# Non removal strategy for outliers in predictive models: The PAELLA algorithm case

**Manuel Castejón-Limas[1,*], Hector Alaiz-Moreton[2], Laura Fernández-Robles[1], Javier Alfonso-Cendón[1], Camino Fernández-Llamas[1], Lidia Sánchez-González[1] and Hilde Pérez[1]**

[1]Department of Mechanical, Computer Science and Aerospace Engineering, Universidad de León, Campus de Vegazana s/n, Léon, 24071, Spain

[2]Department of Electrical, Systems and Automatic Engineering, Universidad de León, Campus de Vegazana s/n, León, 24071, Spain

*Corresponding E-mail: manuel.castejon@unileon.es

E-mails: hector.moreton@unileon.es (Hector Alaiz-Moreton); l.fernandez@unileon.es (Laura Fernández-Robles); javier.alfonso@unileon.es (Javier Alfonso-Cendón); camino.fernandez@unileon.es (Camino Fernández-Llamas); lidia.sanchez@unileon.es (Lidia Sánchez-González); hilde.perez@unileon.es (Hilde Pérez)

## Abstract

This paper reports the experience of using the PAELLA algorithm as a helper tool in robust regression instead of as originally intended for outlier identification and removal. This novel usage of the algorithm takes advantage of the occurrence vector calculated by the algorithm in order to strengthen the effect of the more reliable samples and lessen the impact of those that otherwise would be considered outliers. Following that aim, a series of experiments are conducted in order to learn how to better use the information contained in the occurrence vector. Using a contrively difficult artificial dataset, a reference predictive model is fit using the whole raw dataset. The second experiment reports the results of fitting a similar predictive model but discarding the samples marked as outliers by PAELLA. The third experiment uses the occurrence vector provided by PAELLA in order to classify the observations in multiple bins and fit every possible model changing which bins are considered for fitting and which are discarded in that particular model. The fourth experiment introduces a sampling process before fitting in which the occurrence vector represents the likelihood of being considered in the training dataset. The fifth experiment considers the sampling process as an internal step to be performed interleaved between the training epochs. The last experiment compares our approach using weighted neural networks to a state of the art method.

**Keywords:** probabilistic, sampling, outlier detection, PAELLA, weighted regression

# 1 Introduction

Datasets are the cornerstone upon which machine learning builds predictive models. Most sensibly, companies dealing with the optimization [6, 12, 13] of their factories have discover the rich potential of these resources: large amounts of data are recorded from real manufacturing processes live while operating, and machine learning provides the right tools to manage that complexity and extract useful information. On such scenario, though, the datasets are polluted with outliers and noise more frequently [11]. One drastic approach consists of preprocessing a dataset in order to remove the outliers [3]. A less aggressive approach consists of using a robust algorithm, based on backpropagation learning, for the task at hand that can cope with the presence of such perturbations [19]. This approach takes advantage of all the information comprised on the dataset but, unfortunately, it requires the existence of such robust alternative.

Many studies deal with robust regression without elimination of outliers in many fields such as computer vision to estimate an image background under heavy noise [18], electroencephalogram signal regression [7] and for tribological systems to model friction [8], among many others. At the same time, there is a wide range of possible models that have been used in the literature to perform robust regression, for example support vector regression [10], exponential-type kernel regression [4] and heteroskedastic regression [1]. Particularly, neural network regression is one of the main focuses to perform robust regression in the present time [2, 8, 14, 17, 20]. In this work, we also settle our research on neural networks regression by means of different strategies based on PAELLA algorithm [9] output (occurrence vector) to carry out the regression without dismissing samples of the data.

PAELLA is an outlier identification algorithm [9]. As such, its performance has been reported developing data cleaning operations. Its results come in the form of an occurrence vector which is transformed into a binary classification by applying some suitable threshold thus labeling outliers and core observations.

This paper seeks to determine how to apply the information contained in the results from the PAELLA algorithm in such a way that allows every observation on a dataset to participate at the training stage of a predictive algorithm.

The importance of this work relies to the case in which the data cannot be discarded. One example of such a case is few data samples or when those samples marked as outliers represent a legitimate behaviour in an imbalanced class scenario. The results from this approach can be applied to many fields that work with numeric data. In particular, it can be applied to Security of Information Systems in numerous ways, such as intrusion detection, measuring the foreseeable impact of potential threats, simulating the effectiveness of existing measures to reduce the risk of the threat exploiting the vulnerability, establishing the likelihood of occurrence for a particular threat exploiting a related vulnerability, etc.

## 1.1 The PAELLA algorithm

### 1.1.1 Background and foundations

The PAELLA algorithm has been originally published as an outlier detection and data cleaning technique. It is an iterative method that works well with non-artificial datasets where normality assumptions cannot be established and a priori metric hypothesis might be difficult to define. As originally presented, the algorithm can be considered as a classifier since its output is a binary classification of the input samples into two categories: outliers and the rest of the data. Each subset from this binary classification can be subsequently used by the practitioner with a different aim in each case. On one hand, the researcher can filter the dataset in order to obtain a purified subset with which to feed the rest of the analysis, say building a model for example. On the other hand, having identified the outliers, the researcher can take advantage of the analysis of their features in order to spot the origin of the perturbation and, thus, deploy corrective actions if need be. In industrial processes modeling, such as the example presented in PAELLA's seminal paper, this identification can be of paramount importance.

