ÉLIA YATHIE MATSUMOTO

A METHODOLOGY FOR IMPROVING COMPUTED INDIVIDUAL REGRESSIONS PREDICTIONS

São Paulo

2016

ÉLIA YATHIE MATSUMOTO

A METHODOLOGY FOR IMPROVING COMPUTED INDIVIDUAL REGRESSIONS PREDICTIONS

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Doutor em Ciências.

Área de concentração: Sistemas Eletrônicos

Orientador: Prof. Dr. Emílio Del Moral Hernandez, Livre Docente

São Paulo

2016

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 12 de fevereiro de 2016.

_____

Élia Yathie Matsumoto

_____

Prof. Dr. Emílio Del Moral Hernandez

FICHA CATALOGRÁFICA

Matsumoto, Élia Yathie
A Methodology for Improving Computed Individual Regressions Predictions / E. Y. Matsumoto – versão corr. -- São Paulo, 2016.
118 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Estatística aplicada - em engenharia 2.Regressão 3.Aprendizado computacional 4.Reconhecimento de padrões 5.Redes neurais I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

AGRADECIMENTOS

Ao professor Emílio del Moral Hernandez, orientador deste trabalho.

À professora Maria Cristina Vidal Borba.

Ao professor Martin Edward Weber.

Aos colegas e amigos do Grupo de Pesquisa ICONE-EPUSP (Grupo de Inteligência Computacional, Modelagem e Neurocomputação Eletrônica).

Aos demais professores e funcionários do Departamento de Engenharia Elétrica da Universidade de São Paulo.

Aos colegas e amigos da Opencadd.

À minha família.

RESUMO

Esta pesquisa propõe uma metodologia para melhorar previsões calculadas por um modelo de regressão, sem a necessidade de modificar seus parâmetros ou sua arquitetura. Em outras palavras, o objetivo é obter melhores resultados por meio de ajustes nos valores computados pela regressão, sem alterar ou reconstruir o modelo de previsão original.

A proposta é ajustar os valores previstos pela regressão por meio do uso de estimadores de confiabilidade individuais capazes de indicar se um determinado valor estimado é propenso a produzir um erro considerado crítico pelo usuário da regressão.

O método proposto foi testado em três conjuntos de experimentos utilizando três tipos de dados diferentes. O primeiro conjunto de experimentos trabalhou com dados produzidos artificialmente, o segundo, com dados transversais extraídos no repositório público de dados UCI Machine Learning Repository, e o terceiro, com dados do tipo séries de tempos extraídos do ISO-NE (*Independent System Operator in New England*).

Os experimentos com dados artificiais foram executados para verificar o comportamento do método em situações controladas. Nesse caso, os experimentos alcançaram melhores resultados para dados limpos artificialmente produzidos e evidenciaram progressiva piora com a adição de elementos aleatórios.

Os experimentos com dados reais extraído das bases de dados UCI e ISO-NE foram realizados para investigar a aplicabilidade da metodologia no mundo real. O método proposto foi capaz de melhorar os valores previstos por regressões em cerca de 95% dos experimentos realizados com dados reais.

Palavras-chave: Melhoria em Previsões de Regressões, Estimadores de Confiabilidade de Individuais, Aprendizado de Máquinas, Reconhecimento de Padrões, Redes Neurais Artificiais, Máquina de Comitê de Redes Neurais, Dados Desbalanceados.

ABSTRACT

This research proposes a methodology to improve computed individual prediction values provided by an existing regression model without having to change either its parameters or its architecture. In other words, we are interested in achieving more accurate results by adjusting the calculated regression prediction values, without modifying or rebuilding the original regression model.

Our proposition is to adjust the regression prediction values using individual reliability estimates that indicate if a single regression prediction is likely to produce an error considered critical by the user of the regression.

The proposed method was tested in three sets of experiments using three different types of data. The first set of experiments worked with synthetically produced data, the second with cross sectional data from the public data source UCI Machine Learning Repository and the third with time series data from ISO-NE (Independent System Operator in New England).

The experiments with synthetic data were performed to verify how the method behaves in controlled situations. In this case, the outcomes of the experiments produced superior results with respect to predictions improvement for artificially produced cleaner datasets with progressive worsening with the addition of increased random elements.

The experiments with real data extracted from UCI and ISO-NE were done to investigate the applicability of the methodology in the real world. The proposed method was able to improve regression prediction values by about 95% of the experiments with real data.

Keywords: Improvement of Regression Predictions, Individual Reliability Estimates, Machine Learning, Pattern Recognition, Artificial Neural Networks, Neural Network Committee Machine, Imbalanced Datasets.

SUMMARY

LIST OF SYMBOLS AND ABREVIATIONS 3

LIST OF TABLES.....4

LIST OF FIGURES.....5

1. INTRODUCTION.....8

2. THEORETICAL REVIEW 11

2.1. Foundations.....11

2.2. Related Research.....20

3. METHODOLOGY – PROPOSITION AND DEVELOPMENT.....28

3.1. The Reliability Estimate: Critical Error Scenario Flag (CSFlag) 28

3.2. Regression Predictions Adjustment Procedure Using the CSFlag.....35

3.2.1. Critical Error Scenario Threshold Values Definition 38

4. EXPERIMENTS DESCRIPTION 40

4.1. Data Description 41

4.1.1. Synthetic Working Datasets 41

4.1.2. UCI Working Datasets.....42

4.1.3. ISO-NE Working Datasets.....45

4.2. Regressions modeling 46

4.2.1. Regression Models for the Experiments with Synthetic Data.....47

4.2.2. Regressions Models for the Experiments with Real Data 47

4.3. Critical Error Scenario Threshold Values Definition.....49

4.4. Critical Error Scenario Alert Function Design 50

4.5. Regression Predictions Adjustment Procedure Using the CSFlag.....55

5. NUMERICAL RESULTS – ANALYSIS AND DISCUSSION 58

5.1. Outcomes with the Synthetic Working Datasets 59

5.1.1. Critical Error Scenario Threshold Values Definition 60

5.1.2. Critical Error Scenario Alert Functions' Outcomes 60

5.1.3. Regression Prediction Values Improvement Using the CSFlag 62

5.2. Outcomes with the UCI Working Datasets.....70

5.2.1. Critical Error Scenario Threshold Values Definition 71

5.2.2. Critical Error Scenario Alert Functions' Outcomes 71

5.2.3. Regression Prediction Values Improvement Using the CSFlag 73

5.2.4. Linear and Nonlinear Regression Models: Outcomes Comparison.79

5.3. Outcomes with the ISO-NE Working Datasets.....82

5.3.1. Critical Error Scenario Threshold Values Definition 83

5.3.2. Critical Error Scenario Alert Functions' Outcomes 83

5.3.3. Regression Prediction Values Improvement Using the CSFlag 85

5.3.4. Linear and Nonlinear Regression Models: Outcomes Comparison.88

6. CONCLUSION 90

APPENDIX A – Supplementary Figures.....95

APPENDIX B – Supplementary Tables.....97

REFERENCE LIST 110

LIST OF SYMBOLS AND ABREVIATIONS

α	Probability
δ	Critical Error Scenario Threshold Value
APE	Absolute Percentage Error
ANN	Artificial Neural Networks
CM	Artificial Neural Networks Committee Machines
CSAFunction	Critical Error Scenario Alert Function
CSFlag	Critical Error Scenario Flag
CScenario	Critical Error Scenario
ISO-NE	Independent System Operator in New England
MSE	Mean Squared Error
OLS	Ordinary Least Squared
RMSE	Root Mean Squared Error
UCI	UCI Machine Learning Repository

LIST OF TABLES

Table 1: Adjustment procedure rules. 36

Table 2: (Synthetic) Working dataset sizes. 42

Table 3: (UCI) Original databases descriptions. 43

Table 4: (UCI) Working datasets sizes. 44

Table 5: (ISO-NE) Working database variables. 45

Table 6: (ISO-NE) Working datasets sizes and time frame division. 46

Table 7: (Synthetic) Hypothesis tests outcomes. 63

Table 8: (UCI) RMSE % of improvement. 73

Table 9: (UCI) Linear regression models – Hypothesis tests outcomes 74

Table 10: (UCI) Comparison of the RMSE values generated by the experiments. 80

Table 11: (UCI) – Number of positive RMSE % improvement per group. 82

Table 12: (ISO-NE) RMSE % of improvement. 85

LIST OF FIGURES

Figure 1: **CSAFunction**'s construction process. 30

Figure 2: Observations in the training dataset. 31

Figure 3: Predictions of regression G(X) for the training dataset. 31

Figure 4: **CScenario** definition based on the prediction errors in the training dataset. 32

Figure 5: ***CSFlag*** construction for the training dataset. 32

Figure 6: Predictions of regression G(X) and **CSAFunction** for the testing dataset. 33

Figure 7: **CSAFunction** predictions compared to the new observed values in the testing dataset, assuming the perfect detection of all the critical error scenario cases. 33

Figure 8: Effect of the adjustment procedure on the original predictions for the testing dataset. 37

Figure 9: Original and adjusted predictions compared to the new observed values in the testing dataset. 37

Figure 10: Example of empirical cumulative distribution of the prediction regression error values. 38

Figure 11: Linear regression applied to data produced by a quadratic function. 47

Figure 12: **Cscenario$_{Neg}$** objective function pseudo-code. 49

Figure 13: Summary scheme of the proposed methodology. 55

Figure 14: (Synthetic) RMSE values of the Regression datasets generated by different noise factor values (level of noise). 59

Figure 15: (Synthetic) F-Measure X Noise Factor (Training datasets). 61

Figure 16: (Synthetic) F-Measure X Noise Factor (Testing datasets).....61

Figure 17: (Synthetic) RMSE ($_{Original, Adjusted}$) X Noise Factor (Testing datasets). 62

Figure 18: (Synthetic) RMSE % of improvement X Noise Factor (Testing datasets). 64

Figure 19: (Synthetic) **CSAFunctions'** outcomes for almost clean data. 65

Figure 20: (Synthetic) Adjustment effect on the prediction errors for almost clean data. 66

Figure 21: (Synthetic) **CSAFunctions'** outcomes for highly noisy data. 66

Figure 22: (Synthetic) Adjustment effect on the prediction errors for highly noisy data 67

Figure 23: (Synthetic) Adjustment effect on the regression predictions for almost clean data, with the index of X in ascending order in X axis. 68

Figure 24: (Synthetic) Adjustment effect on the regression predictions for almost clean data. 68

Figure 25: (Synthetic) Adjustment effect on the regression predictions for highly noisy data, with the index of X in ascending order in X axis. 69

Figure 26: (Synthetic) Adjustment effect on the regression predictions for highly noisy data. 69

Figure 27: (UCI) **CSAFunctions'** outcomes.....75

Figure 28: (UCI) Adjustment effect on the prediction errors. 76

Figure 29: (UCI) Adjustment effect on the regression predictions (**CSAFunction$_{Pos}$**) 77

Figure 30: (UCI) Adjustment effect on the regression predictions (**CSAFunction$_{Neg}$**) 77

Figure 31: (UCI) Yacht experiment: Regression prediction error distributions. 78

Figure 32: (ISO-NE) **CSAFunctions'** outcomes. 86

Figure 33: (ISO-NE) Adjustment effect on the prediction errors. 86

Figure 34: (ISO-NE) Adjustment effect on the regression predictions of Obs. # 100 ~ #108. 87

Figure 35: (ISO-NE) Adjustment effect on the regression predictions of Obs. # 148 ~ #155. 87

1. INTRODUCTION

Data is available everywhere, at any time. Not only is data available, but also regression models underlying data analysis tools that help in transforming raw data into useful information. Weather predictors, stock market forecasters, house rental pricing estimators, and even simple medical tests are examples of applications that can currently be easily accessed via websites or smart-phones.

Moreover, these tools are available to anyone. That means everybody potentially has the same set of information, therefore, according to the Perfect Market Theory from Economics (FAMA, 1970), to gain competitive advantage it would be necessary to beat these publicly obtainable predictions.

This research describes the proposal and the development of a methodology to improve computed individual prediction values provided by an existing regression without having to change either its architecture or its parameters. In other words, we are interested in achieving more accurate results by adjusting the computed regression prediction values instead of by constructing a different regression model. As a result, it can be helpful to improve the prediction of a specific observation provided by an existing benchmark regression model or predictor system.

We propose a method to adjust regression predictions using a specific type of point reliability estimates for individual regression predictions also developed in our research, named Critical Error Scenario Flag.

These original ideas were first presented in a previous technical paper (MATSUMOTO; DEL-MORAL-HERNANDEZ, 2014), in which we propose a methodology to create point reliability estimates for individual regression predictions based on critical error scenario definitions determined by the user of the regression.

The definition of a critical error scenario is used to build a reliability estimates as a binary variable able to indicate whether the regression prediction error of an individual observation is likely to be considered critical, according to this predefined

error condition. Then, the construction of the proposed reliability estimate maybe structured as a pattern recognition problem that can be solved using Machine Learning techniques.

Our presumption is that, these reliability estimates are potentially able to correctly identify the critical error scenario cases, and afterwards they can be used to formulate adjustment procedures to improve individual regression predictions as proposed by our study.

To investigate this hypothesis, we carried out experiments using three types of datasets: artificially produced synthetic cross-sectional data, real cross-sectional data extracted from the public data repository UCI Machine Learning Repository, and real time series data extracted from ISO-NE, the Independent System Operator in New England.

The synthetic experiments were performed to verify the response of the proposed method in the case of artificially created clean data, and the effect of the addition of random elements with different amplitudes to the clean data. It was possible to verify that the methodology was able to produce superior results in the case of clean data, while the performance went down as the amplitude of the random elements went up.

The applicability of the proposed methodology in the real world was verified with two sets of real data: eight cross sectional databases from UCI Data Repository and eight time series databases from ISO-NE. Additionally, for real data, the method was used to adjust the prediction values produced by two types of regression models: Linear regression model and Artificial Neural Networks regression model.

As expected, the outcomes of the experiments with real data were similar to the ones obtained with synthetic data with addition of random elements with moderate amplitudes. Although the reliability estimates were capable of correctly identifying fairly small number of critical cases; nonetheless, the adjustment procedure was able to reduce the root squared mean error values of the regression predictions by about 95% of 1600 experiments performed using data from UCI, and in all 16 experiments performed using data from ISO-NE.

These experiments outcomes evidenced that the proposed methodology is able to correctly adjust and to improve computed individual regression predictions of unknown observations.

The rest of this doctoral thesis is composed of five other chapters, two appendices, and a list of references. Chapter 2 contains a theoretical review of the foundations that support our research, and the relevant works currently under development related to the improvement of computed individual regression predictions. The proposition and the development of the methodology are detailed in Chapter 3. In Chapter 4, we describe the working datasets and the procedures employed in the experiments conducted to explore and to observe the behavior of the proposed methodology. The numerical results are analyzed and discussed in Chapter 5. Conclusions are presented in Chapter 6. The appendices contain graphs and tables with supplementary results of the experiments, and are followed by a list of references.

2. THEORETICAL REVIEW

Before we start the discussion about the improvement of computed individual regression predictions, it is worth explaining what regression means in the context of this research. For this reason, this chapter begins with a brief presentation of how the definition of regression is handled by two basic disciplines: Statistics and Machine Learning.

Statistics and Machine Learning are not the only research fields that deal with regressions, they do overlap, and sometimes conflict (BREIMAN, 2001); but we believe that an acquaintance with the specific assumptions and peculiarities of these two approaches is relevant to building the foundations of our work.

To fulfill this theoretical review, we describe what has been recently discussed by the research community specifically about methods to improve computed individual regression predictions.

2.1. Foundations

Regression is usually defined as a process to estimate the relationship between a dependent variable, Y, also called output or response variable; and one or more independent variables, X, or inputs or explanatory variables.

Historically, the first proposition to solve this problem for cross sectional data was the classic linear regression model which assumes that Y is given by a linear combination of the independent variables in X, as defined in Equation (2.1), where β is a vector composed of the coefficients of the regression, and ε is a vector composed of the errors of the regression (WOOLDRIDGE, 2009a).

$$Y = \beta' * X + \varepsilon \quad (2.1)$$
$$where\ \beta'\ is\ the\ transpose\ of\ vector\ \beta$$

This formulation is constructed based on the probabilistic interpretation which states that the estimated value of Y, $\hat{Y}$, is given by the conditional expectation value of Y given X, as defined in Equation (2.2).

$$\hat{Y} = E(Y \mid X) = \beta' * X \qquad (2.2)$$
$$\varepsilon = Y - \hat{Y}$$

The numerical method proposed to solve Equation (2.2), i.e., to estimate the vector β ($\hat{\beta}$), was the Ordinary Least Squared (OLS) method that consists in solving an optimization problem defined as follows: find the values of the elements of β that minimize the sum of the squared errors of the regression, as defined in Equation (2.3).

$$\hat{\beta} = \arg\min \varepsilon^2 = (Y - \beta' * X)^2 \qquad (2.3)$$

For the OLS method applied to solve linear regression problems, there is a theorem named the Gauss-Markov Theorem, which states that the estimators $\hat{\beta}$ given by the OLS method are the best linear unbiased estimators of β, under the following four assumptions (WOOLDRIDGE, 2009a):

(I) The observations are random samples from the same population;

(II) Given X, the conditional expected value of the errors of the regression is zero ($E(\varepsilon \mid X) = 0$);

(III) There are no exact linear relationships among the independent variables in X;

(IV) Given X, the conditional variance of the errors of the regression is constant ($Var(\varepsilon \mid X) = \sigma^2$). In Statistics, this condition is called homoscedasticity, or homoskedasticity, and its opposite state is called heterocedasticity, or heteroscedasticity (no constant variance).

In addition, if the regression error values are normally distributed ($\varepsilon \sim N(0, \sigma^2)$) and independent of the input variables in X; the linear regression method also provides statistical measures, such as P-value, T-statistic, Z-statistic, and F-statistic, which can be used for statistical inference purposes.

Under the five assumptions listed above (the four assumptions of the Gauss-Markov Theorem, and the condition of normally distributed regression errors), the values of the statistics P-value, T-statistic, Z-statistic can be used to verify if a coefficient estimator, $\widehat{\beta_i}$, is statistically valid or not. This information can be used, for instance, to support the features selection process.

The value of the F-statistic is one of the measures provided by Statistics that can be used to relatively compare two different linear models that estimate the same output variable using different sets of input variables from a same set of data. The F-statistic takes into account not only the quality of regression estimation (smaller prediction errors), but also penalizes the increase of the number of input variables, considering that it can reduce the generalization capability of the model, according to the *Occam's Razor* principle, which states that "simpler models should be preferred until the data justifies more complex models" (BALASUBRAMANIAN, 1996).

Additionally, the specification of the linear regression model provides an easy and straightforward interpretation of the causal effect relationship among the dependent variable and the independent variables individually, under the condition known as *Ceteris paribus* ("other factors being equal"), where the coefficient estimator $\widehat{\beta_i}$ informs the linear proportion between the dependent variable Y and the independent variable X_i.

The existence of these characteristics and properties may be an explanation as to why the linear regression model is still one of the most popular regression models currently in use, even after more than two hundred years since its introduction by Carl F. Gauss and Adrien-Marie Legendre, almost simultaneously, around the beginning of the XIX century (CAETANO, 2013).

Regarding numeric computation, the algorithm of the OLS method is easy to implement and it is comparatively faster than other methods. For this reason, the linear regression formulation solved by the OLS method is frequently adopted as a reference benchmark, even when the data does not fully comply with the assumptions of the Gauss-Markov Theorem.

Apparently, the only drawback of the linear regression model is the limitation imposed by its formulation, as it is not able to capture nonlinear relationships among

the input variables. In this regard, aiming at improving the estimation accuracy of the regression models, i.e., reducing the regression prediction errors, other classes of functions, such as, exponential, polynomial, and trigonometric, have been exploited to define different regression model types, generically named nonlinear regression models.

Considering the same set of data, a nonlinear regression model is potentially able to produce smaller prediction errors than a linear regression model, and primarily any approximation function created using curve fitting techniques could be adopted as a formulation for nonlinear regression modeling (HASTIE; TIBSHIRANI; FRIEDMAN, 2011a).

In the case of nonlinear regressions, however, there are virtually infinite nonlinear formulations with countless corresponding methods to calculate the parameters of each of these models; and there is not a theorem similar to the Gauss-Markov to indicate which nonlinear formulation and method for parameters calibration would be the “best” or most recommended. Furthermore, there is no general statistical numeric measurement analogous to P-value or F-statistic to indicate whether a regression parameter estimator is statistically valid, or if one model is comparatively better than another.

Fortunately, to assist the verification of the goodness of fit of regression models in general, Statistics provides regression errors analysis techniques that can be applied to linear and nonlinear models without any assumption about the specificities of the formulation of the models (PEARSON, 2011).

The fundamental presumption of regression errors analysis techniques is that the relationship among the output and the input variables has a deterministic part that can be modeled, and a random nondeterministic part, which cannot be modeled, commonly called noise, uncertainty, or shock.

In theory, the perfect regression model would be able to completely filter the deterministic part of the relationship among the variables, and the regression error, defined as, $\varepsilon = Y - \hat{Y}$, would be the remaining nondeterministic or random part not filtered by the model. For this reason, regression errors are also called regression residuals, and its study is known as residual analysis.

Following this principle, theoretically, the closer the regression errors distribution is to the normal distribution ($\varepsilon \sim N(0, \sigma^2)$), the better the regression model is. Consequently, in regression and residual analysis, the first condition to be checked is if the expected value of the error of the regression prediction conditioned to the input variables is equal to zero (it is the second assumption of the Gauss-Markov Theorem). This means that the regression, on average, is precise enough to represent the relationship between X and Y, otherwise the regression prediction is considered biased. It is important to emphasize that this does not mean error equal to zero or as small as possible.

Another relevant aspect is the shape of the regression prediction errors distribution. Comparatively, a more symmetric distribution indicates less biased regression estimations. In the same way, a more asymmetric and spread regression errors distribution indicates less accurate regression estimations.

In the case of time series data, it is also necessary to verify the correlation among consecutive error values, the auto-correlation of the errors. The existence of high auto-correlation among errors indicates that there is still deterministic information in the past observations correlated to the observations in the future, and this information was not filtered by the model; hence a comparatively less auto-correlated errors sequence would be an indication of a better regression model (ENDERS, 2014a).

Moreover, defining a regression model is not equivalent to solving a data fitting interpolation problem. Regression models are mostly built for estimating and forecasting purposes. Therefore, it is essential to be able to predict the model's generalization capability, in other words, how the model will perform with data outside the known data sample.

The expressions known, seen, in-sample, or training dataset are frequently used to refer to the data sample used to construct the regression model. The data sample used to verify the generalization capability of a regression model is usually called the unknown, unseen, future, out-of-sample, verification or testing dataset.

The construction of a function that perfectly fits a training dataset $\{X, Y\}$ is almost a trivial task; however, the practical usefulness of such model might be virtually zero.

It is like defining a polynomial with $(n-1)$ degree to interpolate n points; it exactly fits all points but has no generalization capacity, because the model is too complex and it is fitting the data instead of capturing the relationship between the input and output variables. The problem of the balance between model complexity and generalization capability is known as overfitting.

One of the most popular tools developed in Statistics, usually adopted to address the overfitting problem, is a model-independent technique named cross-validation (DUDA; HART; STORK, 2001). The first technical papers about cross-validation date back to the 1960s, and the classic version of the cross-validation procedure is performed splitting the known data into two datasets, one to be used to define the model, and to calibrate its parameters; and the other to be set apart to infer about the generalization capability of the model. These datasets are commonly named training and verifying datasets respectively. The division can be done using any proportion; however, the literature recommends setting the training dataset larger than the verifying one, and 80% and 20% are usually suggested proportions (HASTIE; TIBSHIRANI; FRIEDMAN, 2011b).

The operational procedure of the cross-validation technique is to interactively define the regression model structure starting from simpler definitions, and increasing the complexity of the model while monitoring its generalization capability using the verifying dataset that is not involved in the modeling process. The basic concept is using the verifying dataset to simulate the behavior of the model for unknown data; hence, no matter if the regression errors measurement in the training dataset decreases, the interaction stops when the regression errors measurement in the verifying dataset rises. One error measurement traditionally adopted in cross-validation is the root mean squared error (RMSE).

This initial version of cross-validation technique has several variants, mostly related to the way the original data is re-sampled to generate the training and the verifying datasets. The techniques that use random sampling with replacement are generically called Bootstrapping techniques.

The origin of Bootstrapping is the statistical data re-sampling technique called jackknife developed in the mid-1950s to improve average and variance estimation

(DUDA; HART; STORK, 2001). The essential idea of the jackknife method is that, given a data sample D with size N, the arithmetic mean of the arithmetic means of P random samples, S_p, of D, with size $(N-1)$, $\overline{\theta_{Jack}}$, defined as shown in Equation (2.4), is a better estimator of the arithmetic mean of the population than the pure D arithmetic mean, $\bar{\theta}$, defined as shown in Equation (2.5). The jackknife method also provides estimation for the variance of the arithmetic mean estimation.

$$\overline{\theta_{Jack}} = \frac{1}{P}\sum_{p=1}^{P}\overline{\theta_p} \quad ; \; where \; \overline{\theta_p} = \frac{1}{(N-1)}\sum_{i=1}^{N-1} s_i \; ; \; s_i \in S_p \; sample \; of \; D \tag{2.4}$$

$$\bar{\theta} = \frac{1}{N}\sum_{i=1}^{N} d_i \; ; \; d_i \in D \tag{2.5}$$

With the computational performance increase, it was possible to extend the jackknife principle from arithmetic mean estimations to any statistic of interest, and to more complex inferences; and the term “Bootstrap” referring to the collection of these techniques was first introduced by B. Efron, in 1979 (EFRON, 1979).

The theoretical foundation of the cross-validation and the bootstrap techniques applicability is the Central Limit Theorem provided by Statistics. In general terms, this theorem states that, in the case of a population with identically independent distributed random individuals, the arithmetic means of random samples of this population is asymptotically normally distributed, regardless of the original distribution of the population (JAMES, 2009). The Central Limit Theorem has several versions; the first one was postulated by the French mathematician Abraham de Moivre in 1733.

The Central Limit Theorem also supports the statistical hypothesis tests that are frequently used in empirical experiments to verify if the outcomes of two different models are statistically significantly different or not (MAGALHÃES; LIMA, 2004).

Bootstrapping is just one example of several methods that have benefitted from the combination of the Statistics theory and the evolution of computer processing power. Techniques such as the Monte Carlo method (ROBERT; CASELLA, 2010) which is based on the Bayesian Statistics theory and on computer simulations of asymptotic behaviors applied to solve problems of numerical integration, probability

distribution, and scenario forecast among others, became viable thanks to the technological development of the computing power.

Since mid-1980, this merging of Statistics with Computer Science provided a remarkable increase in the numbers of methods built based on computer algorithms developed to extract information from empirical data without any assumption or specification about the data distribution or model specificity, and ultimately allowed the booming of a research field that emerged by late 1960 (SAMUEL, 1959) named Machine Learning.

A primary concern of Machine Learning is extracting useful information exclusively from empirical data when no additional assumption is required, or data specification is available. In Machine Learning, this is known as the Learning Problem (ABU-MOSTAFA; MAGDON-ISMAIL; LIN, 2012a). If we expect a model to find a relationship among any of the variables, this learning process is called Unsupervised Learning. On the other hand, if we want the model to discover an input and output relationship, this is called Supervised Learning.

According to this approach, a regression problem can be defined as a Supervised Learning problem in which the output is a continuous variable. If the output variable can only assume a finite set of values, or labels, it can be handled as a classification problem or pattern recognition problem, in the case of just two labels (MARSLAND, 2014).

In Statistical and also in Supervised Learning methods, the known or training dataset is used to calibrate the parameters of the regression model that can belong to any class of mathematical or computational algorithms; and the choices of the most appropriate model formulation and the most suitable parameter configuration are made using cross-validation techniques.

In both cases, the goal is to overcome the overfitting problem making sure that regression prediction error for the training dataset, E_{Tra}, is small enough ($|E_{Tra}| < \alpha$), and it is close enough to the regression prediction error for the testing dataset,E_{Tes} ($|E_{Tra} - E_{Tes}| < \delta$).

The Vapnik–Chervonenkis (VC) Theory, developed after the 1960s, provides a theoretical bound for the difference between the E_{Tra} and the E_{Tes}, the VC Generalization Bound, defined in Equation (2.6).

$$E_{Tes} \leq E_{Tra} + \sqrt{\frac{8}{N} ln \frac{4m_{\mathcal{H}}(2N)}{\delta}} \quad (2.6)$$

In this equation, N is the number of observations in the training dataset, δ is any fixed tolerance between $[0,1]$, and $m_{\mathcal{H}}(2N)$ is a value given by the Growth Function, a function defined by the VC Theory which numerically expresses the complexity of a predictive model formulation. The predictive model can be a regression or a classification model (ABU-MOSTAFA; MAGDON-ISMAIL; LIN, 2012b).

In the Supervised Learning theory, a class of predictive model formulations receives the name of Hypothesis set ($\mathcal{H}$); linear functions, polynomials functions, regression trees, artificial neural networks are examples of Hypothesis sets. The collection of methods and strategies to calibrate the parameters of a specific predictive model formulation is called the Learning Algorithm. For instance, OLS is a Learning Algorithm used to calibrate the parameters of the Hypothesis set composed of models with linear formulations.

According to Equation (2.6), the difference between the error measurements of the training and the testing datasets (E_{Tra} and E_{Tes}) decreases when the number of observation in the training dataset increases; and this difference increases if the complexity of the model ($m_{\mathcal{H}}(2N)$) increases.

In practice, the VC Generalization bound formalizes the empirically observed fact that, for models with more complex formulations, the difference between the E_{Tra} and the E_{Tes} values is usually higher than the difference observed in the outcomes of models with simpler formulations. The VC Generalization Bound definition is so important that the Machine Learning community considers the VC Theory one of the most significant contributions of Mathematics to the Computational Learning Theory (ABU-MOSTAFA; MAGDON-ISMAIL; LIN, 2012b).

The concepts and the techniques described in this section are the fundamental components that support the theoretical framework of our research, and the empirical

experiments performed to validate its applicability. Next, we present a brief description of the research specifically related to the improvement of computed individual regression predictions.

2.2. Related Research

Compared to the extensive literature available with respect to regression modeling, there is relatively little research about methods to improve computed regression prediction values, and these works are mostly related to the field of reliability estimation of individual regression predictions (BOSNIC; KONONENKO, 2011).

In regression analysis, average error measures, for instance, mean squared error or mean absolute error values are frequently adopted as general purpose prediction evaluation metrics (HAHN; SHAPIRO, 1994). As afore mentioned, in Statistics and Machine Learning (DUDA; HART; STORK, 2001), these average metrics are particularly suitable for use in cross-validation techniques to compare the outcome of models throughout processes such as model formulation choice, features selection, and model parameters calibration. However, these averaged error metrics are less appropriate for estimating the quality of regression predictions for single unseen observations.

The availability of additional information about a specific regression prediction is very desirable in decision-making processes mainly in risk-sensitive areas. For this reason, research in the field of reliability estimation of individual predictions has increased in the last few decades, and the methods addressing this matter are usually divided into two groups in the technical literature (PEVEC; BOSNIC; KONONENKO, 2012).

The first group consists of the methodologies that work with model-specific approaches. In this case, the concepts are based on the mathematical definition of the regression model, and the probabilistic properties of the data; and often analytical solutions are provided. The second group covers model-independent methods that essentially handle the regression model as a “black-box” object, considering just its

inputs and outputs. As a result, these methods can be more widely applied but rarely provide analytical solutions.

The classic linear regression confidence interval method can be considered one of the most conventional examples of model-specific approach. This method assumes that, given the same five assumptions about the data distribution listed in Section 2.1., the prediction errors produced by the linear model closely follow a known normal distribution, and this information can be used to construct a constant interval in which the regression prediction errors for unseen observations are supposed to fall within a certain probability, named confidence degree (MCCULLAGH; NELDER, 1989).

As another example of the model-specific approach, we could mention the Generalized Autoregressive Conditional Heteroscedasticity (GARCH) regression model formulations applied in time series analysis. Unlike the linear regression inference analysis, in the case of GARCH models, constant regression prediction errors variance (homoscedasticity) is not an assumption; on the contrary, these kinds of models estimate the individual variance of the prediction errors of the time series regression model as a function of the variables of the original regression, and the error analysis is handled as a linear autoregressive regression problem (ENDERS, 2014b).

In the case of nonlinear regressions, there are studies developed considering specific properties of predictive model architectures such as Regression Trees (MEINSHAUSEN, 2006), Support Vector Machines (SAUNDERS; GAMMERMAN; VOVK, 1999), and Neural Networks (CARNEY; CUNNINGHAM; BHAGWAN, 1999), among others; and a relatively smaller amount of individual prediction reliability estimation methods belong to the model-independent group, which is likewise usually divided into two groups: prediction interval estimate methods and point estimate methods (PEVEC; BOSNIC; KONONENKO, 2012).

Similarly to the linear regression confidence interval, the prediction interval estimate methods are concerned with estimating an interval in which the regression prediction error for unknown or future observations are supposed to fall, within a

certain probability, but without being necessarily attached to a parametric distribution; as a result, the two methods outcomes have different conceptual interpretations.

In the case of a confidence interval with, for instance, 90% degree of confidence, it means that, considering all regression prediction errors, asymptotically, 90% of them are supposed to fall inside the interval (WOOLDRIDGE, 2009b). On the other hand, the prediction interval methods estimate an interval with a certain probability of containing the regression prediction errors. Prediction intervals are frequently wider than confidence intervals and they can vary for each of the observations (HESKES, 1997).

