithm 1 shows the main steps of fastTracker. Where $X = [X_1, \ldots, X_m]$ is a set of input variables, Y is the target variable, TT^j ($j = 1, \ldots, m$) and ET are the trigger and event thresholds, respectively, and n is the number of batches per analysis span. The output of fastTracker are the sensitivity indices, nSI^j ($j = 1, \ldots, m$), for each input variable.

Algorithm 1: fastTracker methodology [8].

Procedure:

for $k = 1, \ldots, K$ *(for all data)* **do**

  for ***each*** *input variable* X_j **do**

    1. Perform the trigger-event detection of two consecutive batches, i.e. determine if a variable represents a real change in the system state or not.
    2. Determine the $XNOR$, i.e. verify the simultaneous existence or nonexistence of a change in each batch.
    3. Obtain the sensitive index SI_k^j for instant k.
    4. Obtain the normalized sensitivity index, nSI_k^j for instant k.

IV. Regression Models

The target of a soft sensor is to obtain the values of variables that are difficult to measure. In the present work, several models were evaluated to perform the SS function and they are described below.

A. Multiple Linear Regression

The Multiple Linear Regression (MLR) is one of the most popular statistical methods used to relate two or more variables, given its simplicity and easy interpretation and implementation [11]. The goal of MLR is to find a linear function that relates a set of independent variables and the dependent variable. The MLR can be expressed by the equation (5).

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_j x_j + \cdots + \beta_m x_m + \varepsilon \quad (5)$$

Where ε is the model's error term, and regression coefficients β_0 and β_j are determined by the Least Squares method.

B. Ridge Regression

Ridge regression (RR) is a variant of ordinary MLR whose goal is to circumvent the collinearity among independent variables [12]. The RR is a regularization model in which a penalty term is added to a linear least squares loss function. This penalty equals the L2-norm of the coefficient and the regularization strength is controlled by a hyperparameter λ. This and other regularization methods are designed to prevent over-fitting, ensuring a smaller variance in the resulting parameter estimates [13]. The objective of RR is to minimize the penalized residual sum of squares:

$$\sum_{k=1}^{K}\left(y_k-\beta_0-\sum_{j=1}^{m}x_{k,j}\beta_j\right)^2+\lambda\sum_{j=1}^{m}\beta_j^2 \tag{6}$$

Minimizing equation (6), and using the centered $x_{k,j}$, equation (7) is obtained, where $\mathbf{I}$ represents the identity matrix.

$$\boldsymbol{\beta}=(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y} \tag{7}$$

C. Least Absolute Shrinkage and Selection Operator

The Least Absolute Shrinkage and Selection Operator (LASSO) is a regression model with a regulation method. It is almost identical to RR, but it uses the L1-norm of the coefficient, controlled by λ, as the penalty term, as shown in equation (8) [14].

$$\boldsymbol{\beta}=\arg\min_{\boldsymbol{\beta}}\Big\{\frac{1}{2}\sum_{k=1}^{K}\left(y(k)-\beta_0-\sum_{j=1}^{m}x_j(k)\beta_j\right)^2+ +\lambda\sum_{j=1}^{m}|\beta_j|\Big\} \tag{8}$$

LASSO can be used as a feature selection method, as the less important variables can have a null coefficient [14].

D. Elastic Net

The Elastic Net (EN) model combines the capability of variable selection from LASSO and the prediction performance from RR. The EN combines the L1-norm penalty of the LASSO and the L2-norm penalty of RR [15]. The model is estimated by minimizing the following objective function.

$$\arg\min_{\beta_0,\boldsymbol{\beta}}\left\{\frac{1}{K}\sum_{k=1}^{K}y_k\left(\beta_0+\mathbf{x}_k\boldsymbol{\beta}\right)-\log\left(1+e^{\beta_0+\mathbf{x}_k\boldsymbol{\beta}}\right) +\lambda\left(\frac{(1-\alpha)||\boldsymbol{\beta}||^2}{2}+\alpha||\boldsymbol{\beta}||\right)\right\} \tag{5}$$

Where α and λ parameters are responsible for tuning the model, being α ($\alpha\in[0,1]$) responsible for the strength of each penalty term, while non-negative λ controls the trade-off between variance and bias in the estimated parameters.

E. Support Vector Regression

The Support Vector Regression (SVR) considers the following regression function [16]:

$$f(X)=\sum_{k,k'=1}^{K}(\alpha_k^*-\alpha_k)K_e(X_{k'},X_k)+b, \tag{9}$$

where b is a constant, and K_e is the kernel that maps the problem to different dimensions. The Lagrange multipliers α_k^* and α_k, which represent the solution to the above equation, are obtained by maximizing the following function:

$$W(\alpha_k^*,\alpha_k)=-\frac{1}{2}\sum_{k,k'=1}^{K}(\alpha_k^*-\alpha_k)(\alpha_k^*-\alpha_k)K_e(X_k,X_{k'})+ +\sum_{k=1}^{K}y_k(\alpha_k^*-\alpha_k)-\varepsilon\sum_{k=1}^{K}(\alpha_k+\alpha_k). \tag{10}$$

Subject to the following constraints: $\sum_k^K\alpha_k^*=\sum_k^K\alpha_k$ and $0\le\alpha_k^*,\alpha_k\le C$, where C is a cost parameter whose value comes out of cross-validation [16].

F. Gaussian Mixture for Regression

Gaussian Mixture Models (GMM), by the superposition of two or more Gaussians, have been largely used to model real data [17]. The superposition is composed of probabilistic models, where the joint probability distribution $p(X,Y)$ is obtained by Expectation-Maximization method [18], being obtained by equation (11) [18] where G is the number of Gaussian components, the g-th component $\mathcal{N}^g(\cdot)$ is represented by the mean $\boldsymbol{\mu}_{XY}^g$ and variance $\boldsymbol{\Sigma}_{XY}^g$. π^g, where $\sum_{g=1}^K\pi^g=1$, is the component weight.

$$p(X,Y)=\sum_{g=1}^{G}\pi^g\mathcal{N}_g(\mathbf{x}_k,y_k|\boldsymbol{\mu}_{XY}^g,\boldsymbol{\Sigma}_{XY}^g) \tag{11}$$

For the prediction, the Gaussian Mixture for Regression (GMR) computes the conditional distribution $p(Y|X)$ for a given sample, given by equation (12) [18], in which $\pi_{Y|X}^g$ is obtained by equation (13).

$$p(Y|X) = \sum_{g=1}^{G}\pi_{Y|X}^g\mathcal{N}_g(Y|\boldsymbol{\mu}_{Y|X}^g,\boldsymbol{\Sigma}_{Y|X}^g) \tag{12}$$

$$\pi_{Y|X}^g = \frac{\mathcal{N}_g(X|\boldsymbol{\mu}_{\mathbf{x}}^g,\boldsymbol{\Sigma}_X^g)}{\sum_{l=1}^{G}\mathcal{N}_l(\mathbf{x}_k|\boldsymbol{\mu}_X^l,\boldsymbol{\Sigma}_X^l)} \tag{13}$$

V. SoftSensor Methodology

This section presents the methodology of the proposed framework for designing soft sensors. The main steps of the methodology are presented in Algorithm 2, being divided into the following steps: preprocessing, variable expansion, variable selection, and then model training and testing.

Before being used, the dataset is preprocessed (Step 1). Here, empty and semi-empty columns are eliminated as well as columns with non-numeric data. Samples with empty values are also deleted.

Algorithm 2: Soft Sensor Methodology.

Procedure:
Input: dataset.
1. Preprocess the dataset (delete outliers, non numeric variables, and empty variables).
2. Perform the variable expansion.
3. Select the variable selection method (Section III): Pearson's correlation, MI, or fastTracker.
4. Define the model to be used (Section IV): MLR, RR, LASSO, EN, SVR, or GMM.
5. Define the hyperparameters of the selected model.
for $it = 1, \ldots, it_{max}$: **do**
    6. Split randomly the dataset in train (70%) and test (30 %).
    7. Tune hyperparameters using Grid Search and 10-fold Cross-Validation.
    8. Fit the model using the best hyperparameters (from step 7) to the training dataset.
    9. Predict the target variable y for the test dataset.
    10. Calculate the error metrics.
    11. Save the model and hyperparameters values.
12. Obtain the average of the error metrics.
13. Select the model with the best error metrics.

The dataset is divided in two parts: input variables (X) and output variable (Y). The input variables are then expanded (Step 2): the square, the inverse and the root mean squared of each variable X_j are obtained, as well as the product between each two variables. There are two exceptions: if an original variable has negative values the root mean squared is not applied and if there are zero values the inverse is also discarded. With the expanded variables obtained, one of the variable selection methods is selected (Step 3): Pearson Correlation, Mutual Information or fastTracker.

After the variables selection, a model is chosen (Step 4) from the ones presented in Section IV: MLR, RR, LASSO, EN, SVR, or GMM. According to the model selected the possible hyperparameters values are defined (Step 5).

To ensure that the model describes the data well and that results are not biased by the dataset division, the training and testing process will be repeated it_{max} times (iterations). In each iteration it, the dataset is randomly split (Step 6) into two datasets: a train dataset (70 %) and a test dataset (30 %). In this paper, was defined $it_{max} = 30$. Then, the hyperparameters are tuned using a Grid Search with a 10-fold Cross-Validation (Step 7), where each possible hyperparameter combination is tested for the training dataset. The hyperparameters of the models with the lowest Mean Squared Error (MSE) are selected and used to fit the model to the training dataset (Step 8). With this model, the target variable Y is predicted using the input variables from the test dataset (Step 9) and then compared to the true Y. With these values, the error metrics are calculated (Step 10) and saved, as well as the model itself and their hyperparameters (Step 11).

TABLE I: Main characteristics of the datasets.

Dataset	K	$\lvert X\rvert$	**Data type**
Concrete	1030	8	Numeric
Automobile Gas	348	8	Numeric/string
Box–Jenkins	296	8	Numeric
WTP	347	55	Numeric

When finishing all iterations (it_{max}), the average of the error metrics is calculated (Step 12) and the model with the best metrics is selected to be used as a soft sensor (Step 13).

VI. Experimental Results

This section presents the description of the datasets used for the development of the soft sensor, as well as the tests, the considered evaluation metric, and the results obtained.

A. Datasets

For the development of the SS four datasets were used:

- **Concrete**: the objective is to infer the concrete compressive strength, measured in the laboratory, using its age and components.
- **Automobile Gas**: about the automobile gas mileage, concerns a regression problem related to the fuel consumption in miles per gallon, and seven input variables related to the car's characteristics.
- **Box–Jenkins**: the goal is to determine the carbon dioxide (CO_2) concentration from a combustion process of a methane-air mixture.
- **WTP**: from a real urban water treatment plant (WTP), the objective is to estimate the fluoride concentration in the effluent. The sampling rate is 2 hours.

A summary of the datasets' main characteristics is presented in Table I, where $|X|$ is the number of the input variables.

B. Tests and Metrics

For each dataset, the procedure explained in Section V (Algorithm 2) was performed. The datasets were processed: samples with missing values were deleted, as well as non-numeric variables, and no outliers were detected. The datasets were prepared to have the original input variables and the defined expanded variables, with the exception of the WTP dataset, due to computational processing limitations, it wasn't feasible to use all the expanded variables. Tests were performed with 1) no input variables selection (i.e. using all original input variables), 2) the original input variables and the expanded variables with no selection, and using the selected variables by the methods 3) Pearson Correlation, 4) Mutual Information, and 5) fastTracker. Then, possible values for the hyperparameters for each model were defined as presented in Table II to be chosen by Grid Search. It was performed the training of every regression model described in Section IV, i.e. MLR, RR, LASSO, EN, SVR with linear kernel, SVR with radial basis function (rbf) kernel, and GMM. The training and testing were performed 30 times ($it_{max} = 30$) for each model and for each input variables selection to determine the average

TABLE II: Hyperparameters ("**Hyper.**") for the regression models to be chosen by the Grid Search procedure.

Models	**Hyper.**	**Values**
MLR	-	-
RR	λ	1e-10, 1e-4, 1e-3, 1e-2, 1, 2, 5, 10, 20
LASSO	λ	1e-10, 1e-4, 1e-3, 1e-2, 1, 2, 5, 10, 20
EN	λ	1e-10, 1e-4, 1e-3, 1e-2, 1, 2, 5, 10, 20
	α	0:1 (0.1)
SVR (linear)	C	1e-2, 1e-1, 1, 10, 100, 1000
	ϵ	1e-3, 1e-2, 1e-1
SVR (rbf)	C	1e-2, 1e-1, 1, 10, 100, 1000
	ϵ	1e-3, 1e-2, 1e-1
	γ	1e-4, 1e-3, 1e-2, 1e-1, 1, 10
GMM	G	5, 10, 20, 50

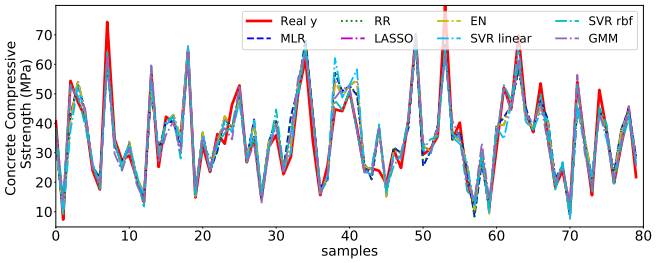
error. The metric used to evaluate the models on the test dataset for the 30 iterations is the Root Mean Square Error (RMSE).

C. Results

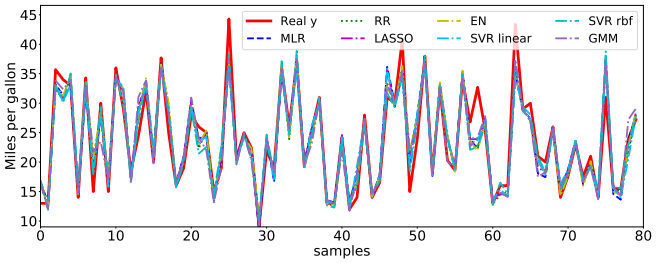
The results are displayed in Table III, where in the "**Variable Selection**" column, "without VS" represents the use of the original input variables and "VE" the use of the expanded variables. The labels "PC", "MI", "ft" and their values represent the thresholds chosen for Pearson Correlation, Mutual Information, and fastTracker, respectively. The third column ($|X|$) indicates the number of selected variables. And, Figure 1 contains the results of the predictions made with the best model found for each regression model for 80 random samples of the test dataset.

From the results, it is possible to see that, for the Concrete and the Automobile Gas datasets, the model with the lowest RMSE is the SVR with rbf kernel combined with the expanded variables. In fact, for these two datasets, the SVR with rbf model has almost the best result for each possible variable selection method. For Box-Jenkins dataset, MLR, RR, LASSO, and EN had the same and best result using the original variables. Furthermore, for the WTP Dataset the SVR with rbf kernel combined with the original variables reach the best model and had the best results for all the variable selection methods. From these results, it is clear that the SVR with rbf kernel performance stood out compared to the others (the best result in 26 of 35) and that, in general, the GMM was the worst option.