PAELLA is an iterative method whose steps can be algorithmically formulated as described in the following three phases

**Pre-processing of the data** The data is divided in $g+1$ subsets $C_k(k = 0, \cdots, g)$ based on the empirical clusters each sample are allocated to. If the cluster strategy allows the presence of noise samples, the subset $k = 0$ represents the samples that cannot be reliably allocated to the rest of clusters.

**Phase 1** The samples of the dataset are fitted into different linear models to create a set of hypersurfaces fitting and coating the dataset. One random sample of each subset is taken as a seed point of the subset. The remaining points in the subset are classified in relation to their Mahalanobis distance to the seed point. The points with the smallest distance, with respect to a given threshold, are added to $G_i$ to infer a model using a robust and affine equivariant fitting. A residual of the sample points not used to infer the model is evaluated against the model and those that report a low quantile function value may be regarded as compliant with the model and added to $G_i$. Previous steps are iterated considering the samples out of $G_i$. In later stages, PAELLA evaluates the so-called goodness-of-fit of the samples in relation to the set of hypersurfaces to evaluate their suitability for the model. Phase 1 reveals potential outliers and distinguishes the samples that follow a general trend from those which does not.

**Phase 2** The remaining samples not yet added to $G_i$ are evaluated against the set of models of each iteration. The smallest residuals of the samples are considered to decide the model to which the samples are associated to, while those samples with largest residuals are considered as possible outliers in the current given trial.

**Phase 3** Phases 1 and 2 are iterated and a vector that represents the frequency of outlierness for every sample is obtained by accumulating the results in every iteration. The accumulated number of times that each sample was considered outlier divided by the total number of iterations provides the frequency vector. The samples with frequencies above a threshold are considered as outliers and separated for further analysis. The result from this phase is a binary classification expressing which samples have been considered outliers by the algorithm.

### 1.1.2 Novel approach based on the occurrence vector

Nevertheless, this approach of filtering the dataset by discarding the observations marked as outliers can be considered as a hard method, and in those experimentations in which the dataset is small it is often times undesirable. As an alternative to filtering, robust methods try to cope with outliers by minimizing the impact of their presence in the parameter fitting phase.

The approach we report in this paper is based on the frequency of outlierness vector before it has been thresholded to be transformed into a binary vector representing a two categories classification. We will call this unprocessed frequency vector the occurrence vector in what follows.

The occurence vector contains information related with the reliability of the information provided by a particular sample, as it is directly related with the number of times it was considered as an outlier by the PAELLA algorithm. This information can be employed by following different strategies in order to take advantage of this occurrence vector. This paper reports a few examples of such possible strategies whose results confirmed that similar or even better results can be obtained without the need of discarding observations.

## 1.2 Structure of the paper

The rest of the paper is structured as follows. Section 2 comprises the core of experiments carried out in this paper. Section 2.1 describes an artificial dataset used during the analyses. Section 2.2 defines the validation metric used to evaluate the different proposed methods. Section 2.3 presents the results of using the raw dataset for building a predictive neural network model. Section 2.4 shows the results of applying PAELLA algorithm and building the former predictive model described in 2.3 using filtered data where outliers have been removed using the PAELLA algorithm. Section 2.5 shows a first trial on generalizing the classification performed on the occurrence vector by binning its values on a number of intervals. Section 2.6 goes one step ahead and uses the occurrences vector values as sampling likelihoods to perform probabilistic sampling regression. Another twist to sampling likelihood is explored on Section 2.7. Section 2.8 presents a robust weighted regression based on neural networks which receives the weights from functions of the PAELLA occurrence vector. In this case, the samples of the dataset are not sampled and they can all participate through a weighted regression in predicting the model. Section 2.9 presents a brief summary of the methods and results obtained through the paper. Finally section 3 sums up the conclusions of this paper.

# 2 Experimental analysis

A set of experiments are carried out in order to determine the suitability of several different approaches. Both the experiments and their results are described in what follows. As each experiment is run, the reasons that motivate the next step to take are detailed, very much following the authors path on understanding the behaviour of the different approaches reported.

## 2.1 Description of the artificial dataset

This dataset comprehends 1000 observations of two variables ($x_1$, $x_2$). Half of these samples were generated following a uniform random distribution. The other half were generated following a simple sinusoidal law slightly altered by some normal noise, as described in Eq. 1:

$$x_2 = \sin(x_1) + \epsilon \tag{1}$$

where $\epsilon \sim \mathcal{N}(\mu = 0,\ \sigma = 0.1)$.

A picture of such dataset is shown in Fig. 1. It is clear that the dataset contains a higher fraction of noise than might be expected in common practice but that might increase the interest on the performance of the techniques used.