In particular, with reference to the construction of prediction intervals for nonlinear regression models, there are studies exploring different approaches, such as, empirical distributions (JØRGENSEN; SJØBERG, 2003), bootstrap techniques (POLIKAR, 2007), maximum likelihood properties (PAPADOPOULOS; EDWARDS; MURRAY, 2000), Bayesian inference theory (HINSBERGEN; LINT; ZUYLEN, 2009), and other distinct frameworks, for example, Lower Upper Bound Estimation (KHOSRAVI et al., 2011), Conformal Prediction (PAPADOPOULOS; HARALAMBOUS, 2011), and non-parametric approaches using supervised learning framework (PEVEC; KONONENKO, 2014)(PEVEC; KONONENKO, 2015), among others.

Another class of reliability estimates for individual regression predictions is the point estimates class that provides a value, instead of an interval, as additional information about individual prediction reliability to help the user of the regression gain auxiliary insight about the individual future predictions.

Most point estimate methods described in the literature are based on estimates generated by the analysis of how modifications in the data affect the outcomes of the models (PEVEC; BOSNIC; KONONENKO, 2012). The previously cited bootstrap and its variants, such as bagging and boosting, are popular examples of this kind of methodology (DRUCKER, 1997)(POLIKAR, 2007).

The basic principle is to repeatedly modify the initial set of the known data (the training or learning dataset) applying bootstrap techniques, and then creating one regression for each of these variations. This collection of regression models is then

combined so as to produce a more accurate model, and also to provide the reliability estimates for the individual regression predictions, usually the average and the standard deviation of the outcomes of the collection of regression models, respectively. Subsequently, these estimates can be used to correct the original regression predictions.

The strategy defined by Wolpert (WOLPERT, 1992) is one example of this kind of approach that uses the combination of variations of the original regression model generated by different samples from a same initial dataset. In his research, Wolpert introduces the stacked generalization concept and defines a strategy on how this concept can be used to estimate and correct regression predictions. According to the author, “when used with multiple generalizers, stacked generalization can be seen as a more sophisticated version of cross-validation”.

Local sensitivity analysis reliability estimates (BOSNIC; KONONENKO, 2007), local cross-validation estimates (ATKESON; MOORE; SCHAAL, 1997), density-based reliability estimates (WAND; JONES, 1995), variance of bagged models (CARNEY; CUNNINGHAM; BHAGWAN, 1999), and local modeling of prediction errors (WELLEN; DANKS, 2012) are another examples of known methods that have been explored and compared in studies such as the ones developed by Bosnic and Kononenko (BOSNIC; KONONENKO, 2008), and by Rodrigues et al. (RODRIGUES et al., 2012).

In Bosnic and Kononenko (2008), the five above mentioned methods were compared using data from 28 different domains, and applied to eight different regression models: regression trees, linear regression, neural networks, bagging, support vector machines, locally weighted regression, random forest, and generalized additive model. The performances of these five different reliability estimates were compared by computing the Pearson correlation between each reliability estimates and the regression prediction errors.

The authors also explored the combined use of two reliability estimates to improve efficiency, and according to their conclusion, the best average performance was achieved by the estimate produced by the variance of bagged models for the regression predictions provided by neural network, bagging, and locally weighted

regression models. One more relevant conclusion by Bosnic and Kononenko (2008) was that the performance of the reliability estimates depends on the dataset domain and the regression model class; this information could be used to automatically select the most appropriate reliability estimate for a given problem.

Another proposal for individual reliability estimate construction is the meta-model approach. The central idea is to use a secondary model to evaluate the main model. In the study developed by Fink et al. (FINK; ZIO; WEIDMANN, 2014), for instance, the authors propose quantifying the reliability of a classifier using two classifiers: the first is the main classifier, which performs the classification; the second is a meta-classifier built to estimate the reliability of the individual outcomes of the main classifier using the concept of randomness and typicalness of pattern based on nearest-neighbors proposed by Vovk et al. (VOVK; GAMMERMAN; SAUNDERS, 1999). In this specific paper, the method was applied to solve a problem of fault diagnosis in railway turnout systems.

The improvement of regression predictions is one of the application fields of the reliability estimates research. For instance, in a more recent article by Bosnic and Kononenko (BOSNIC; KONONENKO, 2010), these authors extended the methodology introduced in their previous work (BOSNIC; KONONENKO, 2007) to define a procedure to improve regression predictions based on local sensitivity analysis and Meta-Model approaches.

In Bosnic and Kononenko (2007, 2010), to measure how much a variation in the input data can influence the output of a system, similar to what is typically done in the case of bootstrapping and other sensitivity analysis techniques, the authors repeatedly modified the initial set of the known data adding new training examples, and creating one regression for each of these variations.

They named the regression model defined by the original training dataset as initial or primary regression model, $f_M(x)$, and its predictions as initial predictions, $f_M(x) = K$. The regression models produced by the modified training datasets received the name of sensitivity regression models, $f_{M'}(x)$, and their predictions were called sensitivity predictions, $f_{M'}(x) = K_\varepsilon$.

Given a known observation from the training dataset, (x, y), the new additional training data was created by adding an arbitrary value to the initial prediction of y, $f_M(x) = K$. In Bosnic and Kononenko's experiments, this arbitrary value was configured as a proportion of the difference between the maximum (b) and minimum (a) values of the output variable y in the training dataset, as shown in Equation (2.7).

$$New\ training\ data\ created\ given\ (x,y):\ (x, K+\delta);\ where\ K = f_M(x) \qquad (2.7)$$
$$\delta = \varepsilon * (b-a);\ \ where\ y \in [a,b]$$

One value of ε was used to create two different sensitivity regression models, one with the addition of a new data defined as $\big(x, K + \varepsilon * (b-a)\big)$; and the other with the addition of other new data defined as $\big(x, K - \varepsilon * (b-a)\big)$. These two sensitivity regression models were used to obtain two new sensitivity predictions of x: K_ε and $K_{-\varepsilon}$.

According to the local-sensitivity analysis theory described in Bosnic and Kononenkos's study, if the initial regression prediction, K, is reliable we could expect to have the two sensitivity predictions K_ε and $K_{-\varepsilon}$ symmetric positioned around the original value K; hence, ideally, we would have the equality expressed in Equation (2.8) satisfied.

$$(K_\varepsilon - K) = (K_{-\varepsilon} - K) \qquad (2.8)$$

In their research, Bosnic and Kononenko described a procedure that starts by defining a set of non-negative values of $\varepsilon \in E,\ E = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N\}$, and by creating two sensitivity regression modes for each of $\varepsilon \in E$ with the addition of the new observations to the training data, using Equation (2.7).

In the sequence, the collection of the K_ε values produced by the sensitivity regression models was used to construct two point reliability estimates for individual regression predictions based on local-sensitivity analysis, SAvar (Sensitivity Analysis – variance) and SAbias (Sensitivity Analysis – bias), as shown in Equations (2.9) and (2.10).

$$\mathrm{SAvar} = \frac{\sum_{\varepsilon \in E} (K_\varepsilon - K_{-\varepsilon})}{N} \qquad (2.9)$$

$$\text{SAbias} = \frac{\sum_{\varepsilon \in E} (K_{\varepsilon} - K) + (K_{-\varepsilon} - K)}{2N} \qquad (2.10)$$

$$E = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N\}$$

The reliability estimate SAvar represents the wideness of the interval of the sensitivity predictions; therefore, it can be interpreted as a measurement of local variance. The formulation of the reliability estimate SAbias informs the average symmetry of the sensitivity predictions (K_{ε} and $K_{-\varepsilon}$) around the initial prediction, K.

To correct the regression predictions, in their more recent paper (BOSNIC; KONONENKO, 2010), the authors proposed the application of the SAvar and SAbias construction procedure to create a collection of differences produced by the different values of ε: of $(K_{\varepsilon} - K)$ and $(K_{-\varepsilon} - K)$. This collection of differences was then used to construct a secondary model, which they named secondary learner or predictor, to estimate the original regression prediction errors. Hence, these estimations could be used to adjust and to correct the original regression predictions. The efficiency of the methodology was evaluated by comparing the RMSE values before and after the correction. The experiments indicated significant increase of the prediction accuracy in only 20% of the cases, but no decrease was detected.

Bosnic and Kononenko dedicated special attention to the process of features selection of the secondary learner. Initially, they tested the system using all the thirty one ($31 = (2^5 - 1)$) possible combinations of $E = \{0.01, 0.1, 0.5, 1.0, 2.0\}$ to compose the set of input parameters using the differences $(K_{\varepsilon} - K)$ and $(K_{-\varepsilon} - K)$. The authors named this process: “exhaustive search of optimal set of attributes”.

Alternatively, to automatically select the input parameters of the secondary learner, they proposed the use of Relief algorithms (ROBNIK-ŠIKONJA; KONONENKO, 2003) which are able to detect conditional dependencies between attributes. In the final conclusions, the authors remarked that their methodology did not produce good results for linear regression prediction, and recommended the use of the method with more complex regression algorithms.

In Rodrigues et al. (RODRIGUES et al., 2012), the afore mentioned strategy was expanded and adapted to handle time-series streaming data, when the data can be read only once. The experiments were performed to improve predictions of electricity

loads; the predictions corrected by this method were compared to the predictions corrected using Kalman filters, and according to the experiments, the proposed method outperformed the Kalman filters improvement.

More recently, in an another paper (BOSNIC et al., 2014), the same group of authors released the outcomes of a new research project involving prediction accuracy improvement and prediction explanation, once again using the local-sensitivity analysis framework with secondary learner.

In our research, we define a method to improve individual regression predictions that can be widely applied, even when local-sensitivity analysis, local cross-validation, variance of bagged models, and prediction errors local modeling techniques cannot be performed.

This happens when no information about the architecture of the initial regression model is available; hence this initial model cannot be retrained with modified training datasets; or when the addition of new observations in the training datasets is not possible or not recommended.

Our proposed methodology for improving computed individual regression predictions is a model-independent method, based on a meta-model approach; it does not depend on the information about the architecture and characteristics of the regression model in question, we just need the inputs and the outputs values of the predictor system. The concepts and the steps of the implementation of our method are fully described in the next chapters.

3. METHODOLOGY – PROPOSITION AND DEVELOPMENT

As initially stated in Chapter 1, this work proposes a methodology for improving computed individual regression predictions using a new type of reliability estimate; the definition of this specific reliability estimate is also a proposal of our research, and its initial concept was inspired by two famous quotes transcribed below extracted from the book “Empirical Model-Building and Response Surfaces” written by the statisticians George Box and Norman Draper (BOX; DRAPER, 1987):

"Essentially, all models are wrong, but some are useful."

“Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful. “

Given that, one strategy to cope with this condition would be try to identify in which cases or for which observations the critical scenarios of “too wrong” regression prediction values are likely to happen, and use this knowledge to adjust them to obtain improved predictions.

In this chapter, we detail our methodology which proposes implement this strategy by constructing reliability estimates for individual regression predictions ideally able to identify critical error scenarios defined by the user of the methodology, in such a manner that these reliability estimates can, in a second step, be applied specifically to adjust and improve computed individual regression predictions.

3.1. The Reliability Estimate: Critical Error Scenario Flag (CSFlag)

In this section, we describe the method to create a point reliability estimate for individual regression predictions, the Critical Error Scenario Flag (**CSFlag**); formerly named Critical Error Flag (**CEFlag**) when it was first proposed in our work presented in 2014 (MATSUMOTO; DEL-MORAL-HERNANDEZ, 2014). In this specific paper, we described the method to create reliability estimates for individual regression predictions by extracting information from the errors produced by the regression model with the training dataset.

Given a regression model, $G(X)$, the basic idea is to define this point reliability estimate, the **CSFlag**, as a binary variable, ***csflag***, to indicate whether the regression prediction of an individual observation in the training dataset, $\widehat{\boldsymbol{y}}_{\boldsymbol{Tra}} = G(\boldsymbol{x}_{\boldsymbol{Tra}})$, produced an error considered critical, according to a previously determined critical error scenario (**CScenario**).

This critical scenario might be established by the user of the regression model and it is supposed to be meaningful for the specific application of the regression system. It can be arbitrarily expressed, for instance: when the regression residual error value falls out of a specific interval or when it is higher than a certain threshold value.

The construction of confidence intervals, prediction intervals and tolerance intervals can be used to specify critical error scenarios conditions; however, the definition of **CScenarios** is not limited to these kinds of statistically defined intervals. As afore mentioned, the conditions can be arbitrarily set, even without a probabilistic meaning, as confidence intervals do, as long as they are useful conditions for defining a risky regression error, for the needs of a specific target application. The only requirement is that we must be able to create an algorithm to define the critical error scenario condition using only the known information available in the training dataset.

The proposition is to use the regression prediction values for the training dataset, $\widehat{\boldsymbol{Y}}_{\boldsymbol{Tra}} = G(\boldsymbol{X}_{\boldsymbol{Tra}})$, and the **CScenario** to determine the **CSFlag** value for each of the individual known observations in the training dataset, setting it to 1 (positive case) if the prediction error of the observation is critical according to the **CScenario**, or 0 (negative case) otherwise, as expressed in Equation (3.1).

$$\boldsymbol{csflag} = \begin{cases} 1\ ;\ if\ Error(y, \hat{y})\ is\ critical, where\ \hat{y}\ = G(x) \\ 0\ ;\ otherwise \end{cases} \qquad (3.1)$$

Thereby, the vector variable ***CSFLAG***, composed of the individual binary variables, ***csflag***, can be used to design a model to separate these two classes of patterns, which, in our research, received the name of Critical Error Scenario Alert Function (**CSAFunction**).

The **CSAFunction**, which is basically a binary pattern recognizer, can be constructed using Supervised Learning techniques, according to the diagram in Figure 1. The inputs are composed of the training dataset, $\boldsymbol{X_{Tra}}$, and the regression prediction values for the training dataset, $G(\boldsymbol{X_{Tra}}) = \widehat{\boldsymbol{Y}}_{\boldsymbol{Tra}}$; the target is the vector variable $\boldsymbol{CSFLAG_{Tra}}$ produced using the training dataset information and the **CScenario**.

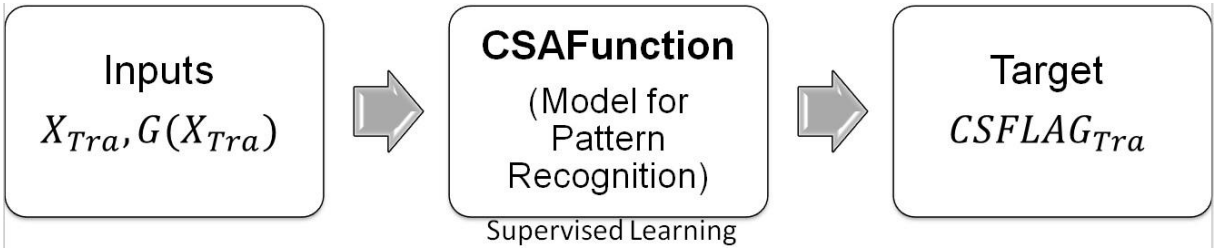

Figure 1: **CSAFunction**'s construction process.

The assumption is that the **CSAFunction** designed using the known information (i.e., $\boldsymbol{X_{Tra}}$ and $G(\boldsymbol{X_{Tra}})$) is an attempt to detect some of the numerical limitations of the regression model, $G(X)$.

As in any other Machine Learning method, having the training and the testing datasets generated by a same process is one of the requirements of our proposed methodology, otherwise the outcomes obtained using the information in the training dataset cannot be used to derive inferences about the observations in the testing dataset (ABU-MOSTAFA; MAGDON-ISMAIL; LIN, 2012b). Therefore, supposing that the training and the testing datasets comply with this main requirement, when applied to the testing dataset, the **CSAFunction** tends to correctly estimating the reliability estimate **CSFlag** values for these unseen observations in the testing dataset; thus, the positive cases (**CSFlag** equal to 1) would indicate the individual observations whose regression predictions, $\widehat{\boldsymbol{y}}_{\boldsymbol{Tes}}$, are likely to generate critical error scenario situations; conversely, the negative cases would signal a low risk of critical error scenario occurrences.

To illustrate the concepts expressed above, we will use figures representing the particular case of a simple regression model where X is composed of just one independent variable, although the described method is designed to handle multiple regression models.

Figure 2 shows an example of a set of $(\boldsymbol{X_{Tra}}, \boldsymbol{Y_{Tra}})$ observed data in the training dataset, and Figure 3 depicts an example of a regression, $G(X)$, that estimates the value of $\boldsymbol{Y_{Tra}}$ $(\widehat{\boldsymbol{Y}}_{\boldsymbol{Tra}} = G(\boldsymbol{X_{Tra}}))$.

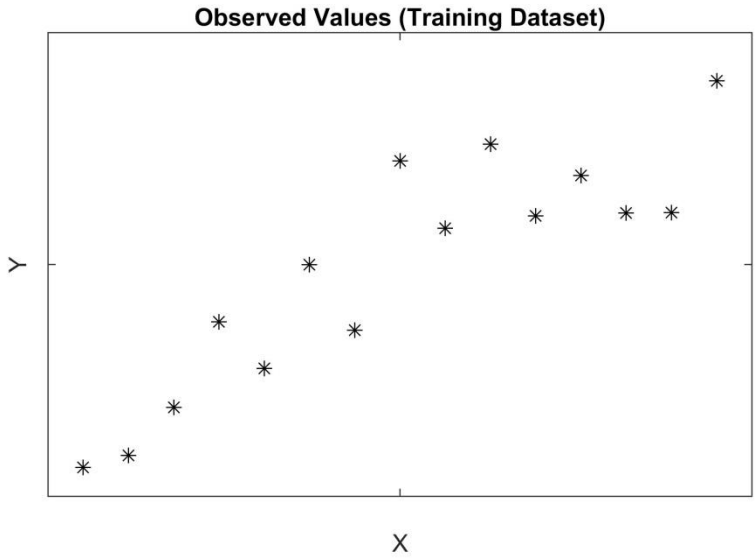

Figure 2: Observations in the training dataset.

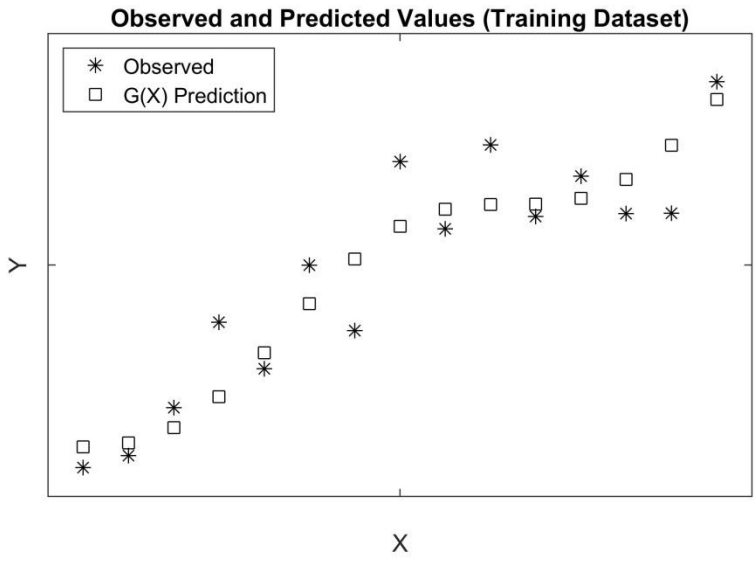

Figure 3: Predictions of regression G(X) for the training dataset.

At this point, the critical error scenario, **CScenario**, is supposed to be defined to establish when the regression prediction error in the training dataset is considered critical by the user of the regression. In this example, we arbitrarily determine the

following: the regression prediction error given by $\boldsymbol{e_{Tra}} = (\boldsymbol{y_{Tra}} - \boldsymbol{\hat{y}_{Tra}})$ is critical if the value observed is out of the limits defined by the dotted lines plotted in Figure 4.

Based on this scenario, we mark the observations in the training dataset identified as critical ones, and construct the binary variables, $\boldsymbol{csflag_{Tra}}$, that assume value 1 (indicated by the black square markers) in the case of critical error scenario occurrence and 0 (indicated by the white square markers), otherwise, as illustrated in Figure 5.

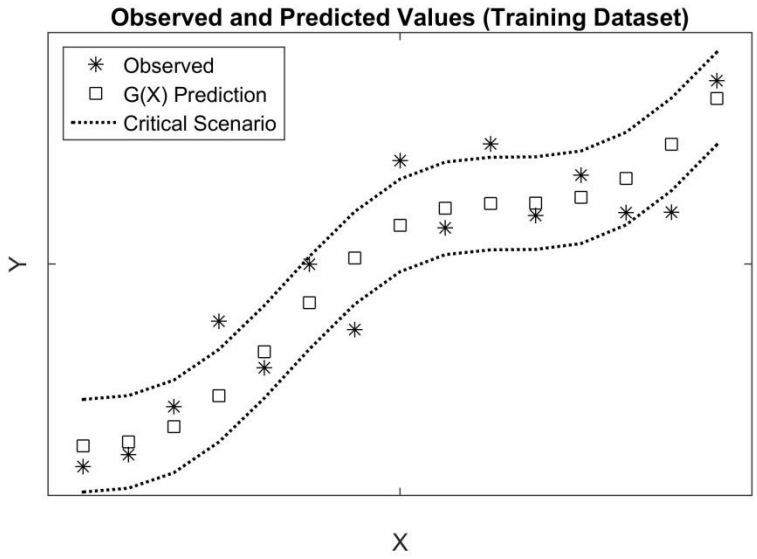

Figure 4: **CScenario** definition based on the prediction errors in the training dataset.

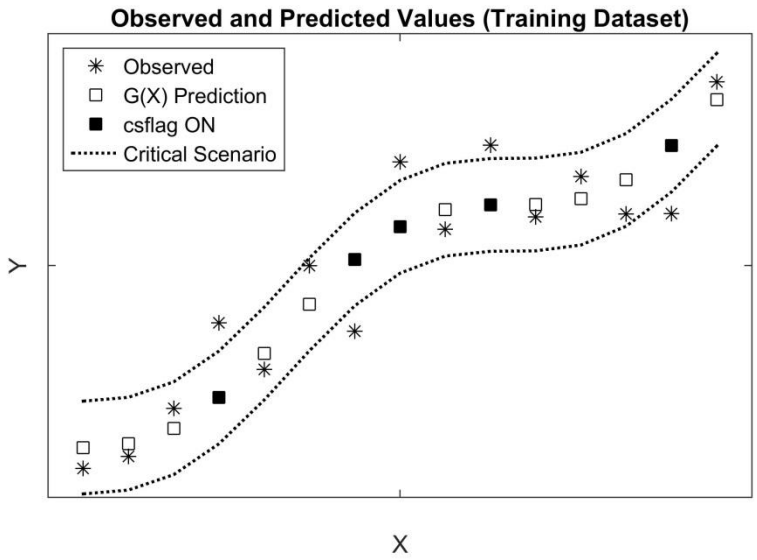

Figure 5: ***CSFlag*** construction for the training dataset.

Subsequently, this set of information from the training dataset is used to construct the **CSAFunction**, as previously illustrated in Figure 1, using $(\boldsymbol{X_{Tra}}, G(\boldsymbol{X_{Tra}}))$ as inputs, and the vector variable $\boldsymbol{CSFlag_{Tra}}$ composed of the $\boldsymbol{csflag_{Tra}}$ binary variables as target. Therefore, the **CSAFunction** can be used to predict the reliability estimate **CSFlag** value for the unknown observations in the testing dataset, $(\boldsymbol{X_{Tes}}, G(\boldsymbol{X_{Tes}}))$, as shown in Figure 6.

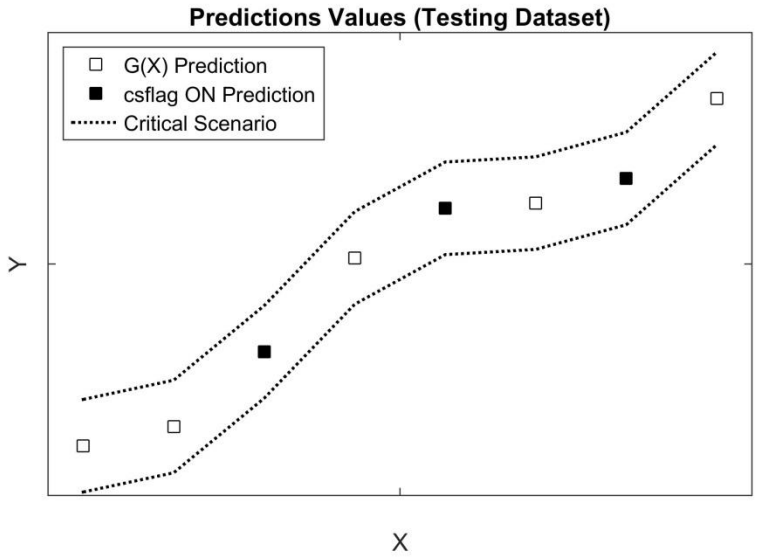

Figure 6: Predictions of regression G(X) and **CSAFunction** for the testing dataset.

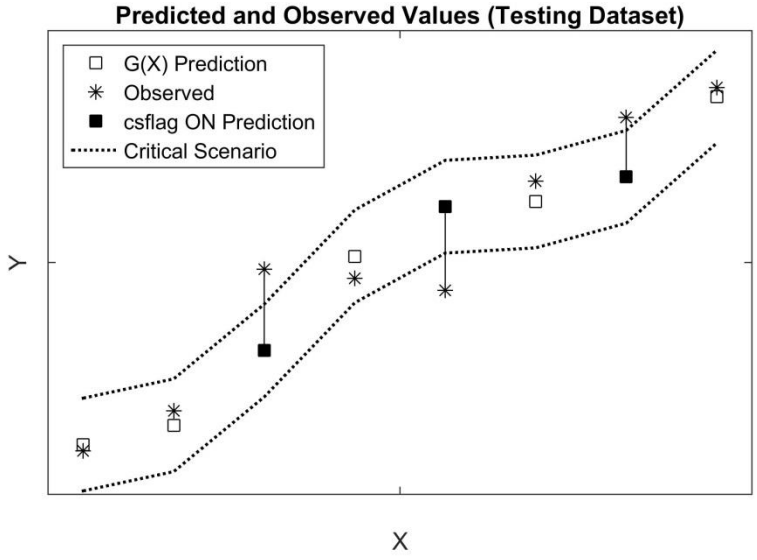

Figure 7: **CSAFunction** predictions compared to the new observed values in the testing dataset, assuming the perfect detection of all the critical error scenario cases.

Of course, as in any Machine Learning strategy, the detection of the cases of interest may be not 100% perfect, but supposing perfect discrimination, Figure 7 depicts what we would ideally obtain when the new observed values, $\boldsymbol{Y_{Tes}}$, in the testing dataset (represented by the star markers), become available: the regression predictions of the unseen observations indicated with the reliability estimate **CSFlag** ON in Figure 6, when compared to the realized values, $\boldsymbol{Y_{Tes}}$, actually produce errors considered critical, as indicated by the solid lines in Figure 7.

Due low reliability, the regression prediction values of the observations in the testing dataset estimated with **CSFlag** equal to 1, could be, for instance, simply discarded before the availability of the new values in the testing dataset, $\boldsymbol{Y_{Tes}}$, in such manner that the critical error scenarios occurrence would be avoided. We have employed this strategy in two of our papers, using two different **CScenario** definitions.

In the article presented in 2013 (MATSUMOTO; DEL-MORAL-HERNANDEZ, 2013), we defined the **CScenario** as follow: the observation is considered critical if the residual error of its regression prediction falls out of the confidence interval, with 90% of confidence level, defined by a Student's t-distribution adjusted to the regression residual errors produced by the training dataset. In a more recent work presented in 2014 (MATSUMOTO; DEL-MORAL-HERNANDEZ, 2014), instead of the regression residual error measurement, we used the regression Absolute Percentage Error (APE), and the **CScenario** was defined by an APE higher than 1.5%.

Conclusively, the outcomes from these two studies demonstrate that the reliability estimates **CSFlag** are able to provide additional assessment about individual regression predictions and can be used to support decision-making tasks.

In the next section, we describe our proposal for how to apply the concept of the critical error scenario definition, **CScenario**, to create the point reliability estimates for individual regression predictions, **CSFlag**, specifically to improve computed regression prediction values for unknown observations.

3.2. Regression Predictions Adjustment Procedure Using the CSFlag

As already stated in Section 3.1, the particular choice of the error scenario to distinguish what is critical or not is supposed to be defined by the user of the regression, according to what it is considered useful and appropriate for the specific application. In our research, to address the problem of improving regression prediction, we propose the following:

1. Using the training dataset, $\boldsymbol{X_{Tra}}$, and the regression prediction for the training dataset, $\boldsymbol{\widehat{Y}_{Tra}}$, to define two critical error scenarios as described below, where $\boldsymbol{\delta_{Pos}}$ and $\boldsymbol{\delta_{Neg}}$ are threshold values that can be arbitrarily set by the user of the methodology:

   - Quite positive prediction error value (**CScenario$_{Pos}$**):
     $$(\boldsymbol{y_{Tra}} - \boldsymbol{\widehat{y}_{Tra}}) > \boldsymbol{\delta_{Pos}}\ (\boldsymbol{\delta_{Pos}} > 0).$$
   - Quite negative prediction error value (**CScenario$_{Neg}$**):
     $$(\boldsymbol{y_{Tra}} - \boldsymbol{\widehat{y}_{Tra}}) < \boldsymbol{\delta_{Neg}}\ (\boldsymbol{\delta_{Neg}} < 0).$$

2. Constructing one reliability estimate **CSFlag** vector variable for each critical error scenario, using the training dataset, as expressed in Equation (3.5).

$$For\ each\ \boldsymbol{y_{Tra}} \in \boldsymbol{Y_{Tra}}$$
$$\boldsymbol{csflag_{Pos}} = \begin{cases} 1\ ;\ if\ (\boldsymbol{y_{Tra}} - \boldsymbol{\widehat{y}_{Tra}}) > \boldsymbol{\delta_{Pos}}\ (CScenario_{\ Pos}) \\ 0\ ;\ otherwise \end{cases} \qquad (3.5)$$
$$\boldsymbol{csflag_{Neg}} = \begin{cases} 1\ ;\ if\ (\boldsymbol{y_{Tra}} - \boldsymbol{\widehat{y}_{Tra}}) < \boldsymbol{\delta_{Neg}}\ (CScenario_{Neg}) \\ 0\ ;\ otherwise \end{cases}$$

3. Designing two models for pattern recognition (**CSAFunction$_{Pos}$** and **CSAFunction$_{Neg}$**) using the training dataset information and the vector variables $\boldsymbol{CSFLAG_{Pos}}$ and $\boldsymbol{CSFLAG_{Neg}}$, composed of the variables $\boldsymbol{csflag_{Pos}}$ and $\boldsymbol{csflag_{Neg}}$, respectively.

4. Using these two **CSAFunctions** to predict the two types of individual reliability estimates ($\boldsymbol{\widehat{C}_{Pos}}$,$\boldsymbol{\widehat{C}_{Neg}}$) for the observations in the testing dataset.

5. Using the two **CSFlag** individual reliability estimates ($\boldsymbol{\widehat{C}_{Pos}}$,$\boldsymbol{\widehat{C}_{Neg}}$), and the threshold values ($\boldsymbol{\delta_{Pos}}$, $\boldsymbol{\delta_{Neg}}$) to individually adjust the regression prediction values produced with the testing dataset, $\boldsymbol{\widehat{y}_{Tes}}$, as expressed in Equation (3.6):

$$
\begin{aligned}
&For\ each\ \hat{\boldsymbol{y}}_{\boldsymbol{Tes}} \in \hat{\boldsymbol{Y}}_{\boldsymbol{Tes}} \\
&\quad \hat{\boldsymbol{c}}_{\boldsymbol{Pos}} = CSAFunction_{Pos}(\boldsymbol{x}_{\boldsymbol{Tes}}, \hat{\boldsymbol{y}}_{\boldsymbol{Tes}}) \\
&\quad \hat{\boldsymbol{c}}_{\boldsymbol{Neg}} = CSAFunction_{Neg}(\boldsymbol{x}_{\boldsymbol{Tes}}, \hat{\boldsymbol{y}}_{\boldsymbol{Tes}}) \\
&\quad if\ \left(\hat{\boldsymbol{c}}_{\boldsymbol{Pos}} = 1 \text{ and } \hat{\boldsymbol{c}}_{\boldsymbol{Neg}} = 0\right) then\ \hat{\boldsymbol{y}}_{\boldsymbol{Adj}} = \hat{\boldsymbol{y}}_{\boldsymbol{Tes}} + \boldsymbol{\delta}_{\boldsymbol{Pos}} \\
&\quad if\ \left(\hat{\boldsymbol{c}}_{\boldsymbol{Pos}} = 0 \text{ and } \hat{\boldsymbol{c}}_{\boldsymbol{Neg}} = 1\right) then\ \hat{\boldsymbol{y}}_{\boldsymbol{Adj}} = \hat{\boldsymbol{y}}_{\boldsymbol{Tes}} + \boldsymbol{\delta}_{\boldsymbol{Neg}}
\end{aligned}
\tag{3.6}
$$

For the cases in which the **CSFlags** variables $\hat{\boldsymbol{c}}_{\boldsymbol{Pos}}$ and $\hat{\boldsymbol{c}}_{\boldsymbol{Neg}}$ are correctly estimated as equal to 1, by construction, the prediction error produced by the adjusted prediction, $\hat{\boldsymbol{y}}_{\boldsymbol{Adj}}$, is smaller than the one produced by the original prediction, $\hat{\boldsymbol{y}}_{\boldsymbol{Tes}}$, $\left(\left|\left(\boldsymbol{y}_{\boldsymbol{Tes}} - \hat{\boldsymbol{y}}_{\boldsymbol{Adj}}\right)\right| \leq \left|\left(\boldsymbol{y}_{\boldsymbol{Tes}} - \hat{\boldsymbol{y}}_{\boldsymbol{Tes}}\right)\right|\right)$, hence the value, $\hat{\boldsymbol{y}}_{\boldsymbol{Adj}}$, adjusted by the proposed methodology can be considered better than the original regression prediction, $\hat{\boldsymbol{y}}_{\boldsymbol{Tes}}$.