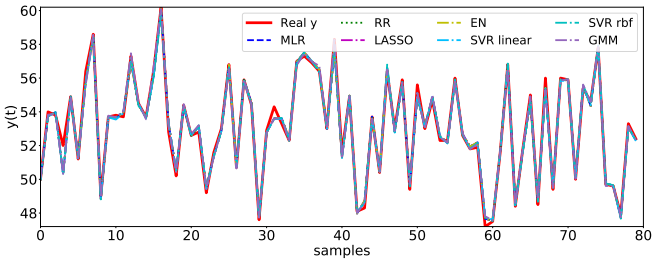
When looking at the variable selection methods, the observations show that the best results are achieved, generally, when using the expanded variables (Concrete and Automobile Gas datasets) or no variable selection method at all (Box-Jenkins and WTP datasets). Within the three variable selection methods, it is hard to conclude which one is the best or worst, since results vary from dataset to dataset, however, the fastTracker method is much faster than the others. It is also possible to conclude that there are regression models that depend more on the input variables than others. For example, the GMM model performance varies significantly depending on the number of input variables (in most cases, a higher $|X|$ results in worse predictions). For the Concrete and Automobile Gas datasets, the variable expansion improved the model's results.



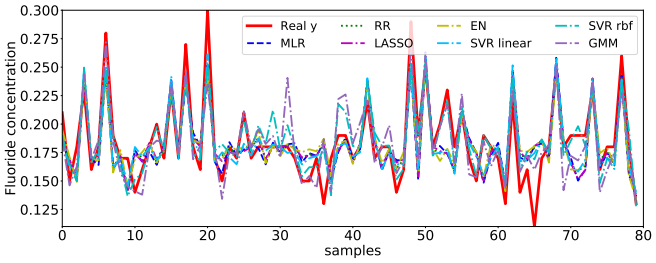
(a) Concrete dataset.



(b) Automobile Gas dataset.



(c) Box-Jenkins dataset.



(d) WTP dataset.

Fig. 1: Forecast results.

VII. Conclusions

This paper presented the development process of a Soft Sensor, doing a comparison between several regression models (MLR, RR, LASSO, EN, SVR, and GMM) and variable selection methods (Pearson Correlation, Mutual Information, and fastTracker) applied to expanded variables. Four datasets were used to compare the SS performance. From the tests, it was concluded that SVR with rbf kernel is the most robust model and that variable expansion tends to improve the model's performance.

TABLE III: RMSE results for the performed tests.

Dataset	Variable Selection		$\|X\|$	MLR	RR	LASSO	EN	SVR (linear)	SVR (rbf)	GMM
Concrete	without VS		8	10.439	10.443	10.441	10.444	10.731	6.082	8.812
	VE	-	57	6.002	5.994	6.230	6.234	6.374	**5.024**	74.599
		PC = 0.2	32	6.593	6.594	6.696	6.698	6.958	5.588	21.182
		PC = 0.3	26	6.739	6.759	6.749	6.760	6.923	5.570	8.870
		PC = 0.4	11	8.690	8.686	8.679	8.684	8.881	6.849	7.822
		MI = 1.0	26	10.641	10.636	10.773	10.764	11.213	7.604	23.669
		MI = 1.3	14	12.798	12.780	12.749	12.755	12.904	12.648	19.433
		fT = 0.5	26	6.792	6.786	7.023	7.022	7.273	5.650	13.741
		fT = 0.6	13	10.851	10.873	10.843	10.843	11.296	7.227	9.977
Automobile Gas	without VS		7	3.446	3.452	3.448	3.452	3.600	2.885	3.201
	VE	-	49	-	2.866	2.884	2.874	2.990	**2.782**	38.479
		PC=0.7	27	-	3.085	3.199	3.197	3.348	2.990	47.759
		PC=0.75	24	3.169	3.103	3.183	3.184	3.350	3.052	11.263
		PC=0.8	9	4.062	4.075	4.066	4.066	4.188	4.058	26.329
		MI = 1.95	24	3.140	3.168	3.173	3.198	3.249	3.083	23.185
		MI = 2.22	7	3.190	3.189	3.190	3.192	3.314	3.109	3.350
		fT = 0.8	22	2.987	2.957	2.911	2.920	2.981	3.005	25.113
		fT = 0.84	9	3.178	3.185	3.181	3.179	3.276	3.032	3.673
Box–Jenkins	without VS		8	**0.267**	**0.267**	**0.267**	**0.267**	0.278	0.280	0.302
	VE	-	52	0.304	0.278	0.273	0.273	0.278	0.296	0.868
		PC = 0.7	28	0.338	0.324	0.319	0.319	0.321	0.320	1.046
		PC = 0.8	23	0.339	0.321	0.318	0.319	0.322	0.320	1.199
		PC = 0.9	18	0.331	0.318	0.317	0.318	0.321	0.323	0.731
		MI = 2.65	28	0.337	0.317	0.317	0.317	0.326	0.314	0.815
		MI = 2.75	18	0.330	0.316	0.318	0.317	0.319	0.318	0.567
		fT = 0.725	28	0.283	0.272	0.272	0.270	0.292	0.274	0.722
		fT = 0.74	16	0.327	0.317	0.323	0.323	0.329	0.322	0.468
WTP	without VS		55	0.024	0.021	0.021	0.021	0.023	**0.019**	0.025
	VE	PC = 0.7	52	0.026	0.023	0.024	0.024	0.025	0.021	0.090
		PC = 0.72	26	0.026	0.025	0.025	0.025	0.026	0.022	0.050
		PC = 0.73	11	0.025	0.025	0.025	0.025	0.026	0.022	0.029
		MI = 0.85	51	0.023	0.021	0.022	0.022	0.021	0.021	0.041
		M2 = 0.87	22	0.022	0.021	0.021	0.021	0.022	0.021	0.034
		fT = 0.725	46	0.023	0.022	0.022	0.022	0.022	0.022	0.063
		fT = 0.73	21	0.025	0.025	0.025	0.025	0.025	0.024	0.034

In future work, other models and datasets from the industry should be compared, and methodologies should be developed to automatically identify which time delay of a given variable is more representative.

REFERENCES

[1] F. Souza, T. Offermans, R. Barendse, G. Postma, and J. Jansen, "Contextual mixture of experts: Integrating knowledge into predictive modeling," *IEEE Transactions on Industrial Informatics*, pp. 1–12, 2022.

[2] F. A. A. Souza, R. Araújo, and J. Mendes, "Review of soft sensors methods for regression applications," *Chemometrics and Intelligent Laboratory Systems*, vol. 152, pp. 69–79, March 2016.

[3] L. Fortuna, S. Graziani, A. Rizzo, M. G. Xibilia *et al.*, *Soft sensors for monitoring and control of industrial processes*. Springer, 2007, vol. 22.

[4] Q. Sun and Z. Ge, "A survey on deep learning for data-driven soft sensors," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 5853–5866, 2021.

[5] ——, "A survey on deep learning for data-driven soft sensors," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 5853–5866, 2021.

[6] M.-T. Puth, M. Neuhäuser, and G. D. Ruxton, "Effective use of pearson's product–moment correlation coefficient," *Animal behaviour*, vol. 93, pp. 183–189, 2014.

[7] T. M. Cover and J. A. Thomas, *Entropy, Relative Entropy and Mutual Information*. John Wiley & Sons, Ltd, 1991, ch. 2, pp. 12–49.

[8] M. Gonçalves, P. Sousa, J. Mendes, M. Danishvar, and A. Mousavi, "Real-time event-driven learning in highly volatile systems: A case for embedded machine learning for scada systems," *IEEE Access*, vol. 10, pp. 50 794–50 806, 2022.

[9] S. Raschka and V. Mirjalili, "Python machine learning: Machine learning and deep learning with python," *Scikit-Learn, and TensorFlow. Second edition ed*, vol. 3, 2017.

[10] F. Souza, R. Araújo, S. Soares, and J. Mendes, "Variable selection based on mutual information for soft sensors applications," in *Proc. 9th Portuguese Conference on Automatic Control (CONTROLO 2010)*, Coimbra, Portugal, September 8-10 2010, pp. 1–6.

[11] J. D. Jobson, *Multiple Linear Regression*. New York, NY: Springer New York, 1991, pp. 219–398.

[12] Q. Hu, C. Xu, and Y. Dong, "Dam deformation analysis based on ridge regression," in *2009 International Conference on Information Engineering and Computer Science*. IEEE, 2009, pp. 1–4.

[13] D. Schreiber-Gregory, "Ridge regression and multicollinearity: An in-depth review," *Model Assisted Statistics and Applications*, vol. 13, 09 2018.

[14] T. Hastie, R. Tibshirani, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, 2nd ed. New York, NY: Springer, 2009.

[15] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[16] C.-H. Wu, J.-M. Ho, and D.-T. Lee, "Travel-time prediction with support vector regression," *IEEE transactions on intelligent transportation systems*, vol. 5, no. 4, pp. 276–281, 2004.

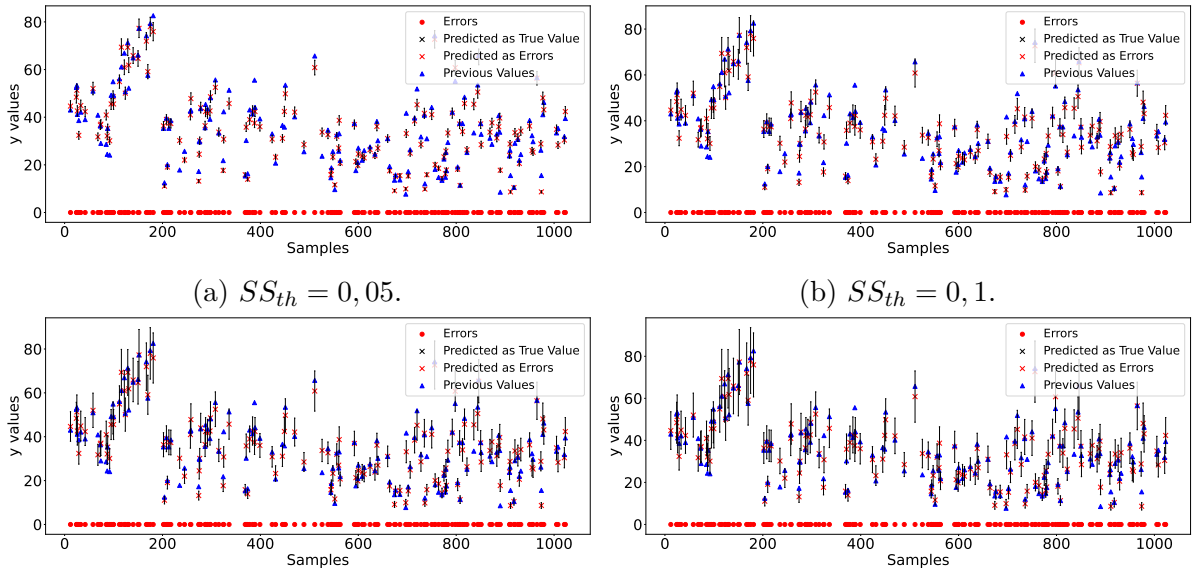
[17] A. Fabisch, "gmr: Gaussian mixture regression," *Journal of Open Source Software*, vol. 6, no. 62, p. 3054, 2021.

[18] R. Salles, J. Mendes, R. Araújo, C. Melo, and P. Moura, "Prediction of key variables in wastewater treatment plants using machine learning models," in *Proc. 2022 IEEE International Joint Conference on Neural Networks (IJCNN 2022), at the 2022 World Congress on Computational Intelligence (WCCI 2022)*, Padova, Italy, July 18-23 2022, pp. 1–9.

Appendix B

Classification and Regression Results Visualization

B.1 Soft Sensor regression plots



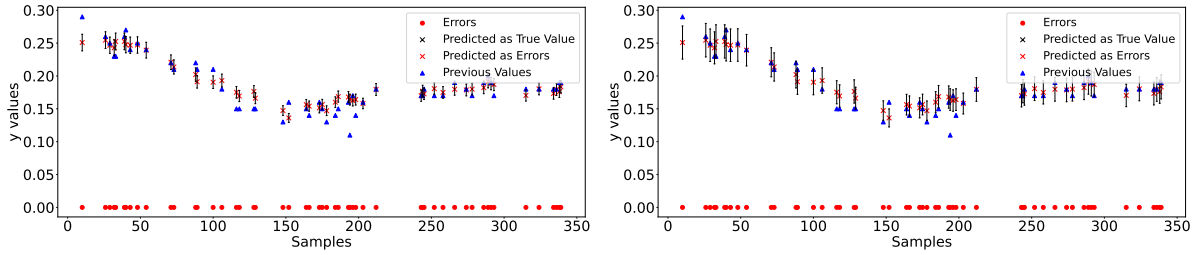
(a) $SS_{th} = 0,05$.

(b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$.

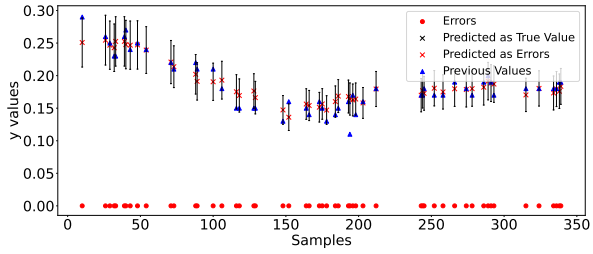
(d) $SS_{th} = 0,2$.

Figure B.1: Error prediction results for Concrete dataset with blank spaces, using Soft Sensors.

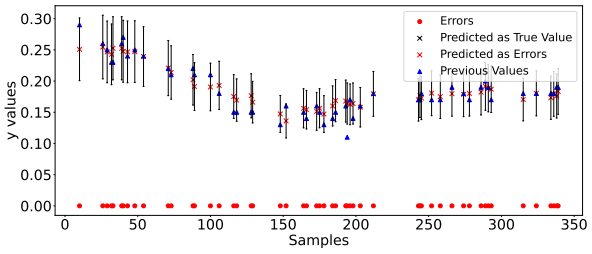


(a) $SS_{th} = 0,05$.

(b) $SS_{th} = 0,1$.

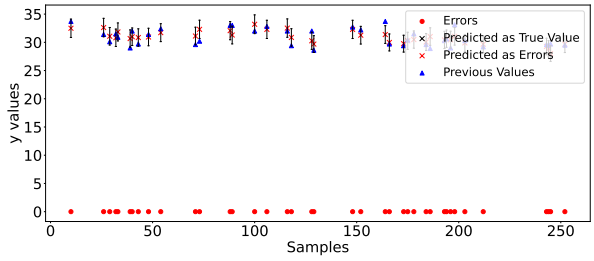


(c) $SS_{th} = 0,15$.

