2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Estimation of Individual Prediction Reliability Using Error Analysis Applied to Short-Term Load Forecasting Problem

Élia Yathie Matsumoto and Emílio Del-Moral-Hernandez

***Abstract* — This work describes the methodology to create a reliability estimate for individual predictions in regressions. This estimate is defined as a binary variable which indicates if the regression prediction error of an individual unseen observation is likely to be critical or not, according to a meaningful criterion previously defined by the regression model user. The approach is based on the construction of a model to separate these two classes of error. The method was evaluated on sixteen experiments applied to short-time load forecasting regression problem using eight databases from ISO New England. In these experiments, the models for pattern recognition were built as ensembles composed of three classification models: K-Nearest Neighbors, Artificial Neural Network Committee Machine, and Support Vector Machine. The obtained results showed that the Ensemble Classifiers were able to detect critical error cases.**

I. Introduction

Regression models performance are commonly estimated by averaged error measures like Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE), or probabilistic confidence measures [1]-[3]. However, having additional information about single prediction reliability would be an unquestionable benefit, manly in risk-sensitive areas.

For this reason, research in the field of evaluation of reliability of individual predictions has significantly increased during the last decades.

The technical literature [4]-[7] usually divides the methods related to this area in two groups. The first group contains the methods that work with model-specific approaches. In this case, the methods are based on the regression model mathematical definition, and can even provide analytical solutions. The second group covers model-independent methods that handle the regression model as a "black-box", considering just its inputs and outputs. As a result, these methods can be more widely applied but rarely provide analytical solutions.

The majority of model-independent methods described in the literature are based on estimates generated using sensitivity analysis of the models outcomes affected by the insertion of perturbed data, such as: local sensitivity analysis reliability estimates [4], estimates generated by variance of bagged models [8], local cross-validation estimates [9], and density-based reliability estimates [10].

The results presented by these studies demonstrate that the methods can provide useful additional assessment, although they can be quite time consuming depending on the size of the dataset.

The methodology presented in this paper is a model-independent method; nonetheless, it uses an approach completely different from the previously mentioned ones.

It does not intend to replace or outperform anyone of them, on the contrary, the goal is to provide complementary information that may be used in combination with them or as an alternative when perturbed data generation is an issue.

The basic idea is to estimate if the regression prediction error of an unseen individual observation will be critical or not, according to a previously determined criterion that defines the critical error condition. Such criterion must be defined by the regression model user and has to be meaningful for the specific application of the regression system. It can be, for example, when the error falls out of a specific confidence interval or when it is higher than a threshold value.

Taking in account this predefined critical error criteria, the method proposes to classify the observations in positive (if the regression prediction error of the observation is critical) or negative (otherwise), and then constructing a model to separate these two classes of pattern.

The model for pattern recognition constructed and calibrated using the training dataset will attempt to capture the numerical limitations of the regression model and, when it is applied to testing dataset observations, it is expected to be able to estimate which ones are positive, i.e., more likely to produce regression prediction errors considered critical.

As it will be detailed ahead, one important challenge in this whole process is the design of the pattern recognition model because the dataset to be handled is probably highly imbalanced, due to the fact that, assuming that the regression model performs reasonably, the percentage of occurrences of critical errors is supposed to be small.

For the purposes of this study, the method was applied to short-term load forecasting regression problems, evaluated on sixteen experiments using databases from ISO New England, a regional transmission organization (RTO), serving Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island and Vermont.

The regression models were built using feedforward multilayered perceptron Artificial Neural Networks and the

Manuscript received January 20th, 2014. The databases used in this work were obtained from ISO New England. At the time of writing, the direct link to the spreadsheet data files was: http://iso-ne.com/markets/hstdata/znl_info/hourly/index.html.

E. Y. Matsumoto (e-mail: elia.matsumoto@usp.br) and E. Del-Moral-Hernandez (e-mail: emilio@lsi.usp.br) are with the Electronic System Department of University of Sao Paulo - Group of Computational Intelligence, Modeling and Electronic Neurocomputing.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

models for pattern recognition were constructed using Ensemble Models composed of three classification models: K-Nearest Neighbors, Artificial Neural Network Committee machine and Support Vector Machine.

The goal of this paper is developing and expanding the initial concept of this methodology that was first described in our previous work presented in 2013 [11].

II. METHODOLOGY

As mentioned before, the method uses pattern recognition techniques applied to errors analysis to provide regression model individual prediction assessment.

The objective is to estimate if the regression prediction error of an individual observation will be critical or not, given a previously defined Critical Error Criterion (CE Criterion); let us call it CE Criterion, from now on.

The proposition is to use the errors produced by the regression model over the training dataset to create a new variable, Critical Error Flag (CEFlag).

The CEFlag of an observation is set to 1 (positive case) if the regression prediction error of that observation is considered critical according to the CE Criterion; otherwise, it is set to 0 (negative case).

Thereby, this new binary variable CEFlag can be used to design a model to separate these two classes of pattern. In this text, this model will be called Critical Error Flag Estimation Function (CEFE Function).