Equation (3.6) follows the adjustment procedure rules detailed in Table 1.

Table 1: Adjustment procedure rules.

Condition	Interpretation	Action
$\hat{\boldsymbol{c}}_{\boldsymbol{Pos}} = 1$ and $\hat{\boldsymbol{c}}_{\boldsymbol{Neg}} = 0$	Quite positive prediction error critical scenario (**CScenario$_{\text{Pos}}$**) identified.	Add the positive scenario threshold value to the regression prediction: $\hat{\boldsymbol{y}}_{\boldsymbol{Adj}} = \hat{\boldsymbol{y}}_{\boldsymbol{Tes}} + \boldsymbol{\delta}_{\boldsymbol{Pos}}$
$\hat{\boldsymbol{c}}_{\boldsymbol{Pos}} = 0$ and $\hat{\boldsymbol{c}}_{\boldsymbol{Neg}} = 1$	Quite negative prediction error critical scenario (**CScenario$_{\text{Neg}}$**) identified.	Add the negative scenario threshold value to the regression prediction: $\hat{\boldsymbol{y}}_{\boldsymbol{Adj}} = \hat{\boldsymbol{y}}_{\boldsymbol{Tes}} + \boldsymbol{\delta}_{\boldsymbol{Neg}}$
$\hat{\boldsymbol{c}}_{\boldsymbol{Pos}} = 0$ and $\hat{\boldsymbol{c}}_{\boldsymbol{Neg}} = 0$	No critical scenario identified.	No action required.
$\hat{\boldsymbol{c}}_{\boldsymbol{Pos}} = 1$ and $\hat{\boldsymbol{c}}_{\boldsymbol{Neg}} = 1$	Conflicting outcomes.	No action taken.

The next figure (Figure 8) depicts the effect of the adjustment procedure on the original regression predictions for the testing dataset, $\hat{\boldsymbol{Y}}_{\boldsymbol{Tes}}$. The adjustments are indicated by the arrows. In Figure 9, we have the outcomes of the adjustment procedure compared to the new observed values, $\boldsymbol{Y}_{\boldsymbol{Tes}}$, in the testing dataset. In the ideal case of perfect discrimination of the positive cases, the adjusted regression predictions, $\hat{\boldsymbol{Y}}_{\boldsymbol{Adj}}$, (indicated by the black circle markers) are closer to the observed values, $\boldsymbol{Y}_{\boldsymbol{Tes}}$, (indicated by the star markers) than the original regression predictions, $\hat{\boldsymbol{Y}}_{\boldsymbol{Tes}}$, (indicated by the white square markers).

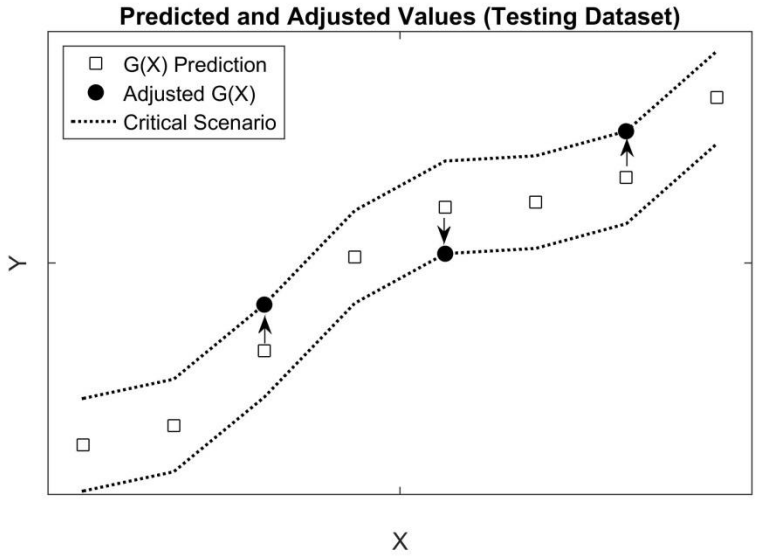

Figure 8: Effect of the adjustment procedure on the original predictions for the testing dataset.

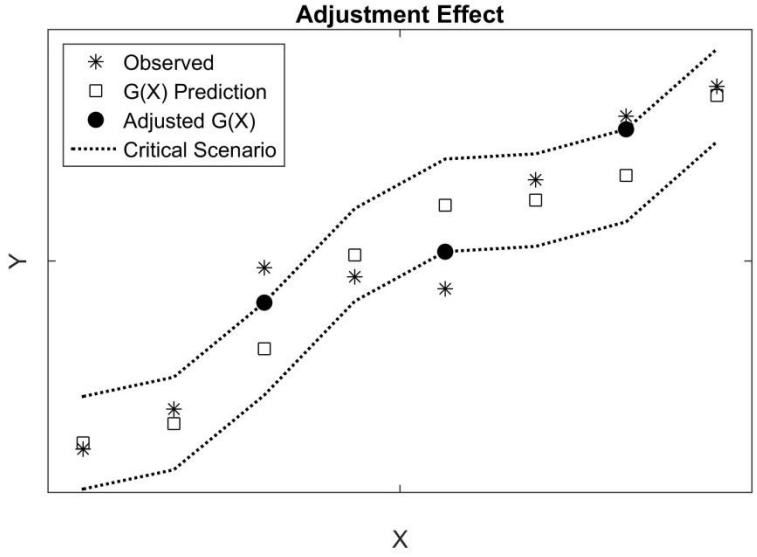

Figure 9: Original and adjusted predictions compared to the new observed values in the testing dataset.

Although the threshold values, $\boldsymbol{\delta_{Neg}}$ and $\boldsymbol{\delta_{Pos}}$, of the critical scenarios may be arbitrarily set, considering the problem of computed regression prediction improvement, we recommend defining these values using the procedure detailed ahead in Item 3.2.1.

3.2.1. Critical Error Scenario Threshold Values Definition

According to the definitions of the two critical error scenarios afore described, the choice of higher absolute threshold values, $|\boldsymbol{\delta_{Pos}}|$ and $|\boldsymbol{\delta_{Neg}}|$, likely implies higher adjustment values but lower numbers of adjustments (numbers of observations with **CSFlag** equal to 1); on the other hand, lower absolute threshold values likely imply the opposite, lower adjustment values but higher numbers of adjustments.

To solve this trade-off, we propose handling this issue by solving the optimization problem defined as: find the threshold values, $\boldsymbol{\delta_{Pos}}$ and $\boldsymbol{\delta_{Neg}}$, that minimize the RMSE value generated by the regression predictions for the training dataset after applying the adjustment procedure shown in Equation (3.6).

Since the threshold values, $\boldsymbol{\delta_{Pos}}$ and $\boldsymbol{\delta_{Neg}}$, depend on the problem domain range, to set the optimization problem with a range independent definition, and supposing that the observations from the training and the testing datasets have the same data distribution; we additionally propose indirectly finding the threshold values by searching the probability values $\boldsymbol{\alpha_{Pos}}$ and $\boldsymbol{\alpha_{Neg}}$ given by the empirical cumulative distribution function of the prediction regression error values for the training dataset (Figure 10).

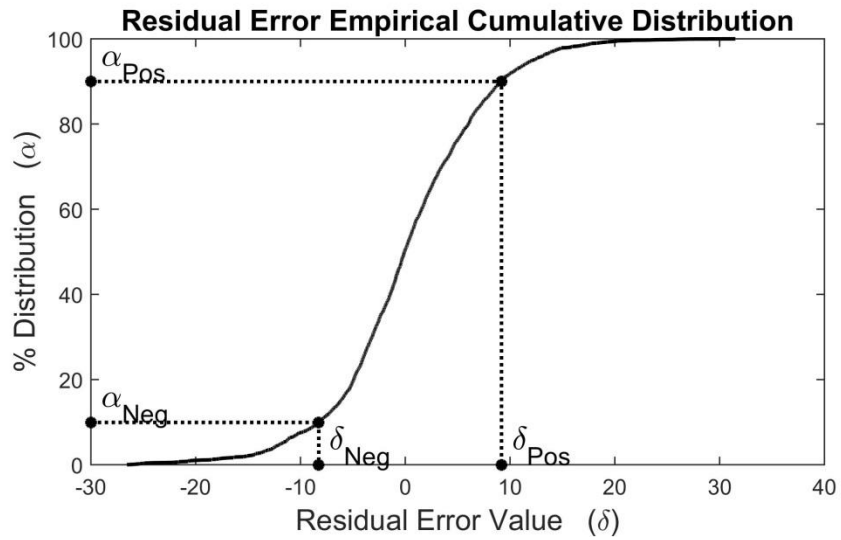

Figure 10: Example of empirical cumulative distribution of the prediction regression error values.

Consequently, the values $\boldsymbol{\alpha_{Pos}}$ and $\boldsymbol{\alpha_{Neg}}$ can be used to redefine the two **CScenario**s as expressed in Equation (3.7), where $\boldsymbol{\alpha_{Pos}}$ and $\boldsymbol{\alpha_{Neg}}$ are always limited

by the interval [0,1] (or [0%,100%]), in other words, the definitions of the **CSCenario**s are range independent.

The values of $\boldsymbol{\delta_{Pos}}$ and $\boldsymbol{\delta_{Neg}}$ are obtained using the inverse of the empirical cumulative distribution function of the prediction regression error values for the training dataset, given the probabilities $\boldsymbol{\alpha_{Pos}}$ and $\boldsymbol{\alpha_{Neg}}$, respectively.

$$For\ each\ \boldsymbol{y_{Tra}} \in \boldsymbol{Y_{Tra}}$$

$$\boldsymbol{csflag_{Pos}} = \begin{cases} 1\,;\, if\ (\boldsymbol{y_{Tra}} - \boldsymbol{\hat{y}_{Tra}}) > \boldsymbol{\delta_{Pos}}\ (CScenario_{Pos}) \\ 0\,;\ otherwise \end{cases}$$

$$\boldsymbol{\delta_{Pos}}\ chosen\ such\ that\ Prob\big((\boldsymbol{y_{Tra}} - \boldsymbol{\hat{y}_{Tra}}) > \boldsymbol{\delta_{Pos}}\big) = (1 - \boldsymbol{\alpha_{Pos}}) \qquad (3.7)$$

$$\boldsymbol{csflag_{Neg}} = \begin{cases} 1\,;\, if\ (\boldsymbol{y_{Tra}} - \boldsymbol{\hat{y}_{Tra}}) < \boldsymbol{\delta_{Neg}}\ \big(CScenario_{Neg}\big) \\ 0\,;\ otherwise \end{cases}$$

$$\boldsymbol{\delta_{Neg}}\ chosen\ such\ that\ Prob\big((\boldsymbol{y_{Tra}} - \boldsymbol{\hat{y}_{Tra}}) < \boldsymbol{\delta_{Neg}}\big) = \boldsymbol{\alpha_{Neg}}$$

Next, we describe the experiments conducted to evaluate and to support the development of the methodology presented in this chapter.

4. EXPERIMENTS DESCRIPTION

In this chapter, we detail the construction of the working datasets and the implementation of the procedures employed in the three sets of experiments performed to verify the ideas detailed in Chapter 3 using three different types of data.

The first set of experiments, using synthetic data, was conducted to examine how the reliability estimates and the adjustment procedure defined by our methodology behave in the case of artificially produced clean data, as well as in the case of the addition of random numbers with different amplitudes, simulating the presence of uncertainties and nondeterministic factors.

The second and the third sets of experiments using several databases from UCI Data Repository and ISO-NE website were performed to observe the outcomes of the proposed methodology with cross sectional real data, and time series real data, respectively.

This chapter begins with the descriptions of the working datasets of each of the three set of experiments, and follows with the descriptions of the procedures that were performed for each of the working datasets. These procedures are:

1. The construction of the regression models to compute the prediction values to be improved by the proposed methodology;
2. The definition of the threshold values of the two critical error scenarios: quite positive prediction error (**CScenario$_{Pos}$**) and quite negative prediction error (**CScenario$_{Neg}$**);
3. The design of the two **CSAFunctions** (**CSAFunction$_{Pos}$** and **CSAFunction$_{Neg}$**) to predict the two reliability estimates **CSFlag$_{Pos}$** and **CSFlag$_{Neg}$**;
4. The execution and the evaluation of the regression predictions adjustment procedure performed using the two reliability estimates **CSFlag$_{Pos}$** and **CSFlag$_{Neg}$**, and the threshold values of the two critical error scenarios.

4.1. Data Description

We intended to simulate the hypothetical condition in which the end user has neither access to the regression modeling process nor the regression training data. For this reason, in the experiments with synthetic data and UCI data, the regressions and the **CSAFunctions** were modeled using different working datasets; however, if the training dataset used to model the regression is available, there is no restriction on using it to model the **CSAFunctions** as well. This situation was illustrated in the case of the experiments with ISO-NE data.

4.1.1. Synthetic Working Datasets

The artificial clean data was generated using the function expressed in Equation (4.1) as the target function to generate the data y ; and the nondeterministic components, here generically named "noise", were produced with the addition of random values with normal distribution, $n \sim N(0, nf^2)$, as expressed in Equation (4.2). The quadratic function $F(x) = x^2$ was intentionally chosen to verify the behavior of the **CSAFunctions** in a very simple and controlled framework, using data with low volatility.

Clean data: $y = F(x) = x^2 \;\; ; \;\; x \sim U(0,1)$ (uniform distribution) (4.1)

Noisy data: $\tilde{y} = y + n \;\; ; n \sim N(0, \boldsymbol{nf}^2)$ (normal distribution) (4.2)

$with\ \boldsymbol{nf} \in [0.000, 0.300[$

The noise factor, $\boldsymbol{nf}$, was used to numerically represent the amplitude of the nondeterministic components, or the level of noise in the y data, and 300 data samples were generated using $\boldsymbol{nf}$ values increasing from 0.000 up to 0.300, using increment value equal to 0.001. Each of the 300 data samples was composed of 250 observations that were split into three working datasets named Regression, Training, and Testing datasets as follows (see Table 2):

- Regression dataset composed of 100 observations, used to model the regressions;
- Training dataset composed of 100 observations, used to model the **CSAFunctions**;

- Testing dataset composed of 50 observations, used to test the **CSAFunctions** and to evaluate the performance of the proposed methodology.

Table 2: (Synthetic) Working dataset sizes.

Synthetic Databases Group	Dataset Size (Number of Observations)			
	Total	Regression Dataset	CSAFunctions Modeling	
			Training Dataset	Testing Dataset
Synthetic Datasets	250	100	100	50

4.1.2. UCI Working Datasets

In the case of cross sectional real data, the methodology was evaluated on eight public multivariate databases provided by the UCI (University of California, Irvine) Machine Learning Repository (BACHE; LICHMAN, 2015) listed below. Table 3 shows general characteristics of the databases.

- **Airfoil**:
  - UCI Name: Airfoil Self-Noise.
  - Source: NASA.
  - Output variable: Scaled sound pressure level, in decibels, observed in airfoils at various wind tunnel speeds and angles of attack.
- **AutoMPG**:
  - UCI Name: Auto MPG.
  - Source: StatLib library which is maintained by the Carnegie Mellow University.
  - Output variable: City-cycle fuel consumption, in miles per gallon.
- **Combined Cycle**:
  - UCI Name: Combined Cycle Power Plant.
  - Source: Faculty of Engineering, Namık Kemal University, in Tekirdağ/Turkey. Data points collected from a Combined Cycle Power Plan over 6 years (2006-2011).
  - Output variable: Load electrical power output.
- **Concrete**:
  - UCI Name: Concrete Compressive Strength.

  - Source: Department of Information Management, Chung-Hua University, Taiwan.
  - Output: Concrete compressive strength, in MPa (mega pascal).
- **Energy**:
  - UCI Name: Energy Efficiency.
  - Source: Oxford Centre for Industrial and Applied Mathematics, University of Oxford, UK.
  - Output: Cooling load.
- **Housing**:
  - UCI Name: Housing
  - Source: StatLib library which is maintained by the Carnegie Mellow University.
  - Output: Median value of owner-occupied homes in $1000's in Boston area.
- **Parkinsons**:
  - UCI Name: Parkinsons Telemonitoring.
  - Source: Oxford Centre for Industrial and Applied Mathematics, University of Oxford, UK.
  - Output: Total UPDRS (Unified Parkinson's disease rating scale).
- **Yacht**:
  - UCI Name: Yacht Hydrodynamics.
  - Source: Ship Hydromechanics Laboratory, Maritime and Transport Technology Department, Technical University of Delft/Nederland.
  - Source: Residuary resistance per unit weight of displacement.

Table 3: (UCI) Original databases descriptions.

UCI Datasets	Attribute Types	Number of Observations	Number of Attributes	Year of Upload
Airfoil	Real	1503	6	2013
AutoMPG	Categorical, Real	398	8	1993
Combined Cycle	Real	9568	4	2014
Concrete	Real	103	10	2009
Energy	Integer, Real	768	8	2012
Housing	Categorical, Integer, Real	506	14	1993
Parkinsons	Integer, Real	5875	26	2009
Yacht	Real	308	7	2013

Information about the variables that compose each of these eight databases is displayed in Appendix B, from Table B01 to B08. These tables show a short description of each of these variables and the following five statistics information: average, standard deviation, minimum, maximum, and volatility (standard deviation divided by average) values.

For each of these eight databases, 100 data samples were created using the bagging sampling method, and each of these 100 data samples was split into three parts named Regression, Training, and Testing datasets, similarly to what was done in the case of the synthetic data, as follows:

- Regression dataset composed of 40% of the observations, used to model the regressions;
- Training dataset composed of 42% of the observations, used to model the **CSAFunctions**;
- Testing dataset composed of 18% of the observations, used to test the **CSAFunctions** and to evaluate the proposed methodology performance.

The percentages were arbitrarily chosen so that around 80% of the total amount of the original data was reserved for modeling, about half for the regression, and half for the **CSAFunctions** modeling; and the remaining part was separated for the **CSAFunctions** testing; and the exact working datasets sizes (in number of observations) are displayed in Table 4.

Table 4: (UCI) Working datasets sizes.

UCI Datasets Groups	Dataset Size (Number of Observations)			
	Total	Regression Dataset	CSAFunctions Modeling	
			Training Dataset	Testing Dataset
Airfoil	1503	602	631	270
AutoMPG	392	157	165	70
Combined Cycle	9568	3828	4018	1722
Concrete	1030	412	433	185
Energy	768	308	322	138
Housing	506	203	213	90
Parkinsons	5875	2350	2468	1057
Yacht	308	124	129	55

4.1.3. ISO-NE Working Datasets

ISO-NE (Independent System Operator in New England) is an independent, non-profit Regional Transmission Organization responsible for operating the grid power system (balancing electricity supply and demand), administering wholesale electricity markets, and planning the power system over New England, in the USA. One of its major duties is to provide tariffs for the prices, terms, and conditions of the energy supply in this area. All this information is public and available in the ISO-NE website (ISO-NE, 2015).

The methodology was evaluated in experiments using data collected from the eight weather stations of ISO-NE Control Area: Boston, Burlington, Bridgeport, Concord, Portland, Providence, Windsor Locks, and Worcester.

From a total of fourteen variables available in the ISO-NE database (listed in Appendix B – Table B09), five of them related to power load, date and weather condition were selected to build the working time series data samples:

- *Date*: date in MM/DD/YYYY format.
- *Hour*: hour ending value.
- *DEMAND*: load used in the settlement process.
- *DryBulb*: dry bulb temperature in degrees Fahrenheit.
- *DewPnt:* dew point temperature in degrees Fahrenheit.

These time series data samples were composed of one dependent variable (output), V_t, and six independent variables (inputs), as displayed in Table 5.

Table 5: (ISO-NE) Working database variables.

Variable	Description
$\boldsymbol{V_t}$	Load variation in t, at the hour.
$\boldsymbol{V_{t-1}}$	Load variation in (t-1), at one hour before.
$\boldsymbol{V_{t-24}}$	Load variation in (t-24), at the hour one day before.
$\boldsymbol{Dr_{t-1}}$	Dry bulb temperature in (t-1), at one hour before.
$\boldsymbol{De_{t-1}}$	Dew point temperature in (t-1), at one hour before.
$\boldsymbol{B_t}$	Boolean flag: is Busy day at the hour? No (0) or Yes (1).
$\boldsymbol{cH_t}$	$cos((Hour_t * \pi)/12)$, $Hour_t$ is an integer value between 0 and 23.

Variables V_t, V_{t-1}, and V_{t-24} derive from the original *DEMAND* variable; the B_t, from the original *Date* variable; the cH_t, from the original *Hour* variable; the Dr_{t-1}, from the *DryBulb*; and De_{t-1}, from the *DewPnt*.

We considered the time frame of hourly data collected during the months of March from 2012 to 2015. One time series data sample was created for each of the eight stations. They were all split into two working datasets, according to the description in TABLE 6.

Table 6: (ISO-NE) Working datasets sizes and time frame division.

Datasets	**# of Obs.**	**Time frame**
Training	2232	From March 1st 2012 (1 am) to March 31st 2012 (0 am), from March 1st 2013 (1 am) to March 31st 2014 (0 am), from March 1st 2014 (1 am) to March 31st 2014 (0 am).
Testing	168	From March 10th 2015 (1 am) to March 16th 2015 (0 am).

Five statistics information of these seven variables (one output and six input variables), considering the data collected from all eight weather stations, is displayed in Appendix B – Table 10: average, standard deviation, minimum, maximum, and volatility (standard deviation divided by average) values.

The Training datasets were used to build the regression models and the **CSAFunctions** models. The Testing datasets were used to verify the performance of the **CSAFunctions** models and the effect of the adjustment procedure.

At the time this document was finished (last quarter of 2015), the direct link to the spreadsheet of the ISO-NE data files was:

http://iso-ne.com/isoexpress/web/reports/pricing/-/tree/zone-info.

In the next sections, we detailed the procedures that were performed for each of the working datasets described in this section.

4.2. Regressions modeling

In this section, we describe the construction of the regression models created to provide the computed prediction value to be improved by the application of the

adjustment procedure using the **CSFlag** reliability estimates proposed by our methodology.

4.2.1. Regression Models for the Experiments with Synthetic Data

As presented in item 4.1.1., in the case of synthetic data, we defined an artificial single regression problem setting the quadratic function, $F(x) = x^2$, as the target function to generate the y data.

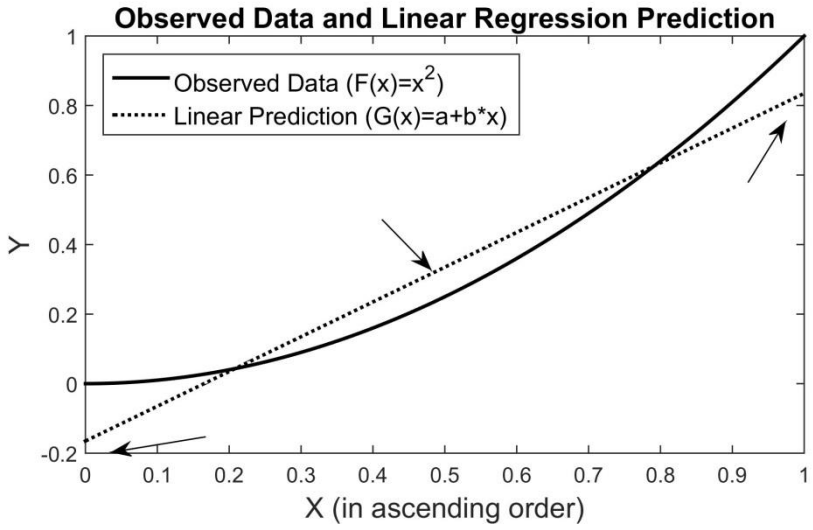

Figure 11: Linear regression applied to data produced by a quadratic function.

To predict the y data generated by $F(x)$, we chose the single linear regression model, $G(x) = a + b * x$, calibrated using the OLS (Ordinary Least Squared) method. This was done on purpose to test if the **CSAFunctions** would be able to capture the numerical limitations of the single linear regression model in a simple outline, clearly observable in Figure 11 (particularly in the regions indicated by the arrows) by examining the response of the reliability estimates (**CSFlag**) produced by the **CSAFunctions**; and also to monitor the effects of the adjustment procedure on datasets with different levels of noise.

4.2.2. Regressions Models for the Experiments with Real Data

The experiments using real databases from UCI and ISO-NE were performed to investigate the potential practical use of the proposed methodology with real data, where the target functions that generate the data are usually unknown, and data are

more volatile and are frequently affected by the presence of nondeterministic components related to all kind of uncertainties.

For these two sets of experiments, we trialed the methodology with two classes of regression models: a linear regression model, the classic linear (Linear) regression model; and a non-linear regression model, the feedforward three layered Artificial Neural Networks (ANN) regression model. Nevertheless, the application of the developed methodology is not restricted to these only two classes of regression models (Linear and ANN).

All Linear regression models were constructed using the linear model formulation, and we calibrated the parameters applying the classical OLS (Ordinary Least Squared) method (WOOLDRIDGE, 2009a).

All ANN regression models were constructed using feedforward three-layered perceptron Artificial Neural Networks (ANN), following an architectural description traditionally recommended by the technical literature (HAYKIN, 2008) to solve problems of regression prediction (KWOK; YEUNG, 1997), time series forecasting (HIPPERT; PEDREIRA; SOUZA, 2001), classification (MATSUMOTO; PINTO, 2009), and general engineering and scientific applications (SAMARASHINGUE, 2006).

The ANN models were trained using the Levenberg-Marquardt backpropagation algorithm, in which we adopted the mean squared error (MSE) as the performance metric function. To complete the ANN architecture definition, we arbitrarily set the following numbers and proportions: the numbers of neurons in the input layer and hidden layer were set as equal to the number of the regression input parameters; and the output layer was set with just one neuron.

The original Training dataset was divided into three bootstrap data sample subsets: a Training subset (composed of 70% of the total dataset) used to calibrate the model parameters; a Control subset (composed of 15% of the total dataset) used for cross-validation to avoid overfitting; and a Verifying subset (composed of 15% of the total dataset) used to choose the “best” ANN (BELL, 2014). To define each regression, ten different ANN regression models were trained and the one among them that produced the smallest MSE value for the Verifying data subset was chosen as the final regression model.

The following procedures described in Sections 4.3, 4.4, and 4.5 were equally executed for the linear and the ANN regression predictions to allow the comparison of the outcomes of the proposed methodology produced in both cases.

4.3. Critical Error Scenario Threshold Values Definition

As previously described in Item 3.2.1, we propose determining the threshold values of the **CScenario**s by handling the trade-off: higher absolute threshold values with less adjustment cases *versus* lower absolute threshold values with more adjustment cases.

Equation (4.3) expresses the formalization of the procedure to determine the negative threshold value in the case of the **CScenario$_{Neg}$** ($\boldsymbol{\delta_{Neg}}$) employed in our experiments.

$$\begin{aligned} \alpha_{Neg}: \; & minimize\ RMSE_{Value} = \delta_{Neg} ObjectiveFunction\left(Y_{Tra}, \hat{Y}_{Tra}, \alpha\right) \\ & \alpha \;\; ; \; subject\ to\ 0 < \alpha < 1 \\ \delta_{Neg} = \; & EmpiricalCumulativeDistribution^{-1}\left(Y_{Tra} - \hat{Y}_{Tra}, \alpha_{Neg}\right) \end{aligned} \tag{4.3}$$

(1) $FUNCTION\ RMSE_{Value} = \delta_{Neg} ObjectiveFunction(Y_{Tra}, \hat{Y}_{Tra}, \alpha)$
(2) $Error_{Tra} = Y_{Tra} - \hat{Y}_{Tra}$ $\%\ Original\ Prediction\ Error$
(3) $\delta_{\alpha} = EmpiricalCumulativeDistribution^{-1}(Error_{Tra}, \alpha)$ $\%\ Threshold$
(4) $FOR\ EACH\ y \in Y_{Tra}$ $\%\ Create\ \hat{Y}_{Adj}$: $prediction\ adjusted\ by\ the\ Threshold$
(5) $\hat{y}_{Adj} = \begin{cases} (\hat{y}_{Tra} + \delta_{\alpha}) & ; \; if\ (y - \hat{y}) < \delta_{\alpha} \\ \hat{y}_{Tra} & ; \quad otherwise \end{cases}$
(6) $END\ OF\ FOR$
(7) $RMSE_{Value} = rmse\left(Y_{Tra} - \hat{Y}_{Adj}\right)$ $\%\ RMSE\ of\ the\ Adjusted\ Prediction\ Error$
(8) $END\ OF\ FUNCION$

Figure 12: **Cscenario$_{Neg}$** objective function pseudo-code.

Figure 12 shows the pseudo-code of the objective function of this optimization problem. This objective function calculates the RMSE value produced by the regression predictions for the Training dataset ($\boldsymbol{\hat{Y}_{Tra}}$) adjusted by applying the

procedure expressed in Equation (3.6), ($\widehat{\boldsymbol{Y}}_{\boldsymbol{Adj}}$), using the threshold value, $\boldsymbol{\delta_{\alpha}}$. The $\boldsymbol{\delta_{\alpha}}$ value is defined by the inverse of the empirical cumulative distribution function of the regression error values produced by the Training dataset, given the probability $\boldsymbol{\alpha}$.

Formula (4.3) and the pseudo-code in Figure 12 refer to the quite negative error value critical scenario (**CScenario$_{Neg}$**). In the case of the quite positive error critical scenario (**CScenario$_{Pos}$**), there are two changes:

1. The optimization procedure to obtain $\boldsymbol{\delta_{Pos}}$ minimizes the value of $(1 - \alpha_{Pos})$.

2. In Line (5) of the afore pseudo-code, the adjustment condition is inverted:

$$(5)\quad \hat{y}_{Adj} = \begin{cases} (\hat{y}_{Tra} + \delta_{\alpha}) & ;\ if\ (y - \hat{y}) > \delta_{\alpha} \\ \hat{y}_{Tra} & ;\ otherwise \end{cases}$$

And the value of δ_{Pos} is defined by the inverse of the empirical cumulative distribution function of the regression error values produced by the training dataset, given the probability α_{Pos}.

Finally, the values of $\boldsymbol{\delta_{Pos}}$ and $\boldsymbol{\delta_{Neg}}$ obtained by following the procedure described above are used to construct the vector variables ***CSFLAG$_{Pos}$*** and the ***CSFLAG$_{Neg}$*** of the Training dataset, as previously expressed in Equation (3.5); and these two vector variables are used to model the **CSAFunctions** according to the description detailed in the next section, employing Machine Learning techniques.

4.4. Critical Error Scenario Alert Function Design

One important element that is in the core of our methodology is the design of the **CSAFunction**, the computational algorithm constructed to recognize the critical error scenario patterns and predict the reliability estimate **CSFlag**. Hence, regardless of the type of the model adopted to implement the **CSAFunction**, we must face the overfitting problem previously mentioned in Section 2.1, i.e., we have to find the balance between the model complexity and the model generalization capability. In the experiments preformed in this research, the overfitting problem was handled using the cross-validation technique.

Furthermore, as illustrate in Figure 1 (Chapter 3), the **CSAFunctions** are supposed to be constructed applying Supervised Learning techniques, using as inputs the input variables of the training dataset, $\boldsymbol{X_{Tra}}$, and the regression prediction values for the training dataset, $\boldsymbol{G(X_{Tra}) = \widehat{Y}_{Tra}}$; and as output the vector variable $\boldsymbol{CSFLAG_{Tra}}$, composed of the reliability estimate binary variable, $\boldsymbol{csflag_{Tra}}$, that indicate for which observations of the training dataset the regression predictions produced critical error scenario condition.

Therefore, presuming that the regression model quality is reasonably good, the vector variable $\boldsymbol{CSFLAG_{Tra}}$ is supposed to contain imbalanced data, with much fewer ones than zeros. To handle this specific condition, the design of the **CSAFunction** requires additional and special attention because standard classification algorithms usually adopt strategies that minimize the MSE (Mean Squared Error) value, and this operation tends to work well with balanced data, but to be biased towards the majority class in the case of imbalanced data (KOTSIANTIS; KANELLOPOULOS; PINTELAS, 2010).

Moreover, with reference to the evaluation of the **CSAFunction** model performance, among several evaluation metrics described in the specialized literature on classification with imbalanced datasets (GOUTTE; GAUSSIER, 2005), we chose three of them to be observed: Precision, Sensitivity, and F-Measure.

The first metric, Precision, defined as shown in Equation (4.4), measures the proportion of positive predictions made by the model that are correct. It indicates how good the positive predictions of the model are.

$$Precision = \frac{Number\ of\ True\ Positives}{(Number\ of\ True\ Positives + Number\ of\ False\ Positives)} \quad (4.4)$$

The second metric, Sensitivity (also called Recall), defined as shown in Equation (4.5), measures the proportion of true positive patterns that are correctly detected by the model. It indicates how good the model is in detecting positive patterns.

$$Sensitivity = \frac{Number\ of\ True\ Positives}{(Number\ of\ True\ Positives + Number\ of\ False\ Negative)} \quad (4.5)$$

The third metric, F-Measure (also called F-Score), defined as shown in Equation (4.6), is the harmonic mean between Precision and Sensitivity.

$$F\text{–}Measure = \frac{2 * Precision * Sensitivity}{(Precision + Sensitivity)} \quad (4.6)$$

The harmonic mean of two numbers tends to be closer to the smaller one; hence higher F-Measure values imply more balanced Precision and Sensitivity values. According to the literature, in the case of extremely imbalanced datasets, the Sensitivity value, i.e., the number of positive patterns correctly identified, is often very low. In practice, it means that rare cases are usually hard to identify.