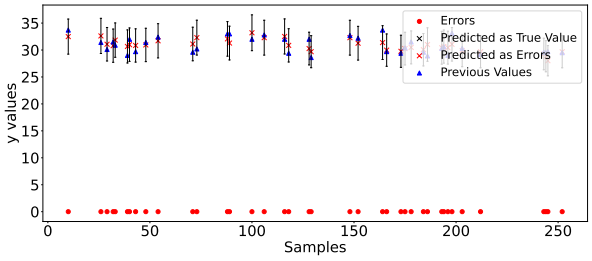


(d) $SS_{th} = 0,2$.

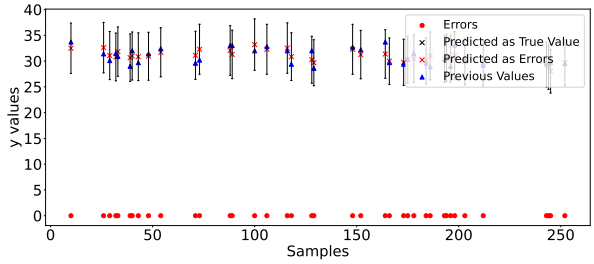
Figure B.2: Error prediction results for WTP dataset with blank spaces, using Soft Sensors.



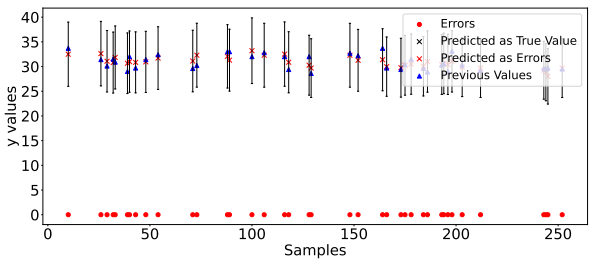
(a) $SS_{th} = 0,05$.



(b) $SS_{th} = 0,1$.

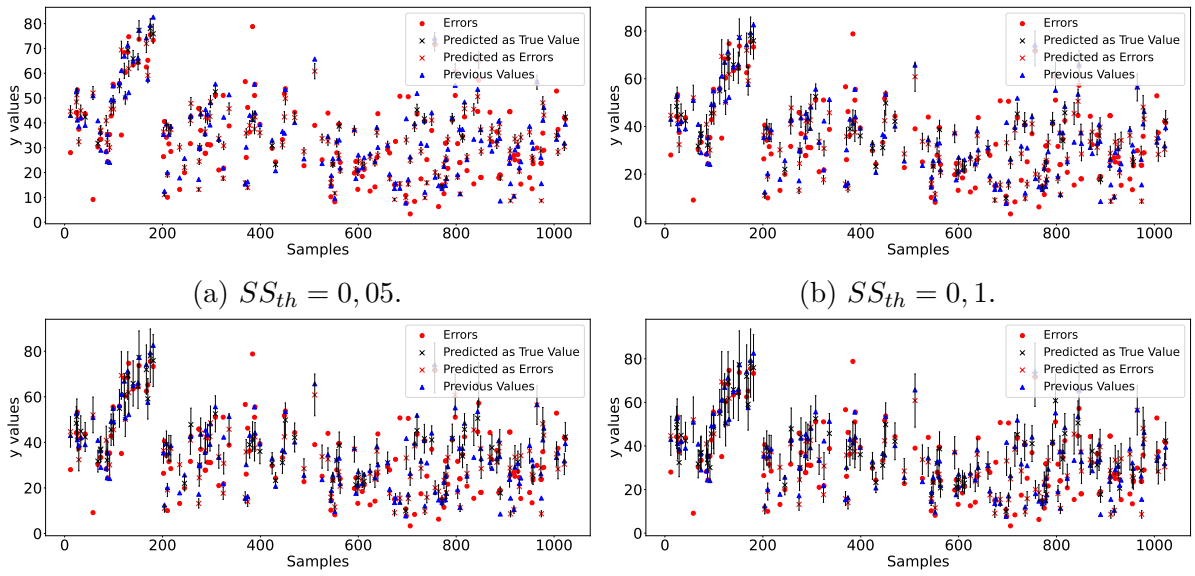


(c) $SS_{th} = 0,15$.



(d) $SS_{th} = 0,2$.

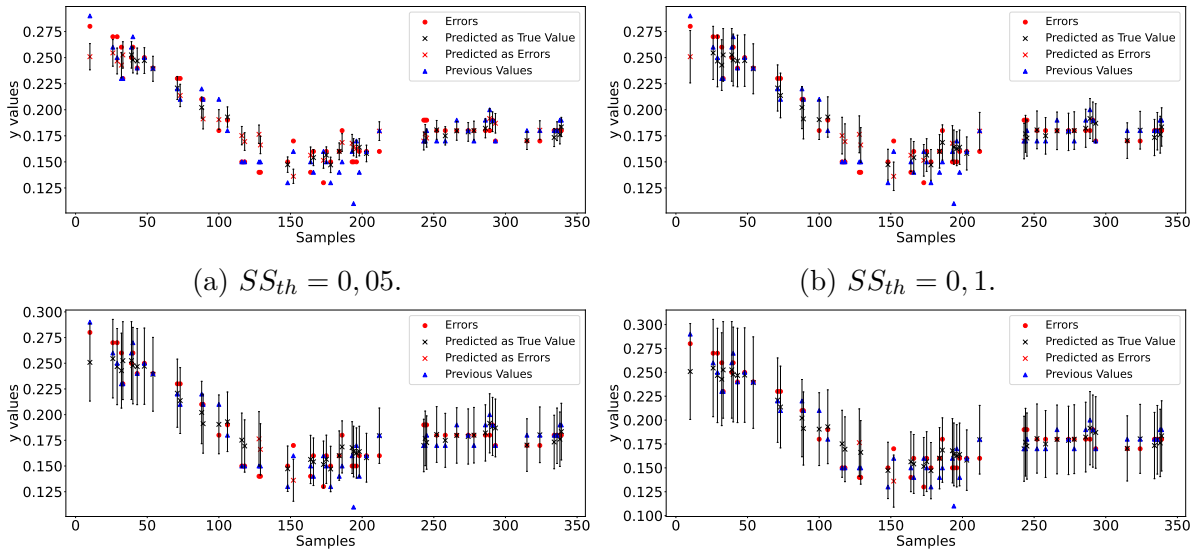
Figure B.3: Error prediction results for Cement dataset with blank spaces, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

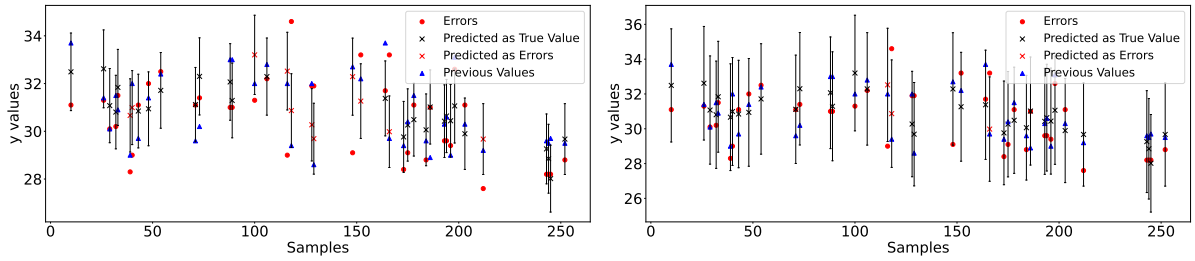
Figure B.4: Error prediction results for Concrete dataset with doubles, using Soft Sensors.



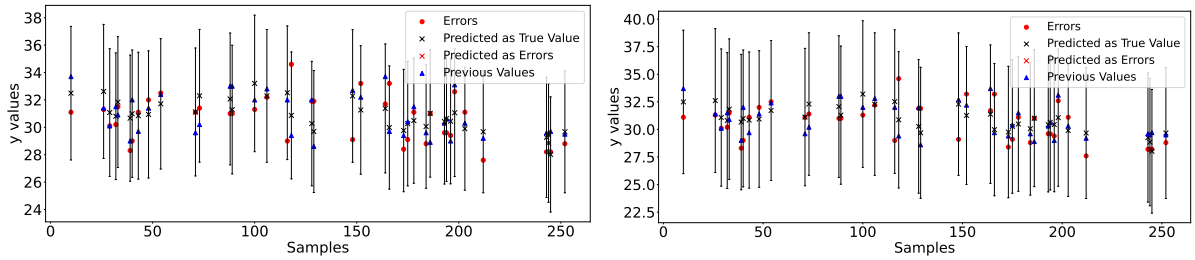
(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

Figure B.5: Error prediction results for WTP dataset with doubles, using Soft Sensors.

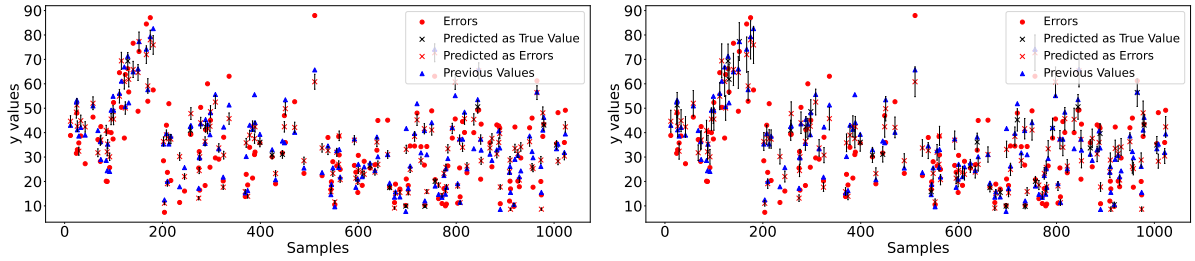


(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

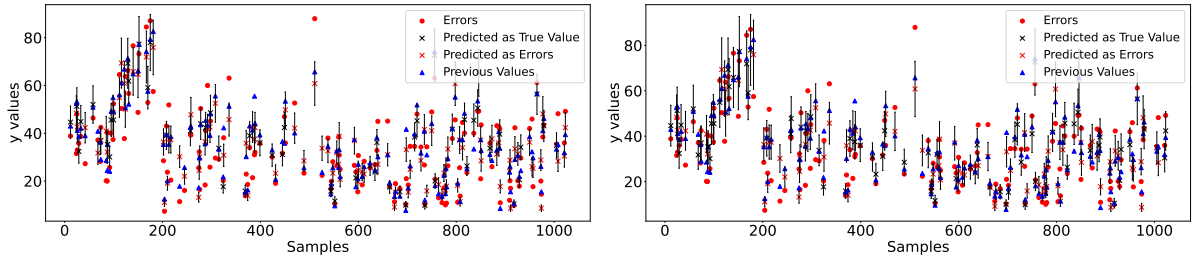


(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

Figure B.6: Error prediction results for Cement dataset with doubles, using Soft Sensors.

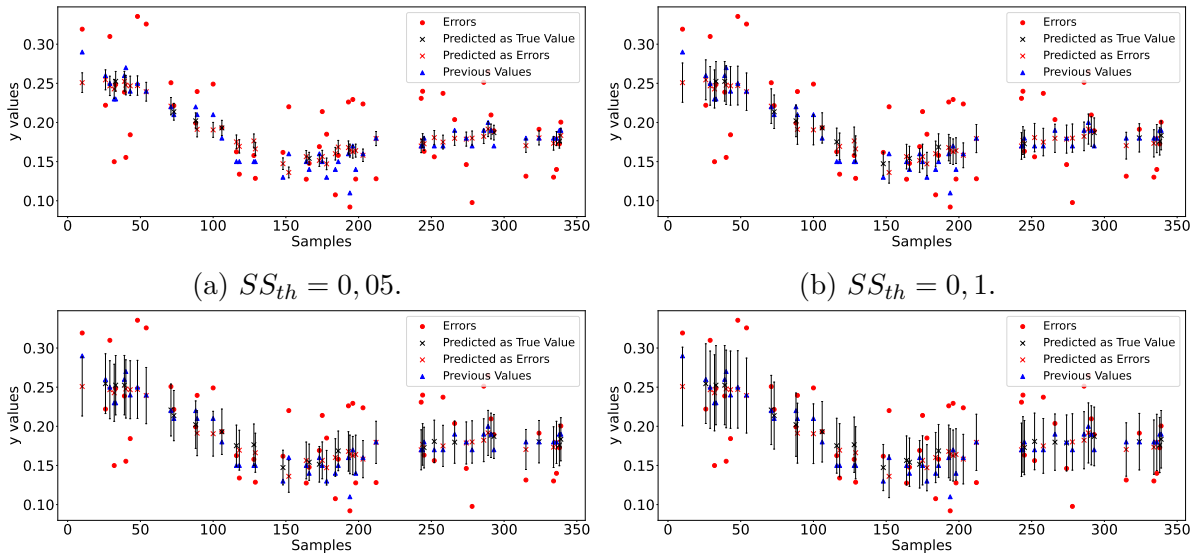


(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.