The CEFE Function, as shown in (1), is calibrated using:
-- As input, the regression model training dataset, X_{train}, and its outcome, $\widehat{Y_{train}}$;
-- And, as output, the CEF Flag of the training dataset.

$$CEFlag_{train} = CEFE\ Function(X_{train}, \widehat{Y_{train}}) \quad (1)$$

Thus, the CEFE Function is conditioned to the constraints and limitations imposed by the regression model and the training dataset.

Assuming that training and testing datasets observations were generated by the same process, when applied to the testing dataset, the CEFE Function is prone to estimate the CEFlag values of the unseen observations.

In this case, the positive case (CEFlag equal 1) would indicate that the regression prediction error of the observation is likely to be critical, contrarily; the negative case would signalize low risk of critical error occurrence.

As a result of this new reliability estimate availability, the CEFlag prediction, we are able to treat distinctively the observations estimated as positive cases: they can be analyzed in detail for better support of decision-making process, or, when viable, just be discarded due low reliability. This is done ahead in the experiments dealing with real data presented in Section IV.

Supposing that the regression model accuracy is acceptable, the number of positive cases is supposed to be much lower than the negative, and, for this reason, one relevant task to be accomplished is the design of the model for pattern recognition, because standard pattern recognition algorithms that use MSE (Mean Squared Error) value optimization strategy tend to work well with balanced data, but to be biased towards the majority class in the case of imbalanced data [12].

For this reason, it is not recommended to use MSE value to evaluate or compare the CEFE Functions performance.

In this study, among several evaluation metrics described in the specialized literature [13]-[15], three of them in particular are observed: **Precision,** **Sensitivity**, and **FMeasure**.

The first metric, **Precision**, measures the percentage of positive predictions made by the model that are correct (2).

$$Precision = \frac{True\ Positive}{(True\ Positive + False\ Positive)} \quad (2)$$

The second metric, **Sensitivity**, also called **Recall**, measures the percentage of true positive patterns that are correctly detected by the model, or the accuracy on the positive cases (3).

$$Sensitivity = \frac{True\ Positive}{(True\ Positive + False\ Negative)} \quad (3)$$

The third metric, **FMeasure** or **FScore**, represents a harmonic mean between **Precision** and **Sensitivity** (4).

$$FMeasure = \frac{2 * Precision * Sensitivy}{(Precision + Sensitivity)} \quad (4)$$

According to the literature [13]-[15], in the case of extremely imbalanced datasets, **Sensitivity** values are often very low. In practice, it means that rare cases are usually hard to identify. In theory, this metric could be improved if lower **Precision** values were tolerated for the sake of higher **Sensitivity** values.

The harmonic mean of two numbers tends to be closer to the smaller one, so higher **FMeasure** values imply more balanced **Precision** and **Sensitivity** values.

In the experiments developed in this study, we compared the outcomes using **Prediction** and **FMeasure** metrics to define the "best" specific parameters for each one of the classifier models.

III. EXPERIMENTS

Although model performance optimization is not the main focus of this research, all models were constructed to achieve reasonable performance, but they were not ultimately optimized.

The proposed method was applied to the acknowledged time-series regression problem of short-term load forecasting that consists in forecast load variation one hour in advance [16]. The regression models were constructed based on Feedforward Multilayered Perceptron Artificial Neural Network (ANN).

The models for pattern recognition (the CEFE Functions) were constructed using Ensemble Models (EM) composed of three classification models recognized by the machine learning community [21]: K-Nearest Neighbors Classifier (KNN), feedforward multilayered perceptron artificial Neural Network Committee Machine Classifier (NCC) and Support Vector Machine Classifier (SVM).

A. ANN Regression Models

Following the architecture described and successfully proven by the several studies [17]-[20], the ANN regression models were constructed using three layers: Input, Hidden and Output. The number of neurons in the Input layer was defined by the number of input parameters. For the Hidden layer, twice the number of input parameters was used. The Output layer was created with one neuron.

The training method applied to calibrate the ANN was the Levenberg-Marquardt backpropagation algorithm with MSE performance function.

The ANN regression models were trained using subsets of the training datasets defined according to the Bagging (bootstrap aggregating) [8] ensemble learning method. For each ANN, the original training dataset was divided into three bootstrap sample subsets: a training subset with 70% of the original data used to calculate the weights and bias of the neural network; a control subset with the 15% of the original data used for cross-validation to avoid overfitting; and a verifying subset with the remaining 15% used to choose the "best" ANN. The selection criterion was the smallest MSE value for the verifying subset.

B. Critical Error Criterion (CE Criterion)

The particular choice for the criterion for differentiating what is considered a critical level of error and what is not is defined by the user of the regression model, according to what he/she considers useful and appropriate for the specific application and in consonance with the practices in the specific field.

The regression models were constructed to forecast load variation one hour in advance, however, in the field of load forecasting, regression model accuracy is conventionally evaluated by the measure of the Mean Absolute Percentage Error (MAPE) of the load value forecast, and a MAPE value around 1% is usually considered an indication of a high degree of accuracy [17].

For this reason, the error measurement adopted for the critical error criterion definition was the Absolute Percentage Error (APE) of the load value, instead of the Squared Error (SE) of the load variation. The CE Criterion for the experiments was defined as follows: if the load APE value is higher than 1.5% then the error is considered critical, and the CE Flag is set to positive.