In all the experiments, the **CSAFunctions** were designed as Neural Networks Committee Machines (CM) composed of 11 (eleven) individuals ANN models constructed as three layered perceptron. T

The number of 11 individuals ANN models to compose the CM was arbitrarily set; however the choice of the CM formulation to implement the **CSAFunctions** was based on the ANN and CM improved prediction quality (BASAWARAJ; SUBHASH; CHANDRASEKHAR, 2012), strong robustness (DIETTERICH, 2000), and generalization potential (ZHOU; WU; TANG, 2002) attested by the literature. Nonetheless, any kind of model for pattern recognition or classification, such as K-Nearest Neighbors or Support Vector Machine (MATSUMOTO; DEL-MORAL-HERNANDEZ, 2014), could be used to implement the **CSAFunctions**.

Notwithstanding the class of the model chosen to implement the **CSAFunctions**, as considered afore, the model overfitting and the classification of imbalanced datasets are two important pattern recognition problems that are present in the essence of the **CSAFunction** design.

Moreover, for cases in which imbalanced datasets occurrences are detected in the **CSFlag** variables, to reduce the bias toward negative cases (the majority class) and improve the precision with imbalanced datasets, in the CM modeling process, we applied two techniques recommended by the literature (SUN et al., 2009):

- Adapted Bootstrap Sampling: For the cross-validation process, we adapted the bootstrap sampling to produce subsets with the same, or similar as

possible, imbalanced proportion of the full training dataset (NGUYEN; BOUZERDOUM; PHUNG, 2009).

- Cost-sensitive Learning: We replaced the standard MSE performance function by a weighted MSE function that assigned a penalty value for false-negative cases (ZADROZNY; LANGFORD; ABE, 2003).

These techniques were adopted to calibrate the parameters of the ANN models (weights and synapses values), and this process was executed in two steps following the procedure presented in our previous article (MATSUMOTO; DEL-MORAL-HERNANDEZ, 2013), and described below:

1. **False-negative penalty value definition**: It consists in training the model several times with different penalty values ($\boldsymbol{fnpenalty}$) for false-negative error cases, and choose the "best" one by applying the cross-validation technique using the Precision metric for performance comparison. To define the range of the $\boldsymbol{fnpenalty}$ values to be tested, we used the proportion of positive cases in the Training dataset ($\boldsymbol{prop}$), as shown in Equation (4.7):

$$\boldsymbol{prop} = \frac{Number\ of\ Positive\ Cases}{Total\ Number\ of\ Cases}$$
$$\boldsymbol{maxpenalty} = \left(\frac{1}{\boldsymbol{prop}} - 1\right)$$
$$\boldsymbol{fnpenalty} \in [1, \boldsymbol{maxpenalty}] \qquad (4.7)$$

The interval is defined between the initial value 1, which means no penalty, and the maximum penalty defined as the inverse of the positive cases proportion minus 1, the weight that compensates the imbalanced proportion between the two data patterns.

In our study presented in 2014 (MATSUMOTO; DEL-MORAL-HERNANDEZ, 2014), for the performance comparison in the cross-validation tests, we replaced the Precision metric by the F-Measure metric, that allowed us to take the Sensitivity metric into account. This change can be recommended when it is possible tolerating more incorrect indications (more false positives, i.e., lower Precision) for the sake of more positive correct identifications (more true positives, i.e., higher Sensitivity).

2. **Final Model Training**: It is the final calibration of the parameters of the model, using the "best' $\boldsymbol{fnpenalty}$ value defined in Step 1.

The individuals ANN models were built using architecture similar to the one described in Item 4.2.2 to construct the ANN regression models, but using different numbers of neurons in the hidden layer. To define this number, we tested five different numbers starting with the number of neurons in the hidden layer equal to the numbers of input parameters, and then progressively increasing it to twice, three, four, and five times the original number. The choice of the best configuration was made using the cross-validation technique.

Alternatively, instead of arbitrating the number of neurons in the hidden layer and compare different ANN architectures, another option could be the adoption of training functions based on the Bayesian regularization methods which can be used to automatically define the number of the neurons in the hidden layer, according to the research developed by MacKay (MACKAY, 1992), and detailed by Foresee and Hagan (FORESEE; HAGAN, 1997).

Bayesian regularization is the name given to the neural network training algorithm in which the optimization process minimizes the error and the number of parameters (weight and bias) of the ANN. This is done adapting the performance function by adding a component that penalizes the performance value with the increase of the number of parameters.

In the experiments, the output of the individual ANN model was conventionally defined to be equal to 1, in the case of critical scenario patter identification, and 0 otherwise. The final outcome of the **CSAFunction** composed of 11 (eleven) individual ANN models was then defined by simple majority voting, i.e., if 6 (six) or more individual outcomes were equal to 1 then the **CSAFunction** outcome was set to 1 (on/positive); otherwise it was set to 0 (off/negative).

Next, we present a summary scheme describing how the **CSFlag** values predicted by the **CSAFunctions** can be used to adjust the computed individual regression predictions. We also describe the metrics and tests employed in the experiments to measure the effect of the adjustment procedure on the original regression prediction error for the testing dataset.

4.5. Regression Predictions Adjustment Procedure Using the CSFlag

The steps listed in Section 3.2 are schematically represented in Figure 13, which displays the process to adjust the individual regression predictions for the testing datasets.

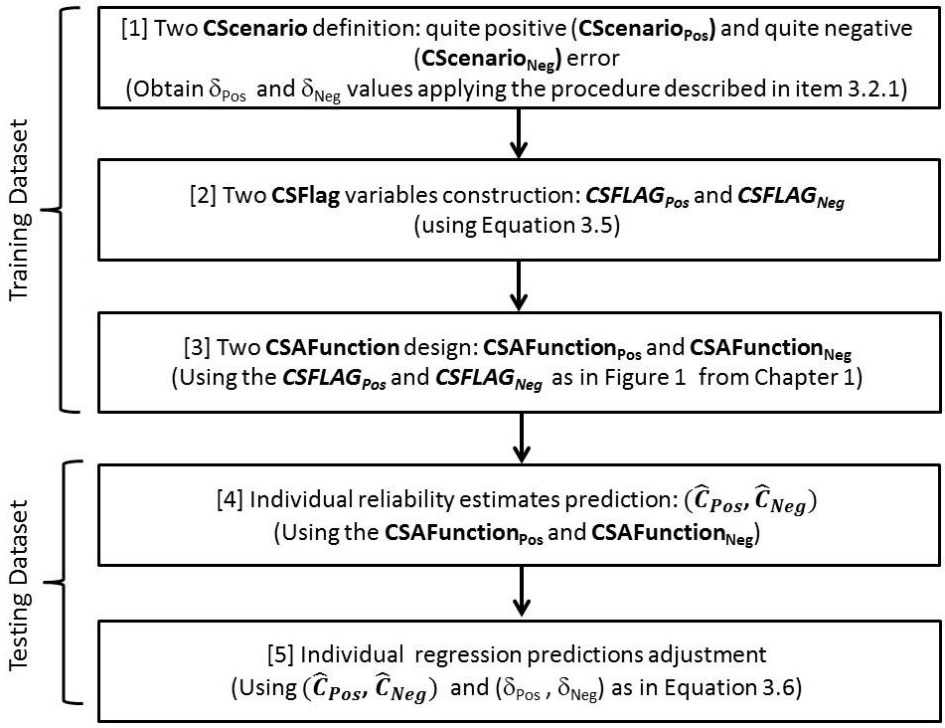

Figure 13: Summary scheme of the proposed methodology.

The observations in the training dataset are used to define the two Critical Error Scenario conditions, **CScenario$_{Pos}$** and **CScenario$_{Neg}$**, and their respective threshold values, $\boldsymbol{\delta_{Pos}}$ and $\boldsymbol{\delta_{Neg}}$. These threshold values ($\boldsymbol{\delta_{Pos}}$ and $\boldsymbol{\delta_{Neg}}$) are then used to construct the **CSFlag** vector variables for the training dataset, $\boldsymbol{CSFlag_{Pos}}$ and $\boldsymbol{CSFlag_{Neg}}$, which are employed to build the two **CSAFunctions, CSAFunction$_{Pos}$** and **CSAFunction$_{Neg}$**.

The outputs of these two **CSAFunctions** for the testing dataset, i.e., the **CSFlags** variables, $\hat{\boldsymbol{c}}_{\boldsymbol{Pos}}$ and $\hat{\boldsymbol{c}}_{\boldsymbol{Neg}}$, are used to adjust the original regression prediction for the testing dataset according to the procedure described by Equation (3.6) from Section 3.2.

In the case of perfectly separable situations for the testing dataset, the reliability estimates, $\hat{\boldsymbol{c}}_{\boldsymbol{Pos}}$ and $\hat{\boldsymbol{c}}_{\boldsymbol{Neg}}$, would properly discriminate all critical cases from non-

critical, and the RMSE value obtained after the application of the adjustment procedure ($\boldsymbol{rmse_{Adjusted}}$) would be always smaller than the RMSE value produced by the original regression prediction ($\boldsymbol{rmse_{Original}}$).

As remarked in Section 3.1, however, as in any Machine Learning strategy, the perfect recognition of the critical scenario condition cannot be assured by the proposed methodology. For this reason, we adopted the average error metric, RMSE (root mean squared error), to evaluate the improvement generated by the methodology in the experiments.

We compared the RMSE values produced by the regression predictions for the testing datasets before ($\boldsymbol{rmse_{Original}}$) and after the adjustment procedure execution ($\boldsymbol{rmse_{Adjusted}}$) by calculating the RMSE percentage of improvement as defined in Equation (4.8). Notice that an original RMSE value ($\boldsymbol{rmse_{Original}}$)) equal to zero cannot be improved; hence, under this condition, Equation (4.8) is not needed.

$$\begin{aligned} \boldsymbol{rmse_{Original}} &= rmse\left(Y_{Tes} - \hat{Y}_{Tes}\right) \\ \boldsymbol{rmse_{Adjusted}} &= rmse\left(Y_{Tes} - \hat{Y}_{Adj}\right) \\ \boldsymbol{rmse_{Improv}} = 100 * \frac{\left(\boldsymbol{rmse_{Original}} - \boldsymbol{rmse_{Adjusted}}\right)}{\boldsymbol{rmse_{Original}}} &; for\ \boldsymbol{rmse_{Original}} \neq 0 \end{aligned} \tag{4.8}$$

In the experiments, we created three vector variables for improvement evaluation purposes: $\boldsymbol{RMSE_{Original}}$, a vector composed of the $\boldsymbol{rmse_{Original}}$ values; $\boldsymbol{RMSE_{Adjusted}}$, a vector composed of the $\boldsymbol{rmse_{Adjusted}}$ values; and $\boldsymbol{RMSE_{Improv}}$, a vector composed of the $\boldsymbol{rmse_{Improv}}$ values.

In the case of the Synthetic data, we worked with predictions produced by Linear regression models, and we ran 300 experiments using data with different levels of noise, and examined the correlation between the $\boldsymbol{rmse_{Improv}}$ values produced by the methodology and the level of noise in the datasets. For the cross section data from the eight UCI databases, to verify the response of the methodology, for each database, we performed 100 experiments using 100 different working datasets. We followed the same steps to improve the predictions produced by Linear and by ANN regression models. Hence, we performed 200 experiments for each of the eight UCI databases, which total 1600 experiments.

The improvement was evaluated by applying two statistical hypothesis tests (LEHMANN; ROMANO, 2007) using the three vector variables afore mentioned:

- 2-Sample hypothesis test: we used this test decision for the null hypothesis that the values in $\boldsymbol{RMSE_{Original}}$ and $\boldsymbol{RMSE_{Adjusted}}$ come from independent random samples from normal distributions with equal means, with 5% significance level.
- One-Sided (right tail) hypothesis test: we used this test decision for the null hypothesis that the values in $\boldsymbol{RMSE_{Improv}}$ come from a population with normal distribution with mean equal to 0 (zero), with 5% significance level.

As distinctive improvement evidence, we would expect to have both null hypothesis tests rejected:

- The 2-Sample null hypothesis rejected in favor of the alternate hypothesis that the values in $\boldsymbol{RMSE_{Original}}$ and $\boldsymbol{RMSE_{Adjusted}}$ come from populations with unequal means.
- The One-Side (right tail) null hypothesis rejected in favor of the alternate hypothesis that the values in $\boldsymbol{RMSE_{Improv}}$ come from a population with a mean greater than 0 (zero).

The 2-Sample and One-Sided hypothesis tests were also used to compare the outcomes of the methodology applied to the Linear and to the ANN regression models.

For the ISO-NE data, differently from the case with UCI data, we did not work with resampled datasets, because in time series data the observations are usually correlated, making it necessary to preserve the chronological sequence of the data (ENDERS, 2014a). In this case, we performed only eight experiments, one for each database from the eight ISO-NE weather stations, and we compared the eight pairs of RMSE values before and after the application of the adjustment procedure: $(\boldsymbol{rmse_{Original}}, \boldsymbol{rmse_{Adjusted}})$.

Next, we analyze and discuss the outcomes obtained with the execution of the procedures described in this chapter performed using the working datasets described in Section 4.1.

5. NUMERICAL RESULTS – ANALYSIS AND DISCUSSION

In this chapter, we present and analyze the numerical outcomes of the experiments carried out according to the description detailed in Chapter 4.

For the experiments with Synthetic working datasets, to better monitor the correlation between the outcomes of the models and the level of noise in the data, the results are presented in graphic format having the 300 noise factor values ($\boldsymbol{nf}$) values, on the X axis, and the metrics of interest on the Y axis.

For the UCI Databases, to inspect the consistency of the methodology, the results are displayed in tables with the information aggregated per group of experiments for each of the eight UCI Databases. Each group of experiments consists of a collection of 100 experiments repeated using 100 different data samples (working datasets) from the same database. In the cells tables with two values, the first one is the average value of the metric of interest for the 100 experiments in the group; and the second, displayed between squared brackets, is the standard deviation value.

In the case of the ISO-NE data, the tables display the final outcomes obtained from experiments with the eight working datasets, one for each of the eight weather stations of ISO-NE control area.

To avoid excessive interruptions in the text with the insertion of too many tables with numerical information, the tables with intermediate outcomes of the experiments, such as the performance metrics of the regression models, or the threshold values of the **CScenarios**, are displayed in Appendix B – Supplementary Tables.

Regarding the numerical computational implementation developed in this research, all the models and procedures were implemented in MATLAB (R2015a) using functions from Statistics & Machine Learning Toolbox, Optimization Toolbox, Global Optimization Toolbox, and Neural Networks Toolbox (MATHWORKS, 2015).

To solve the optimization problem designed to determine the threshold values of the **CScenario**s, as described in Section 4.3, to better handle the local minima problem obstacle, the optimization algorithms were implemented with a hybrid

scheme to optimize the objective functions using genetic algorithm and gradient-descendent methods (GOLDBERG, 1989).

All **CSAFunctions** were constructed following the design and the procedure described in Section 4.4. Although pattern recognition is not the main focus of this work, the **CSAFunctions** were built up to achieve good performance, but they were not ultimately optimized.

5.1. Outcomes with the Synthetic Working Datasets

For each of the 300 synthetic working datasets generated as described in Item 4.1.1, one single linear regression model was built as described in Item 4.2.1.

As expected, the performance of the regression models decreases for higher noise factor values. This can be monitored in Figure 14, which displays the RMSE values generated by the predictions of each of the 300 regression models for the Regression working datasets. The values gradually rise and spread out keeping pace with the noise increase.

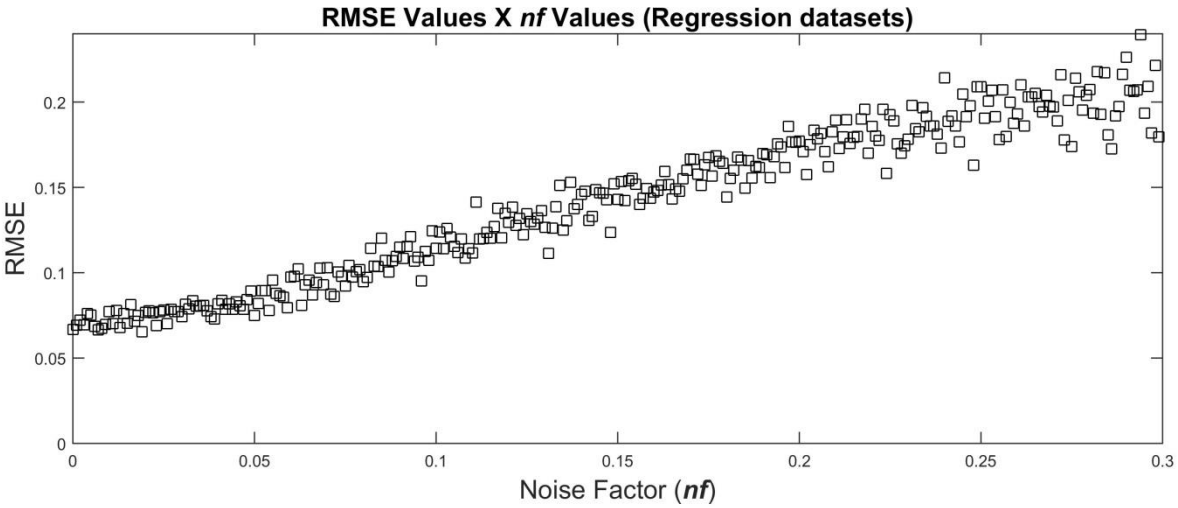

Figure 14: (Synthetic) RMSE values of the Regression datasets generated by different noise factor values (level of noise).

The RMSE values generated by the regressions predictions with the Training and the Testing datasets followed the same behavior. The RMSE values are in Appendix B – Table B11, which displays the average, standard deviation, minimum, and

maximum values of the RMSE generated with the Regression, the Training, and the Testing datasets of the Synthetic Datasets Group.

5.1.1. Critical Error Scenario Threshold Values Definition

For each of the 300 synthetic working datasets, we applied the procedure described in Section 4.3 to define the "best" threshold values of the **CScenario$_{Pos}$** and the **CScenario$_{Neg}$**, $\boldsymbol{\delta_{Pos}}$ and $\boldsymbol{\delta_{Neg}}$, i. e., the threshold values that minimize the RMSE value of the regression predictions for the Training dataset after applying the adjustment procedure expressed in Equation (3.6) from Section 3.2.

The average of the threshold values determined for the critical scenario of quite positive error (**CScenario$_{Pos}$)** was 0.152. In the case of quite negative error (**CScenario$_{Neg}$**), the value was -0.137. Next, these threshold values were used to construct the **CSFlag** variables; and in terms of percentage of positive cases, the numbers varied between 5% and 44%.

Detailed information about the threshold values of the critical scenarios obtained in the case of Synthetic Dataset Groups is in Appendix B – Tables B12 and B13.

5.1.2. Critical Error Scenario Alert Functions' Outcomes

Since the majority of the percentages of positive cases were below 44%, as mentioned in Section 4.4, the **CSAFunctions** modeling was handled as an imbalanced data classification problem, and, as described in Section 4.4, the cost-sensitive learning technique was applied to solve it.

Since the F-Measure metric combines the two other metrics, Precision and Sensitivity, to empirically monitor the performance of the **CSAFunctions**, we examined the F-Measure values produced by the **CSAFunction$_{Pos}$** and the **CSAFunction$_{Neg}$** against the level of noise.

This information is represented in Figures 15 and 16, which respectively display the F-Measure values produced by the **CSAFunctions** with the Training and with the Testing datasets. In the graphs, the F-Measure values of the **CSAFunction$_{Pos}$** are represented by upward-pointing black triangle markers with a trend curve plotted with dashed line; and the F-Measure values of the **CSAFunction$_{Neg}$** are represented by

downward-pointing white triangle markers with a trend curve plotted with dotted line. The same class of information for the Precision and the Sensitivity metric values are depicted in the four graphs in Appendix A (A01, A02, A03, A04).

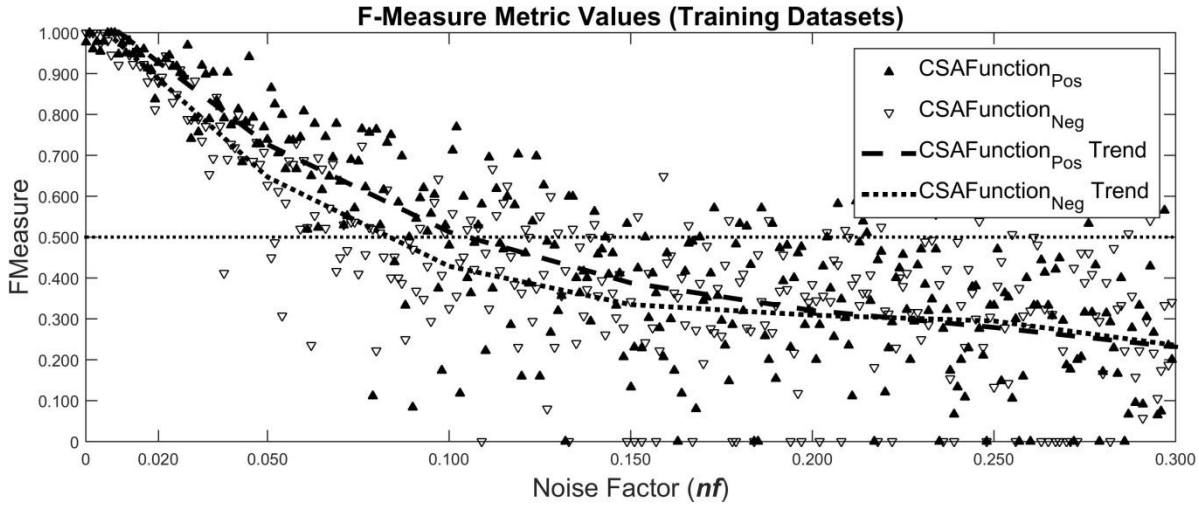

Figure 15: (Synthetic) F-Measure X Noise Factor (Training datasets).

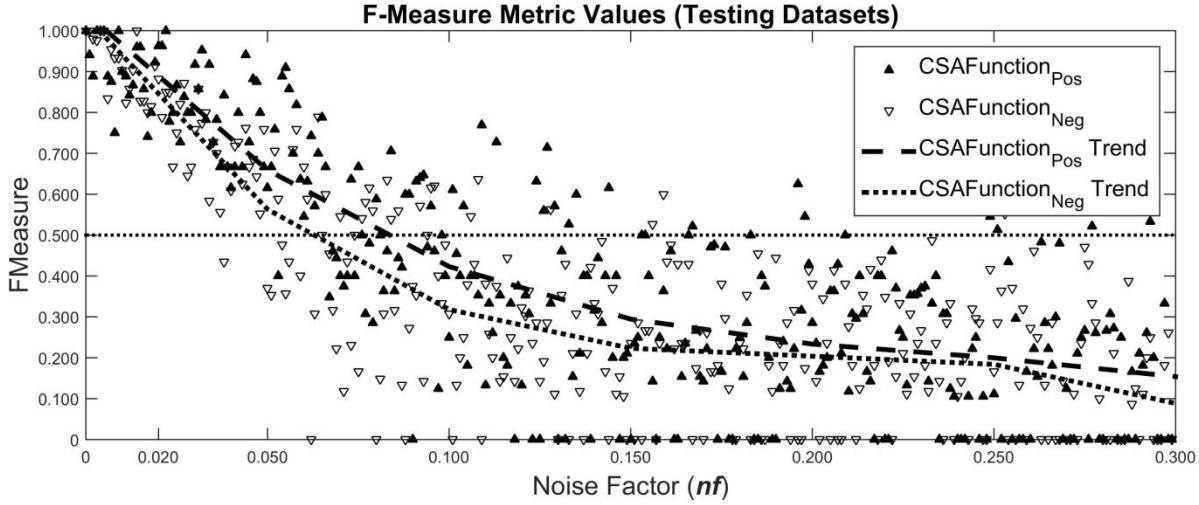

Figure 16: (Synthetic) F-Measure X Noise Factor (Testing datasets).

Figure 15 shows the outcomes of the **CSAFunctions** obtained with the Training datasets. The F-Measure values are verified to have a trend to decline as the level of noise goes up, starting from F-Measure values closer to 1.0, which shows almost perfect discrimination for lower values of $\boldsymbol{nf}$; and spreading out around F-Measure values lower than 0.50, which indicates very low performance for higher values of $\boldsymbol{nf}$.

Figure 16 shows the F-Measure values generated by the **CSAFunctions** with the Testing datasets. In this case, the F-Measure values show trend curves similar to the

ones obtained with the Training datasets, but the values start to spread out for much lower values of $\boldsymbol{nf}$, which indicates that for the Testing datasets, as expected, the **CSAFunctions** achieved lower performance in comparison to the outcomes with the Training datasets.

5.1.3. Regression Prediction Values Improvement Using the CSFlag

The adjustment procedure described in Section 3.2 was applied to all 300 synthetic working datasets to adjust the regression prediction values for the unknown data in the Testing datasets.

Once again, to empirically examine the effect of the noise on the regression prediction improvement produced by the proposed methodology, we created two graphs shown in Figures 17 and 18.

Figure 17 displays the RMSE values generated by the original regression prediction values (indicated by the white square markers), and the RMSE values generated by the regression prediction values adjusted by the proposed methodology (indicated by the black circle markers) for each of the 300 experiments, starting with $\boldsymbol{nf}$ equal to zero (clean data) up to 0.299.

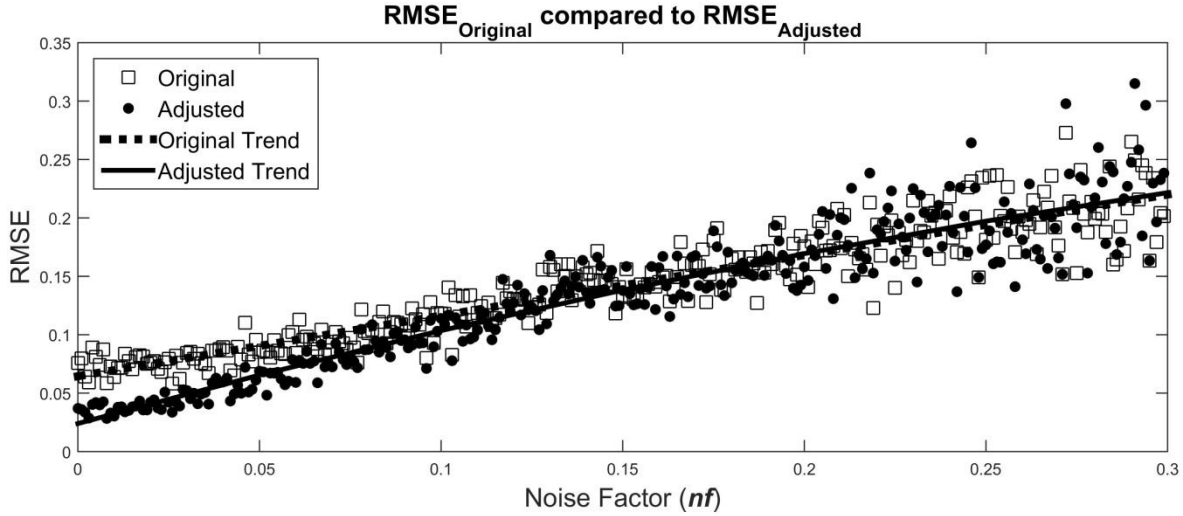

Figure 17: (Synthetic) RMSE ($_{\text{Original, Adjusted}}$) X Noise Factor (Testing datasets).

Note that the black markers start below the white markers for lower values of $\boldsymbol{nf}$, and the two markers merge with the increase of the values of $\boldsymbol{nf}$, as indicated by the trend lines. This can be interpreted as an indication that, for this set of experiments,

the proposed adjustment procedure better performed with data with low level of noise and lost efficiency for data with higher level of noise.

This statement was statistically confirmed by the application of the two statistical hypothesis tests described in Section 4.5 considering two different subsets of experiments:

- First half subset (lower noise): composed of the first 150 experiments (values of $\boldsymbol{nf}$ from 0 to 0.150);
- Second half subset (higher noise): composed of the last 150 experiments (values of $\boldsymbol{nf}$ from 0.151 to 0.300).

According to the outcomes of the tests displayed in Table 7, the two null hypothesis were rejected in the case of the First half subset which means that for the experiments performed with lower level of noise, in average, the RMSE original values were higher than the adjusted RMSE values, and the percentages of improvement were positive. On the other hand, the outcome for the Second half subset indicated the opposite, i.e., that it was not possible to reject neither the hypothesis that the two samples came from samples with equal means nor the hypothesis that the percentage of improvement average was positive.

Table 7: (Synthetic) Hypothesis tests outcomes.

Synthetic Experiments Subset	**Two-Sample Hypothesis**		**One-Sided (right tail) Hypothesis**	
	Null hypothesis Equal means	**P-Value**	**Null hypothesis Mean equal to zero**	**P-Value**
First half (lower noise)	Rejected	0.000	Rejected	0.000
Second half (higher noise)	Not rejected	0.912	Not rejected	0.647

Figure 18 displays the RMSE percentage of improvement calculated as defined in Equation (4.8) from Section 4.5, once more, against the level of noise numerically represented by the noise factor value ($\boldsymbol{nf}$). In this case, the RMSE percentage of improvement values can be seen to be significantly higher (starting with values close to 60%) for lower $\boldsymbol{nf}$ values and to go down as the $\boldsymbol{nf}$ values increase.

The variation of the percentages of improvement reflects the outcomes of the **CSAFunctions** shown in Figure 16 (Item 5.1.2). For lower values of $\boldsymbol{nf}$, the high values achieved by the F-Measure metric indicate almost perfect discrimination, and

consequently higher percentages of improvement. As the $\boldsymbol{nf}$ value goes up, the percentage of improvement values start oscillating and spreading out around zero, because the F-Measure metric values generated by the **CSAFunctions** went down to values lower than 0.500.

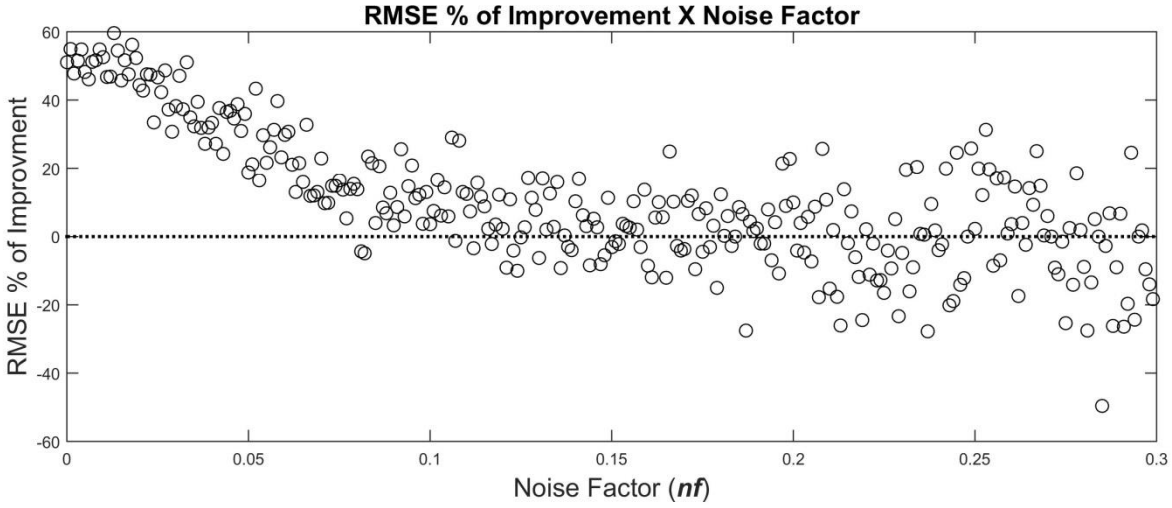

Figure 18: (Synthetic) RMSE % of improvement X Noise Factor (Testing datasets).

These two extreme situations are illustrated in the next four graphs, which depict the **CSAFunctions** outcomes and the effect of the adjustment procedure on the regression prediction errors for the experience with $\boldsymbol{nf}$ equal to 0.005, representing the case of almost clean data (Figures 19 and 20); and for the experience with $\boldsymbol{nf}$ equal to 0.295, representing the case of highly noisy data (Figures 21 and 22).

In these four figures (19, 20, 21, and 22), axis X is the index of the observations, and axis Y shows the regression prediction errors instead of the values of the output variable and the regression predictions as in the figures used to explain the concept of the methodology in Chapter 3. The representation having the prediction error values of the observations in axis Y is more compact, allows a clearer depicting the critical scenarios conditions, and can be used in the case of multiple regressions.

Again, in these four figures, the original regression prediction errors for the Testing dataset are represented by solid vertical lines; the observations for which the **CSAFunction$_{\mathbf{Pos}}$** outcomes are equal to 1 are indicated by upward-pointing black triangle markers; the observations for which the **CSAFunction$_{\mathbf{Neg}}$** outcomes are equal to 1 are indicated by downward-pointing white triangle markers; and the critical

scenario limits, i.e., the threshold values of the **CScenario$_{Pos}$** ($\boldsymbol{\delta_{Pos}}$) and **CScenario$_{Neg}$** ($\boldsymbol{\delta_{Neg}}$) are both represented by horizontal dotted lines. Figure 20 displays the regression prediction errors for the Testing dataset after the execution of the adjustment procedure.

The adjustment procedure consists in adding up a positive value, $\boldsymbol{\delta_{Pos}}$, (the threshold value of the **CScenario$_{Pos}$**) to the original regression prediction in the case of quite positive errors to reduce positive errors; and adding up a negative value, $\boldsymbol{\delta_{Neg}}$, (the threshold value of the **CScenario$_{Neg}$**) in the case of quite negative errors to reduce negative errors.