(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

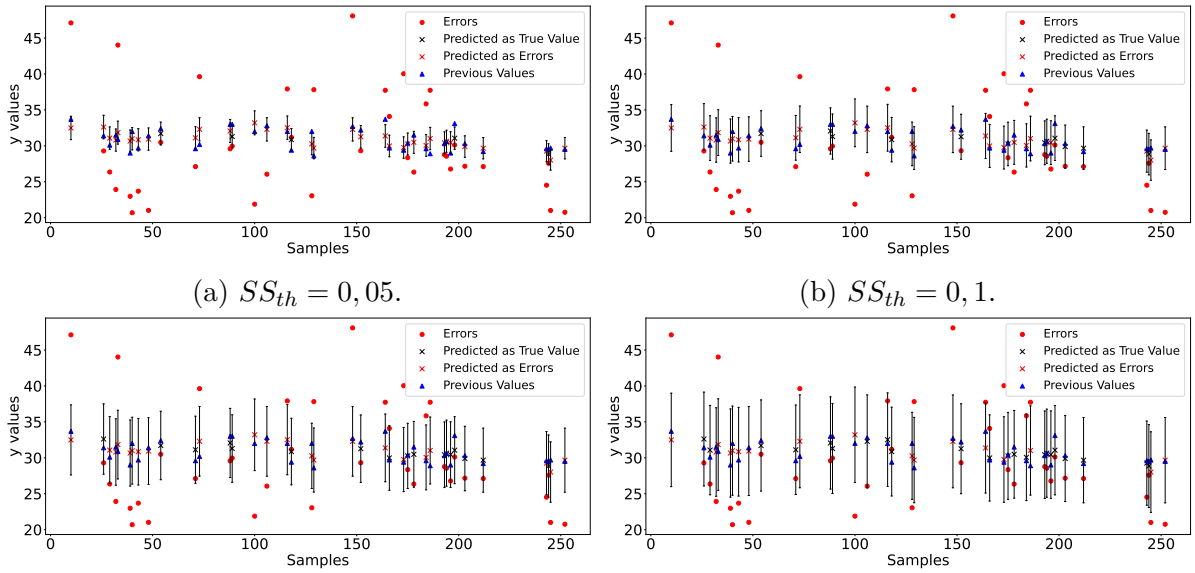
Figure B.7: Error prediction results for Concrete dataset with measurement errors, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

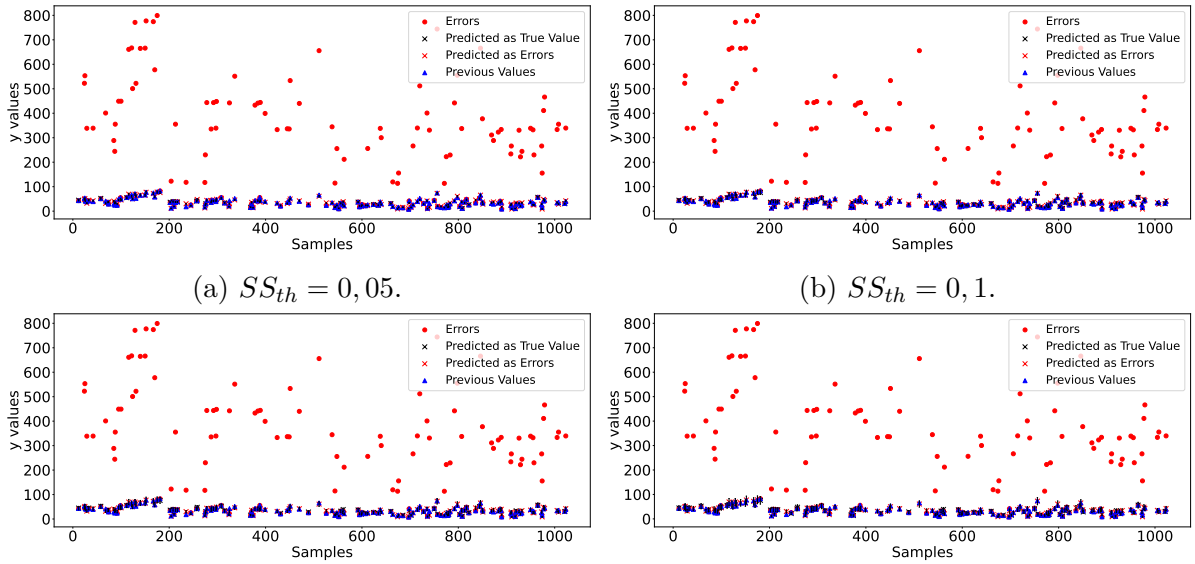
Figure B.8: Error prediction results for WTP dataset with measurement errors, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

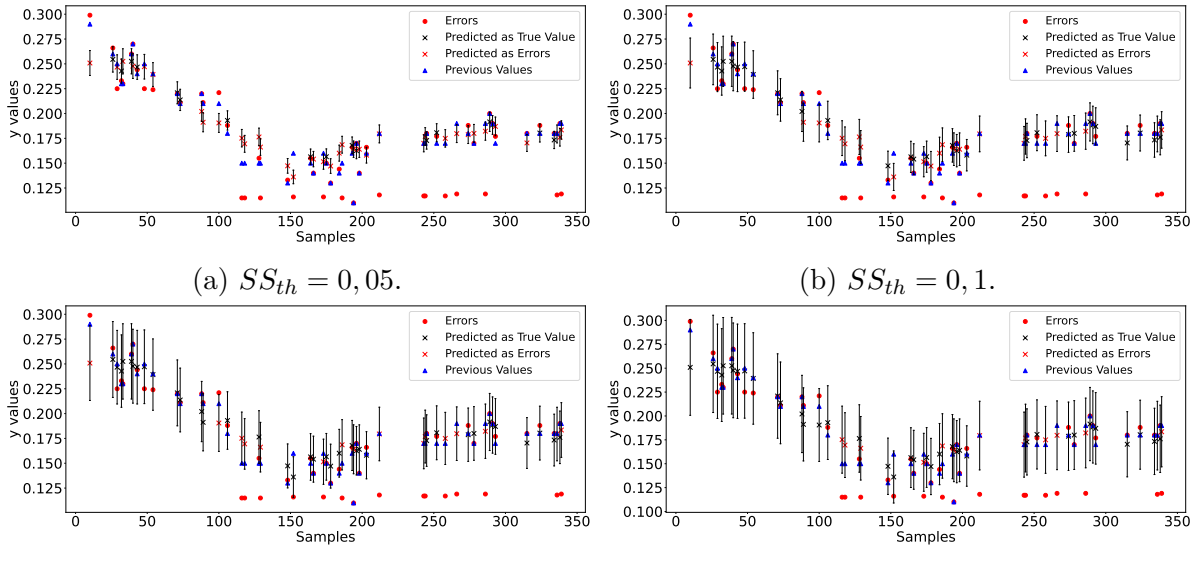
Figure B.9: Error prediction results for Cement dataset with measurement errors, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

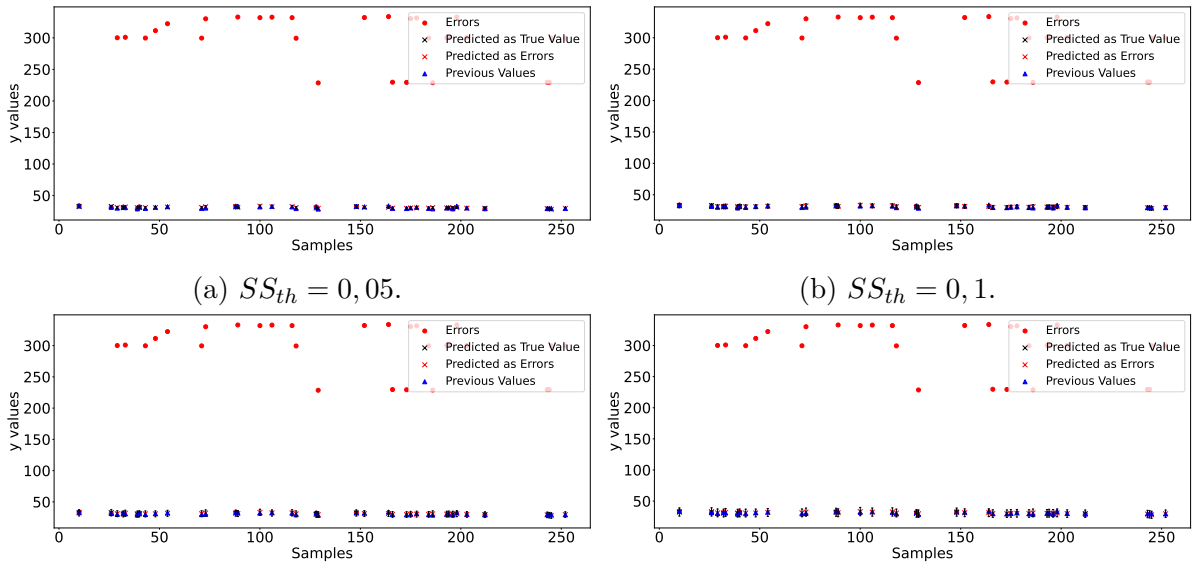
Figure B.10: Error prediction results for Concrete dataset with extra numbers, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

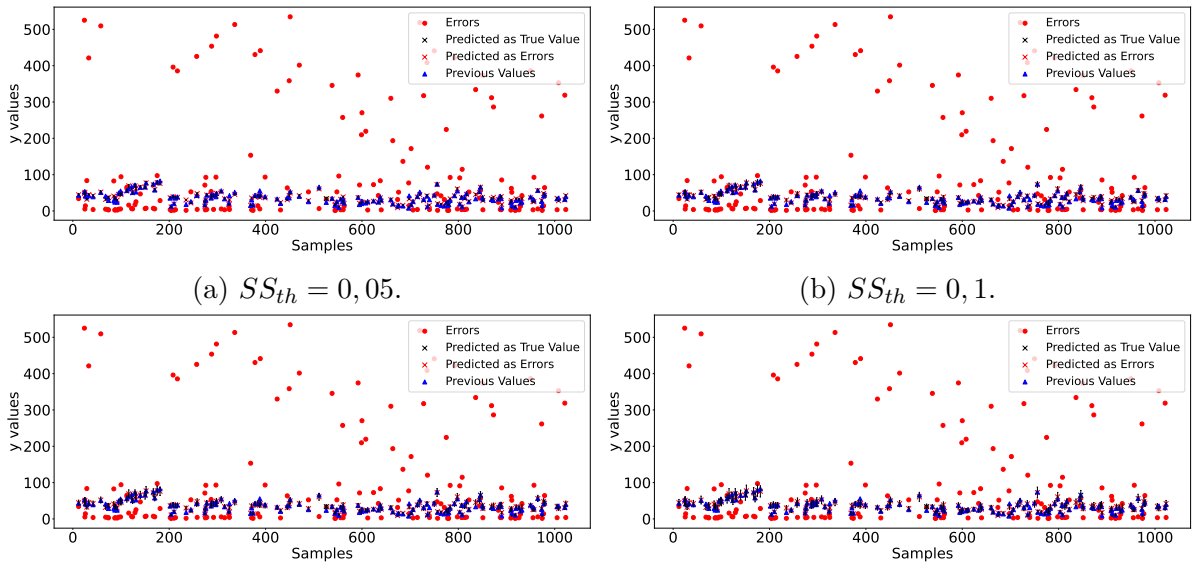
Figure B.11: Error prediction results for WTP dataset with extra numbers, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

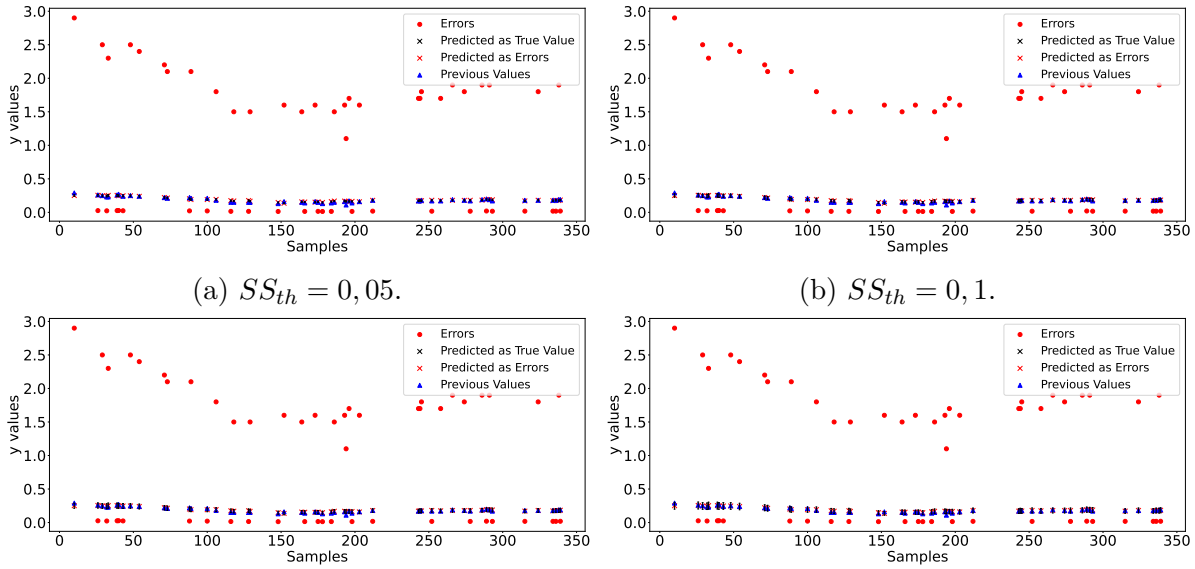
Figure B.12: Error prediction results for Cement dataset with extra numbers, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

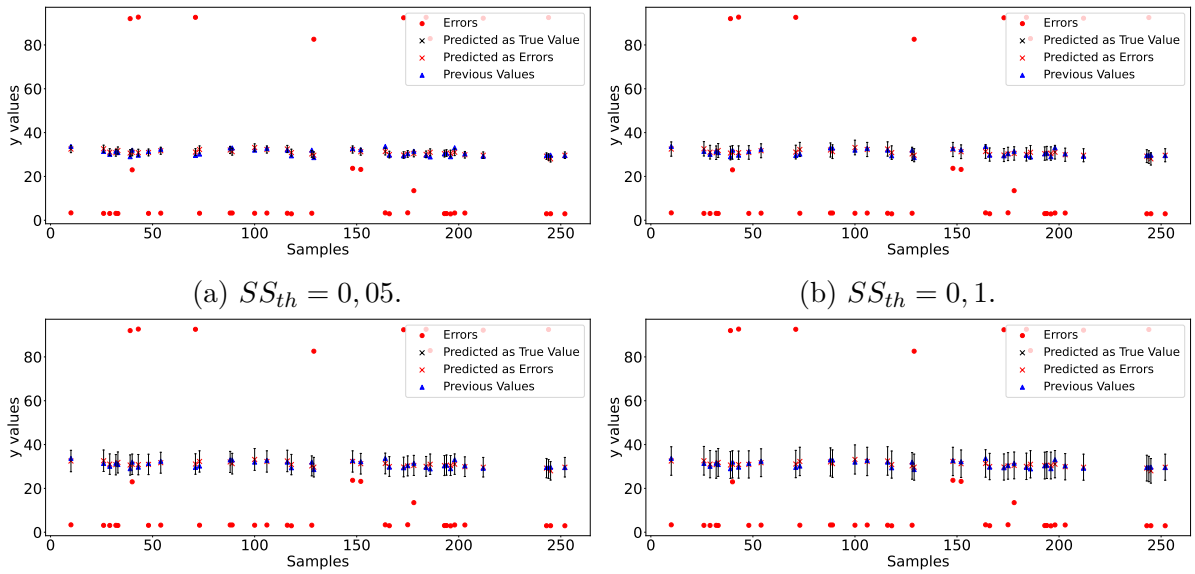
Figure B.13: Error prediction results for Concrete dataset with order errors, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

Figure B.14: Error prediction results for WTP dataset with order errors, using Soft Sensors.