C. Design of the Critical Error Flag Estimation Function (CEFE Function)

Classification of imbalanced dataset is an important problem in data mining that is present in the core of the design of the CEFE Functions.

In our previous work [9], where the foundation of this methodology was first presented, the CEFE Function was constructed as a NCC classifier composed of twenty individual ANN classifiers.

In this present work, this NCC classifier was combined with two other classifiers, K-Nearest Neighbors (KNN) and Support Vector Machine Gaussian radial basis function kernel (SVM), to build up the CEFE Function as an Ensemble Model (EM) with the outcome defined by simple majority voting.

The choice of a classifier, using EM architecture, was based on its improved prediction, accuracy, strong robustness and generalization capability attested to by numerous researches [20]-[24].

The training to calibrate the three classification models composing the ensemble was processed in two steps:

Step 1 - Specific parameters calibration: it consists in define the values of specific parameters for each kind of model, for instance: the number of K-neighbors, in the case of KNN; the cost function matrix values to be applied during the ANN training process; or the soft margin value of the SVM. It involves training the model with different parameter values and choosing the "best" one. In order to avoid overfitting [22], the training dataset was split in two: a cross-validation subset (20%) and training subset (80%). The criterion applied to choose the "best" parameter value was selecting the one that produced the highest **FMeasure** metric for the cross-validation subset. The same criterion was applied using the **Precision** metric.

Step 2 – Final Model training: it is the final calibration of the model using the "best" parameters defined in Step 1, and the complete training dataset.

Also, aiming to improve accuracy with imbalanced datasets, three techniques were applied to the training:

1) Cost-sensitive learning [12]: instead of using the standard MSE function, we used a weighted MSE function considering a cost matrix (5) to balance the false-positive and false-negative misclassification. *FP penalty* is the value that penalizes the false-positive outcomes, and *FN penalty*, the false-negatives.

$$Cost\ Matrix = \begin{bmatrix} 0 & FP\ penalty \\ FN\ penalty & 0 \end{bmatrix} \quad (5)$$

2) Cross-validation: besides its adoption to avoid overfitting, during ANN´s training process, cross-validation was also used to define: the numbers of K-neighbors and the values of the cost matrix of KNN models; the values of the cost matrix of the NCC Models; the soft-margins size of the SVM models.
3) Adapted bootstrap sampling [8]: in cross-validation, the bootstrap sampling was adapted to produce subsets with the same imbalanced proportion of the whole training dataset.

Additionally, in order to define the values of the specific parameters for the training runs described in *Step 1*, we propose the use of the proportion of positive cases from the training dataset, *Prop*, given by (6), as the underlying information to define the intervals of values to be tested.

$$Prop = \frac{Number\ of\ Positive\ Cases}{Total\ Number\ of\ Cases} \quad (6)$$

C1. KNN Classifier: the numbers of K-neighbors were defined by testing values starting from a maximum numbers of neighbors, MAX_N, decreasing it until two. The number MAX_N was arbitrarily defined as the nearest integer to the inverse of *Prop* (7), i.e., the theoretical number of negative case neighbors of each positive case observation [26].

$$MAX_N = fix\left(\frac{1}{Prop}\right) \quad (7)$$

The values of the cost matrix were defined by keeping the *FP Penalty* value fixed at one, and testing values for *FN Penalty* produced by values ranging from *Prop* until 0.5, as defined in (8):

$$FN\ Penalty = \frac{1}{2 * Pr}, Pr \in [Prop, 0.5]. \quad (8)$$

Pr equal 0.5 means no penalty (*FN Penalty = 1*). The maximum *FN Penalty* value is given by the half to the inverse of the positive case proportion [25].

C2. NCC Classifier: The same procedure described to define the *FP Penalty* in the case of *KNN Classifiers* was applied to the *NCC Classifiers*.

C3. SVM Classifier: The values of the soft margins, SM_1 and SM_0, were defined by testing values according to the (9) and (10).

$$SM_1 = \frac{1}{2 * Pr} \quad (9) \qquad SM_0 = \frac{1}{2 * (1 - \mathrm{Pr})} \quad (10)$$

Again, the testing values were produced ranging *Pr* from *Prop* until 0.5.

In the case of *Pr* equal 0.5, both values, SM_1 and SM_0 are equal one that indicates symmetric soft margins. Smaller values of *Pr* produce wider soft margin for positive cases (higher SM_1 values), and narrower soft margin for negative cases (lower SM_0 values) [27].

IV. Data Description and Numerical Results

The methodology was evaluated on sixteen experiments using public data accessible on ISO New England website. From a total of sixteen variables available in the original data files (see APPENDIX I), five were selected to build the working datasets:

-- 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 datasets were composed with one dependent variable (output), V_t, and eight independent variables (inputs), as shown in TABLE I.