All four graphs, from Figure 19 to 22, have the same legend markers, but they are displayed only in Figures 20 and 22 to avoid data visualization blocking in Figures 19 and 21.

Figures 19 and 20 illustrate the experience with almost clean data (with $\boldsymbol{nf}$ equal to 0.005). Figure 19 shows an example of perfect discrimination of the critical scenario cases that generated all performance metrics values equal to 1 (Precision, Sensitivity and F-Measure). Figure 20 graphically represents the reduction of the regression prediction errors for the Testing dataset produced by the application of the adjustment procedure. Numerically, the RMSE value reduction in the Testing dataset was from 0.091 to 0.044, which means a percentage of improvement equal to 52.3%.

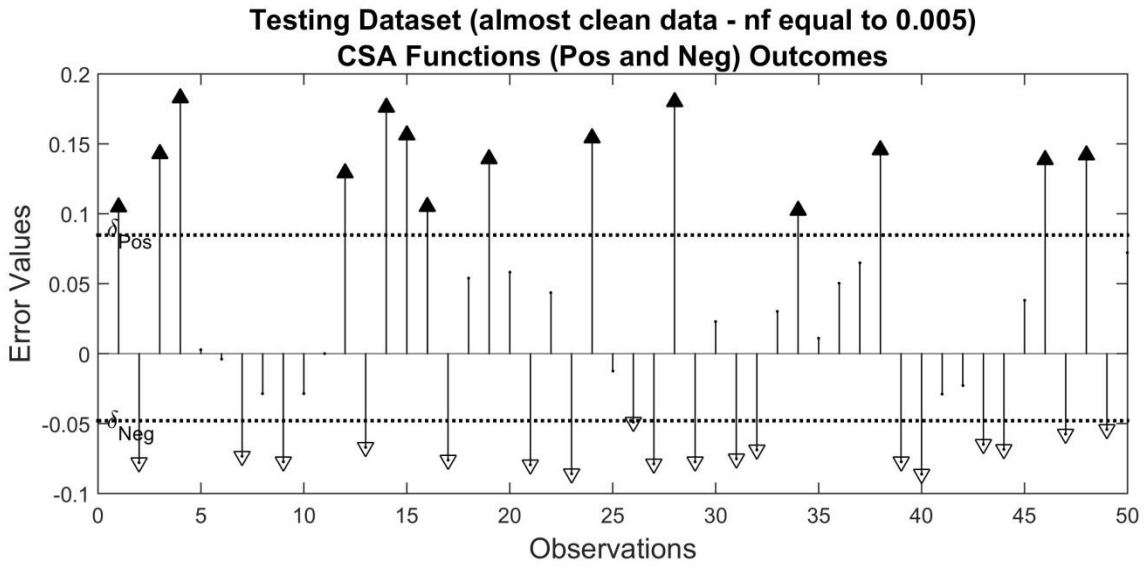

Figure 19: (Synthetic) **CSAFunctions'** outcomes for almost clean data.

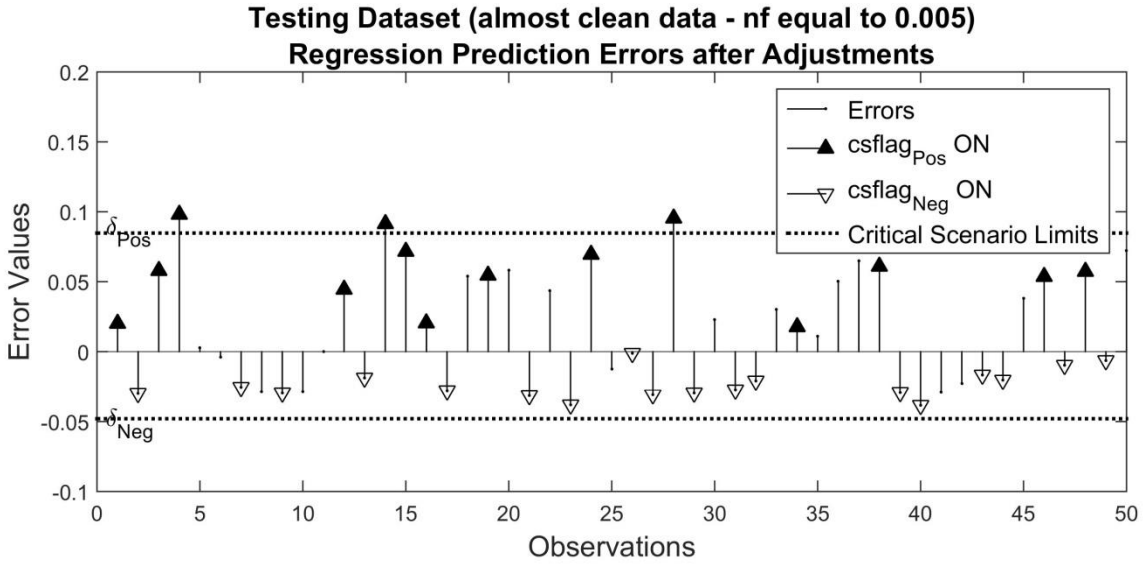

Figure 20: (Synthetic) Adjustment effect on the prediction errors for almost clean data.

On the other hand, the graphs in Figures 21 and 22 display an example of experiment with highly noisy data (produced by $\boldsymbol{nf}$ equal to 0.295). Figure 21 graphically represents the outcomes of the **CSAFunction$_{\mathbf{Pos}}$** and the **CSAFunction$_{\mathbf{Neg}}$** for the Testing dataset of an experiment that produced all performance metric values below 0.220.

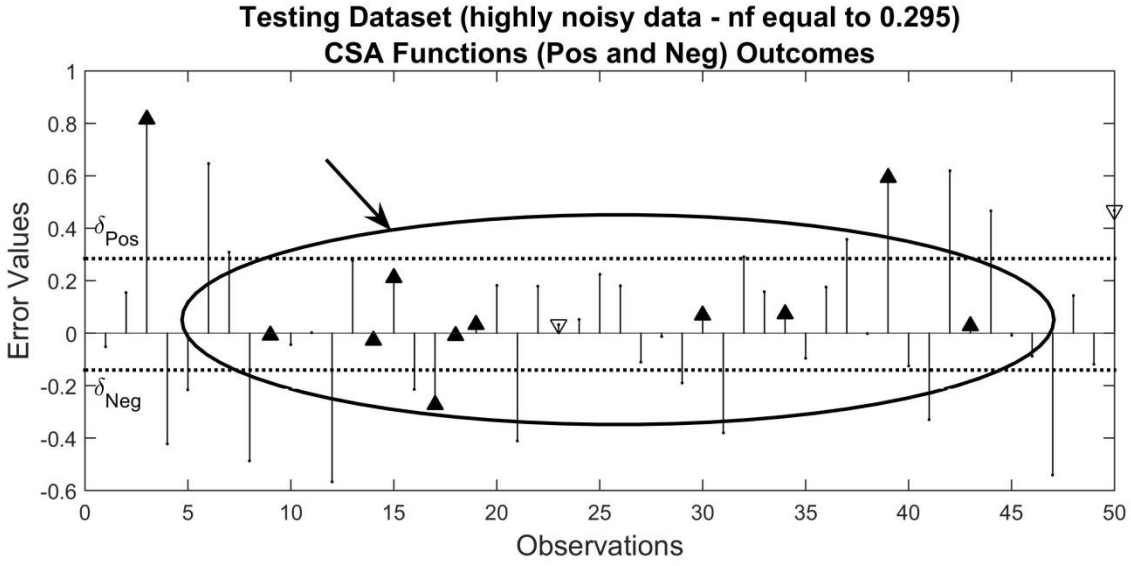

Figure 21: (Synthetic) **CSAFunctions'** outcomes for highly noisy data.

The arrow and the ellipse in Figure 21 indicate the occurrences of false-positives. In this specific case, the **CSAFunction$_{\mathbf{Pos}}$** produced a Precision metric value equal to 0.182. As a consequence, the adjustment procedure increased the individual prediction error values instead of decrease them, as indicated by the arrow and the

ellipse in Figure 22. The RMSE value increase in the Testing dataset was from 0.306 to 0.313, a percentage of improvement equal to -2.3% (a worsening by 2.3%); in other word, our methodology failed in the case of data with high level of noise.

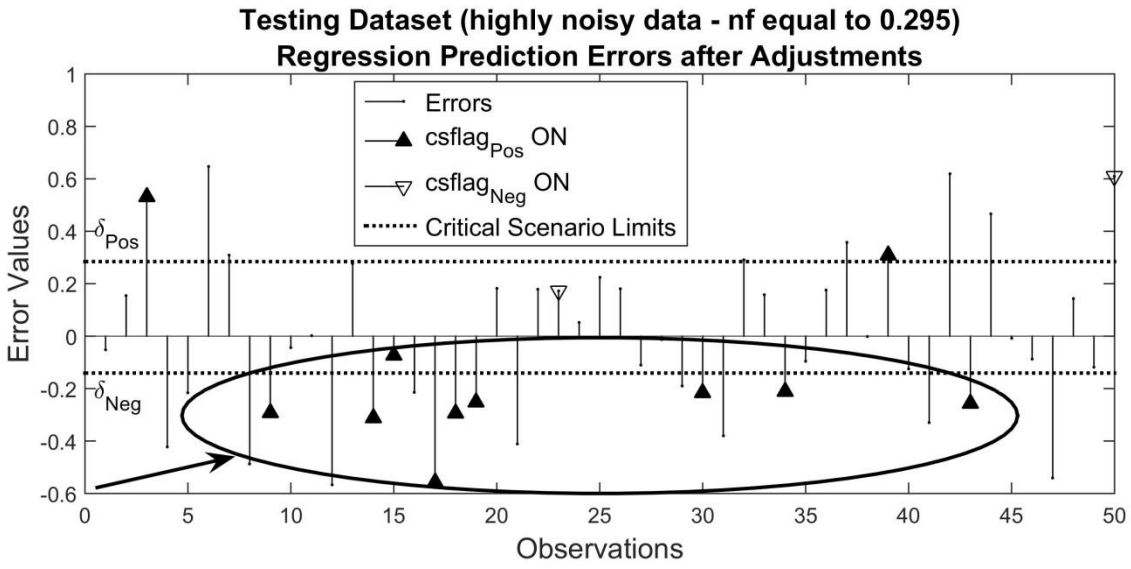

Figure 22: (Synthetic) Adjustment effect on the prediction errors for highly noisy data

In the next four figures (23, 24, 25, and 26), we show the effect of the adjustment procedure on the regression predictions of the output variable for the Testing dataset. In these four graphs, in axis Y, the actual observed values, $\boldsymbol{Y_{Tes}}$, in the Testing dataset are represented by the star markers; the original Linear regression prediction values, $\widehat{\boldsymbol{Y}}_{\boldsymbol{Tes}}$, by the white square markers; and the adjusted prediction values, $\widehat{\boldsymbol{Y}}_{\boldsymbol{Adj}}$, by the black circle markers.

In Figures 23 and 24, we show the outcome for the experience with almost clean data (with $\boldsymbol{nf}$ equal to 0.005). In Figure 23, to help better discern the effect of the adjustment procedure on individual observations, in axis X, we are using the indices of the 50 values of the X variable in ascending order. To allow the recognition of the parabolic shape of the Y variable, in axis X of Figure 24, we are using just the values of X variable.

The same set of information obtained with the experiments with highly noise data (with $\boldsymbol{nf}$ equal to 0.295) are illustrated in Figures 25 and 26. Once again, in Figure 25, in axis X, we have the index of the X variable in ascending order; and, in Figure 26, in axis X, again, we have the value of X variable.

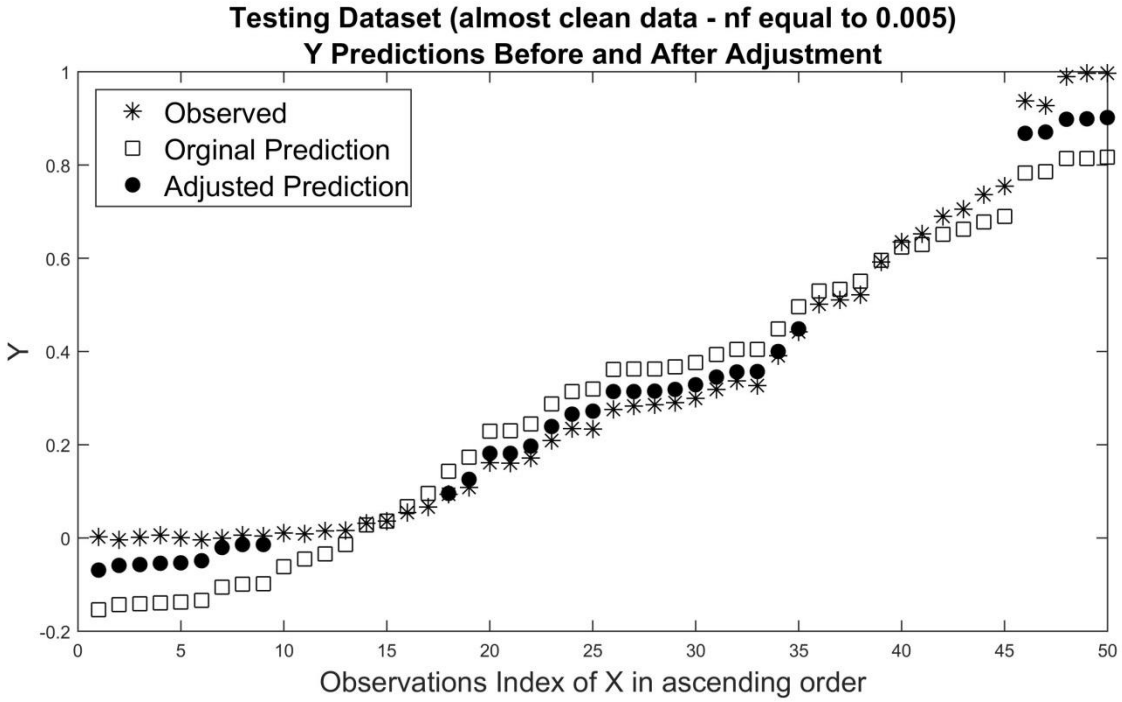

Figure 23: (Synthetic) Adjustment effect on the regression predictions for almost clean data, with the index of X in ascending order in X axis.

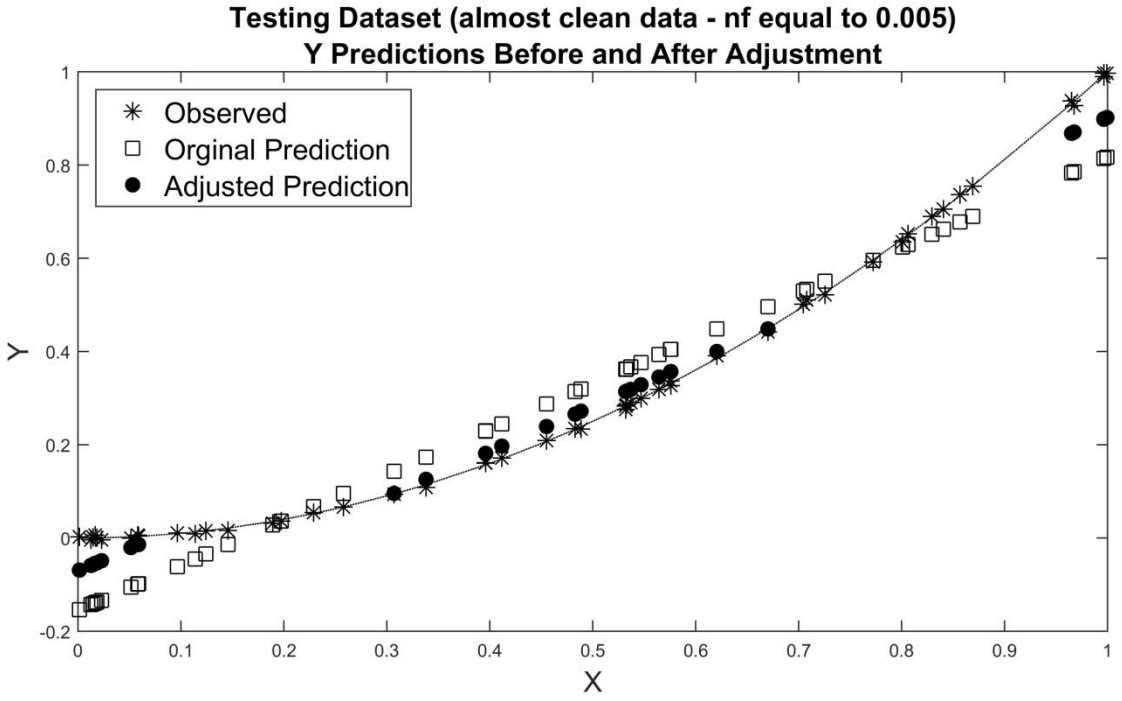

Figure 24: (Synthetic) Adjustment effect on the regression predictions for almost clean data.

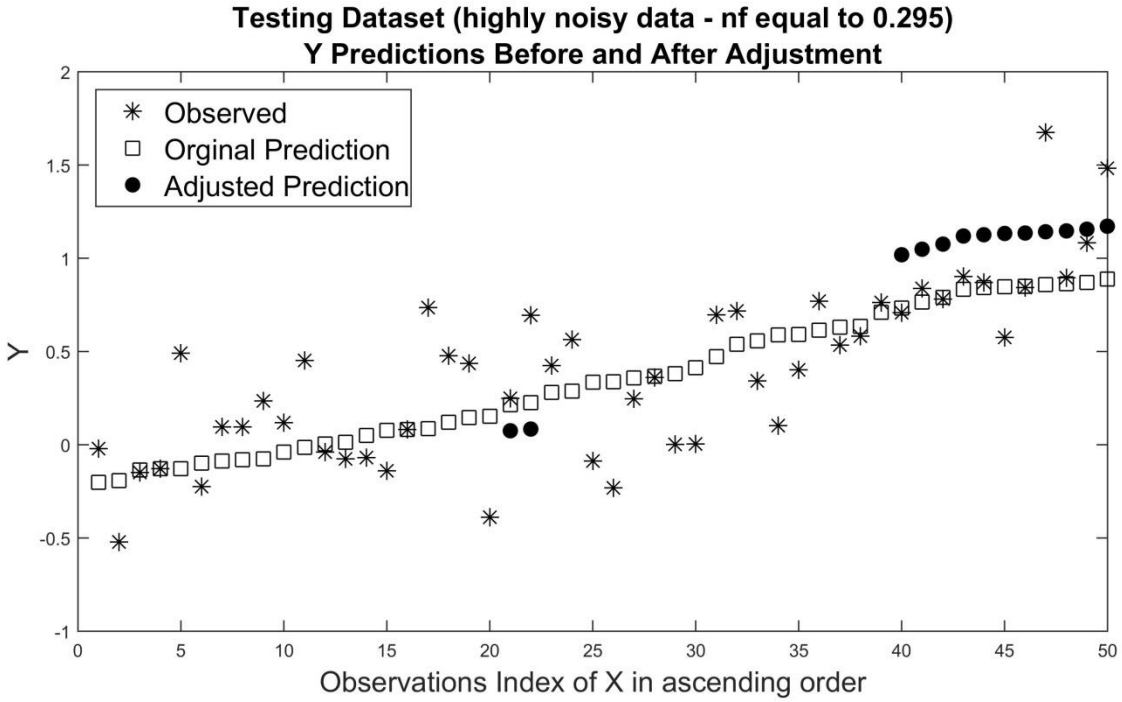

Figure 25: (Synthetic) Adjustment effect on the regression predictions for highly noisy data, with the index of X in ascending order in X axis.

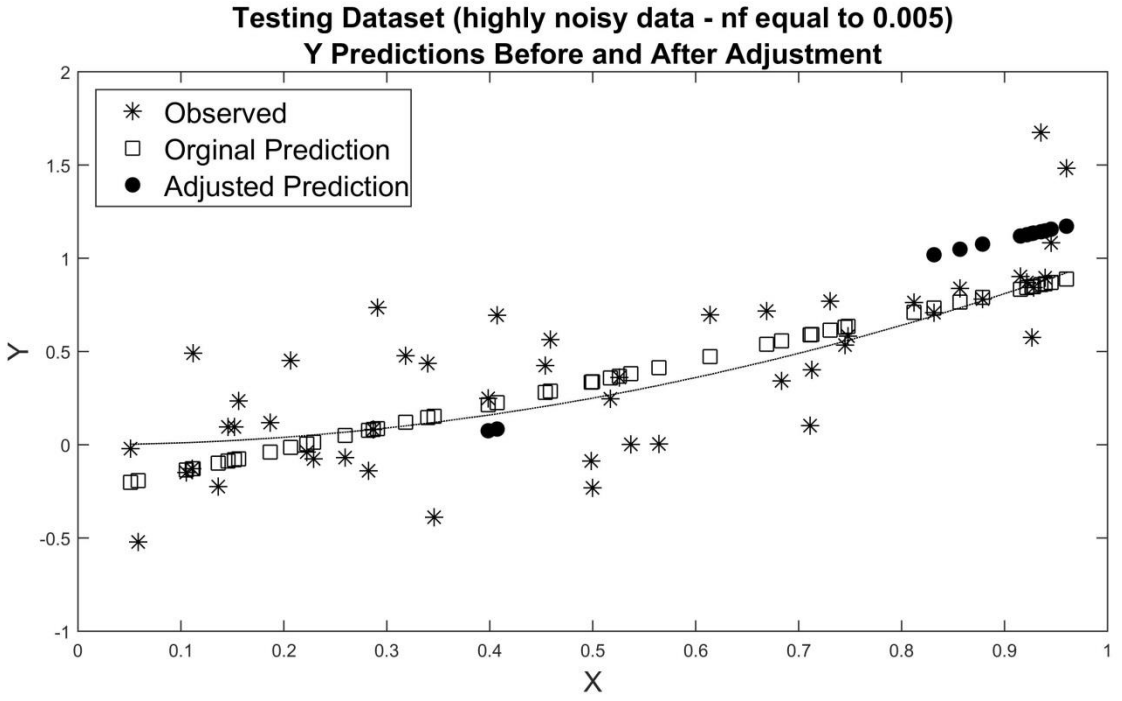

Figure 26: (Synthetic) Adjustment effect on the regression predictions for highly noisy data.

In Figures 23 and 24, for the case of almost clean data, we can observe that the adjusted predictions (black circle markers) are closer to the actual observed values (star markers) than the original regression predictions (white square markers). In Figures 25 and 26, we can see the random effect of the adjustment procedure for the case of highly noisy data.

Next, we detail the outcomes of the experiments performed using real cross sectional data from the UCI Machine Learning Data Repository.

5.2. Outcomes with the UCI Working Datasets

For each experiment of each UCI Dataset group, we calibrated the parameters of a Linear regression model, and an ANN regression model following the procedures described in Item 4.2.2. To calibrate these two classes of models, we used the Regression datasets constructed as the description in Item 4.1.2.

For the experiments with UCI data, regarding features selection, the sets of input parameters were defined using the sequential forward selection (WOOLDRIDGE, 2009c) in which input variables are sequentially added to an initial empty set until the addition of a new variable fails to improve the outcome of the regression model. The models designed with the different sets of input variables were compared using 10 k-folds cross-validation technique (HASTIE; TIBSHIRANI; FRIEDMAN, 2011c).

The summary of features selection process is displayed in Appendix B - Table B14. For each dataset group, this table lists the total number of available input variables and the number of the input variables selected for each type of model.

The average and standard deviation of RMSE values produced by the Linear regression models and the ANN regression models are in Appendix B, and respectively in Table B15 and Table B16. These tables display the outcomes for the Regression, the Training, and the Testing datasets of the UCI Datasets Groups; the ANN regression models seem to have surpassed the Linear regression models in all cases. This is statistically confirmed ahead in Item 5.2.4. Although the experiments in our research were conducted to improve the predictions provided by Linear and ANN regression models, once again, we emphasizes that our methodology can be applied to other kinds of regression models with different formulations.

5.2.1. Critical Error Scenario Threshold Values Definition

For all the experiments with the UCI Datasets Groups, we performed the procedure described in Section 4.3 to define the "best" threshold values for the **CScenario$_{Pos}$** and the **CScenario$_{Neg}$**. These threshold values were then used to create the **CSFlag** variables. The information about the critical scenarios is in Appendix B, from Table B17 to Table B20. These tables display the average and standard deviation values (in square brackets) of the following information produced by the threshold values of the **CScenarios** chosen as the "best" ones:

- α , the "best" probability value given by the empirical cumulative distribution function of the prediction regression error values in the Training dataset;
- δ, the "best" threshold value, which is equal to the error value corresponding to α , the "best" probability;
- Number of positive cases (# of ON) in the Training and the Testing datasets defined by δ;
- Percentage of positive cases (% of ON) in the Training and the Testing datasets.

Tables B17 and B18 display the **CScenario$_{Pos}$** and **CScenario$_{Neg}$** "best" threshold choices information for the Linear regression models; and Tables B19 and B20, for the ANN regression models.

The "best" threshold values in Tables B17 and B18 are observed to be slightly higher than in Tables B19 and B20. This means that the distributions of the errors produced by the Linear regression models have a wider spread than the ones produced by the ANN regression models. It may be interpreted as another indication that the ANN regression models surpassed the Linear models.

5.2.2. Critical Error Scenario Alert Functions' Outcomes

In the tables that display the information about the critical scenarios (Appendix B - Tables B17 to B20), the averages of the percentage of positive cases in the Training datasets (fourth column of the tables) vary from 11.1% to 39.4%, in the case of Linear regression models; and from 9.1% to 21.4%, for ANN regression models. These low percentage values characterize typical cases of the imbalanced data

classification problem, and, once again, the cost-sensitive learning technique was applied to handle it.

In the case of the experiments performed using the data from the UCI Repository, the **CSAFunction**s were constructed according to the description in Section 4.4, using CM (Neural Network Committee Machines) composed of ANN models designed with the number of neurons in the hidden layers equal to twice the number of input parameters. The average and standard deviation values of the $\boldsymbol{fnpenalty}$ computed and applied to the **CSAFunction**$_{\mathbf{Pos}}$ and the **CSAFunction**$_{\mathbf{Neg}}$ are displayed in Appendix B – Tables B21.

For the UCI Datasets Groups, to evaluate the performance of the **CSAFunction**s, we verified the average and standard deviation values of the three performance metrics **Precision**, **Sensitivity**, and **F-Measure**, per Group of experiments.

Table B22 in Appendix B displays the summary of the **CSAFunctions**$_{\mathbf{Pos}}$ metrics for the Linear regression models: the average and standard deviation values of the three performance metrics produced by the Training and by the Testing datasets of each group. The same class of information about the **CSAFunctions**$_{\mathbf{Neg}}$ metrics is in Appendix B - Table B23. The summary for the ANN regression models is in Appendix B - Tables B24 and B25.

As expected, the F-Measure metric average values produced by the UCI Datasets Groups are similar to the ones produced by the Synthetic datasets mildly noisy datasets ($\boldsymbol{nf}$ around 0.05). However, by more closely examining the outcomes from both sets of experiments, in the case of the UCI Datasets Groups, the Sensitivity metric values are verified to be comparatively lower, and the difference between the Training datasets metrics average and the Testing datasets metrics average in the case of the UCI Datasets Groups are comparatively higher than the difference observed in the experiment with Synthetic datasets, especially for the experiments with ANN regression models.

There are several plausible causes of the more drastic performance decrease in the UCI Testing datasets than in the Synthetic Testing datasets. One of them is the nature of the noise in the Synthetic datasets used to model the nondeterministic component in the data, since in the real world the nondeterministic part of the data arguably do not fit the normal distribution, and they are also probably embedded in

the input variables, making the differentiation between randomness and real world data volatility more difficult for the **CSAFunctions**. Another factor is the information provided by the set of the input variables because it may be not enough to fully explain the behavior of the output variables; or maybe there is just not enough data to explain the input and output relationship.

5.2.3. Regression Prediction Values Improvement Using the CSFlag

Once more, we applied the procedure expressed in Equation (3.6) to adjust the regression predictions produced by the Linear and by the ANN regression models with the Testing datasets.

Table 8 displays the average and standard deviation of the following metrics produced by the adjustment procedure for the two classes of regression model (Linear and ANN): the RMSE generated by the original regression predictions, the RMSE generated by the adjusted regression predictions; and the RMSE percentage of improvement.

Table 8: (UCI) RMSE % of improvement.

UCI Datasets Group	Linear Regression Models			ANN Regression Models		
	RMSE Original	RMSE Adjusted	RMSE % of Improvement	RMSE Original	RMSE Adjusted	RMSE % of Improvement
Airfoil	4.813 [0.199]	3.883 [0.206]	19.3% [2.8%]	3.507 [0.437]	3.290 [0.358]	5.9% [3.9%]
AutoMPG	3.514 [0.325]	3.224 [0.343]	8.3% [4.0%]	3.097 [0.389]	2.985 [0.356]	3.4% [5.5%]
Combined Cycle	4.576 [0.116]	4.390 [0.123]	4.1% [0.8%]	4.219 [0.127]	4.194 [0.127]	0.6% [0.4%]
Concrete	10.566 [0.479]	8.599 [0.491]	18.6% [4.3%]	7.094 [1.427]	6.543 [1.161]	7.5% [4.1%]
Energy	1.986 [0.137]	1.753 [0.157]	11.6% [6.7%]	1.852 [0.234]	1.676 [0.208]	9.2% [7.5%]
Housing	4.948 [0.622]	4.397 [0.622]	11.2% [5.4%]	4.494 [1.040]	4.214 [0.930]	5.8% [5.9%]
Parkinsons	9.822 [0.196]	7.716 [0.205]	21.4% [1.3%]	6.898 [0.487]	6.446 [0.472]	6.5% [3.3%]
Yacht	8.932 [0.878]	5.447 [0.943]	39.3% [5.8%]	1.115 [0.369]	0.968 [0.270]	11.1% [11.5%]

Although the average values of the RMSE percentage of improvement are all positive for all cases, this information is not enough to state that the proposed

methodology was able to improve the regression predictions. For this reason, as previously mentioned, we applied the two statistical hypothesis tests described in Section 4.5 to compare the outcomes of the methodology for the two classes of regression models, and the evidence of improvement would be the rejection of the two null hypotheses:

- 2-Sample null hypothesis: the values of two different samples come from normal distributions with equal means.
- One-Sided (right tail) null hypothesis: the sample values come from a population with normal distribution with mean equal to 0 (zero).

Table 9 displays the outcomes of the two hypothesis tests performed to verify the improvement obtained by the application of the proposed methodology in the case of the Linear and the ANN regression models for all the UCI Dataset Groups.

Table 9: (UCI) Linear regression models – Hypothesis tests outcomes

UCI Database Groups	**Linear Regression Models**		**ANN Regression Models**	
	2-Sample (P-value)	**One-Sided (P-value)**	**2-Sample (P-value)**	**One-Sided (P-value)**
Airfoil	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)
AutoMPG	Rejected (0.000)	Rejected (0.000)	Rejected (0.035)	Rejected (0.000)
Combined Cycle	Rejected (0.000)	Rejected (0.000)	**Not rejected (0.169)**	Rejected (0.000)
Concrete	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)
Energy	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)
Housing	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)
Parkinsons	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)
Yacht	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)

The rejection of the two null hypotheses indicates that, statistically, the average of the hundred RMSE values produced by the adjusted regression predictions is different (unequal means) and smaller than the average of the hundred RMSE values

produced by the original regression predictions; hence, it can be considered a consistent demonstration that the proposed methodology was able to improve the original regression prediction (LEHMANN; ROMANO, 2007) . In the case of linear regression models, this supportive condition was achieved by all the UCI Datasets Groups. For ANN regression models, we obtained it in seven out of eight groups.

The only “null hypothesis not rejected” condition happened in the equal means test (2-Sample test) of the Combined Cycle Group for ANN regression models. This indicates that, in this case, the improvement was positive, yet not enough to change the RMSE values on average. This result was anticipated by the low values of the Sensitivity and F-Measure performance metrics produced by the **CSAFunction**s with the Training datasets from the Combined Cycle Group highlighted in Table B24 – Appendix B.

For the Training datasets, the Combined Cycle Group achieved the lowest F-Metric values among the UCI Datasets Groups with reasonable Precision, but extremely low Sensitivity. As a result, for the Testing datasets, positive cases were more correctly identified than misclassified ones, but in numbers that were not enough to improve the average RMSE values.

Next, from Figure 25 to 28, we show the outcome of one of the UCI experiments.

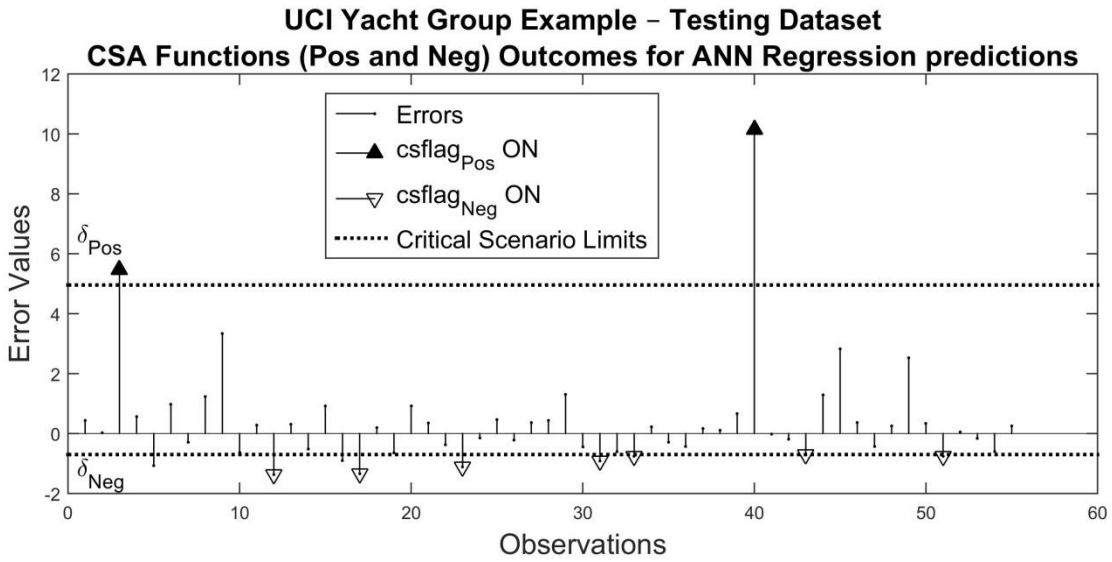

Figure 27: (UCI) **CSAFunctions’** outcomes.