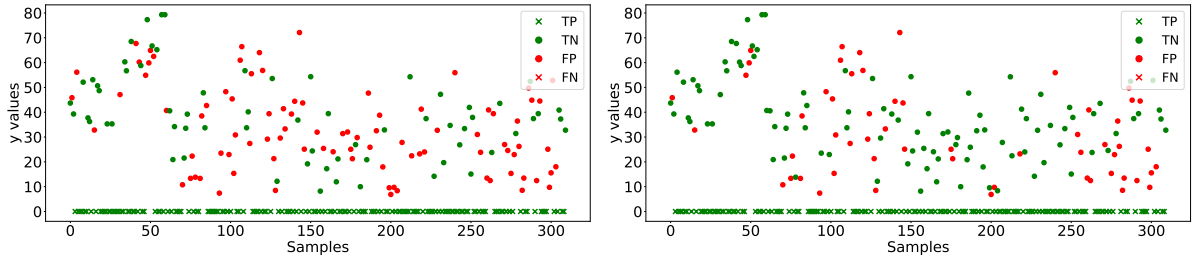


(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

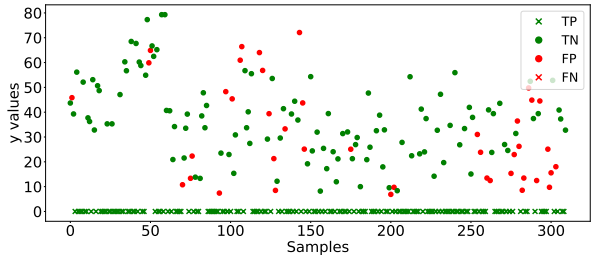
Figure B.15: Error prediction results for Cement dataset with order errors, using Soft Sensors.

B.2 Soft Sensor classification plots

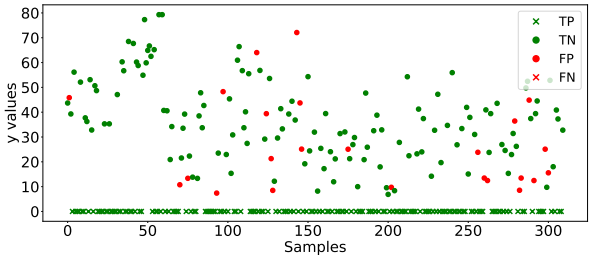


(a) $SS_{th} = 0,05$.

(b) $SS_{th} = 0,1$.

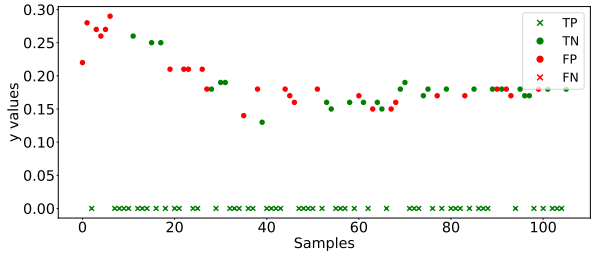


(c) $SS_{th} = 0,15$.

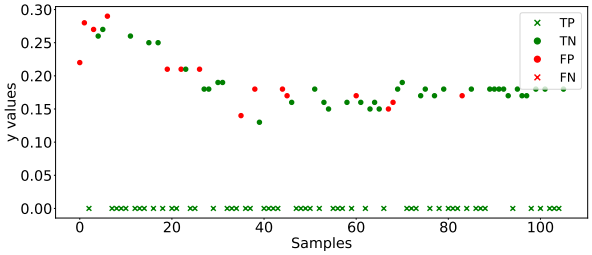


(d) $SS_{th} = 0,2$.

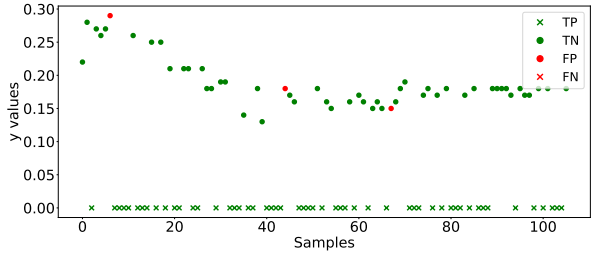
Figure B.16: Classification results for Concrete dataset with blank spaces, using Soft Sensors.



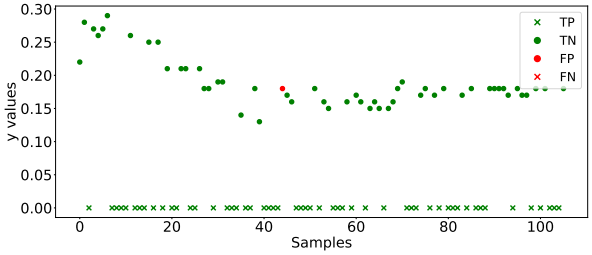
(a) $SS_{th} = 0,05$.



(b) $SS_{th} = 0,1$.

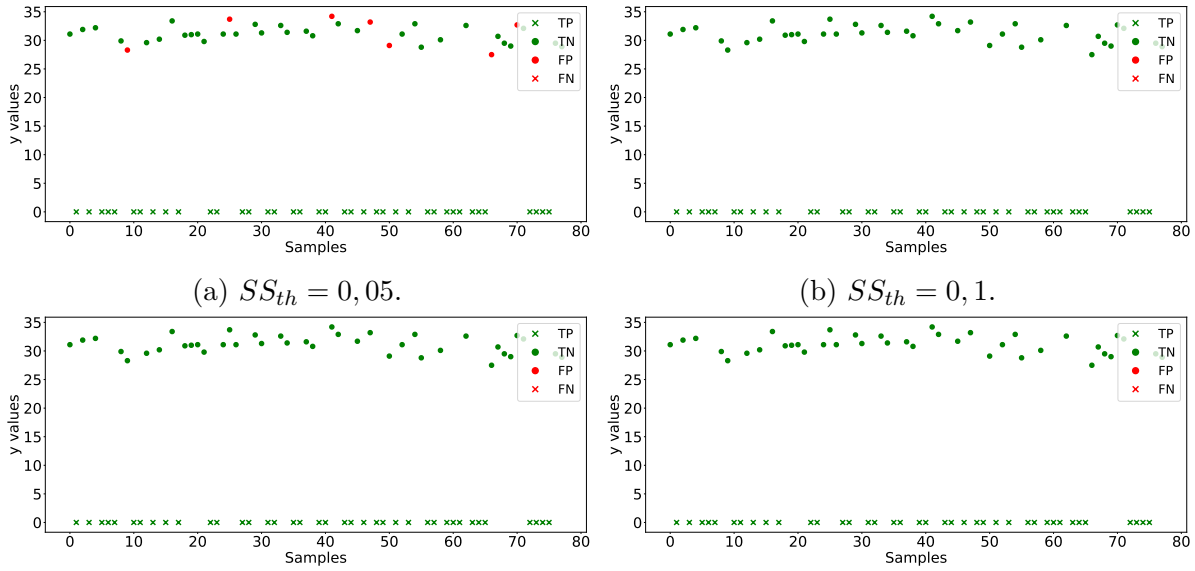


(c) $SS_{th} = 0,15$.



(d) $SS_{th} = 0,2$.

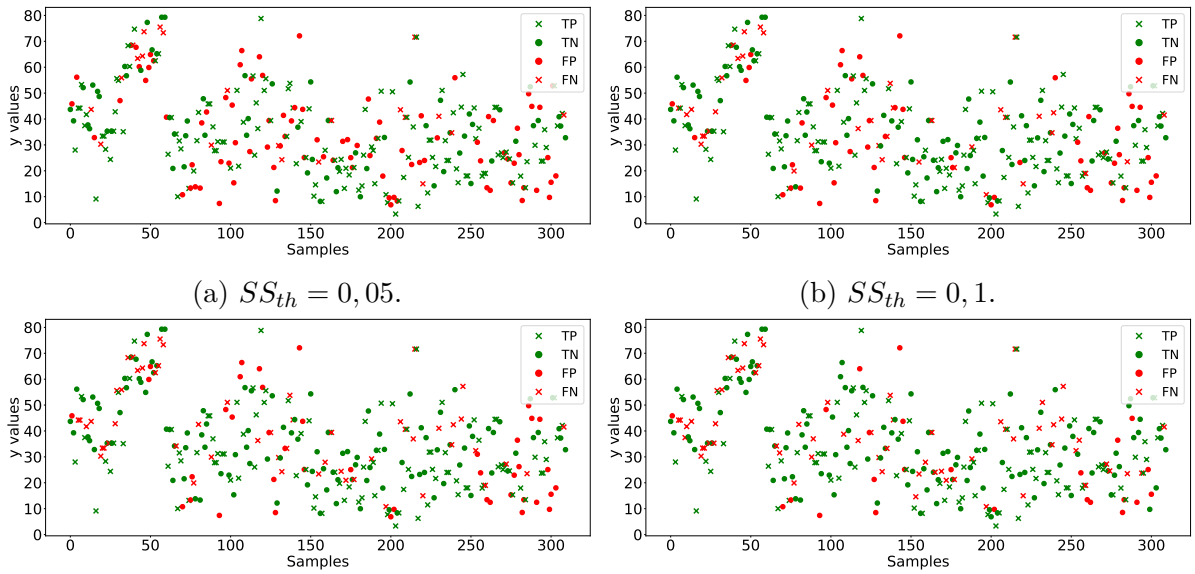
Figure B.17: Classification results for WTP dataset with blank spaces, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

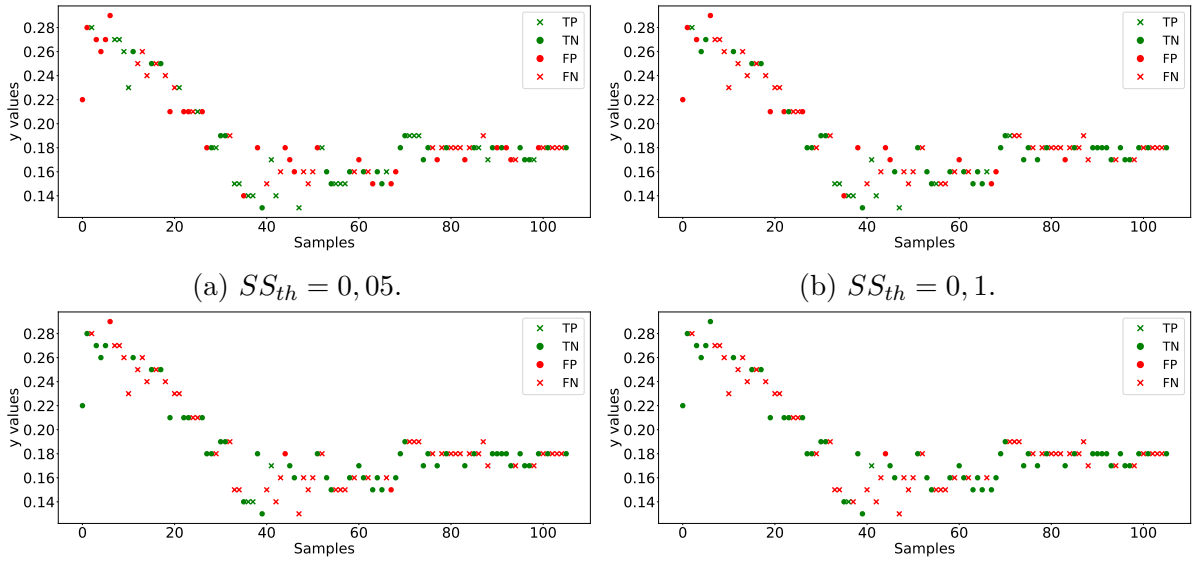
Figure B.18: Classification results for Cement dataset with blank spaces, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

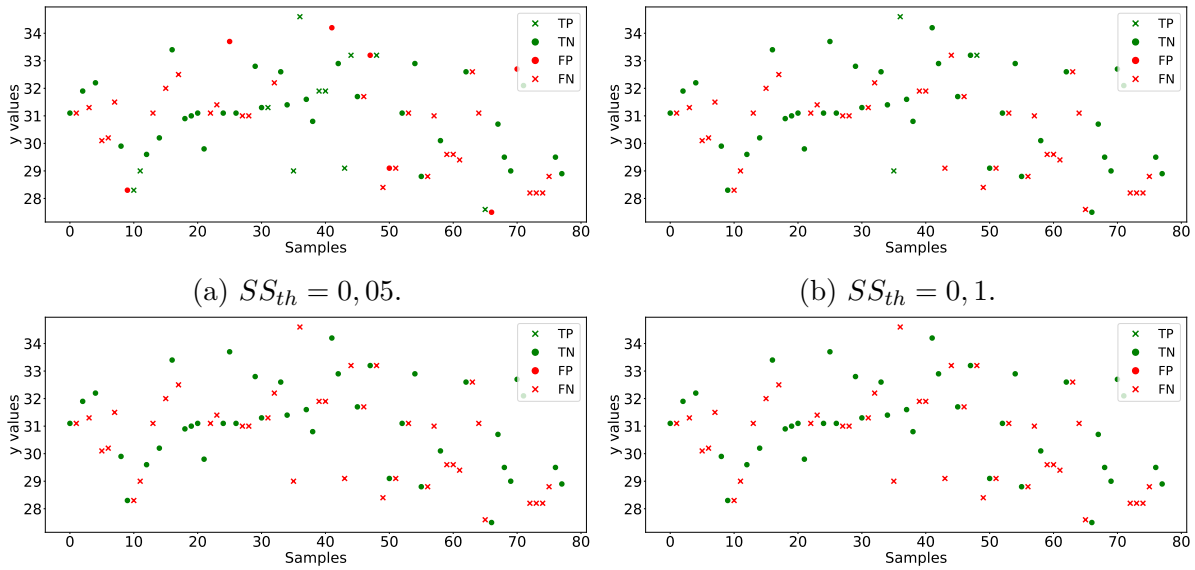
Figure B.19: Classification results for Concrete dataset with doubles, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

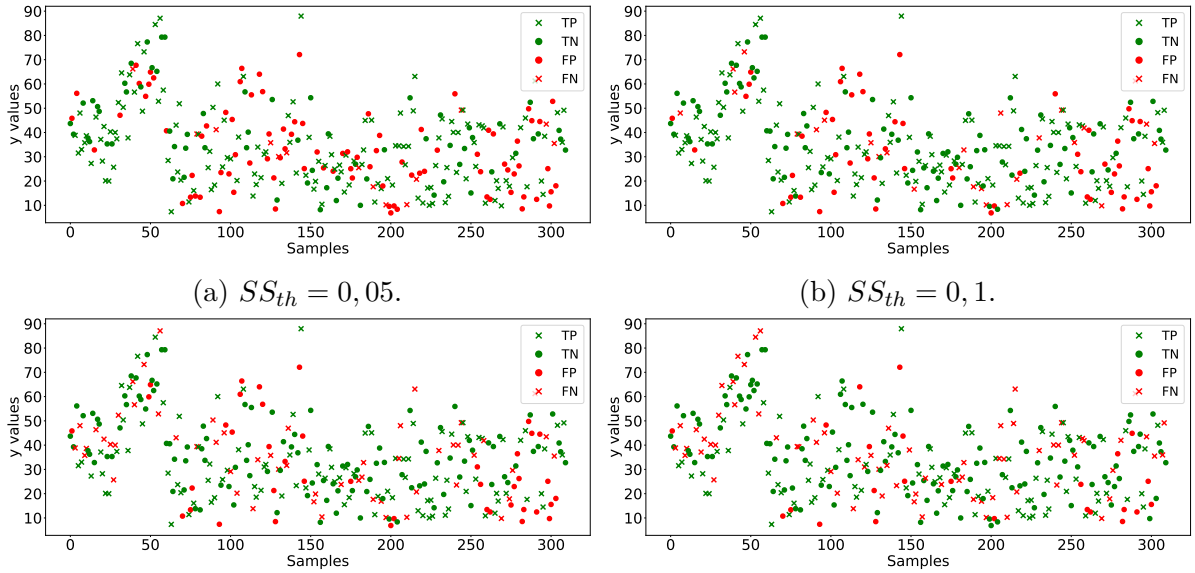
Figure B.20: Classification results for WTP dataset with doubles, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