TABLE I
MODEL VARIABLES

Variable	Description
V_t	Load Variation in t, one hour in advance
V_{t-1}	Load Variation in (t-1), at the hour
V_{t-24}	Load Variation in (t-24), at the hour one day before
V_{t-168}	Load Variation in (t-168), at the hour one week before
L_{t-1}	Load in (t-1), at the hour
cH_{t-1}	*cos*(Hour(t-1)*π/12); Hour values between 1 and 24
B_{t-1}	Boolean flag: is Busy day in (t-1)? No (0) or Yes(1)
Dr_{t-1}	Dry bulb temperature in (t-1) in degrees Fahrenheit
De_{t-1}	Dew point temperature in (t-1) in degrees Fahrenheit

The variables $V_t, V_{t-1}, V_{t-24}, V_{t-168}$, and L_{t-1} were derived from the original DEMAND variable; cH_{t-1}, from the original Hour variable; B_{t-1}, from the original Date variable; Dr_{t-1}, from the DryBulb; and De_{t-1}, from the DewPnt.

The data were collected from eight stations of ISO New England Control Area: Boston, Bridgeport, Burlington, Concord, Portland, Providence, Windsor Locks and Worcester.

Two working datasets were created for each one of the eight stations, one to predict the load variation during winter season and another to predict it during summer. In this way, we have a total of sixteen predictive regressors. Each of the sixteen dataset was then split into two sub-datasets: training and testing.

TABLE II shows the general description of the sub-datasets composition.

TABLE II
SUB-DATASETS DESCRIPTION

Dataset	In-Sample sub-dataset	Out-of-Sample sub-dataset
Winter	From Nov 1st 2011, 1a Until Nov 30th 2012, 12p (9504 observations)	From Dec 1st 2011, 1a Until Nov 30th 2012, 12p (744 observations)
Summer	From May 1st 2012, 1a Until May 31st 2013, 12p (9504 observations)	From Jun 1st 2013, 1a Until Jun 30th 2013, 12p (720 observations)

In this paper, the experiments will be identified by the first three letters of the control area followed by "W" for the winter datasets, and "S" for the summer.

A. ANN Regression Models Outcomes

All ANN regressions were structured as Integrated Auto Regressive with Exogenous Inputs models [3], as in formula (11) using the variables listed in TABLE I.

$$V_t = G(V_{t-1}, V_{t-24}, V_{t-168}, X_{t-1}) \quad (11)$$

Exogenous Inputs: $X_{t-1} = [L\ cH\ B\ Dr\ De]_{t-1}$

They were implemented in MATLAB with Neural Network Toolbox, and constructed according to the design decisions described in Session III.A, and following best practices recommended by the literature [28], [29] to improve generalization, such as, early stopping and regularization.

TABLE A and TABLE B in APPENDIX II, show the ANN regression models performance metrics obtained in our experiments for training and testing datasets, respectively: load variation RMSE, Adjusted R-Squared, and load value MAPE.

According to these metrics, all ANN regression models achieved good performance in both cases (Adj-RSquared values higher than 0.940).

Fig. 1. shows the scatter plot (observation against prediction) for training and testing datasets for the Portland control area forecast in winter (PorW experiment).

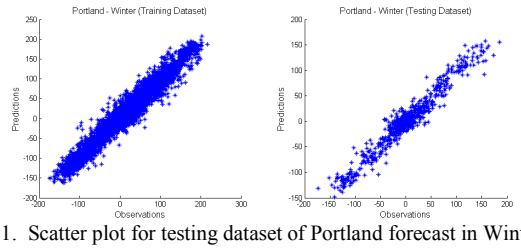

Fig. 1. Scatter plot for testing dataset of Portland forecast in Winter.

This regression was the one that produced the lowest Adjusted R-squared value in training datasets. It is possible to visually confirm that, even in this case, the regression achieved reasonable effectiveness.

TABLE III shows the performance metrics average in testing datasets.

TABLE III
REGRESSION METRICS AVERAGE IN TESTING DATASETS

RMSE (Load Var.)	Adj-RSqr (Load Var.)	MAPE (Load)
21.566	0.958	0.788%

B. Critical Error Criterion (CE Criterion)

As mentioned in Session III.B., the CE Criterion adopted in the experiments was the load APE higher than 1.5% (12).

$$L_t = V_t + L_{t-1} \quad (actual\ Load\ value)$$

$$\widehat{L_t} = \widehat{V_t} + L_{t-1}\ (forecasted\ Load\ value)$$

$$APE_t = \frac{abs(L_t - \widehat{L_t})}{L_t} \quad (Load\ APE)$$

$$C\varepsilon_t = \begin{cases} 0\,;if\ APE_t \le 1.5\% \\ 1\,;otherwise \end{cases} \qquad (12)$$

In average, 10.901% of the observations in training datasets were classified as positive, and 13.532% in testing datasets. The classifications defined by the CE Criterion were used to construct the models for pattern recognition.

C. Critical Error Flag Estimation Function (CEFE Function)

All CEFE Functions were implemented in MATLAB with Neural Network Toolbox and Statistics Toolbox, and constructed according to the references mentioned in Session III.C.

The specific parameters of each one of the classifiers, KNN, NCC, and SVM, were defined according to the design process described in Session III.C. As mentioned, the final outcome was defined by simple majority voting.

The values obtained using **FMeasure** and **Precision** metrics as selecting criterion, are listed on TABLE C and D in APPENDIX II.

TABLE IV shows the metrics average produced by the EM CEFE Function for training and testing datasets, using **FMeasure** metric as specific parameters selection criterion.