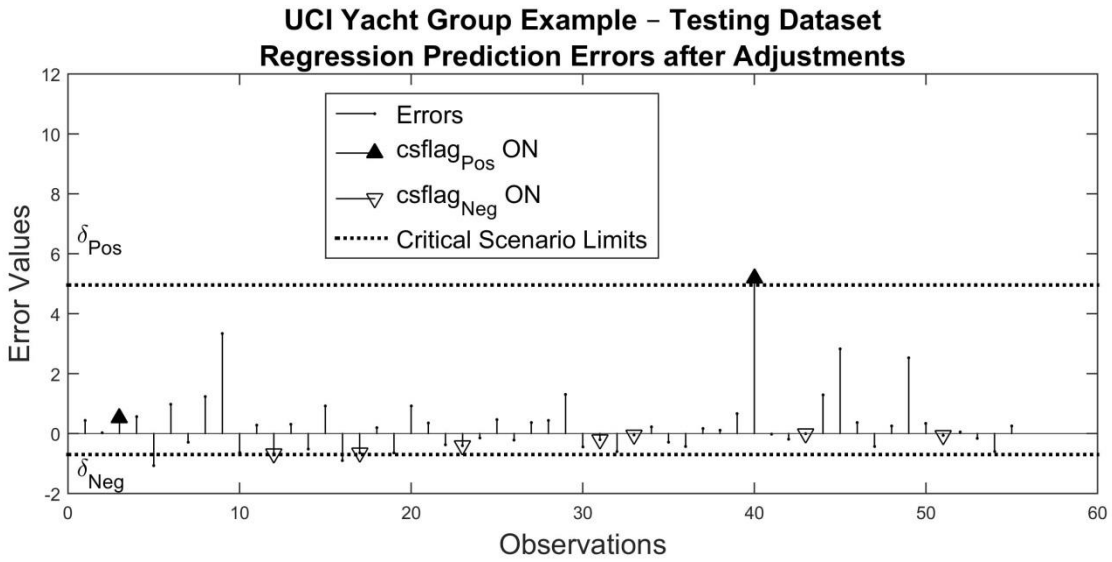

Figure 28: (UCI) Adjustment effect on the prediction errors.

As an example of the adjustment effect on the regression prediction errors, Figures 25 and 26 respectively display the **CSAFunctions** outcomes, and the effect of the adjustment procedure in the Testing Dataset of one of the 100 experiments of the UCI Yacht Group applied to a nonlinear (ANN) regression model. The graphs use the same graphic representation employed in Figures 19 and 20, for the Synthetic data in Item 5.1.3.

The adjustment effect on the predictions of the output variable of the Yacht experiment, Residuary Resistance, is illustrated in Figures 27 and 28.

In these graphs, in axis Y, we show the same class of information displayed in Figures 23 and 24, in Item 5.1.3: the actual observed Residuary Resistance values in the Testing dataset, $\boldsymbol{Y_{Tes}}$ (star markers); the original Linear regression prediction values, $\widehat{\boldsymbol{Y}}_{\boldsymbol{Tes}}$ (white square markers); and the adjusted prediction values, $\widehat{\boldsymbol{Y}}_{\boldsymbol{Adj}}$ (black circle markers). However, since in this set of experiment we are working with multiple regressions, we adopted the observations indices in axis X.

Figure 27 shows the two observations that were correctly identified as positive cases by the **CSAFunction$_{\mathbf{Pos}}$**, (quite positive residual error values indicated by the black triangles in Figures 25 and 26). The arrows indicate the direction of the adjustment produced by the addition of the $\boldsymbol{\delta_{Pos}}$ value.

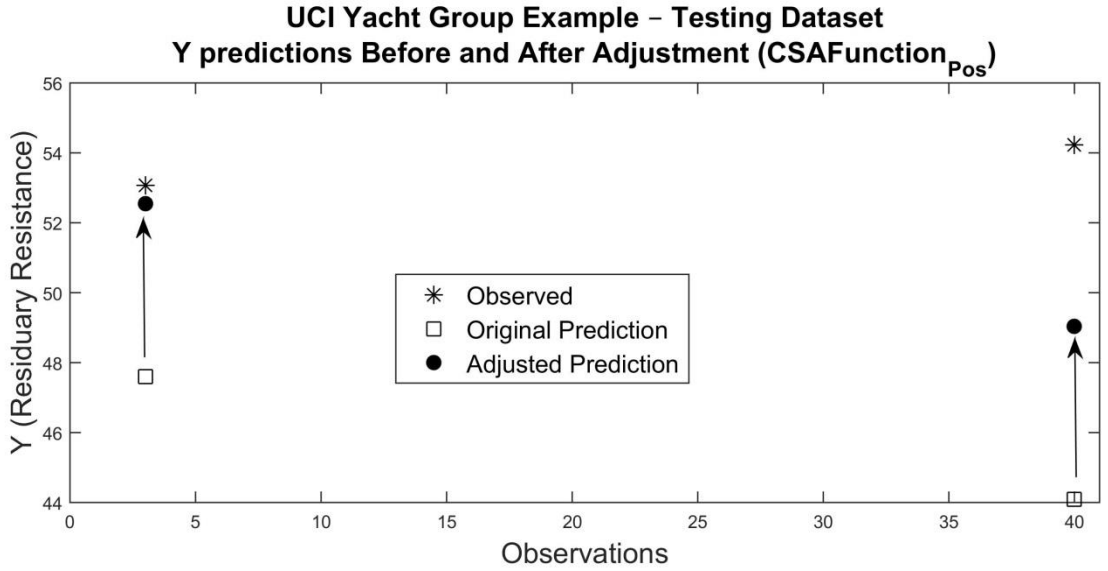

Figure 29: (UCI) Adjustment effect on the regression predictions (**CSAFunction$_{Pos}$**)

Figure 28 shows information about the seven observations that were correctly identified as positive cases by the **CSAFunction$_{Neg}$**, (quite negative residual error values indicated by the seven white triangles in Figures 25 and 26). The arrows indicate the direction of the adjustment produced by the addition of the $\boldsymbol{\delta_{Neg}}$ value.

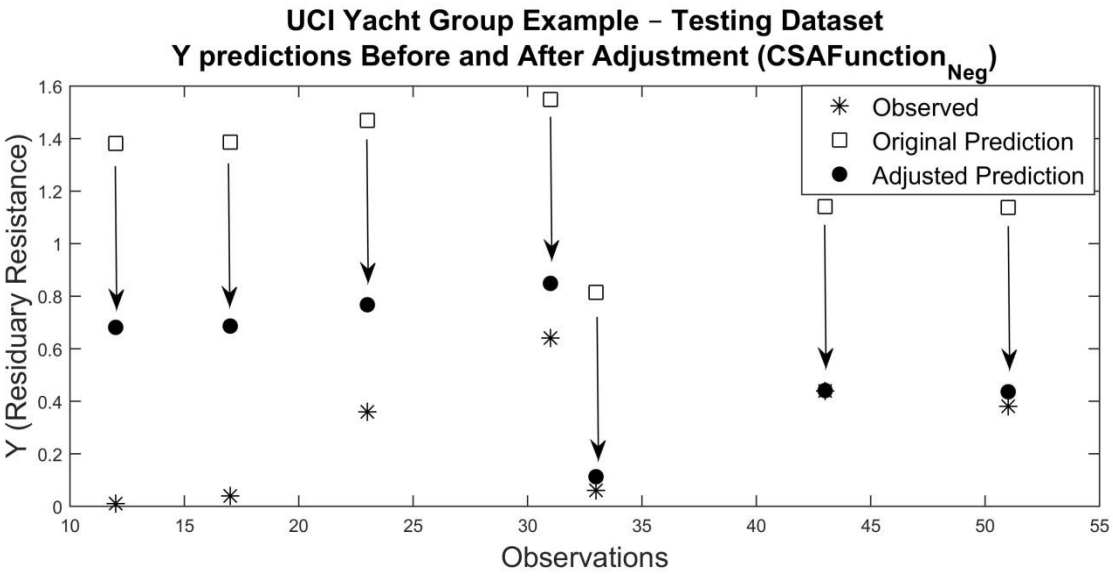

Figure 30: (UCI) Adjustment effect on the regression predictions (**CSAFunction$_{Neg}$**)

In the three graphs, in Figures 26, 27 and 28, that depict the effect of the adjustment procedure it is possible to clearly distinguish the reduction in the individual error values, and the improvement on the regression prediction values of the observations in the Testing dataset correctly identified as positive by the **CSAFunctions.**

Regarding the RMSE percentage of improvement, to illustrate the favorable condition of the rejection of the two null hypotheses, Figure 29 displays information about the RMSE values produced by regression predictions for the Testing datasets of the 100 experiences from the UCI Yatch Group.

The graph shows the following elements:

- Two histograms: one generated by the RMSE values produced by the original regression predictions (dotted lined bars), and another generated by the RMSE produced by the adjusted regression predictions (solid line bars).
- Two curves: one representing the normal distribution adjusted to the histogram generated by the RMSE values produced by the original regression predictions (dotted line curve), and another representing the normal distribution adjusted to the histogram generated by the RMSE values produced by the adjusted regression predictions (solid line curve).
- Two star markers indicating the average values of the two normal distributions.

The average of the hundred RMSE values produced by the adjusted regression predictions is smaller (0.968) than the average of the hundred RMSE values produced by the original regression predictions (1.115).

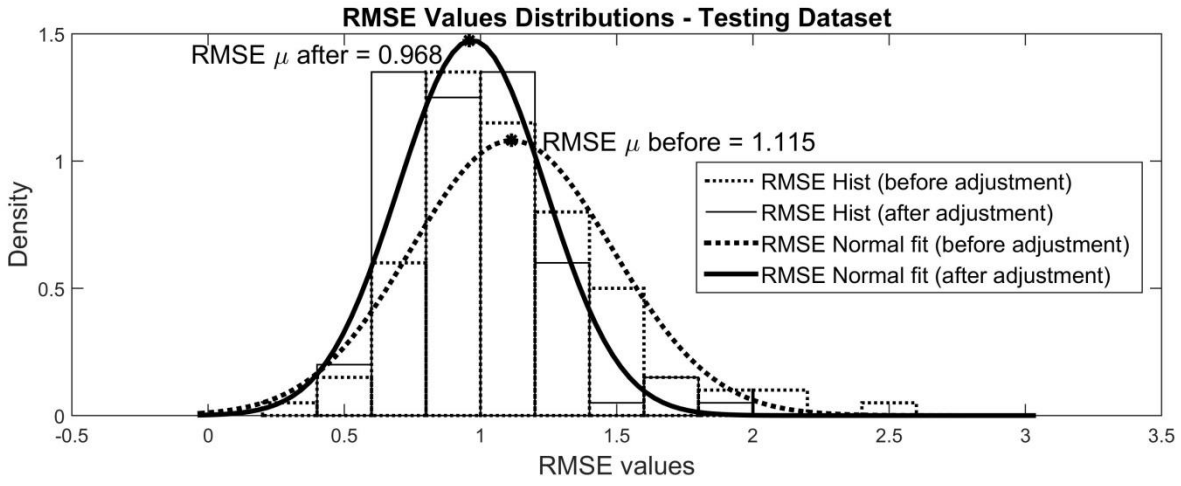

Figure 31: (UCI) Yacht experiment: Regression prediction error distributions.

5.2.4. Linear and Nonlinear Regression Models: Outcomes Comparison

One of the hypotheses of our proposed methodology is that the **CSAFunction**, via the definition of the critical scenario, **CScenario**, is able to extract information available in the regression prediction errors for the Training dataset to predict the value of the individual reliability estimate **CSFlag** of the Testing datasets; consequently the performance of the **CSAFunction** depends on the remaining deterministic part of the relationship among the variables that was not filtered by the regression model.

Following this rationale, since the ANN regression models performed better than the Linear regression models, in theory, the **CSAFunctions** would have more deterministic information available to be extracted from the errors produced by the Linear regression models than from the ones produced by the ANN regression models, for this reason, we could expect to obtain higher percentages of improvement in the case of the Linear regression models.

The outcomes endorsed that this happened in the experiments with data from the UCI databases: the percentages of improvements provided by the proposed methodology in the case of the Linear regression models were higher than in the case of the ANN regressions, as we can verify comparing the third and sixth column of Table 7 – RMSE % of improvement, in Item 5.2.3.

To more closely compare the outcomes of the experiences with the Linear and the ANN regression models, we constructed and analyzed four vector variables composed of the hundred RMSE values produced by the experiments for the Testing datasets, for each of the eight UCI Dataset groups:

i. $\boldsymbol{RMSE_{LinearOriginal}}$: composed of the 100 RMSE values produced by the original Linear regression predictions.
ii. $\boldsymbol{RMSE_{LinearAdjusted}}$: composed of the 100 RMSE values produced by the Linear regression predictions adjusted by the proposed methodology.
iii. $\boldsymbol{RMSE_{ANNOriginal}}$: composed of the 100 RMSE values produced by the original ANN (nonlinear) regression predictions.
iv. $\boldsymbol{RMSE_{ANNAdjusted}}$: composed of the 100 RMSE values produced by the ANN (nonlinear) regression predictions adjusted by the proposed methodology.

These four vector variables were used in three sets of hypothesis tests composed of the same two hypothesis tests described in Section 4.5.

The first set compared the RMSE produced by the two original regression predictions to verify if the ANN regression models really outperformed the Linear regression models in all cases ($\boldsymbol{RMSE_{LinearOriginal}}\ X\ \boldsymbol{RMSE_{ANNOriginal}}$).

The second set compared the RMSE produced by the adjusted Linear regression predictions with the RMSE produced by the original ANN regression predictions to verify if the adjusted Linear regression predictions outperformed the original ANN regression predictions ($\boldsymbol{RMSE_{LinearAdjusted}}\ X\ \boldsymbol{RMSE_{ANNOriginal}}$).

The third set compared the RMSE produced by the two adjusted regression predictions to verify if the adjusted Linear regression predictions outperformed the adjusted ANN regression predictions ($\boldsymbol{RMSE_{LinearAdjusted}}\ X\ \boldsymbol{RMSE_{ANNAdjusted}}$).

Table 10 displays the summary of the three hypothesis test results. In the first set of tests (first and second columns), the null hypotheses were rejected in all cases. That indicates, as expected, that the ANN regression models outperformed the Linear regression models.

Table 10: (UCI) Comparison of the RMSE values generated by the experiments.

UCI Database Groups	**RMSE Linear Original X ANN Original**		**RMSE Linear Adjusted X ANN Original**		**RMSE Linear Adjusted X ANN Adjusted**	
	2-Sample (P-value)	**One-Sided (P-value)**	**2-Sample (P-value)**	**One-Sided (P-value)**	**2-Sample (P-value)**	**One-Sided (P-value)**
Airfoil	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)
AutoMPG	Rejected (0.000)	Rejected (0.000)	Rejected (0.015)	Rejected (0.003)	Rejected (0.000)	Rejected (0.000)
Combined Cycle	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)
Concrete	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)
Energy	Rejected (0.000)	Rejected (0.000)	Rejected (0.001)	**Not Rejected (1.000)**	Rejected (0.004)	Rejected (0.000)
Housing	Rejected (0.000)	Rejected (0.000)	**Not Rejected (0.423)**	**Not Rejected (0.903)**	**Not Rejected (0.103)**	Rejected (0.017)
Parkinsons	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)
Yacht	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)	Rejected (0.000)

The second set of tests (third and fourth columns) has two null hypothesis “Not Rejected” cases. The One-sided null hypothesis “Not Rejected” in the Energy Datasets Group indicates that the averages of the RMSE values are different, and the difference between them is positive; that means that, in this case, the adjusted Linear regression predictions outperformed the original ANN regression predictions.

The two “Not Rejected” cases in the Housing Datasets Group indicates that the averages of the RMSE values are not different, and the difference between them is negative; that means that the adjusted Linear Regression predictions are a little bit better than the ANN Regression predictions, but not enough to change the average of the RMSE values.

The One-sided null hypothesis “Not Rejected” case in the third set of test (fifth and sixth columns), again in the Housing Datasets Group, indicates that the averages of the RMSE values are still not different, but, this time, the difference between them is positive; that means that the adjusted ANN Regression predictions are a little bit better than the Linear Regression predictions, but not enough to change the average of the RMSE values.

In conclusion, the methodology has proved to be effective in improving a comparatively weaker regression model, but there is no guarantee that its application surpasses the outcomes provided by more suitable regression models.

Moreover, an outstanding percentage of improvement, after the execution of the adjustment procedure may be a sign that the original regression model could be superseded by a better one. Conversely, the occurrence of extremely low Sensitivity metric values in the training datasets, can be an indication that the **CSAFunctions** may be facing problems in identifying positive cases, and the process might be reaching its limit, i.e., it is no longer possible to extract enough deterministic information from the data.

Finally, Table 11 displays the number of positive RMSE percentage of improvements in the 100 experiments per group. According to this information, the methodology was able to improve the original Linear regression predictions in 98.9% of the experiments (791 in 800), and in 90.5% (724 in 800) of the experiments, in the case of ANN regression predictions.

Table 11: (UCI) – Number of positive RMSE % improvement per group.

UCI Datasets Groups	Number of positive RMSE % Improvement in 100 experiments	
	Linear Regression Models	ANN Regression Models
Airfoil	100	97
AutoMPG	99	76
Combined Cycle	100	97
Concrete	100	99
Energy	95	87
Housing	97	85
Parkinsons	100	100
Yacht	100	83
Total	791 (98.9%)	724 (90.5%)

Next, we present the outcomes of the experiments performed using real time series data from the ISO-NE.

5.3. Outcomes with the ISO-NE Working Datasets

The routine of experiments performed with the data from the UCI Repository was repeated with the data from ISO-NE Database to analyze how the proposed methodology behaves when applied to time series data.

One set of experiments was defined for each of ISO-NE database, and the Training datasets constructed as described in Item 4.1.3 were used to design and to calibrate the parameters of a Linear regression model and an ANN regression model for each of the eight weather observation stations following the procedure described in Item 4.2.3.

As previously mentioned in Chapter 4, in this case, we used the Training datasets to model the regressions and the **CSAFunctions**. The regression models were formulated as an Integrated Auto Regressive with Exogenous Inputs model as defined in Equation (5.1), based on the work developed by Tee et al. (TEE; CARDELL; ELLIS, 2009) to forecast the load variation one hour in advance.

$$V_t = G(V_{t-1}, V_{t-24}, X1_{t-1}, X2_t) \tag{5.1}$$

$$Exogenous\ Inputs: X1_{t-1} = [Dr_{t-1}, De_{t-1}]\ and\ X2_t = [B_t, cH_t]$$

The RMSE values produced by the Linear and the ANN regression models with the Training and the Testing datasets, for each of ISO-NE datasets are displayed in Appendix B, Table B26. Again, the ANN regression models outperformed the Linear regression models in all cases.

5.3.1. Critical Error Scenario Threshold Values Definition

For all the experiments with ISO-NE data, one more time, the procedure described in Section 4.3 was applied to define the “best” threshold values for the **CScenario$_{Neg}$** and the **CScenario$_{Pos}$**. These threshold values were hence used to create the **CSFlag** variables.

The information about the critical scenarios is in Appendix B, from Table B27 to Table B30. These tables contain the same class of information displayed in the UCI experiments, but with no standard deviation values, because only one experiment were performed in each of the cases.

The “best” threshold values in Tables B27 and B28 are slightly higher than the numbers in Tables B29 and B30 indicating that, similarly to what happened in the UCI experiments, the errors produced by the Linear regression models are noticed to be more spread out than the ones produced by the ANN regression models

5.3.2. Critical Error Scenario Alert Functions’ Outcomes

The percentage of positives cases obtained in the Training datasets for the ISO-NE experiments were all lower than 20.0%. This numbers are displayed in the fourth column of Tables B27 up to B30 in Appendix B. So, once again, the cost-sensitive learning technique was thus applied to handle the problem of pattern recognition with imbalanced datasets.

The **CSAFunctions** were constructed according to the description in Section 4.4, and regarding the number of neurons in the hidden layer, as mentioned before in Section 4.4, five different numbers were tested, starting with the number of neurons in the hidden layer equal to the numbers of input parameters, and then progressively increasing it to twice, three, four, and five times the original number.

The choice of the best configuration was made using the cross-validation technique, and, for the set of experiments using ISO-NE data, instead of twice the number of input parameters, like in the case of the experiments with UCI data, the **CSAFunction$_{Neg}$** were implemented using CM (Neural Network Committee Machines) composed of ANN models designed with the number of neurons in the hidden layers equal to three times the numbers of the input parameters. In the case of the **CSAFunction$_{Pos}$**, the number of neurons in the hidden layers was set to five times the numbers of input parameters.

So, comparatively, more neurons were necessary to build the **CSAFunctions** in the case of the experiments with ISO-NE data, probably because of the high volatility of the load variation variables observable in the fourth column of Table B10 – Appendix B. These values are much higher than the volatility values of the variables from the UCI databases observable in the fourth column of Tables B01 up to B08 – Appendix B.

The values of the $\boldsymbol{fnpenalty}$ obtained by applying the procedure described in 4.4 are displayed in Appendix B – Table B31. These values were used to calibrate the parameters of the **CSAFunction$_{Neg}$** and **CSAFunction$_{Pos}$**.

To analyze the performance of these two **CSAFunction**s, we compared the metric values **Precision**, **Sensitivity**, and **F-Measure** generated by each of them in the eight experiments. Table B32 in Appendix B displays the performance metric values produced by the the **CSAFunctions$_{Neg}$** for the Training and the Testing datasets with Linear regression models. The same set of type information about the **CSAFunctions$_{Pos}$** is in Appendix B - Table B33. The summaries for the ANN regression models are in Table B34 and B35, both in Appendix B.

In these four tables (from B32 up to B35), we are able to observe that although the F-Measure metric values produced by the ISO-NE datasets are similar to the F-Measure averages produced by the experiments with the UCI Repository data; in the case of ISO-NE experiments using ANN regressions, comparatively, the Precision metric values are slightly higher, and the Sensitivity metric values are clearly lower.

5.3.3. Regression Prediction Values Improvement Using the CSFlag

In the case of the experiments with data from ISO-NE, as previously explained at the beginning of this section, for each regression model type, only eight experiments were performed; one for each database.

Table 12 displays the following metric values produced by the Linear and the ANN regression models for the Testing datasets: the RMSE generated by the original regression predictions, the RMSE generated by the regression predictions adjusted by the procedure described in Section 4.5; and the RMSE percentage of improvement.

According to the information in Table 12, the average RMSE percentage of improvement for the experiments using Linear regression models is 19.2% (with 2.3% of standard deviation), and 5.0% (with 3.9% of standard deviation) in the case of the ANN regression models. Although the percentages of improvement values are all positive, the percentages of improvement in the case of ANN regression models comparatively presented high variation.

Table 12: (ISO-NE) RMSE % of improvement.

ISO-NE Dataset	Linear Regression Models			ANN Regression Models		
	RMSE Original	RMSE Adjusted	RMSE % of Improvement	RMSE Original	RMSE Adjusted	RMSE % of Improvement
Boston	42.158	33.180	21.3%	26.112	24.260	7.1%
Bridgeport	57.617	45.382	21.2%	32.076	28.460	11.3%
Burlington	12.431	10.278	17.3%	7.656	7.541	1.5%
Concord	22.478	18.027	19.8%	12.549	12.282	2.1%
Portland	23.664	19.769	16.5%	16.043	15.559	3.0%
Providence	14.082	11.801	16.2%	9.032	9.027	0.1%
Windsor Locks	59.529	46.466	21.9%	37.300	35.062	6.0%
Worcester	32.495	26.239	19.3%	21.609	19.724	8.7%

As an example of the adjustment effect on the prediction, Figures 30 and 31 display the **CSAFunctions**' outcomes and the effect of the adjustment procedure obtained in the experiments with the Testing dataset from Windsor Locks station applied to a Linear regression model. These graphs display the same class of information depicted in Figures 19 and 20.

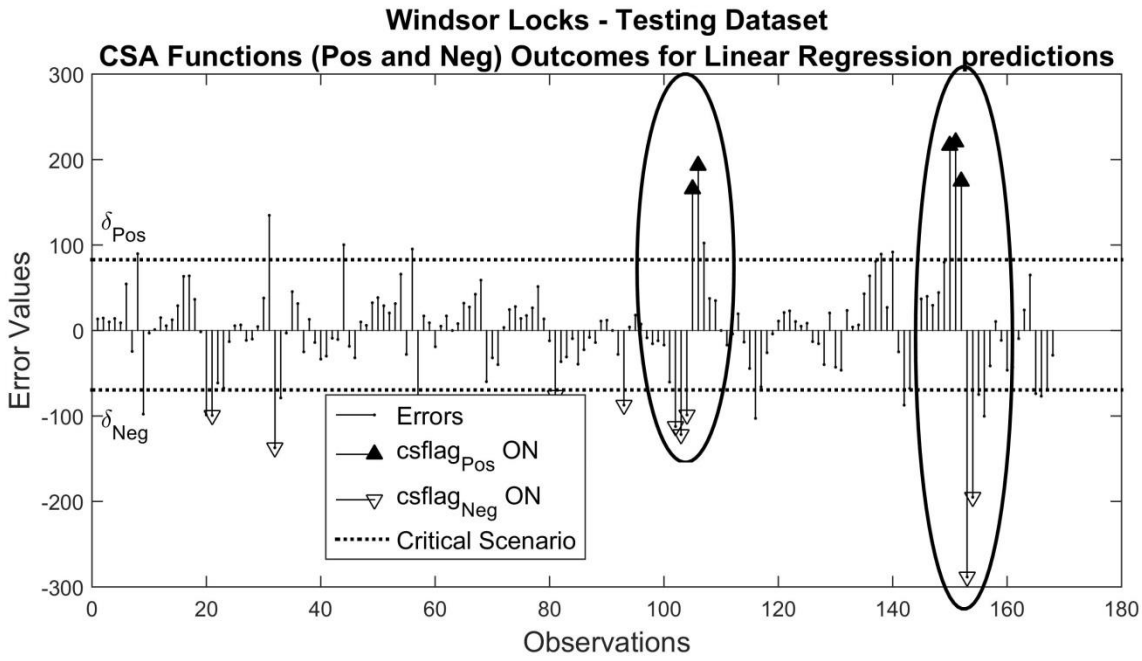

Figure 32: (ISO-NE) **CSAFunctions'** outcomes.

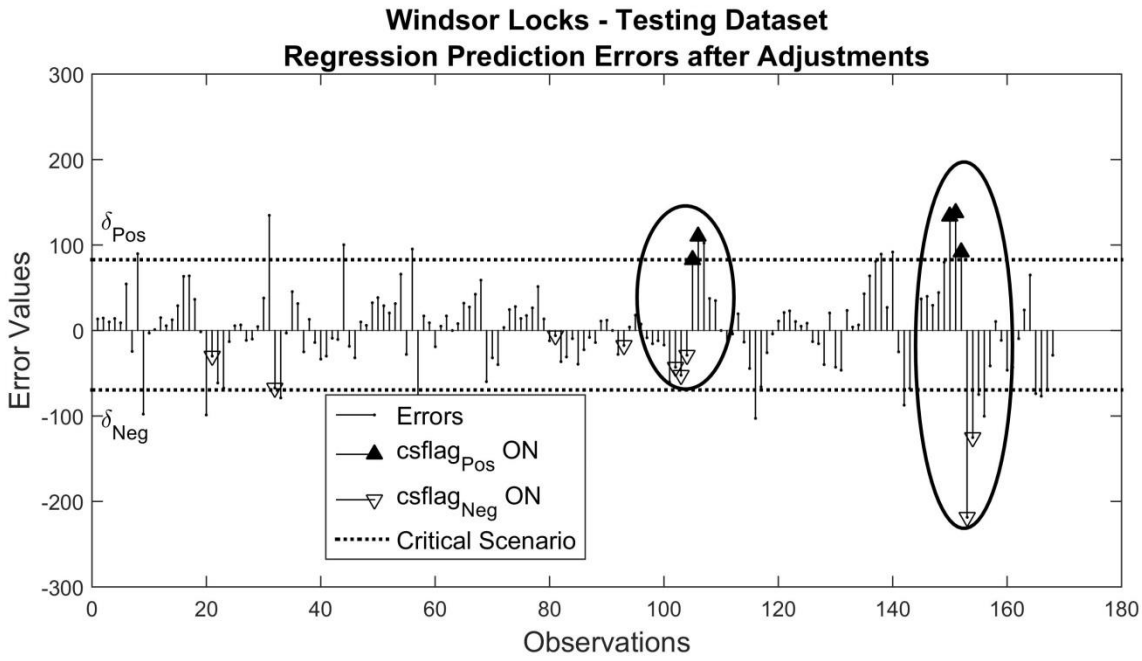

Figure 33: (ISO-NE) Adjustment effect on the prediction errors.

The adjustment effects on the load variation prediction values in the two observations intervals highlighted by the two ellipses in Figures 31 and 32 (from 100 to 108 and from 148 to 155) are respectively illustrated in Figures 33 and 34.

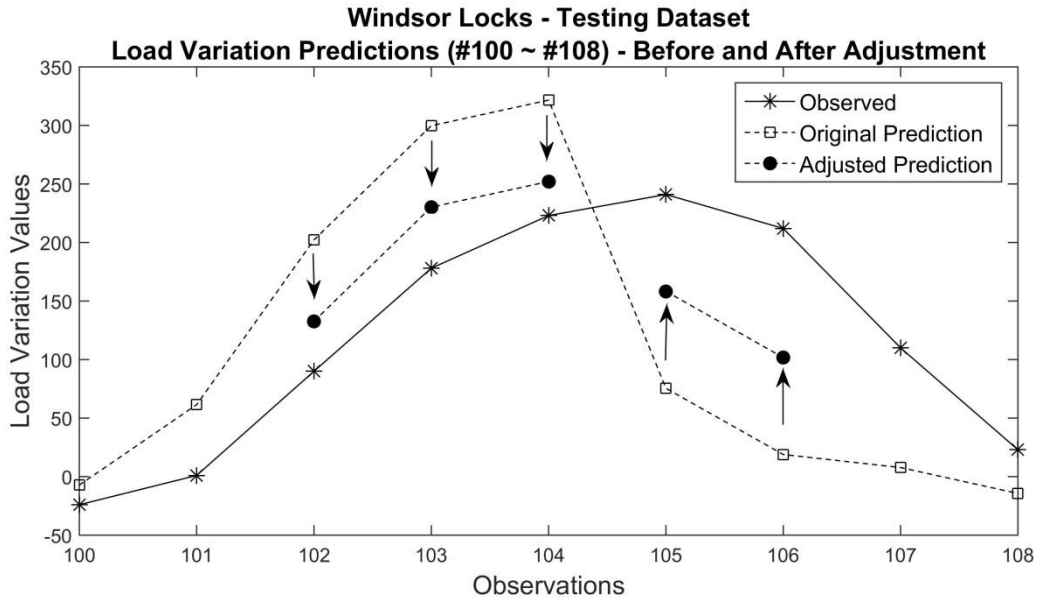

Figure 34: (ISO-NE) Adjustment effect on the regression predictions of Obs. # 100 ~ #108.

In these graphs, in axis Y, the actual observed load variation values are represented by a solid line with star markers; the original Linear regression prediction values by a dashed line with white square markers; and the adjusted prediction values by a dashed line with black circle markers. The arrows indicate the direction of the adjustment.

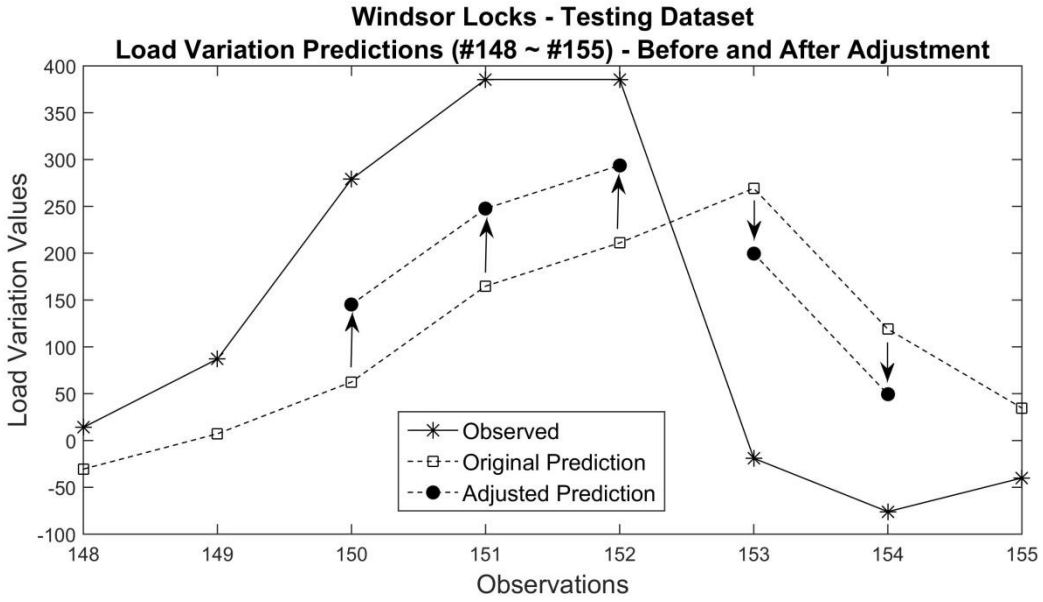

Figure 35: (ISO-NE) Adjustment effect on the regression predictions of Obs. # 148 ~ #155.