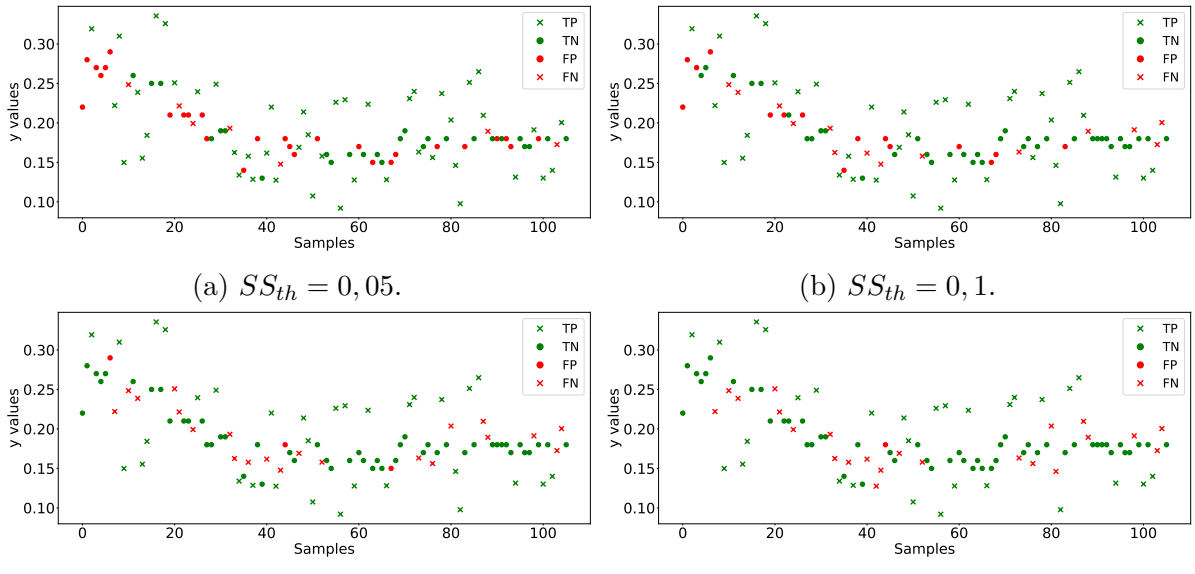
Figure B.21: Classification results for Cement dataset with doubles, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

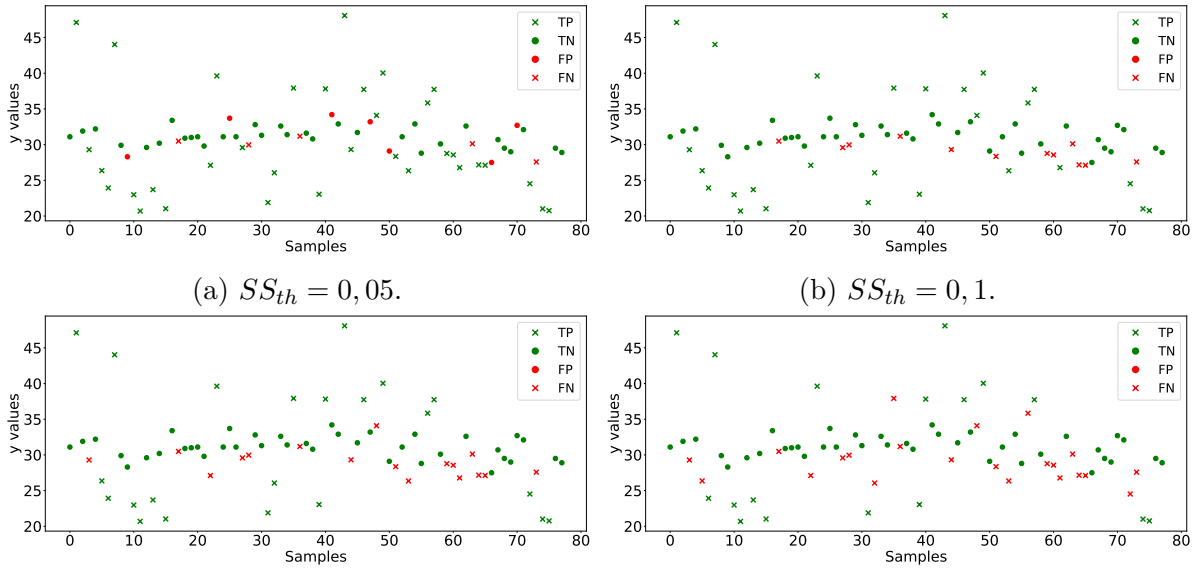
Figure B.22: Classification results for Concrete dataset with measurement errors, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

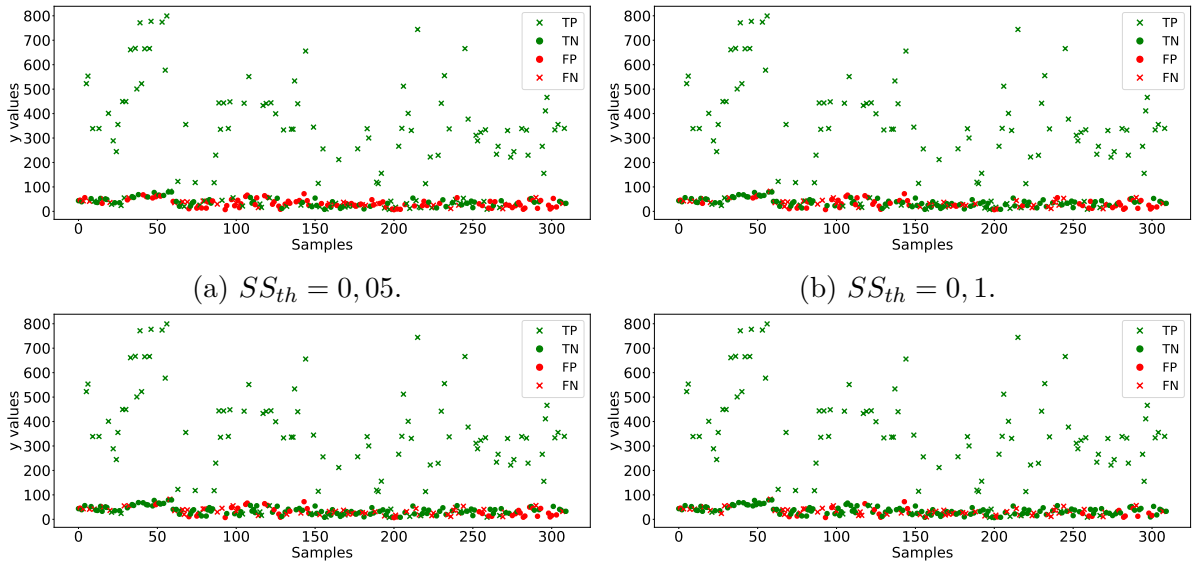
Figure B.23: Classification results for WTP dataset with measurement errors, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

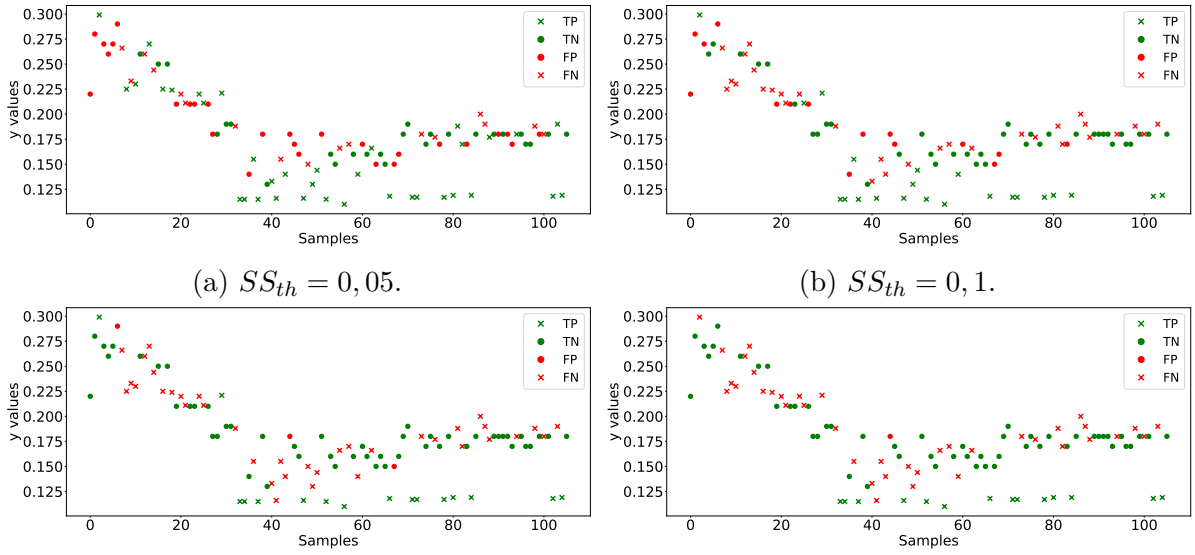
Figure B.24: Classification results for Cement dataset with measurement errors, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

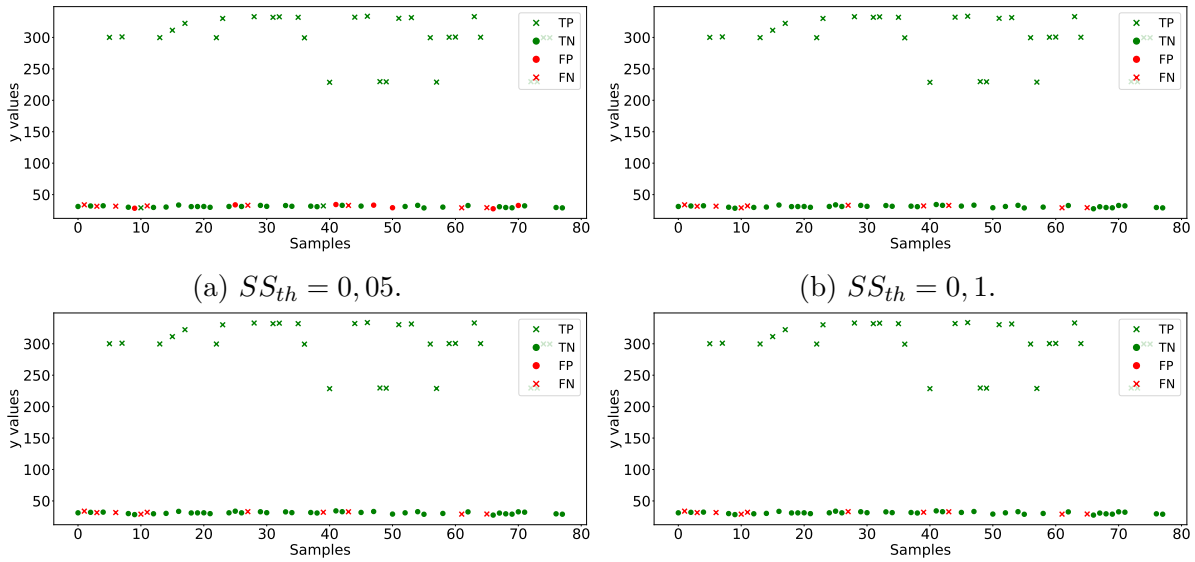
Figure B.25: Classification results for Concrete dataset with extra number errors, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

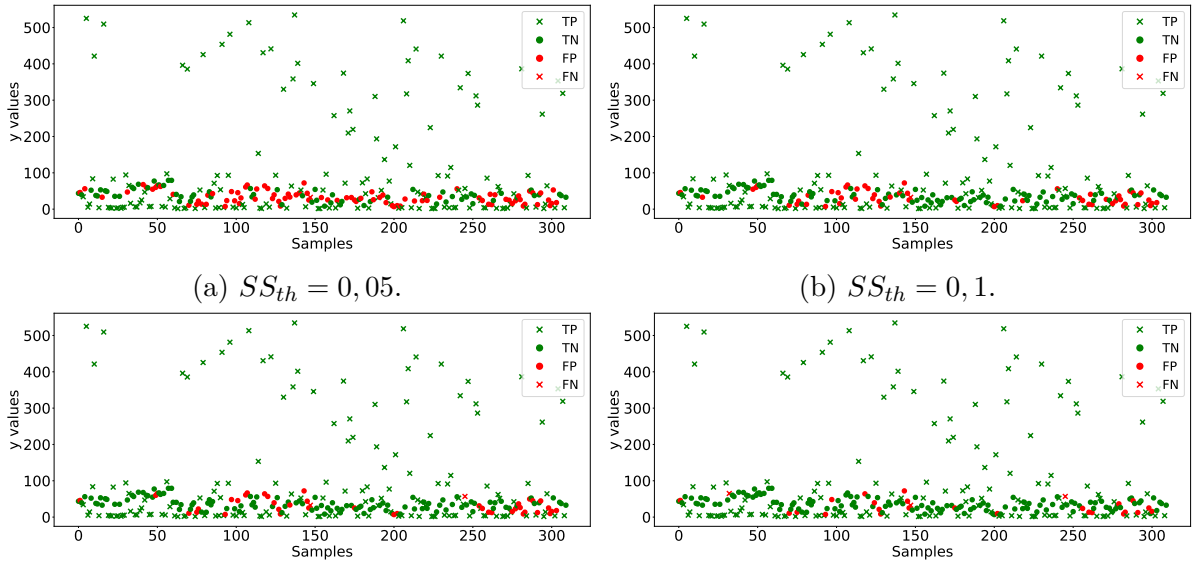
Figure B.26: Classification results for WTP dataset with extra number errors, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