TABLE IV
ENSEMBLE MODEL CEFE FUNCTION METRICS AVERAGE (FMEASURE)

Dataset	Precision	Sensitivity	FMeasure
Training	44.556%	59.474%	0.503
Testing	37.349%	40.367%	0.381

As expected, the EM CEFE Function performed better for the training datasets than for the testing datasets. Even so, the testing datasets **Prediction** metric average value (37.349%) was close to three times the theoretical random drawing rate given by the average proportion of the positive cases (13.532%), and the **Sensitivity** metric average value indicates that more than 40% of the critical error occurrences correctly were detected.

TABLE V shows the same metrics using **Precision** metrics.

TABLE V
ENSEMBLE MODEL CEFE FUNCTION METRICS AVERAGE (PRECISION)

Dataset	Precision	Sensitivity	FMeasure
Training	85.090%	8.966%	0.160
Testing	60.318%	5.453%	0.095

In this case, as a consequence, higher **Precision** metric values were achieved but with much lower **Sensitivity** metric rates.

The EM CEFE Functions performances for each one of the sixteen experiments for testing datasets are listed in TABLE E and F in APPENDIX II.

Some of the central concepts of the methodology are illustrated in Fig. 2, 3, and 4.

Fig. 2. shows an example of seventy two hours of actual observed load values against ANN regression predictions (predicted load variation plus one hour before observed load value) in the Bridgeport control area during winter, from Dec 26th 2012 (2am) until Dec 29th 2012 (1am).

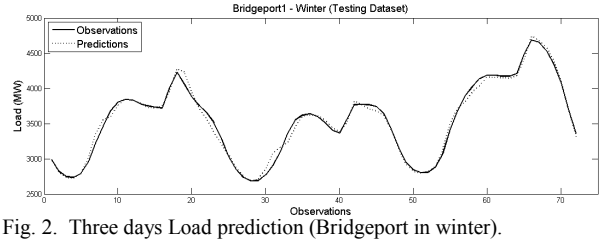

Fig. 2. Three days Load prediction (Bridgeport in winter).

Fig. 3. depicts the APE load produced by the regression forecast, during the same time frame, the critical error threshold value (1.5%), the estimated positive cases (CEFlag equal 1), as well the negative cases (CEFlag equal 0), using the **FMeasure** metric.

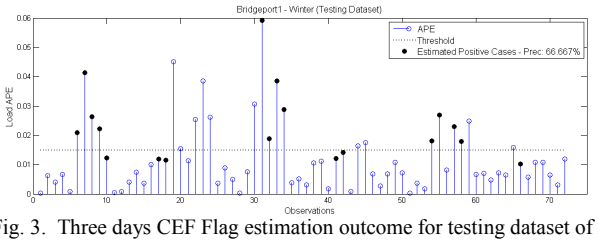

Fig. 3. Three days CEF Flag estimation outcome for testing dataset of Bridgeport in Winter, using the **FMeasure** metric.

Fig. 4. shows the same information, using the **Precision** metric.

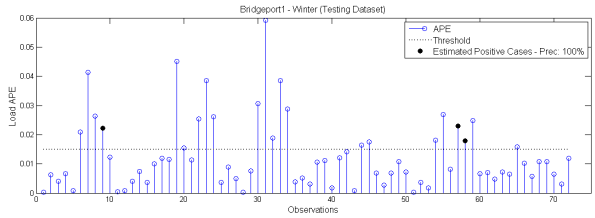

Fig. 4. Three days CEF Flag estimation outcome for testing dataset of Bridgeport in Winter, using the **Precision** metric.

In load demand management, the CEFlag prediction may be used in several areas. For instance, in operations, it can be

checked to trigger load balance procedures with one hour in advance instead of in real-time, which would usually be riskier and more expensive.

In maintenance, it can help to identify instrument failures or noisy data, since unseen observations with values affected by these kinds of problems tend to produce wrong predictions and to be estimated as positive cases. In energy trading, there is the option of do no trade when the information is not reliable, so the estimated positive cases could be just discarded. The exclusion of positive cases is supposed to avoid wrong actions or operations based in low reliable information.

TABLE VI summarizes the effect of the exclusion of the observations estimated as positive case over the regression performance metrics in testing datasets. It shows the percentage of improvement over the regression metrics values in TABLE III.

TABLE VI
REGRESSION METRICS AVERAGE IN TESTING DATASETS AFTER POSITIVE ESTIMATED CASES EXCLUDED (FMEASURE)

RMSE (% improv.)	Adj-RSqr (% improv.)	MAPE (% improv.)
18.395 (17.240%)	0.962 (0.434%)	0.691% (12.340%)

For comparison purposes, TABLE VII shows the same information, in the case of the use of **Prediction** metric.

TABLE VII
REGRESSION METRICS AVERAGE IN TESTING DATASETS AFTER POSITIVE ESTIMATED CASES EXCLUDED (PRECISION)

RMSE (% improv.)	Adj-RSqr (% improv.)	MAPE (% improv.)
21.101 (2.158%)	0.959 (0.128%)	0.773% (1.889%)

These results demonstrate that the use of **FMeasure** produced higher improvement rates and it would be recommended when the balance between sensitivity and precision is important.

The improvement rates in all sixteen experiments are displayed in TABLE F - APPENDIX II.

V. CONCLUSION

The results of the experiments provide evidences that the models for pattern recognition were able to estimate individual positive cases in testing datasets.