One more time, it is possible to clearly distinguish the reduction in the individual regression prediction error values in Figure 31, especially, in the two intervals detailed in Figures 32 and 33.

5.3.4. Linear and Nonlinear Regression Models: Outcomes Comparison

In the case of the ISO-NE datasets, comparing the values in the first and fourth columns of Table 11, the ANN regression models were observed to greatly surpass the Linear regression models in all the eight experiments. In the same table, the percentages of improvement provided by the proposed methodology in the case of the Linear regression models (values on the third column of the table) were also verified to be much higher than in the case of the ANN regression (sixth column of the table) for all datasets.

Accordingly, the rationale developed for the UCI experiments, in Item 5.2.4, regarding how the performance of the **CSAFunctions** is related to the performance of the regression model was confirmed by the outcomes of the experiments with ISO-NE data. Hence, in the case of the Linear regression models, the high RMSE percentages of improvement achieved after the execution of the adjustment procedure in the Testing datasets may be interpreted as an indication that the Linear regression models could be replaced by better models. Moreover, the high variance in the RMSE percentages of improvement achieved in the Training datasets for the ANN regression models could be interpreted as an alert that the process of trying to extract more deterministic information from the available data is reaching its limit.

This condition may be caused by the same set of factors mentioned in the case of the UCI experiments. Specifically, in the case of the Short-Term Load Forecasting problem (STLF), the random events that affect the electricity load variation do not have normal distribution and they also appear in the time series data that compose the set of input variables; furthermore the electricity load variation information is very volatile making the distinction between uncertainties and data trend more difficult.

Additionally, maybe we would need larger training datasets in numbers of observations or in amount of input information to get better results. In STLF, the definition of a precise hypothesis of the input-output relationship is considered a major challenge by the technical literature (JAVED et al., 2012); hence, the six input variables described in Item 4.1.3 could not be enough to fully explain the complex behavior of the load variation time series data, and equally insufficient to properly estimate the **CSFlag** values. It is also possible that the weather conditions in March

2015 were so different from the previous years that the Training and the Testing datasets ended up being generated by different process.

The overall conclusion according to the outcomes obtained in the three sets of experiments is that, despite having to cope with the same challenges faced by any other Machine Learning method, the proposed methodology is potentially able to improve individual predictions of an existing regression model, using only the information about its inputs and outputs for the training dataset.

6. CONCLUSION

We proposed and tested a method to improve computed individual regression predictions using a new type of point reliability estimate named **CSFlag**, a binary variable designed to indicate whether the regression prediction of an individual unseen observation is likely to produce an error condition considered critical, according to a scenario (**CScenario**) defined by the user of the regression. The original concept and the development of the methodology to construct the reliability estimate **CSFlag** is also a contribution of our research.

Aiming at improving regression predictions, we specify the construction of the two reliability estimates **CSFlag** based on the definition of two critical error scenarios, a quite positive residual error scenario (**CScenario$_{Pos}$**), and a quite negative residual error scenario (**CScenario$_{Neg}$**). Furthermore, in our proposition, the threshold values of these two critical error scenarios which are used to define the critical boundaries of the regression prediction errors are also adopted to adjust and improve the regression prediction values for the unseen observations forecasted as likely to produce critical error scenarios, i.e., with the reliability estimates **CSFlag** equal to 1.

In this context, to define the threshold values for both critical error scenarios, we indicate the use of optimization techniques to handle the following trade-off: higher absolute threshold values implying fewer adjustment cases *versus* lower absolute threshold values implying more adjustment cases.

The improvement of individual regression prediction values is achievable if distinguishing the critical scenario cases is feasible. Thus, one important task to be accomplished is the construction of the model to recognize the critical error pattern, the **CSAFunction** classifier.

The efficacy of the proposed methodology ultimately depends on the performance of the **CSAFunction**. Considering that if the regression models have reasonable quality, the proportion of the number of critical cases is supposed to be small, we recommend, when necessary, the adoption of strategies that address the problem of imbalanced dataset classification in the design of the **CSAFunctions**.

The main hypothesis of the presented methodology is that it is possible to construct two effective models for pattern recognition, the **CSAFunctions,** to predict two reliability estimates, the **CSFlags**, for individual regression predictions to be used to distinctively adjust and correct the computed regression predictions for unseen observations; moreover, the performance of the **CSAFunctions** and the efficiency of our methodology are undermined among other factors by the amount of randomness in the data and the degree of volatility of the data.

To empirically verify this hypothesis, we performed three sets of experiments using three different types of data: synthetic artificially produced data; real cross-sectional 8 databases extracted from the UCI (University of California, Irvine, Machine Learning Repository); and 8 real time series data obtained from ISO-NE (Independent System Operator in New England). The outcomes from these three sets of experiments effectively supported our initial conjectures.

The Synthetic datasets were artificially created to test the methodology in a very simple regression problem framework, arbitrarily choosing a single linear regression to predict a single quadratic function, and using datasets with different noise levels.

In this case, all the observed performance metrics (Precision, Sensitivity, and F-Measure) generated by the **CSAFunctions** in the experiments with cleaner datasets achieved very high values in the training datasets, and most of all in the testing datasets.

In consequence, for cleaner datasets, the method was able to correctly indicate most of the critical scenario cases and to improve the regression predictions of observations in the testing datasets. Conversely, as the level of noise went up, not only did the metric values of the **CSAFunctions** go down, but the difference between the outcomes in the training and in the testing datasets drastically increased, up to the point at which the **CSAFunctions** totally failed, and the adjustment procedure worsened the regression predictions instead of improving them.

The experimental outcomes reinforce the hypothesis that, in the ideal condition of artificial clean and low volatile data, it is possible to correctly forecast the reliability estimates, **CSFlag**, and to individually adjust and to improve the regression prediction values of all the observations forecasted with **csflag** equal to 1.

In the case of real data, the methodology was evaluated with two types of regression model, Linear and Artificial Neural Networks (ANN); and, as expected, since real data contains uncertainties, the **CSAFunctions** produced performance metric values comparable to the outcomes from the Synthetic moderately noisy databases.

For real data, the adjustment procedure defined by the developed method was able to improve computed individual regression predictions, and to reduce the RMSE values for the testing datasets, achieving positive percentage of improvement in about 95% of the experiments with real data: in 1515 from a total of 1600 experiments with UCI datasets; and in all 16 experiments with ISO-NE datasets.

In all experiments with real data, the methodology provided higher RMSE percentages of improvement for predictions produced by Linear regression models; and this result is related to another presumption of the methodology. This presumption is that the **CSAFunctios** use the deterministic part of the relationship between the output and the input variables not filtered by the regression model. Hence, in theory, the **CSAFunction** would perform comparatively better with regression prediction errors containing more deterministic information, i.e., with weaker regression models.

Since, in all experiments with real data, the Linear regression models were outperformed by the ANN regression models, we consider that the higher RMSE percentages of improvement with Linear regression models was an expected outcome, because the **CSAFunctions** had more deterministic information available to be extracted from the prediction errors of the Linear regression models than from the ANN regression models.

In other words, the experiments provided evidence that the methodology is able to improve computed individual regression predictions of unknown observations, but, like any Statistical or Machine Learning strategy, it is subject to limitations and uncertainties imposed by the real world, where data is subject to random events, it is usually very volatile, and it is rarely generated by a known function.

Regarding the implementation of the methodology, although, all **CSAFunctions** were built using Neural Networks Committee Machines, we again emphasize that any

kind of model for pattern recognition or classification could be adopted, such as, Support Vector Machines and K-Nearest Neighbors.

Furthermore, the proposed methodology has no restrictions on the type of regression model, and does not require any additional information about the architecture of the regression model. The regressions are handled as “black-box” prediction systems where only inputs and outputs values are available.

In the experiments, the two critical error scenarios, **CScenario$_{Pos}$** and **CScenario$_{Neg}$**, were specifically defined to be used in the adjustment procedure; however, our methodology to construct the **CSFlag** is supposed to work as long the critical error scenario definition is algorithmically describable and dependable only on the information available in the training dataset.

The only essential assumptions are the ones that are acknowledged by most Machine Learning methods: the training and the testing datasets are supposed to be generated by the same process; and there is enough data to build the models (ABU-MOSTAFA; MAGDON-ISMAIL; LIN, 2012c).

To extend the proof of concept trials, new experiments using synthetic artificially data produced with high volatility could be performed to try out the robustness of the proposed methodology under more challenging conditions.

As an extension of our research, a plausible route to follow is the development of reliability estimates for classification models aimed at improving computed individual classification predictions. And, with regard to the nature of the individual reliability estimate **CSFlag**, it would be interesting to investigate the potential advantages of defining it as a softer index, such as a graduation from 0 to 1, instead of as a binary variable.

Further research could explore the use of the reliability estimate **CSFlag** defined by other types of Critical Scenarios conditions (**CScenario**), aiming at solving problems in different applications, for instance, detection of errors in data measurement, identification of outliers; and other general data analysis tasks.

As additional empirical experiments in the field of prediction improvement, we plan benchmark the proposed methodology against existing predictions provider

systems. Among several possibilities, we could try to surpass the load variation forecast available on the ISO-NE website(ISO-NE, 2015), or monthly rent prices predictions estimated by smartphone applications such as Rent Zestimate provided by the real estate company Zillow (ZILLOW, 2015).

We could also continue to advance the present work by investigating the challenging research field of prediction explanation, similarly to what was proposed by Kononenko et al. (KONONENKO et al., 2013), and Bosnic et al. (BOSNIC et al., 2014) for regression models; and by Strumbelj and Kononenko (STRUMBELJ; KONONENKO, 2010) for classification models.

Finally, our research could advance to solve the following problem: “How to make the methodology viable when we do not know which variables are used as inputs of the regression”.

In the case of time series data, one possible alternative to be explored would be trying to use the auto regressive regression errors to compose the set of input variables of the **CSAFunction.**

APPENDIX A – Supplementary Figures

Section 6.2. Supplementary Figures

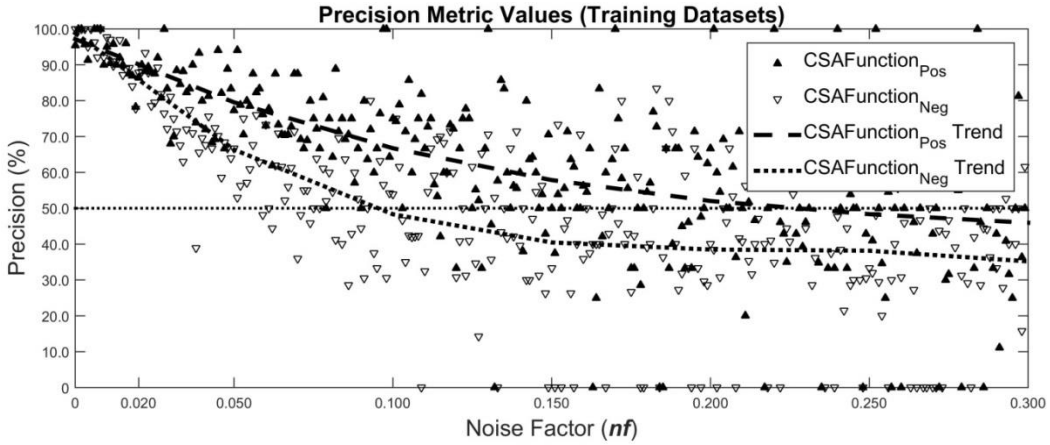

Figure A01: (Synthetic) Precision X Noise Factor (Training datasets).

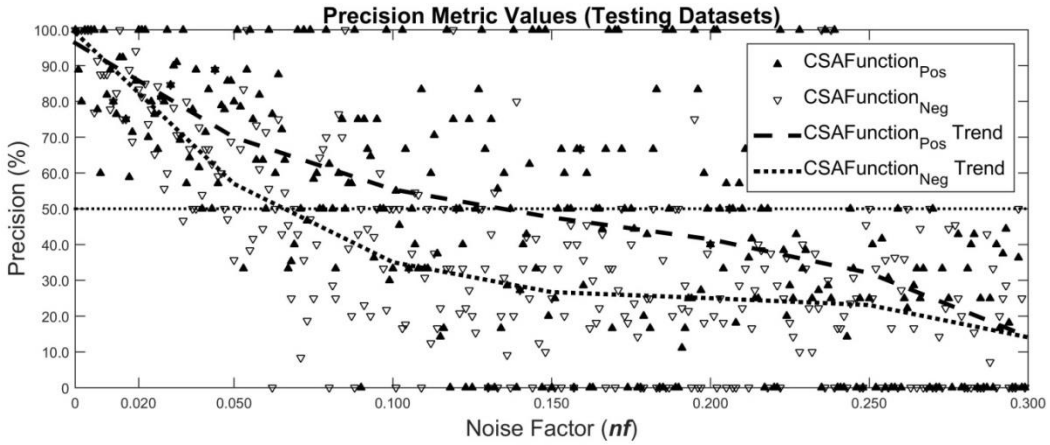

Figure A02: (Synthetic) Precision X Noise Factor (Testing datasets).

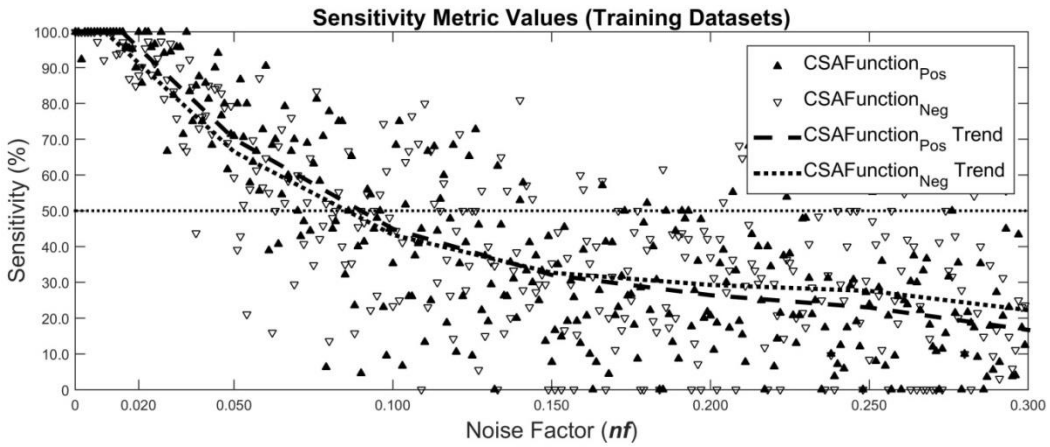

Figure A03: (Synthetic) Sensitivity X Noise Factor (Training datasets).

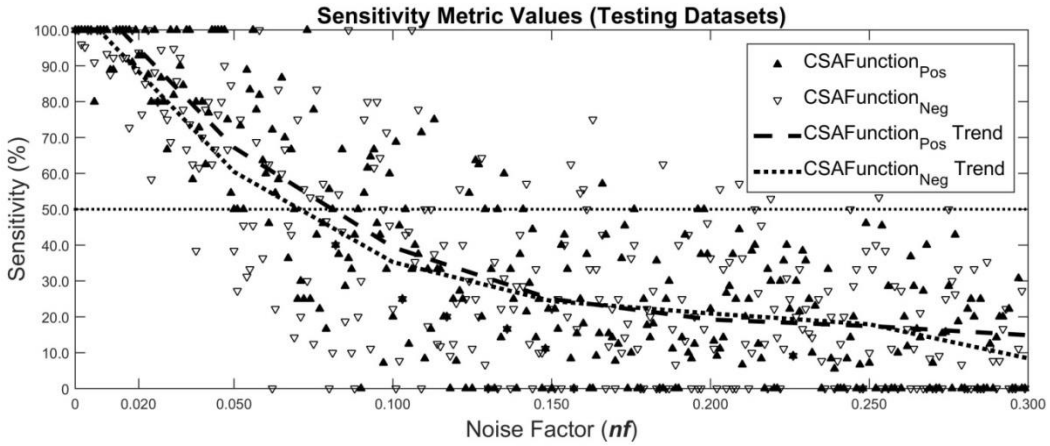

Figure A04: (Synthetic) Sensitivity X Noise Factor (Testing datasets).

APPENDIX B – Supplementary Tables

Section 4.1. Data Description

Table B01: (UCI) Airfoil Database Statistics

Variable Description	Avg.	Std.	Min.	Max.	Vol.
(Y) Scaled sound pressure level	124.836	6.899	103.380	140.987	0.055
(X1) Frequency	2886.381	3152.573	200.000	20000.000	1.092
(X2) Attack angle	6.782	5.918	0.000	22.200	0.873
(X3) Chord length	0.137	0.094	0.025	0.305	0.685

Table B02: (UCI) AutoMPG Database Statistics

Variable Description	Avg.	Std.	Min.	Max.	Vol.
(Y) Miles per gallon	23.446	7.805	9.000	46.600	0.333
(X1) Number of cylinders	5.472	1.706	3.000	8.000	0.312
(X2) Displacement	194.412	104.644	68.000	455.000	0.538
(X3) Horse power	104.469	38.491	46.000	230.000	0.368
(X4) Weight	2977.584	849.403	1613.000	5140.000	0.285
(X5) Acceleration	15.541	2.759	8.000	24.800	0.178
(X6) Model Year	75.980	3.684	70.000	82.000	0.048

Table B03: (UCI) Combined Cycle Database Statistics

Variable Description	Avg.	Std.	Min.	Max.	Vol.
(Y) Electrical energy	454.365	17.067	420.260	495.760	0.038
(X1) Ambient temperature	19.651	7.452	1.810	37.110	0.379
(X2) Exhaust vacuum	54.306	12.708	25.360	81.560	0.234
(X3) Ambient pressure	1013.259	5.939	992.890	1033.300	0.006
(X4) Relative humidity	73.309	14.600	25.560	100.160	0.199

Table B04: (UCI) Concrete Database Statistics

Variable Description	Avg.	Std.	Min.	Max.	Vol.
(Y) Compressive strength	35.818	16.706	2.332	82.599	0.466
(X1) Cement	281.166	104.507	102.000	540.000	0.372
(X2) Blast	73.895	86.279	0.000	359.400	1.168
(X3) Fly Ash	54.187	63.996	0.000	200.100	1.181
(X4) Water	181.566	21.356	121.750	247.000	0.118

(X5) Superplasticizer	6.203	5.973	0.000	32.200	0.963
(X6) Coarse aggregate	972.919	77.754	801.000	1145.000	0.080
(X7) Fine aggregate	773.579	80.175	594.000	992.600	0.104
(X8) Age	45.662	63.170	1.000	365.000	1.383

Table B05: (UCI) Energy Database Statistics

Variable Description	Avg.	Std.	Min.	Max.	Vol.
(Y) Cooling load	24,588	9,513	10,900	48,030	0,387
(X1) Relative compactness	0,764	0,106	0,620	0,980	0,138
(X2) Surface area	671,708	88,086	514,500	808,500	0,131
(X3) Wall area	318,500	43,626	245,000	416,500	0,137
(X4) Roof area	176,604	45,166	110,250	220,500	0,256
(X5) Overall height	5,250	1,751	3,500	7,000	0,334
(X6) Orientation	3,500	1,119	2,000	5,000	0,320
(X7) Glazing area	0,234	0,133	0,000	0,400	0,568
(X8) Glazing area distribution	2,813	1,551	0,000	5,000	0,551

Table B06: (UCI) Housing Database Statistics

Variable Description	Avg.	Std.	Min.	Max.	Vol.
(Y) Rental median value	22,533	9,197	5,000	50,000	0,408
(X1) Per capita crime rate	3,614	8,602	0,006	88,976	2,380
(X2) Prop. of residential land zoned	11,364	23,322	0,000	100,000	2,052
(X3) Prop. of non-retail business	11,137	6,860	0,460	27,740	0,616
(X4) Charles River boolean variable	0,069	0,254	0,000	1,000	N/A (*)
(X5) Nitric oxides concentration	0,555	0,116	0,385	0,871	0,209
(X6) Average number of rooms	6,285	0,703	3,561	8,780	0,112
(X7) Prop. of owner-occupied units	68,575	28,149	2,900	100,000	0,410
(X8) Dist. from employment centers	3,795	2,106	1,130	12,127	0,555
(X9) Accessibility to highways	9,549	8,707	1,000	24,000	0,912
(X10) Property-tax rate	408,237	168,537	187,000	711,000	0,413
(X11) Pupil-teacher ratio	18,456	2,165	12,600	22,000	0,117
(X12) Proportion of blacks	356,674	91,295	0,320	396,900	0,256
(X13) % lower status of the populat.	12,653	7,141	1,730	37,970	0,564

(*) N/A: not applicable.

Table B07: (UCI) Parkinsons Database Statistics

Variable Description	Avg.	Std.	Min.	Max.	Vol.
(Y) Total UPDRS (*)	29.019	10.700	7.000	54.992	0.369
(X1) Age	64.805	8.822	36.000	85.000	0.136
(X2) Sex	0.318	0.466	0.000	1.000	1.465
(X3) Jitter DDP	0.009	0.009	0.001	0.173	1.046
(X4) Shimmer	0.034	0.026	0.003	0.269	0.759
(X5) NHR (noise ratio)	0.032	0.060	0.000	0.748	1.858
(X6) HNR (noise ratio)	21.679	4.291	1.659	37.875	0.198
(X7) RPDE (dynamical complexity)	0.541	0.101	0.151	0.966	0.187
(X8) DFA (signal fractal scal. exp.)	0.653	0.071	0.514	0.866	0.109
(X9) PPE (fundamental freq. var.)	0.220	0.091	0.022	0.732	0.417

(*)UPDRS - Unified Parkinson's disease rating scale

Table B08: (UCI) Yacht Database Statistics

Variable Description	Avg.	Std.	Min.	Max.	Vol.
(Y) Residuary resistance	10.495	15.160	0.010	62.420	1.444
(X1) Longitudinal position	-2.382	1.513	-5.000	0.000	-0.635
(X2) Prismatic coefficient	0.564	0.023	0.530	0.600	0.041
(X3) Length-displacement ratio	4.789	0.253	4.340	5.140	0.053
(X4) Beam-draught ratio	3.937	0.548	2.810	5.350	0.139
(X5) Length-beam ratio	3.207	0.248	2.730	3.640	0.077
(X6) Froude number	0.288	0.101	0.125	0.450	0.351

Table B09: (ISO-NE) Variables description provided by ISO New England

Variable Name	Description
Date	Date in MM/DD/YYYY format.
DA_DEMD	Day ahead demand.
DEMAND	Load used in the settlement process.
DA_LMP	Day ahead location marginal price.
DA_EC	Energy component of the day ahead price.
DA_CC	Congestion component of the day a head price.
DA_MLC	Marginal loss component of the day ahead price.
RT_LMP	Real time locational marginal price.
RT_EC	Energy component of the real time price.
RT_CC	Congestion component of the real time price.
RT_MLC	Marginal loss component of the real time price.

DryBulb	Dry bulb temperature in degrees Fahrenheit.
DewPnt	Dew point temperature in degrees Fahrenheit.
RegCP	Regulation clearing price.

Table B10: (ISO-NE) All Datasets Statistics

Variable Description	Avg.	Std.	Min.	Max.	Vol.
(Y) V_t	-0.331	115.893	-391.000	497.000	-349.928
(X1) V_{t-1}	-0.348	115.929	-391.000	497.000	-333.209
(X2) V_{t-24}	-0.326	116.460	-391.000	497.000	-357.422
(X3) Dr_{t-1}	36.034	12.448	-7.000	84.000	0.345
(X4) De_{t-1}	22.077	13.942	-16.000	57.000	0.632
(X5) B_t **(boolean variable)**	0.000	0.707	0.000	1.000	N/A (*)
(X6) cH_t	0.680	0.466	0.000	1.000	0.686

(*) N/A: not applicable.

Section 5.1. Outcomes with the Experiments with Synthetic Working Datasets

Table B11: (Synthetic Datasets Groups) Linear regression models RMSE statistic metrics

Dataset Type	Average	Standard Deviation	Minimum	Maximum
Regression	0.171	0.073	0.067	0.318
Training	0.176	0.074	0.067	0.337
Testing	0.175	0.076	0.066	0.362

Table B12: (Synthetic Datasets Groups) **CScenario$_{Pos}$** Threshold Values

Information	Average	Standard Deviation	Minimum	Maximum
α_{Pos}	79.1%	5.5%	62.7%	92.5%
δ_{Pos}	0.152	0.066	0.057	0.417
Training Dataset: # *csflag* ON	21.390	5.439	8.000	38.000
% *csflag* ON	21.4%	5.4%	8.0%	38.0%
Testing Dataset: # *csflag* ON	10.173	3.699	0.000	20.000
% *csflag* ON	20.4%	7.4%	0.0%	40.0%

Table B13: (Synthetic Datasets Groups) **CScenario$_{Neg}$** Threshold Values

Information	Average	Standard Deviation	Minimum	Maximum
α_{Neg}	25.4%	7.0%	6.7%	45.0%
δ_{Neg}	-0.137	0.065	-0.425	-0.042

Training Dataset: # *csflag* ON	23.893	6.988	5.000	44.000
% *csflag* ON	23.9%	7.99%	5.0%	44.0%
Testing Dataset: # *csflag* ON	11.480	4.313	2.000	25.000
% *csflag* ON	23.0%	8.6%	4.0%	50.0%

Section 5.2. Outcomes with the Experiments with UCI Working Datasets

Table B14: (UCI) Features Selection

UCI Datasets Group	Total Number of Input Variables	Selected for the Linear Regression Models	Selected for the ANN Regression Models
Airfoil	6	5	5
AutoMPG	8	2	4
Combined Cycle	4	4	4
Concrete	10	6	8
Energy	8	7	5
Housing	14	12	6
Parkinsons	26	9	8
Yacht	7	1	4

Table B15: (UCI) Linear regression models RMSE Metrics.

UCI Datasets Group	Regression Dataset RMSE [Avg./Std Dev.]	Training Dataset RMSE [Avg./Std Dev.]	Testing Dataset RMSE [Avg./Std Dev.]
Airfoil	4.786 [0.122]	4.839 [0.114]	4.813 [0.199]
AutoMPG	3.352 [0.182]	3.494 [0.199]	3.514 [0.325]
Combined Cycle	4.553 [0.069]	4.556 [0.063]	4.576 [0.116]
Concrete	10.260 [0.290]	10.555 [0.317]	10.566 [0.479]
Energy	1.939 [0.069]	1.980 [0.069]	1.986 [0.137]
Housing	4.843 [0.353]	5.164 [0.360]	4.948 [0.622]
Parkinsons	9.811 [0.097]	9.852 [0.093]	9.822 [0.196]
Yacht	8.857 [0.537]	9.082 [0.539]	8.932 [0.878]

Table B16: (UCI) ANN regression models RMSE Metrics

UCI Database Group	Regression Dataset RMSE [Avg./Std Dev.]	Training Dataset RMSE [Avg./Std Dev.]	Testing Dataset RMSE [Avg./Std Dev.]
Airfoil	3.238 [0.445]	3.466 [0.415]	3.507 [0.437]
AutoMPG	2.670 [0.310]	3.124 [0.324]	3.097 [0.389]
Combined Cycle	4.169 [0.084]	4.195 [0.082]	4.219 [0.127]
Concrete	5.750 [1.111]	7.207 [1.385]	7.094 [1.427]

Energy	1.617 [0.225]	1.864 [0.219]	1.852 [0.234]
Housing	3.591 [0.541]	4.769 [1.483]	4.494 [1.040]
Parkinsons	6.557 [0.460]	6.928 [0.465]	6.898 [0.487]
Yacht	0.873 [0.357]	1.169 [0.408]	1.115 [0.369]

Table B17: (UCI) Linear regression models **CScenario$_{Pos}$** "Best" Threshold

UCI Datasets Group	CScenario$_{Pos}$ "Best" Threshold		Training Dataset		Testing Dataset	
	α_{Pos}	δ_{Pos}	# of ON	% of ON	# of ON	% of ON
Airfoil	80.5% [2.6%]	4.075 [0.386]	123.600 [16.087]	19.6% [2.5%]	51.740 [9.280]	19.2% [3.4%]
AutoMPG	83.2% [5.0%]	3.037 [0.660]	28.170 [8.295]	17.1% [5.0%]	11.840 [4.777]	16.9% [6.8%]
Combined Cycle	79.5% [1.2%]	3.923 [0.183]	823.250 [46.277]	20.5% [1.2%]	352.760 [25.586]	20.5% [1.5%]
Concrete	73.9% [4.2%]	6.747 [0.985]	113.310 [18.301]	26.2% [4.2%]	48.270 [8.085]	26.1% [4.4%]
Energy	89.1% [2.8%]	2.519 [0.494]	35.680 [8.991]	11.1% [2.8%]	14.240 [4.297]	10.3% [3.1%]
Housing	89.1% [3.8%]	6.002 [1.773]	23.810 [8.189]	11.2% [3.8%]	9.550 [4.342]	10.6% [4.8%]
Parkinsons	68.4% [1.8%]	4.716 [0.507]	781.500 [44.473]	31.7% [1.8%]	332.990 [24.385]	31.5% [2.3%]
Yacht	77.2% [4.3%]	6.473 [1.313]	29.990 [5.498]	23.2% [4.3%]	12.370 [3.329]	22.5% [6.1%]

Table B18: (UCI) Linear regression models **CScenario$_{Neg}$** "Best" Threshold

UCI Datasets Group	CScenario$_{Neg}$ "Best" Threshold		Training Dataset		Testing Dataset	
	α_{Neg}	δ_{Neg}	# of ON	% of ON	# of ON	% of ON
Airfoil	16.7% [4.4%]	-4.621 [0.977]	103.810 [27.397]	16.5% [4.3%]	43.300 [13.523]	16.0% [5.0%]
AutoMPG	24.7% [4.1%]	-2.643 [0.415]	39.230 [6.769]	23.8% [4.1%]	15.770 [3.824]	22.5% [5.5%]
Combined Cycle	22.7% [1.3%]	-3.474 [0.147]	909.780 [53.358]	22.6% [1.3%]	387.030 [30.575]	22.5% [1.8%]
Concrete	13.5% [2.9%]	-12.106 [1.904]	56.890 [12.496]	13.1% [2.9%]	23.050 [7.328]	12.5% [4.0%]
Energy	15.8% [4.7%]	-1.894 [0.473]	49.470 [15.063]	15.4% [4.7%]	20.060 [7.541]	14.5% [5.5%]
Housing	24.9% [4.1%]	-3.316 [0.452]	51.590 [8.751]	24.2% [4.1%]	21.150 [5.050]	23.5% [5.6%]
Parkinsons	39.5% [2.3%]	-4.238 [0.465]	972.940 [57.973]	39.4% [2.3%]	415.710 [29.207]	39.3% [2.8%]
Yacht	32.8% [4.0%]	-6.356 [0.823]	40.860 [5.162]	31.7% [4.0%]	17.440 [3.491]	31.7% [6.3%]

Table B19: (UCI) ANN regression models **CScenario$_{Pos}$** "Best" Threshold

UCI Datasets Group	CScenario$_{Pos}$ "Best" Threshold		Training Dataset		Testing Dataset	
	α_{Pos}	δ_{Pos}	# of ON	% of ON	# of ON	% of ON
Airfoil	79.5% [4.3%]	2.706 [0.409]	129.890 [27.231]	20.6% [4.3%]	55.290 [14.373]	20.5% [5.3%]
AutoMPG	86.1% [5.5%]	3.091 [0.915]	23.430 [9.189]	14.2% [5.6%]	8.990 [4.373]	12.8% [6.2%]
Combined Cycle	78.6% [1.3%]	3.349 [0.168]	859.150 [51.204]	21.4% [1.3%]	365.860 [31.352]	21.2% [1.8%]
Concrete	84.5% [3.2%]	6.808 [1.596]	67.450 [13.932]	15.6% [3.2%]	27.130 [6.435]	14.7% [3.5%]
Energy	89.8% [3.0%]	2.399 [0.455]	33.380 [9.555]	10.4% [3.0%]	13.310 [4.382]	9.6% [3.2%]
Housing	86.1% [6.2%]	4.152 [1.533]	30.100 [13.278]	14.1% [6.2%]	11.940 [6.237]	13.3% [6.9%]
Parkinsons	82.9% [2.7%]	6.076 [0.707]	422.700 [66.520]	17.1% [2.7%]	177.070 [28.920]	16.8% [2.7%]
Yacht	91.3% [4.9%]	1.610 [1.078]	11.760 [6.314]	9.1% [4.9%]	4.340 [2.865]	7.9% [5.2%]

Table B20: (UCI) ANN regression models **CScenario$_{Neg}$** "Best" Threshold

UCI Datasets Group	CScenario$_{Neg}$"Best" Threshold		Training Dataset		Testing Dataset	
	α_{Neg}	δ_{Neg}	# of ON	% of ON	# of ON	% of ON
Airfoil	15.9% [3.2%]	-3.225 [0.633]	98.860 [20.416]	15.7% [3.2%]	40.810 [9.255]	15.1% [3.4%]
AutoMPG	18.7% [6.8%]	-2.667 [0.683]	29.410 [11.107]	17.8% [6.7%]	12.090 [5.850]	17.3% [8.4%]
Combined Cycle	19.9% [1.4%]	-3.385 [0.173]	799.500 [54.354]	19.9% [1.4%]	341.400 [29.006]	19.8% [1.7%]
Concrete	18.1% [4.8%]	-6.242 [1.629]	76.790 [20.666]	17.7% [4.8%]	30.610 [10.233]	16.5% [5.5%]
Energy	11.6% [3.2%]	-2.057 [0.490]	35.990 [10.430]	11.2% [3.2%]	15.410 [5.944]	11.2% [4.3%]
Housing	15.8% [6.3%]	-4.084 [1.168]	32.130 [13.470]	15.1% [6.3%]	12.510 [6.160]	13.9% [6.8%]
Parkinsons	17.9% [2.5%]	-5.761 [0.679]	441.230 [62.499]	17.9% [2.5%]	187.010 [27.815]	17.7% [2.6%]
Yacht	11.7% [5.1%]	-1.023 [0.388]	13.540 [6.631]	10.5% [5.1%]	5.030 [3.465]	9.1% [6.3%]

Table B21: (UCI) Linear and ANN regression models (average, stand.deviation) of FN Penalty Values.