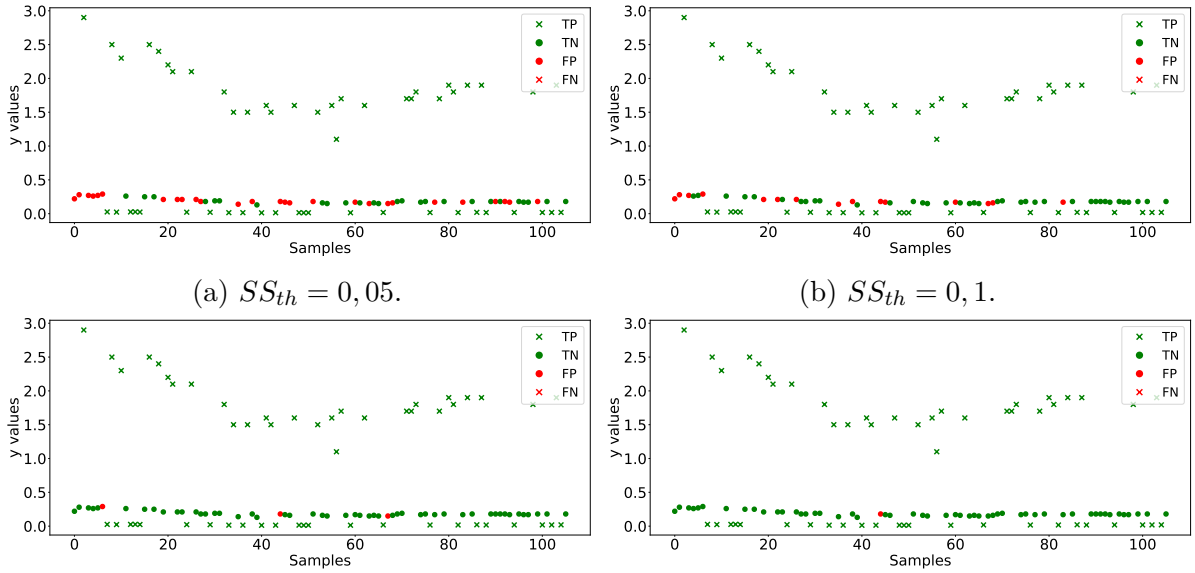
Figure B.27: Classification results for Cement dataset with extra number errors, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

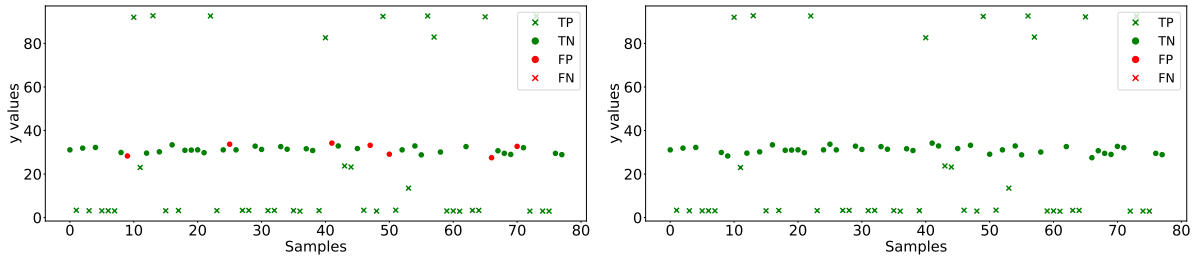
Figure B.28: Classification results for Concrete dataset with order errors, using Soft Sensors.



(a) $SS_{th} = 0,05$. (b) $SS_{th} = 0,1$.

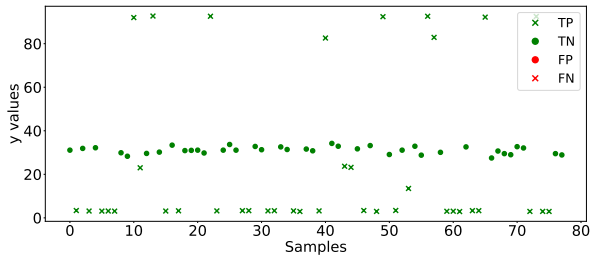
(c) $SS_{th} = 0,15$. (d) $SS_{th} = 0,2$.

Figure B.29: Classification results for WTP dataset with order errors, using Soft Sensors.

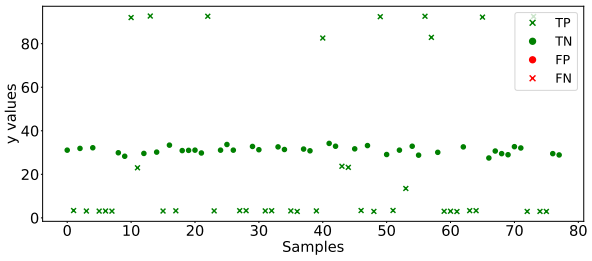


(a) $SS_{th} = 0,05$.

(b) $SS_{th} = 0,1$.



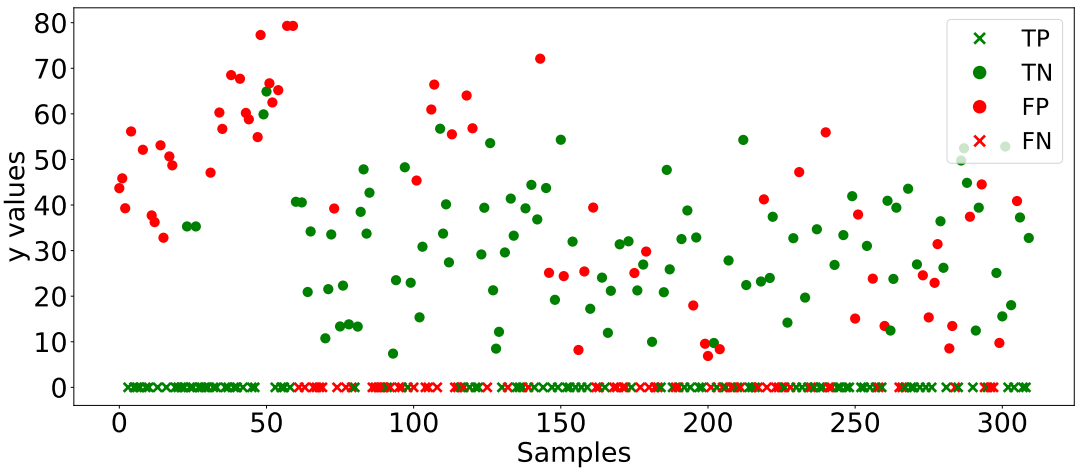
(c) $SS_{th} = 0,15$.



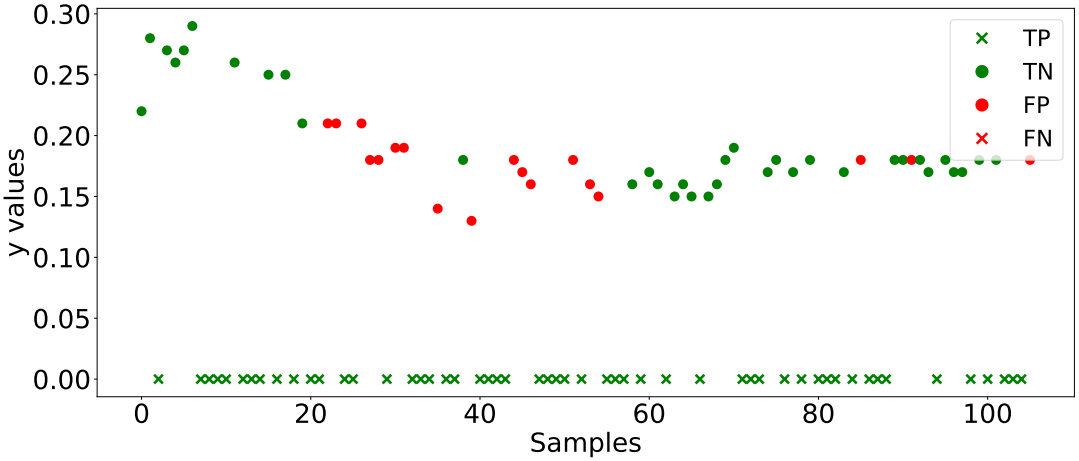
(d) $SS_{th} = 0,2$.

Figure B.30: Classification results for Cement dataset with order errors, using Soft Sensors.

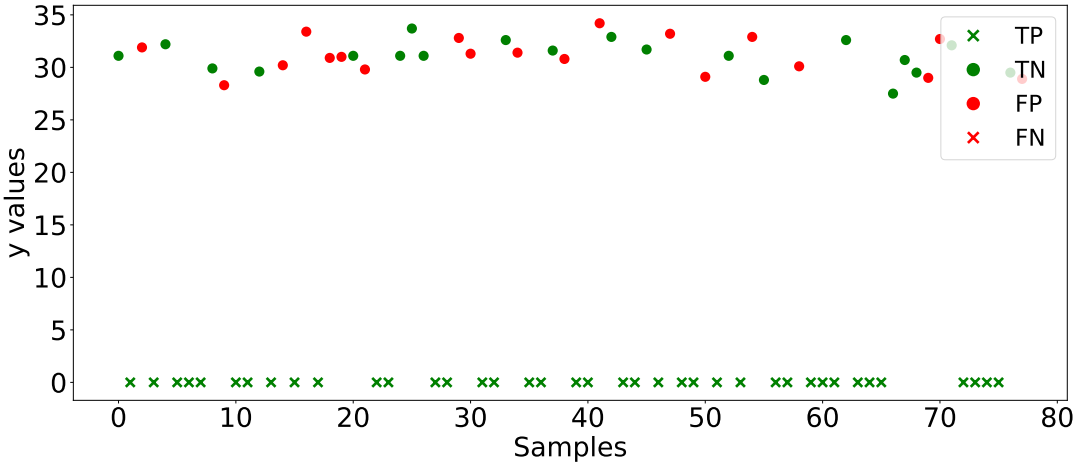
B.3 PCA classification plots



(a) Concrete dataset.

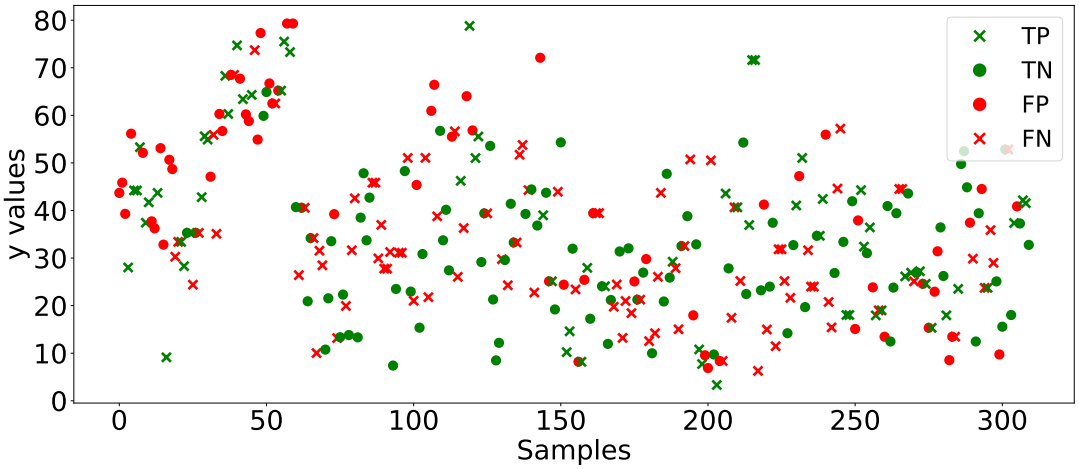


(b) WTP dataset.

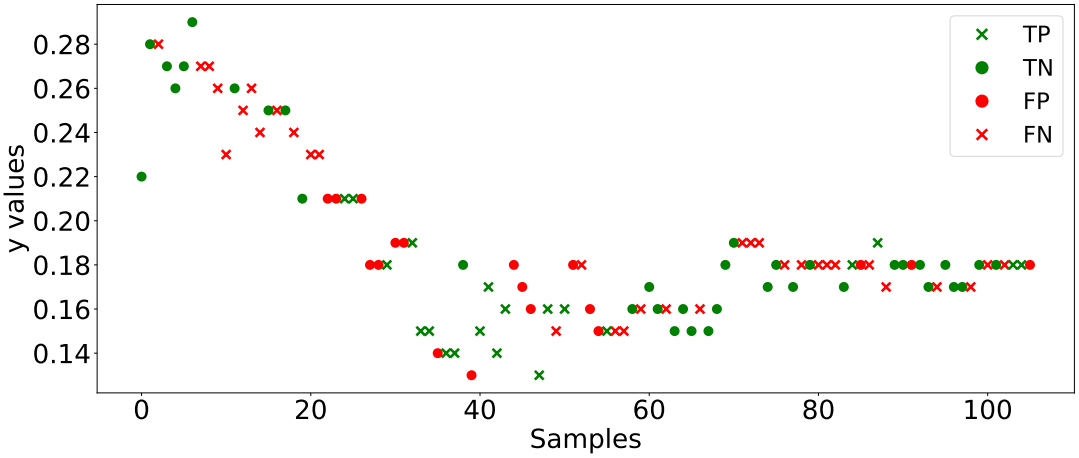


(c) Cement dataset.

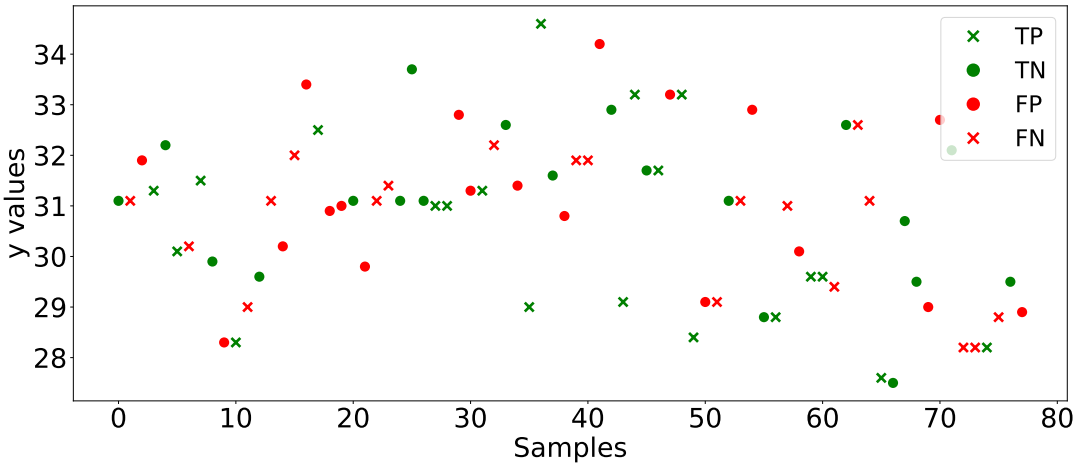
Figure B.31: PCA classification results for blank spaces.



(a) Concrete dataset.