In other words, they demonstrated that the presented methodology is capable to provide reliability estimate for individual predictions in regressions, the CEFlag, which can be used as an additional assessment to better support tasks like decision making, measurement error detection, noisy data, outlier identification, and general data analysis and investigation.

The procedure described in this paper imposes no restrictions on the type of the regression model to be used or the critical error criterion (CE Criterion) to be adopted. The only assumption is that training and testing datasets observations are generated by the same process.

Due to space constraints, the outcomes and the evaluation metrics of the individual KNN, NCC, and SVM classification models are not present in this document.

However, it is worth mentioning that, in the experiments, the Ensemble Model classifiers (EM CEFE Functions) achieved the best balanced performance reaching highest FMeasure values. In average, SVM classifiers accomplished higher Precision values and lower Sensitivity. On the other hand, NCC and KNN classifiers performed just the opposite.

This indicates that CEFE Function modeling is still a work in progress. Further research should evaluate other strategies to solve the imbalanced class problem.

An additional contribution of this work is the proposal of the used of the proportion of positive cases in the training dataset as underlying information to define the value of specific parameters of the classifiers: the numbers of K-neighbors, the penalty values and the soft margin values.

In this study, we observed the effects of the exclusion of observations estimated as positive case only from the testing datasets. Further research should investigate how the CEFlag information of the training datasets could be applied to help improving regression models accuracy.

With regard to the nature of the estimate information, alternatively, we could explore the potential advantages of using softer index instead of the binary CEFlag.

Finally, the research could continue in the direction of extend the utilization of the presented methodology in combination with other reliability estimate methods cited in Section I.

As just mentioned, only the results of EM classifiers are listed in this paper. However, all experimental results of each one of the three individual classifiers are available and can be provided by the authors on request, as well the working datasets and the programming code used in the experiments.

APPENDIX I: VARIABLES DESCRIPTION PROVIDED BY ISO NEW ENGLAND

Date: date in MM/DD/YYYY format.
Hour: hour ending value.
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.
SYSLoad: actual system load.
RegCP: Regulation clearing price.

APPENDIX II: COMPLEMENTARY TABLES

TABLE A
ANN REGRESSION MODELS OUTCOMES FOR TRAINING DATASET

Experiment	RMSE (Load Var.)	Adj-RSqr (Load Var.)	MAPE (Load)
BosW	26.241	0.963	0.630%
BosS	25.811	0.964	0.623%
BriW	37.028	0.963	0.756%
BriS	36.680	0.963	0.732%
BurW	6.274	0.961	0.692%
BurS	6.207	0.962	0.685%
ConW	13.039	0.968	0.714%
ConS	12.309	0.971	0.672%
PorW	14.463	0.946	0.874%
PorS	14.068	0.948	0.840%
ProW	10.756	0.952	0.814%
ProS	10.356	0.955	0.789%
WinW	37.780	0.961	0.759%
WinS	35.947	0.965	0.715%
WorW	17.936	0.967	0.644%
WorS	17.453	0.968	0.626%

TABLE B
ANN REGRESSION MODELS OUTCOMES FOR TESTING DATASET

Experiment	RMSE (Load Var.)	Adj-RSqr (Load Var.)	MAPE (Load)
BosW	27.060	0.960	0.668%
BosS	30.679	0.957	0.779%
BriW	36.026	0.966	0.737%
BriS	41.890	0.960	0.862%
BurW	7.857	0.949	0.830%
BurS	6.731	0.951	0.765%
ConW	13.903	0.968	0.739%
ConS	13.831	0.964	0.761%
PorW	15.660	0.948	0.904%
PorS	13.722	0.944	0.853%
ProW	10.219	0.958	0.802%
ProS	11.602	0.950	0.908%
WinW	37.220	0.963	0.776%
WinS	41.672	0.960	0.880%
WorW	17.505	0.972	0.605%
WorS	19.481	0.961	0.738%

TABLE C
CLASSIFIERS: SPECIFIC PARAMETERS (FMEASURE)

Experiment	KNN #Neighbors	KNN Penalty	NCC Penalty	SVM Soft Margin (Pos.Case)
BosW	11	6.155	6.155	2.689
BosS	10	6.092	6.092	2.680
BriW	6	2.332	4.194	2.332
BriS	6	2.416	4.578	2.416
BurW	10	2.562	5.345	2.562
BurS	10	2.557	5.315	5.315
ConW	9	4.884	4.884	2.478
ConS	4	5.788	5.788	2.635
PorW	6	2.013	3.040	2.013
PorS	6	2.064	3.198	2.064
ProW	7	3.562	3.562	2.171
ProS	4	3.817	3.817	2.240
WinW	7	4.090	4.090	2.307
WinS	4	4.771	4.771	2.456
WorW	6	2.682	2.682	2.682
WorS	3	2.760	6.674	2.760

TABLE D
CLASSIFIERS: SPECIFIC PARAMETERS (PRECISION)