UCI Datasets Group	Linear Regression Models		ANN Regression Models	
	$CSAFunction_{Pos}$ FN Penalty Avg. [Std.]	$CSAFunction_{Neg}$ FN Penalty Avg. [Std.]	$CSAFunction_{Pos}$ FN Penalty Avg. [Std.]	$CSAFunction_{Neg}$ FN Penalty Avg. [Std.]
Airfoil	1.044 [0.056]	1.084 [0.141]	1.074 [0.106]	1.187 [0.288]
AutoMPG	1.345 [0.635]	1.073 [0.127]	1.584 [1.154]	1.362 [1.084]
Combined Cycle	1.038 [0.056]	1.038 [0.046]	1.092 [0.119]	1.150 [0.171]
Concrete	1.039 [0.048]	1.118 [0.281]	1.142 [0.737]	1.061 [0.102]
Energy	1.206 [0.361]	1.105 [0.150]	1.225 [0.354]	1.172 [0.296]
Housing	1.383 [0.787]	1.062 [0.079]	1.431 [1.265]	1.356 [1.221]
Parkinsons	1.018 [0.027]	1.014 [0.018]	1.061 [0.119]	1.048 [0.059]
Yacht	1.347 [0.563]	1.177 [0.240]	2.037 [2.124]	1.683 [1.568]

Table B22: (UCI) Linear regression models – Performance metrics information of **$CSAFunction_{Pos}$**

UCI Dataset Groups	$CSAFunction_{Pos}$ Metrics (Avg. [Std.]) In Training Dataset			$CSAFunction_{Pos}$ Metrics (Avg. [Std.]) In Testing Dataset		
	Precision	Sensitivity	F-Measure	Precision	Sensitivity	F-Measure
Airfoil	86.1% [4.5%]	49.9% [8.3%]	0.626 [0.066]	71.0% [10.9%]	39.1% [10.3%]	0.496 [0.098]
AutoMPG	88.9% [8.3%]	47.2% [15.8%]	0.595 [0.143]	66.2% [26.2%]	31.7% [18.4%]	0.398 [0.182]
Combined Cycle	69.0% [2.6%]	21.2% [5.0%]	0.321 [0.060]	65.1% [5.8%]	20.0% [5.0%]	0.302 [0.060]
Concrete	88.3% [3.1%]	64.2% [9.5%]	0.740 [0.070]	73.1% [8.4%]	52.1% [11.3%]	0.600 [0.089]
Energy	91.5% [9.2%]	51.9% [19.4%]	0.636 [0.177]	72.7% [24.2%]	40.5% [19.6%]	0.492 [0.178]
Housing	94.8% [5.4%]	67.2% [12.4%]	0.777 [0.092]	69.4% [27.2%]	40.3% [19.8%]	0.486 [0.198]
Parkinsons	87.9% [1.2%]	77.0% [3.2%]	0.820 [0.018]	83.8% [2.4%]	73.1% [4.0%]	0.780 [0.024]
Yacht	98.6% [2.2%]	98.9% [2.4%]	0.987 [0.016]	91.5% [9.0%]	90.3% [10.7%]	0.905 [0.080]

Table B23: (UCI) Linear regression models – Performance metrics information of **$CSAFunction_{Neg}$**

UCI Datasets Group	$CSAFunction_{Neg}$ Metrics (Avg. [Std.]) In Training Dataset			$CSAFunction_{Neg}$ Metrics (Avg. [Std.]) In Testing Dataset		
	Precision	Sensitivity	F-Measure	Precision	Sensitivity	F-Measure
Airfoil	93.4% [3.2%]	66.4% [7.2%]	0.773 [0.053]	81.3% [7.7%]	56.1% [9.6%]	0.659 [0.084]
AutoMPG	78.4%	40.7%	0.517	56.5%	29.4%	0.366

	[9.2%]	[14.2%]	[0.131]	[18.2%]	[14.8%]	[0.147]
Combined Cycle	66.6% [3.1%]	10.1% [3.3%]	0.173 [0.046]	59.9% [6.9%]	9.0% [3.3%]	0.154 [0.048]
Concrete	92.6% [4.3%]	70.4% [14.6%]	0.788 [0.105]	68.5% [14.6%]	48.2% [16.4%]	0.550 [0.142]
Energy	84.0% [8.5%]	64.0% [13.8%]	0.717 [0.103]	72.5% [15.6%]	57.6% [15.7%]	0.623 [0.129]
Housing	84.6% [6.6%]	39.6% [13.6%]	0.524 [0.126]	60.1% [19.3%]	28.0% [14.7%]	0.363 [0.151]
Parkinsons	84.5% [1.5%]	79.0% [2.8%]	0.816 [0.019]	81.2% [2.7%]	74.8% [3.5%]	0.778 [0.024]
Yacht	99.0% [1.4%]	99.6% [1.1%]	0.993 [0.009]	93.2% [6.4%]	96.4% [5.2%]	0.946 [0.042]

Table B24: (UCI) ANN regression models – Performance metrics information of **CSAFunction$_{Pos}$**

UCI Datasets Group	**CSAFunction$_{Pos}$** Metrics (Avg. [Std.]) In Training Dataset			**CSAFunction$_{Pos}$** Metrics (Avg. [Std.]) In Testing Dataset		
	Precision	Sensitivity	F-Measure	Precision	Sensitivity	F-Measure
Airfoil	84.4% [14.8%]	27.5% [15.8%]	0.391 [0.182]	63.2% [21.1%]	19.6% [13.6%]	0.279 [0.166]
AutoMPG	77.0% [31.8%]	27.5% [21.0%]	0.367 [0.218]	31.6% [37.8%]	10.8% [15.5%]	0.145 [0.191]
Combined Cycle	**81.1% [15.9%]**	**3.3% [2.1%]**	**0.062 [0.039]**	67.5% [23.4%]	2.7% [1.9%]	0.051 [0.034]
Concrete	93.5% [6.1%]	37.2% [14.7%]	0.514 [0.146]	58.4% [22.4%]	21.6% [13.5%]	0.298 [0.151]
Energy	80.9% [23.0%]	46.5% [26.1%]	0.551 [0.257]	57.0% [32.1%]	35.8% [25.4%]	0.410 [0.252]
Housing	92.3% [14.9%]	40.0% [21.1%]	0.527 [0.212]	46.3% [35.6%]	20.7% [20.1%]	0.268 [0.232]
Parkinsons	86.2% [5.4%]	28.1% [11.4%]	0.411 [0.134]	75.5% [9.7%]	23.4% [11.0%]	0.344 [0.131]
Yacht	92.5% [12.4%]	70.9% [24.4%]	0.776 [0.190]	56.8% [38.7%]	44.0% [35.3%]	0.457 [0.323]

Table B25: (UCI) ANN regression models – Performance metrics information of **CSAFunction$_{Neg}$**

UCI Datasets Group	**CSAFunction$_{Neg}$ Metrics (Avg. [Std.]) In Training Dataset**			**CSAFunction$_{Neg}$ Metrics (Avg. [Std.]) In Testing Dataset**		
	Precision	Sensitivity	F-Measure	Precision	Sensitivity	F-Measure
Airfoil	89.4% [15.6%]	23.9% [14.5%]	0.357 [0.176]	67.0% [26.1%]	15.8% [11.5%]	0.240 [0.145]
AutoMPG	79.3% [26.7%]	30.3% [20.9%]	0.399 [0.210]	47.4% [34.1%]	18.3% [16.6%]	0.238 [0.184]
Combined	**79.6%**	**2.1%**	**0.039**	59.3%	1.5%	0.029

Cycle	**[23.6%]**	**[1.9%]**	**[0.035]**	[30.7%]	[1.8%]	[0.031]
Concrete	91.3% [5.4%]	40.6% [14.4%]	0.546 [0.136]	63.0% [18.6%]	25.5% [14.1%]	0.345 [0.152]
Energy	82.8% [12.0%]	53.7% [21.1%]	0.624 [0.180]	62.6% [21.2%]	45.4% [23.0%]	0.498 [0.195]
Housing	88.0% [13.5%]	36.0% [19.3%]	0.480 [0.185]	44.2% [33.7%]	20.7% [19.6%]	0.261 [0.214]
Parkinsons	84.0% [4.9%]	25.0% [10.9%]	0.372 [0.129]	72.1% [10.0%]	19.8% [9.6%]	0.298 [0.118]
Yacht	87.7% [24.1%]	56.2% [29.0%]	0.646 [0.266]	44.7% [38.1%]	38.7% [34.2%]	0.382 [0.314]

Section 5.3. Outcomes with the Experiments with ISO-NE Working Datasets

Table B26: (ISO-NE) Linear and ANN regression models RMSE Metrics.

ISO-NE Dataset	**Linear Regression Models**		**ANN Regression Models**	
	Training Dataset	**Testing Dataset**	**Training Dataset**	**Testing Dataset**
Boston	41.331	42.158	25.588	26.112
Bridgeport	56.513	57.617	36.675	32.076
Burlington	10.357	12.431	6.905	7.656
Concord	21.549	22.478	14.062	12.549
Portland	21.730	23.664	15.843	16.043
Providence	14.208	14.082	9.676	9.032
Windsor Locks	57.480	59.529	40.132	37.300
Worcester	30.880	32.495	20.546	21.609

Table B27: (ISO-NE) Linear regression models **$CScenario_{Pos}$** "Best" Threshold

ISO-NE Dataset	$CScenario_{POS}$"Best" Threshold		Training Dataset		Testing Dataset	
	α_{Pos}	δ_{Pos}	# of ON	% of ON	# of ON	% of ON
Boston	92.0%	48.445	179	8.0%	14	8.3%
Bridgeport	88.6%	51.353	254	11.4%	18	10.7%
Burlington	94.7%	14.615	119	5.3%	8	4.8%
Concord	93.2%	25.363	153	6.9%	10	6.0%
Portland	83.4%	17.656	370	16.6%	26	15.5%
Providence	89.1%	13.763	244	10.9%	19	11.3%
Windsor Locks	94.9%	82.822	114	5.1%	12	7.1%
Worcester	94.5%	40.534	124	5.6%	11	6.5%

Table B28: (ISO-NE) Linear regression models **CScenario$_{Neg}$** "Best" Threshold

ISO-NE Dataset	CScenario$_{Neg}$"Best" Threshold		Training Dataset		Testing Dataset	
	α_{Neg}	δ_{Neg}	# of ON	% of ON	# of ON	% of ON
Boston	13.4%	-43.728	298	13.4%	23	13.7%
Bridgeport	10.1%	-69.649	224	10.0%	18	10.7%
Burlington	10.6%	-10.611	235	10.5%	27	16.1%
Concord	11.0%	-23.119	243	10.9%	19	11.3%
Portland	10.9%	-25.591	242	10.8%	22	13.1%
Providence	14.4%	-13.495	320	14.3%	21	12.5%
Windsor Locks	9.9%	-69.525	220	9.9%	20	11.9%
Worcester	13.0%	-30.815	289	12.9%	23	13.7%

Table B29: (ISO-NE) ANN regression models **CScenario$_{Pos}$** "Best" Threshold

ISO-NE Dataset	CScenario$_{Pos}$"Best" Threshold		Training Dataset		Testing Dataset	
	α_{Pos}	δ_{Pos}	# of ON	% of ON	# of ON	% of ON
Boston	79.6%	19.923	455	20.4%	29	17.3%
Bridgeport	81.2%	32.750	420	18.8%	29	17.3%
Burlington	81.9%	5.532	404	18.1%	32	19.0%
Concord	84.5%	12.780	347	15.5%	23	13.7%
Portland	83.9%	14.536	359	16.1%	31	18.5%
Providence	86.1%	9.355	311	13.9%	23	13.7%
Windsor Locks	84.2%	33.284	353	15.8%	22	13.1%
Worcester	84.5%	18.569	345	15.5%	21	12.5%

Table B30: (ISO-NE) ANN regression models **CScenario$_{Neg}$** "Best" Threshold

ISO-NE Dataset	CScenario$_{Neg}$"Best" Threshold		Training Dataset		Testing Dataset	
	α_{Neg}	δ_{Neg}	# of ON	% of ON	# of ON	% of ON
Boston	16.0%	-23.280	356	15.9%	28	16.7%
Bridgeport	16.8%	-34.453	374	16.8%	20	11.9%
Burlington	17.2%	-6.071	382	17.1%	35	20.8%
Concord	13.0%	-13.927	288	12.9%	15	8.9%
Portland	19.8%	-12.757	440	19.7%	40	23.8%
Providence	1.65%	-8.138	368	16.5%	31	18.5%
Windsor Locks	15.9%	-35.432	354	15.9%	29	17.3%
Worcester	18.7%	-17.495	415	18.6%	31	18.5%

Table B31: (ISO-NE) Linear and ANN regression models of FN Penalty Values.

ISO-NE Dataset	Linear Regression Models		ANN Regression Models	
	$CSAFunction_{Pos}$ FN Penalty	$CSAFunction_{Neg}$ FN Penalty	$CSAFunction_{Pos}$ FN Penalty	$CSAFunction_{Neg}$ FN Penalty
Boston	1.229	1.000	1.000	1.082
Bridgeport	1.216	1.216	1.000	1.080
Burlington	1.357	1.357	1.000	1.000
Concord	1.000	1.000	1.083	1.090
Portland	1.095	1.095	1.000	1.072
Providence	1.000	1.000	1.191	1.000
Windsor Locks	1.098	1.098	1.082	1.000
Worcester	1.000	1.000	1.000	1.075

Table 32: (ISO-NE) Linear regression models – Performance metrics information of **$CSAFunction_{Pos}$**

ISO-NE Dataset	$CSAFunction_{Pos}$ Metrics In Training Dataset			$CSAFunction_{Pos}$ Metrics In Testing Dataset		
	Precision	Sensitivity	F-Measure	Precision	Sensitivity	F-Measure
Boston	80.0%	58.1%	0.673	75.0%	64.3%	0.692
Bridgeport	93.3%	44.1%	0.599	90.9%	55.6%	0.690
Burlington	94.4%	42.9%	0.590	100.0%	50.0%	0.667
Concord	92.6%	41.2%	0.570	100.0%	50.0%	0.667
Portland	81.6%	27.6%	0.412	75.0%	23.1%	0.353
Providence	89.5%	31.6%	0.467	87.5%	36.8%	0.519
Windsor Locks	95.4%	54.4%	0.693	100.0%	41.7%	0.588
Worcester	91.3%	50.8%	0.653	100.0%	54.5%	0.706

Table 33: (ISO-NE) Linear regression models – Performance metrics information of **$CSAFunction_{Neg}$**

ISO-NE Dataset	$CSAFunction_{Neg}$ Metrics In Training Dataset			$CSAFunction_{Neg}$ Metrics In Testing Dataset		
	Precision	Sensitivity	F-Measure	Precision	Sensitivity	F-Measure
Boston	87.7%	69.1%	0.773	85.7%	52.2%	0.649
Bridgeport	88.5%	62.1%	0.730	100.0%	50.0%	0.667
Burlington	78.5%	45.1%	0.573	88.9%	29.6%	0.444
Concord	86.2%	51.4%	0.644	88.9%	42.1%	0.571
Portland	86.3%	44.2%	0.585	81.8%	40.9%	0.545
Providence	83.9%	52.2%	0.644	81.8%	42.9%	0.563
Windsor Locks	90.6%	56.8%	0.698	100.0%	45.0%	0.621
Worcester	84.6%	53.3%	0.654	88.9%	34.8%	0.500

Table 34: (ISO-NE) ANN regression models – Performance metrics information of **$CSAFunction_{Pos}$**

ISO-NE Dataset	$CSAFunction_{Pos}$ Metrics In Training Dataset			$CSAFunction_{Pos}$ Metrics In Testing Dataset		
	Precision	**Sensitivity**	**F-Measure**	**Precision**	**Sensitivity**	**F-Measure**
Boston	80.2%	33.8%	0.476	70.0%	24.1%	0.359
Bridgeport	86.5%	39.5%	0.542	73.3%	37.9%	0.500
Burlington	82.6%	4.7%	0.089	0.0%	0.0%	0.000
Concord	86.3%	18.2%	0.300	100.0%	8.7%	0.160
Portland	66.7%	2.8%	0.053	100.0%	6.5%	0.121
Providence	100.0%	0.6%	0.013	0.0%	0.0%	0.000
Windsor Locks	94.0%	17.8%	0.300	50.0%	9.1%	0.154
Worcester	82.9%	33.6%	0.478	66.7%	28.6%	0.400

Table 35: (ISO-NE) ANN regression models – Performance metrics information of **$CSAFunction_{Neg}$**

ISO-NE Dataset	$CSAFunction_{Neg}$ Metrics In Training Dataset			$CSAFunction_{Neg}$ Metrics In Testing Dataset		
	Precision	**Sensitivity**	**F-Measure**	**Precision**	**Sensitivity**	**F-Measure**
Boston	84.3%	28.7%	0.428	69.2%	32.1%	0.439
Bridgeport	91.5%	23.0%	0.368	66.7%	20.0%	0.308
Burlington	81.3%	19.4%	0.313	50.0%	5.7%	0.103
Concord	74.7%	19.4%	0.309	25.0%	6.7%	0.105
Portland	78.9%	3.4%	0.065	66.7%	5.0%	0.093
Providence	86.8%	9.0%	0.163	50.0%	3.2%	0.061
Windsor Locks	82.4%	25.1%	0.385	66.7%	20.7%	0.316
Worcester	75.4%	25.1%	0.376	70.0%	22.6%	0.341

REFERENCE LIST

1. ABU-MOSTAFA, Y. S.; MAGDON-ISMAIL, M.; LIN, H.-T. Chapter 1 - The Learning Problem. In: **Learning From Data**. USA: AM Book, 2012a. p. 201.

2. ABU-MOSTAFA, Y. S.; MAGDON-ISMAIL, M.; LIN, H.-T. Chapter 2 - Training versus Testing. In: **Learning From Data**. USA: AM Book, 2012b. p. 201.

3. ABU-MOSTAFA, Y. S.; MAGDON-ISMAIL, M.; LIN, H.-T. Chapter 5 - Three Learning Principles. In: **Learning From Data**. USA: AM Book, 2012c. p. 201.

4. ATKESON, C. G.; MOORE, A. W.; SCHAAL, S. Locally Weighted Learning for Control. In: **Lazy Learning**. New York: Springer Link, 1997. p. 75–113.

5. BACHE, K.; LICHMAN, M. **UCI Machine Learning Repository**. Disponível em: <https://archive.ics.uci.edu/ml/datasets.html>. Acesso em: 1 dez. 2015.

6. BALASUBRAMANIAN, V. Statistical Inference, Occam's Razor and Statistical Mechanics on The Space of Probability Distributions. **Cornell University Library**, p. 17, 1996.

7. BASAWARAJ, G.; SUBHASH, K.; CHANDRASEKHAR, B. Novel Ensemble Neural Network Models for better Prediction using Variable Input Approach. **International Journal of Computer Applications**, v. 39, n. 18, p. 37–45, 2012.

8. BELL, J. Chapter 5 - Artificial Neural Networks. In: **Machine Learning: Hands-On for Developers and Technical Professionals**. USA: John Wiley & Sons, Inc., 2014. p. 91–116.

9. BOSNIC, Z. et al. Enhancing data stream predictions with reliability estimators and explanation. **Engineering Applications of Artificial Intelligence**, v. 34, p. 178–192, 2014.

10. BOSNIC, Z.; KONONENKO, I. Estimation of individual prediction reliability using the local sensitivity analysis. **Applied Intelligence**, v. 29, n. 3, p. 187–203, 2007.

11. BOSNIC, Z.; KONONENKO, I. Comparison of approaches for estimating reliability of individual regression predictions. **Data & Knowledge Engineering**, v. 67, n. 3, p. 504–516, 2008.

12. BOSNIC, Z.; KONONENKO, I. Correction of Regression Predictions Using the Secondary Learner on the Sensitivity Analysis Outputs. **Computing and Informatics**, v. 29, n. 6, p. 929–946, 2010.

13. BOSNIC, Z.; KONONENKO, I. Chapter 14 – Reliability Estimates for Regression Predictions: Performance Analysis. In: **Integrations of Data Warehousing, Data Mining and Database Technologies**. USA: IGI Global, 2011. p. 320–339.

14. BOX, G. E. P.; DRAPER, N. R. **Empirical Model-Building and Response Surfaces**. USA: John Wiley & Sons, Inc, 1987.

15. BREIMAN, L. Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). **Statistical Science**, v. 16, n. 3, p. 199–231, 2001.

16. CAETANO, M. A. L. **Mudanças Abruptas no Mercado Financeiro**. Sao Paulo/Brazil: Editora Erica, 2013.

17. CARNEY, J. G.; CUNNINGHAM, P.; BHAGWAN, U. Confidence and prediction intervals for neural network ensembles. **IJCNN'99. International Joint Conference on Neural Networks**, v. 2, p. 1215–1218, 1999.

18. DIETTERICH, T. G. Ensemble Methods in Machine Learning. **First International Workshop, Multiple Classifier Systems - MCS2000**, v. 1857, p. 1–15, 2000.

19. DRUCKER, H. Improving Regressors using Boosting Techniques. **ICML '97 Proceedings of the Fourteenth International Conference on Machine Learning**, p. 107–115, 1997.

20. DUDA, R. O.; HART, P. E.; STORK, D. G. Chapter 9 - Algorithm-Independent Machine Learning. In: **Pattern Classification**. USA: John Wiley & Sons, Inc., 2001. p. 680.

21. EFRON, B. Bootstrap Methods: Another Look at the Jackknife. **The Annals of Statistics**, v. 7, n. 1, p. 1–26, 1979.

22. ENDERS, W. Chapter 2 - Stationary Time-Series Models. In: **Applied Econometric Time Series**. USA: John Wiley & Sons, Inc., 2014a. p. 496.

23. ENDERS, W. Chapter 3 - Modeling Volatility. In: **Applied Econometric Time Series**. USA: John Wiley & Sons, Inc., 2014b. p. 496.

24. FAMA, E. F. Efficient Capital Markets: A Review of Theory and Empirical Work. **The Journal of Finance**, v. 25, n. 2, p. 383–417, 1970.

25. FINK, O.; ZIO, E.; WEIDMANN, U. Quantifying the reliability of fault classifiers. **Information Sciences**, v. 266, p. 65–74, 2014.

26. FORESEE, F. D.; HAGAN, M. T. Gauss-Newton approximation to Bayesian regularization. **The 1997 International Joint Conference on Neural Networks (IJCNN)**, p. 1930–1935, 1997.

27. GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. USA: Addison-Wesley Publishing Company, 1989.

28. GOUTTE, C.; GAUSSIER, E. A Probabilistic Interpretation of Precision, Recall and F -Score, with Implication for Evaluation. **27th European Conference on IR Research, ECIR 2005, Santiago de Compostela**, v. 3408, p. 345–359, 2005.

29. HAHN, G. J.; SHAPIRO, S. S. **Statistical Models in Engineering**. USA: John Wiley & Sons, Inc., 1994.

30. HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. Chapter 9 - Additive Models, Trees, and Related Methods. In: **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. USA: Springer, 2011a. p. 739.

31. HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. Chapter 7 - Model Assessment and Selection. In: **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. USA: Springer, 2011b. p. 739.

32. HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. Chapter 8 - Model Inference and Averaging. In: **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. USA: Springer, 2011c. p. 739.

33. HAYKIN, S. O. Chapter 4 - Multilayer Perceptrons. In: **Neural Networks and Learning Machines**. Thrid ed. USA: Prentice Hall, 2008. p. 936.

34. HESKES, T. Pratical Confidence and Prediction Intervals. **Advances in Neural Information Processing Systems 9**, p. 176–182, 1997.

35. HINSBERGEN, C. P. I.; LINT, J. W. C.; ZUYLEN, H. J. Bayesian committee of neural networks to predict travel times with confidence intervals. **Transportation Research Part C: Emerging Technologies**, v. 17, n. 5, p. 498–509, out. 2009.

36. HIPPERT, H. S.; PEDREIRA, C. E.; SOUZA, R. C. Neural networks for short-term load forecasting: a review and evaluation. **IEEE Transactions on Power Systems**, v. 16, n. 1, p. 44–55, 2001.

37. ISO-NE. **ISO New England**. Disponível em: <http://www.iso-ne.com/>. Acesso em: 1 dez. 2015.

38. JAMES, B. Capítulo 7 - O Teorema Central do Limite. In: **Probabilidade: Um Curso em Nível Intermediário**. Rio de Janeiro/Brazil: IMPA, 2009. p. 299.

39. JAVED, F. et al. Forecasting for demand response in smart grids: An analysis on use of anthropologic and structural data and short term multiple loads forecasting. **Applied Energy**, v. 96, p. 150–160, 2012.

40. JØRGENSEN, M.; SJØBERG, D. I. K. An effort prediction interval approach based on the empirical distribution of previous estimation accuracy. **Information and Software Technology**, v. 45, n. 3, p. 123–136, 2003.

41. KHOSRAVI, A. et al. Lower upper bound estimation method for construction of neural network-based prediction intervals. **IEEE transactions on neural networks**, v. 22, n. 3, p. 337–46, 2011.

42. KONONENKO, I. et al. Explanation and Reliability of Individual Predictions. **Informatica**, v. 37, n. 1, p. 41–48, 2013.

43. KOTSIANTIS, S.; KANELLOPOULOS, D.; PINTELAS, P. Handling imbalanced datasets : A review. **GETS International Transactions on Computer Science and Engineering**, v. 30, n. 1, p. 25–36, 2010.

44. KWOK, T. Y.; YEUNG, D. Y. Constructive algorithms for structure learning in feedforward neural networks for regression problems. **IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council**, v. 8, n. 3, p. 630–45, 1997.

45. LEHMANN, E. L.; ROMANO, J. P. **Testing Statistical Hypotheses**. USA: Springer, 2007.

46. MACKAY, D. J. C. Bayesian Interpolation. **Neural Computation**, v. 4, n. 3, p. 415–447, 1992.

47. MAGALHÃES, M. N.; LIMA, A. C. P. Capítulo 8 - Inferência Estatística - Testes de Hipóteses. In: **Noções de Probabilidade e Estatística**. São Paulo/Brazil: EDUSP, 2004. p. 392.

48. MARSLAND, S. **Machine Learning: An Algorithmic Perspective, Second Edition**. 2. ed. New York: Chapman and Hall/CRC, 2014.

49. MATHWORKS. **MATLAB Documentation**. Disponível em: <http://www.mathworks.com/help/index.html>. Acesso em: 1 dez. 2015.

50. MATSUMOTO, E.; PINTO, A. C. Aplicação de Redes Neurais na Classificação de Rentabilidade Futura de Empresas. **IX CBRN - Congresso Brasileiro de Redes Neurais**, 2009.

51. MATSUMOTO, E. Y.; DEL-MORAL-HERNANDEZ, E. Using neural networks committee machines to improve outcome prediction assessment in nonlinear

regression. **The 2013 International Joint Conference on Neural Networks (IJCNN)**, p. 1–8, 2013.

52. MATSUMOTO, E. Y.; DEL-MORAL-HERNANDEZ, E. Estimation of individual prediction reliability using error analysis applied to short-term load forecasting problem. **The 2014 International Joint Conference on Neural Networks (IJCNN)**, p. 4206–4313, 2014.

53. MCCULLAGH, P.; NELDER, J. A. **Generalized Linear Models**. USA: Chapman & Hall, 1989.

54. MEINSHAUSEN, N. Quantile Regression Forests. **Journal of Machine Learning Research**, v. 7, p. 983–999, 2006.

55. NGUYEN, G. H.; BOUZERDOUM, A.; PHUNG, S. L. Chapter 10 - Learning Pattern Classification Tasks with Imbalanced Data Sets. In: **Pattern Recognition**. China: Peng-Yeng Yin, 2009. p. 193–208.

56. PAPADOPOULOS, G.; EDWARDS, P. J.; MURRAY, A. F. Confidence Estimation Methods for Neural Networks. **ESANN'2000 - European Symposium on Artificial Neural Networks**, p. 75–80, 2000.

57. PAPADOPOULOS, H.; HARALAMBOUS, H. Reliable prediction intervals with regression neural networks. **Neural networks : the official journal of the International Neural Network Society**, v. 24, n. 8, p. 842–51, out. 2011.

58. PEARSON, R. **Exploring Data in Engineering, the Sciences, and Medicine**. United Kingdom: Oxford University Press, 2011.

59. PEVEC, D.; BOSNIC, Z.; KONONENKO, I. Chapter 3 - Individual Reliability Estimates in Classification and Regression. In: **Intelligent Data Analysis for Real-Life Application: Theory and Practice**. USA: IGI Global, 2012. p. 35–56.

60. PEVEC, D.; KONONENKO, I. Input dependent prediction intervals for supervised regression. **Intelligent Data Analysis**, v. 18, n. 5, p. 873–887, 2014.

61. PEVEC, D.; KONONENKO, I. Prediction intervals in supervised learning for model evaluation and discrimination. **Applied Intelligence**, v. 42, n. 4, p. 790–804, 2015.

62. POLIKAR, R. Bootstrap - Inspired Techniques in Computation Intelligence. **IEEE Signal Processing Magazine**, v. 24, n. 4, p. 59–72, 2007.

63. ROBERT, C. P.; CASELLA, G. **Monte Carlo Statistical Methods**. USA: Springer, 2010.

64. ROBNIK-ŠIKONJA, M.; KONONENKO, I. Theoretical and Empirical Analysis of ReliefF and RReliefF. **Machine Learning**, v. 53, n. 1-2, p. 23–69, 2003.

65. RODRIGUES, P. P. et al. Chapter 2 - Estimating Reliability for Assessing and Correcting Individual Streaming Predictions. In: **Reliable Knowledge Discovery**. USA: Springer, 2012. p. 29–49.

66. SAMARASHINGUE, S. Chapter 9 - Neural Networks for Time-Series Forecasting. In: **Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition**. USA: Auerbach Publications, 2006. p. 570.

67. SAMUEL, A. L. Some Studies in Machine Learning Using the Game of Checkers. **IBM Journal of Research and Development**, v. 3, n. 3, p. 210–229, 1959.

68. SAUNDERS, C.; GAMMERMAN, A.; VOVK, V. Transduction with Confidence and Credibility. **IJCAI '99 Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence**, p. 722–726, 1999.

69. STRUMBELJ, E.; KONONENKO, I. An Efficient Explanation of Individual Classifications using Game Theory. **The Journal of Machine Learning Research**, v. 11, p. 1–18, 2010.

70. SUN, Y. et al. Classification of Imbalanced Data: a Review. **International Journal of Pattern Recognition and Artificial Intelligence**, v. 23, n. 04, p. 687–719, 2009.

71. TEE, C. Y.; CARDELL, J. B.; ELLIS, G. W. **Short-term load forecasting using artificial neural networks**41st North American Power Symposium. **Anais**...IEEE, 2009

72. VOVK, V.; GAMMERMAN, A.; SAUNDERS, C. Machine-Learning Applications of Algorithmic Randomness. **Sixteenth International Conference on Machine Learning**, p. 444–453, 1999.

73. WAND, M. P.; JONES, M. C. **Kernel Smoothing**. United Kingdom: Chapman and Hall/CRC, 1995.

74. WELLEN, S.; DANKS, D. **Learning Causal Structure through Local Prediction-error Learning**COGSCI 2012 - Cognitive Science Society. **Anais**...Sapporo/Japan: 2012

75. WOLPERT, D. H. Stacked generalization. **Neural Networks**, v. 5, n. 2, p. 241–259, 1992.

76. WOOLDRIDGE, J. M. Chapter 3 - Multiple Regression Analysis: Estimation. In: **Introductory Econometrics: A Modern Approach**. USA: Thomson - South Western, 2009a. p. 881.

77. WOOLDRIDGE, J. M. Chapter 4 - Multiple Regression Analysis: Inference. In: **Introductory Econometrics: A Modern Approach**. USA: Thomson - South Western, 2009b. p. 881.

78. WOOLDRIDGE, J. M. Chapter 6 - Multiple Regression Analysis: Further Issues. In: **Introductory Econometrics: A Modern Approach**. US: Thomson - South Western, 2009c. p. 881.

79. ZADROZNY, B.; LANGFORD, J.; ABE, N. Cost-Sensitive Learning by Cost-Proportionate Example Weighting. **ICDM '03 Proceedings of the Third IEEE International Conference on Data Mining**, p. 435, 2003.

80. ZHOU, Z.-H.; WU, J.; TANG, W. Ensembling neural networks: Many could be better than all. **Artificial Intelligence**, v. 137, n. 1-2, p. 239–263, 2002.

81. ZILLOW. **What is Rent Zestimate?** Disponível em: <http://www.zillow.com/wikipages/What-is-a-Rent-Zestimate/>. Acesso em: 1 dez. 2015.