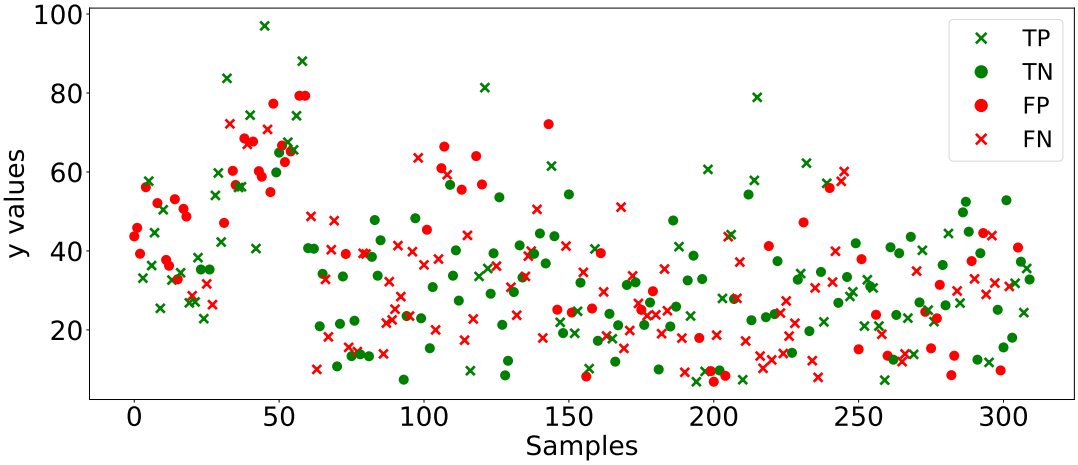


(b) WTP dataset.

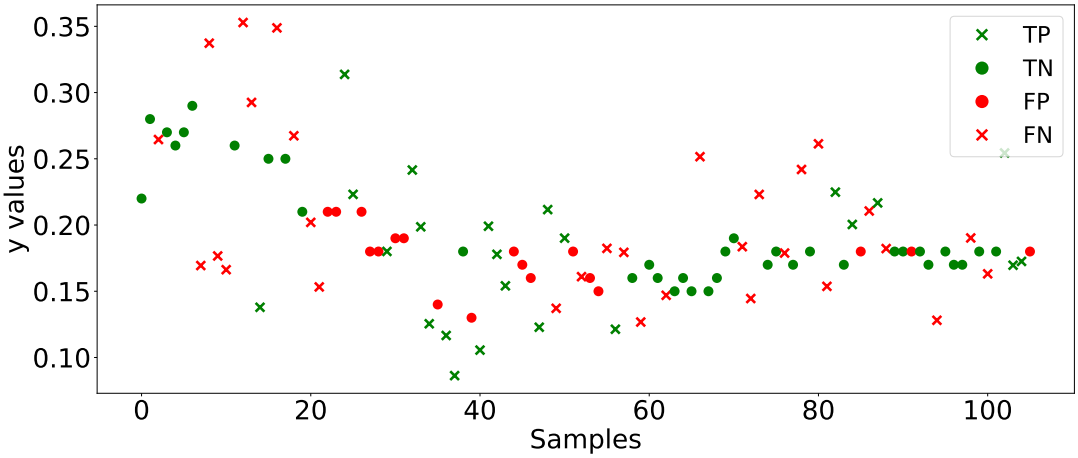


(c) Cement dataset.

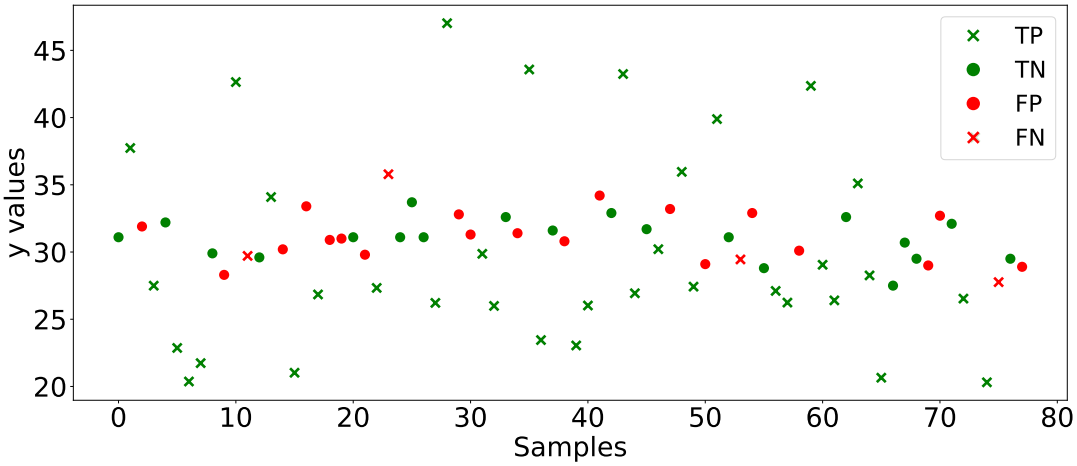
Figure B.32: PCA classification results for doubles.



(a) Concrete dataset.

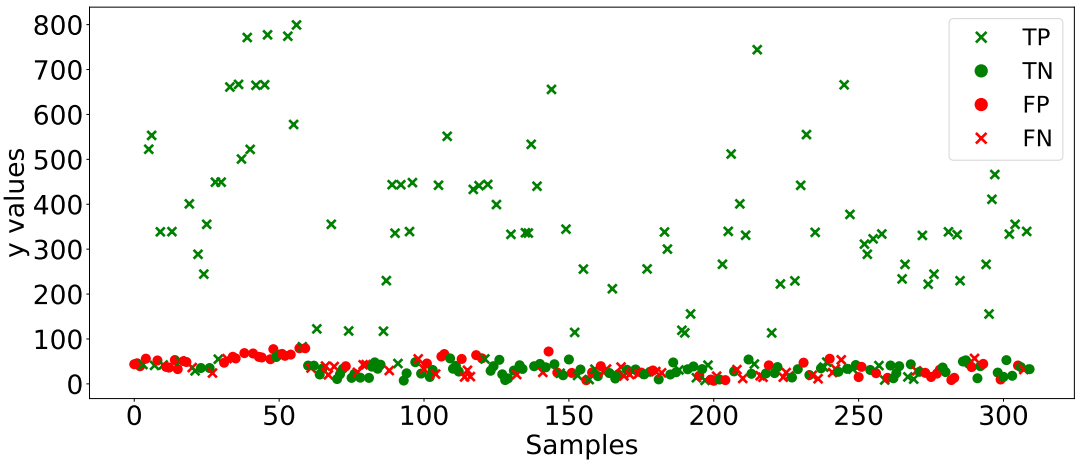


(b) WTP dataset.

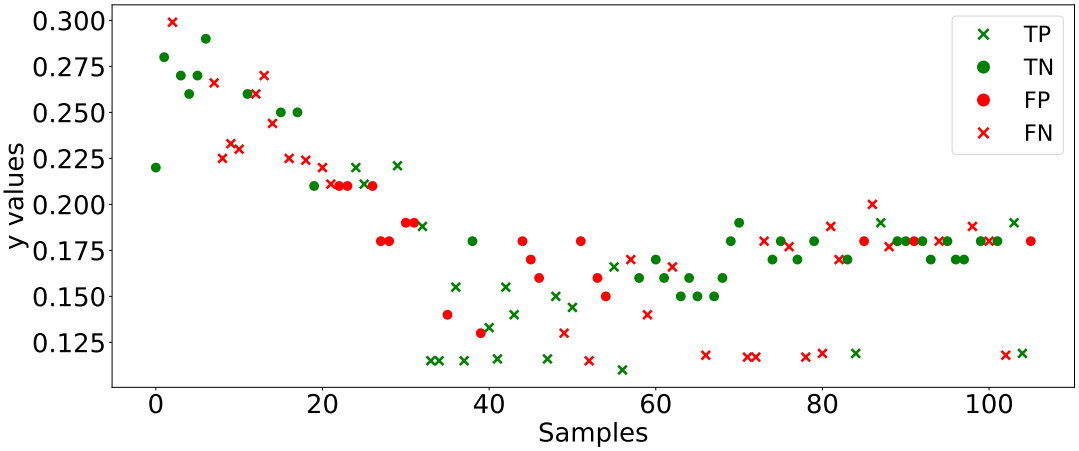


(c) Cement dataset.

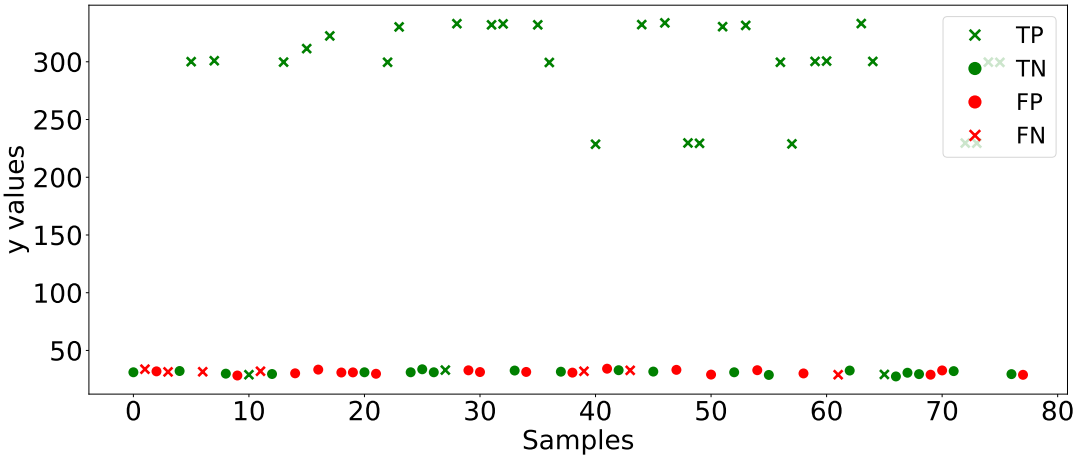
Figure B.33: PCA classification results for measurement errors.



(a) Concrete dataset.

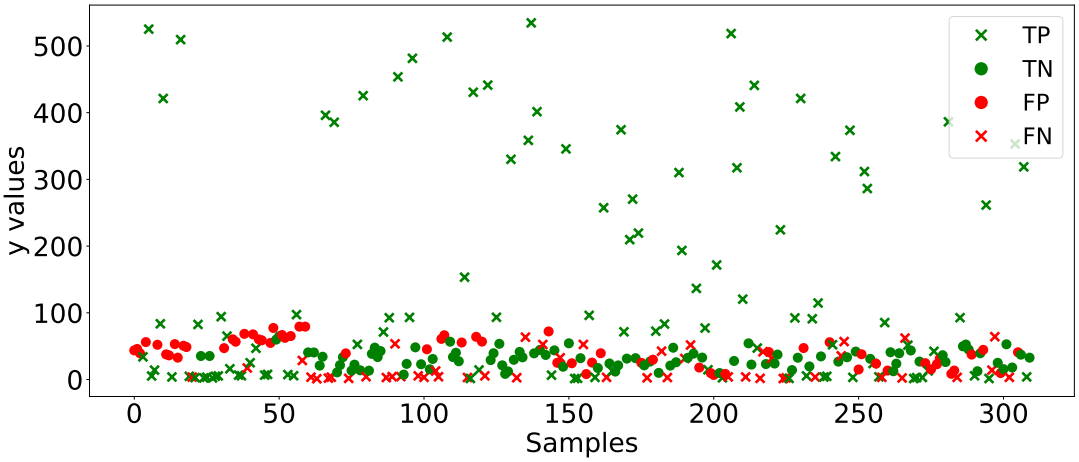


(b) WTP dataset.

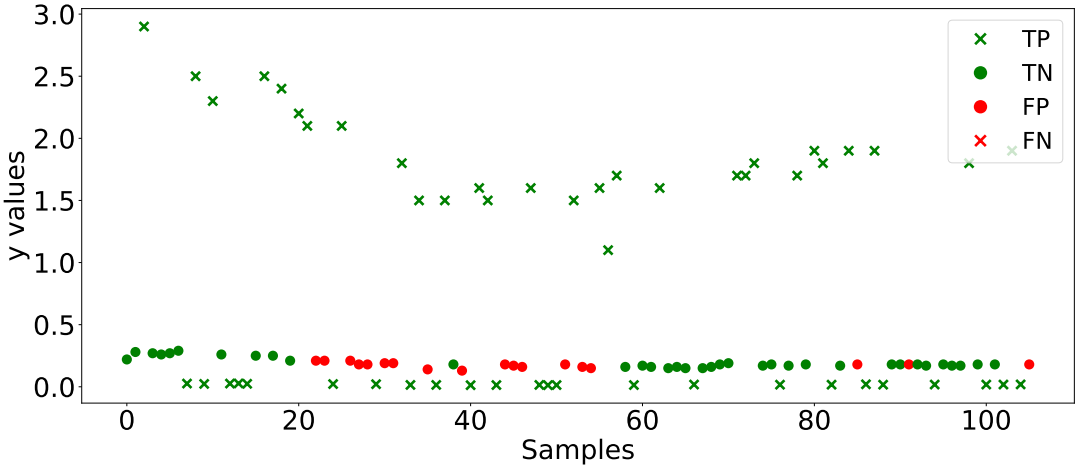


(c) Cement dataset.

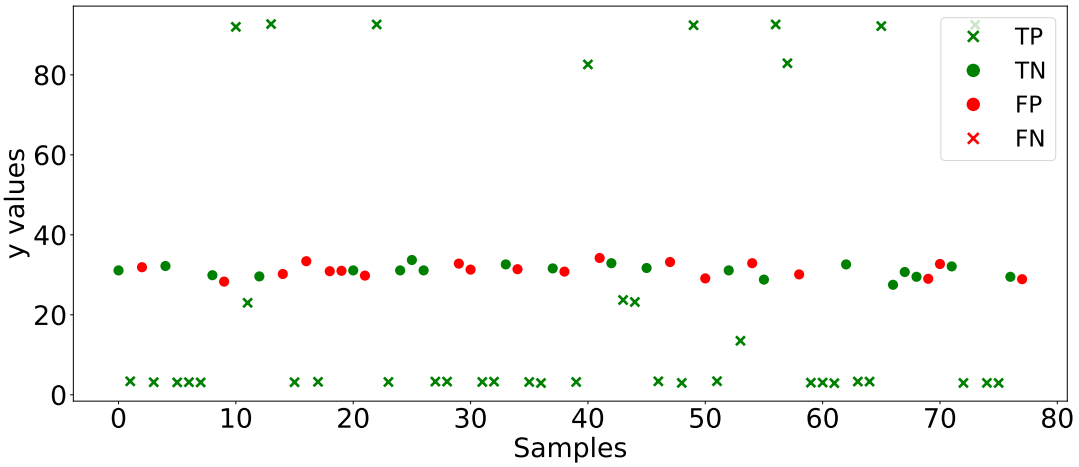
Figure B.34: PCA classification results for extra numbers.



(a) Concrete dataset.



(b) WTP dataset.



(c) Cement dataset.

Figure B.35: PCA classification results for order errors.