Experiment	KNN #Neighbors	KNN Penalty	NCC Penalty	SVM Soft Margin (Pos.Case)
BosW	10	1,265	1,720	1,000
BosS	8	1,000	1,000	1,000
BriW	6	1,000	1,000	1,000
BriS	2	1,000	1,641	1,243
BurW	8	1,000	1,255	1,000
BurS	5	1,000	1,683	1,000
ConW	8	1,000	1,000	1,000
ConS	6	1,000	1,000	1,705
PorW	3	1,000	1,000	1,505
PorS	4	1,000	1,207	1,524
ProW	6	1,000	1,000	1,000
ProS	5	1,000	1,000	1,000
WinW	6	1,000	1,000	1,000
WinS	4	1,000	1,246	1,246
WorW	10	1,000	1,264	1,000
WorS	8	1,000	1,000	1,270

TABLE E
EM CEFE FUNCTION METRICS FOR TESTING DATASETS (FMEASURE)

Experiment	Precision	Sensitivity	F-measure
BosW	33.981%	47.945%	0.398
BosS	31.481%	36.170%	0.337
BriW	50.820%	68.889%	0.585
BriS	40.816%	35.398%	0.379
BurW	36.036%	38.095%	0.370
BurS	26.582%	23.596%	0.250
ConW	36.036%	50.000%	0.419
ConS	35.484%	36.264%	0.359
PorW	42.857%	26.087%	0.324
PorS	28.302%	13.636%	0.184
ProW	39.286%	42.308%	0.407
ProS	39.535%	40.157%	0.398
WinW	42.953%	64.000%	0.514
WinS	38.961%	46.154%	0.423
WorW	38.462%	44.643%	0.413
WorS	36.000%	32.530%	0.342

TABLE F
EM CEFE FUNCTION METRICS FOR TESTING DATASETS (PRECISION)

Experiment	Precision	Sensitivity	F-measure
BosW	75,000%	4,110%	0,078
BosS	0,000%	0,000%	0,000
BriW	68,750%	12,222%	0,208
BriS	0,000%	0,000%	0,000
BurW	62,500%	4,762%	0,088
BurS	0,000%	0,000%	0,000
ConW	80,000%	10,000%	0,178
ConS	100,000%	4,396%	0,084
PorW	60,870%	10,145%	0,174
PorS	75,000%	2,727%	0,053
ProW	66,667%	9,615%	0,168
ProS	80,000%	3,150%	0,061
WinW	62,963%	17,000%	0,268
WinS	100,000%	0,769%	0,015
WorW	33,333%	7,143%	0,118
WorS	100,000%	1,205%	0,024

TABLE G
REGRESSION METRICS IN TESTING DATASETS AFTER POSITIVE ESTIMATED CASES EXCLUDED (IMPROV. OVER VALUES IN TABLE B)

Experiment	RMSE (Load Var.)	Adj-RSqr (Load Var.)	MAPE (Load)
BosW	21.748	0.968	0.549%
(Improv.%)	19.632%	0.841%	17.841%
BosS	27.533	0.961	0.704%
(Improv.%)	10.255%	0.381%	9.620%
BriW	27.079	0.977	0.566%
(Improv.%)	24.835%	1.144%	23.267%
BriS	38.587	0.963	0.776%
(Improv.%)	7.885%	0.286%	9.957%
BurW	6.577	0.957	0.736%
(Improv.%)	16.300%	0.843%	11.348%
BurS	6.172	0.954	0.713%
(Improv.%)	8.306%	0.351%	6.716%
ConW	11.246	0.975	0.633%
(Improv.%)	19.112%	0.670%	14.447%
ConS	12.007	0.965	0.679%
(Improv.%)	13.189%	0.094%	10.806%
PorW	13.732	0.951	0.827%
(Improv.%)	12.315%	0.378%	8.483%
PorS	13.492	0.941	0.830%
(Improv.%)	1.678%	-0.354%	2.713%
ProW	21.748	0.968	0.549%
(Improv.%)	19.632%	0.841%	17.841%
WinW	8.534	0.964	0.684%
(Improv.%)	16.491%	0.586%	14.742%
WinS	10.730	0.941	0.800%
(Improv.%)	7.520%	-0.898%	11.878%
WorW	27.763	0.973	0.593%
(Improv.%)	25.409%	1.046%	23.594%
WorS	35.991	0.966	0.754%
(Improv.%)	13.634%	0.584%	14.326%
WinW	8.534	0.964	0.684%
(Improv.%)	16.491%	0.586%	14.742%
WinS	10.730	0.941	0.800%
(Improv.%)	7.520%	-0.898%	11.878%
WorW	27.763	0.973	0.593%
(Improv.%)	25.409%	1.046%	23.594%
WorS	35.991	0.966	0.754%
(Improv.%)	13.634%	0.584%	14.326%

REFERENCES

[1] P. McCullagh and J. Nelder, *Generalized Linear Models*, Chapman & Hall, 1994.

[2] J. Wooldridge, *Introductory Econometrics: A Modern Approach, 2nd* Ed., Thomson South-Western, 2003.

[3] W. Enders, *Applied Econometric Time Series*, John Wiley & Sons, Inc, 2004.

[4] Z. Bosnic and I. Kononenko - "Estimation of individual prediction reliability using the local sensitivy analysis" – Applied Intelligence, Vol. 29, Issue 3, pp. 187-203, Springer, 2008.

[5] Z. Bosnic and I. Kononenko, "*Comparision of approaches for estimating reliability of individual regression predictions*" – Data & Knowledge Engineering 67, pp. 504-516, Elsevier, 2008.

[6] Z. Bosnic and I. Kononenko, *"Chapter 14 - Reliability Estimates for Regression Predictions: Performance Analysis"* in D. Taniar and L. Chen, *Integrations of Data Warehousing, Data Mining and Database Technologies*, pp. 320-339, IGI Global, 2011.

[7] P. P. Rodrigues, Z. Bosnic, J. Gama, and I. Kononenko – *"Chapter 2 - Estimating Reliability for Assessing and Correcting Individual Streaming Predictions"* – in H. Dai, J. Lui, and E. Smirnov – Reliable Knowledge Discovery, Springer, 2012.

[8] R. Polikar – "*Bootstrap-Inspired Technique in Computational Intelligence*" - IEEE SIGNAL IEEE Signal Processing Magazine, 59, May/2007.

[9] D. Girard – *"Estimating the accuracy of (local) cross-validation via randomized GCV choices in kernel or smoothing spline regression"* – Journal of Nonparametric Statistics, Volume 22, Issue 1, Taylor Francis Online, 2010.

[10] H. Dai, H. Zhang and W. Wang – *"A support vector density-based importance sampling for reliability assessment"* – Reliability Engineering & System Safety, Volume 106, pp. 86-93, Elsevier, 2012.

[11] E. Y. Matsumoto and E. Del-Moral-Hernandez – *"Using Neural Networks Committee Machines to Improve Outcome Prediction Assessment in Nonlinear Regression"* – in Proc. ICJNN 2013, Dallas, USA, 2013.

[12] G. H. Nguyen, A. Bouzerdoum, and S. L. Phung - "Chapter 10 - Learning Pattern Classification Tasks with Imbalanced Data Sets" - in Pattern Recognition, Peng-Yeng Ying (Ed.), 2009.

[13] S. Kotsiantis, D. Kanellopoulos and P. Pintelas – *"Handling imbalanced datasets: A Review"* – GESTS International Transactions on Computer Science and Engineering, Vol. 30, 2006.

[14] Y. Sun, A. Wong and M. Kamel – *"Classification of Imbalanced Data: A Review"* – International Journal of Pattern Recognition and Artificial Intelligence, Vol. 23, No 4, pp. 687-719, World Scientific Publishing Company, 2009.

[15] C. Goutte and E. Gaussier – "*A Probabilistic Interpretation of Precision, Recall and F-score, with Implication for Evaluation*" – Proceedings of the European Colloquium on IR Research (ECIR'05), LLNCS 3408, pp. 345-359, Springer, 2005.

[16] H. Hahn, S. Meyer-Nieberg, and S. Pickl – "Electric Load Forecasting Methods: Tools for Decision Making" – European Journal of Operational Research, Vol. 199, Issue 3, pp. 902-907, 2009.

[17] S. K. Sheikh and M. G. Unde, *"Short-Term Load Forecasting using ANN Technique"* – International Journal of Engineering Sciences & Emerging Technologies, Vol. 1, Issue 2, pp. 97-107, 2012.

[18] H. S. Hippert, C. E. Pedreira and R. C. Souza – *"Neural Networks for Short-term Load Forecasting: a review and evaluation"* – Power Systems, IEEE Transaction, Vol. 16, Issue 1, pp. 44-55, 2001.

[19] Z. H. Osman, M. L. Awad and T. K. Mahmoud – *"Neural Network Based Approach for Short-term Load Forecasting"* – Power Systems Conference and Exposition, PSCE '09, pp. 1-8, 2009.

[20] K. Siwek, S. Osowski and R. Szupiluk – *"Ensemble Neural Network Approach for Accurate Load Forecasting in a Power System"* – International Journal of Applied Mathematics and Computer Science, Vol 19, Number 2, pp. 303-315, 2009.

[21] R. Duda, D. Stork, P. Hart, *Pattern Classification* – John Wiley & Sons (2005).

[22] Z.H. Zhou, J.Wu and W.Tang – "*Ensembling neural networks: Many could be better than all*" – Science Direct, 2001.

[23] M. Sewell – *"Ensemble Learning"* – University College London Department of Computer Science, 2011.

[24] A.M.S.Yaser, I.M.Magdon, H.T.Lin, *Learning From Data*, AMLBook, 2012.

[25] B. J. Zadrozny, J. Langford and N. Abe – *"Cost-Sensitive Learning by Cost-Proportionate Example Weighting"* - ICDM 2003 - Third IEEE International Conference on Data Mining, Melbourne/FL, USA, 2003.

[26] N. Bhatia – *"Survey of Nearest Neighbor Techniques"* – IJCSIS (International Journal of Computer Science and Information Security), Vol. 8, No 2, pp. 302-305, 2010.

[27] A. Ben-Hur and J. Weston – *"A User's Guide to Support Vector Machines"* – Data Mining Techniques for the Life Sciences Methods in Molecular Biology, Vol. 609, pp.223-239, 2010.

[28] S. Haykin, *Neural Network and Learning Machines (3rd Edition),* Pearson Education, Inc. 2009.

[29] S. Samarasinghe, *Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition*, Auerbach Publications, 2007.