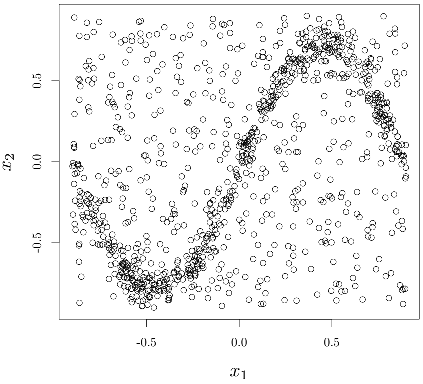

FIGURE 1: Artificial dataset used in the experiments

## 2.2 Evaluation metrics

We used the mean squared error (MSE) to measure the performance of the different methods. MSE formuale is defined in Eq. 2.

$$MSE = \frac{1}{N}\sum_{i=1}^{n}\left(x_{2_i} - \hat{x_{2_i}}\right)^2 \tag{2}$$

where $\hat{x_{2_i}}$ is the predicted value of a sample in the dataset of $N$ samples and $x_{2_i}$ is the corresponding actual value.

TABLE 1
Raw data model training parameters for grid search

| | Values |
|---|---|
| Replicates | 30 |
| Number of hidden neurons | 2, 3, 4, 5, 6 |
| Learning rate | 0.001, 0.005, 0.010 |
| Momentum | 0.001, 0.005, 0.010 |

## 2.3 Raw data model

In this experiment, a set of multilayer perceptron (MLP) models is trained using the raw data in order to have a reference of how MLP models would perform with all data and the artificial neural network (ANN) algorithm. The result is useful as a reference afterwards. The neural networks were trained using the adaptive gradient descent with momentum method [16]. We split the training set in two subsets, training (66%) and validation (34%), for avoiding overfitting by means of the use of the early stopping technique. The hyper-parameters of the neural network were chosen by means of an exhaustive grid search with the values for the parameters shown in Table 1.

As the replicates parameter states, each configuration is trained 30 times in order to assess the performance of different values of the initial weights and biases. Considering all the different combinations of the training parameters, a total of 1,350 different neural networks were trained.

The best neural network model obtained following these conditions yielded a MSE equal to 0.2799. As will be seen momentarily, there is plenty of room for improvement yet.

## 2.4 PAELLA filtered data model

The previous section provided a reference of the quality expected from a neural network model using the dataset without further preprocessing. Even though neural networks are well known for its robustness [15], some additional measures might help increasing the quality of the model. The first option a practitioner might choose could be using an outlier identification model in order to separate the core observations from those belonging to the surrounding noise.

In order to evaluate this approach, we applied the PAELLA algorithm to identify the outlier samples, as it was initially developed. Firstly, the raw dataset is pre-processed using finite normal mixture modeling via the expectation-maximization (EM) algorithm and Bayesian Information Criterion (BIC) to cluster the samples into 1 to 10 groups. Then, the samples are processed using the PAELLA algorithm as described in Section 1.1.1 with a 99% threshold value. As Fig. 2 clearly shows, the dataset is cleanly split into two categories: one category containing mostly those points belonging to sinusoidal function and another with the rest. Then, the dataset is filtered in the two categories and only the samples not labeled as outliers are used to train MLP models. The same experiment as described in Section 2.3 is repeated, but now using only those observations that passed the 99% threshold. As expected, the results greatly improve and the best model provides a MSE equal to 0.0200.

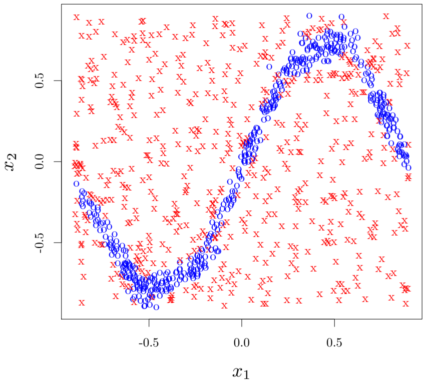

FIGURE 2: Color-coded by PAELLA running in outlier identification mode using a 99% threshold. The red crosses mark the identified outliers whereas the rest of samples are mark with blue circles.

Nevertheless, this approach did not include all the observations in the training–test pair of datasets. The noisy samples were just removed after splitting the observations into two categories according to the results obtained from PAELLA and a choice of a suitable threshold. This approach poses a challenge especially in those cases where the dataset is small. Instead of discarding information, the practitioner might choose a different ap-

TABLE 2
Where $N$ represents the number of bins, $2^N$ is the number of combinations to evaluate for such number of bins, Networks is the number of networks trained.

| $N$ | $2^N$ | Networks | MSE |
|---|---|---|---|
| 5 | 32 | 960 | 0.0893 |
| 10 | 1,024 | 30,720 | 0.0288 |
| 15 | 32,768 | 983,040 | 0.0298 |
| 20 | 1,048,576 | 31,457,280 | 0.0277 |

proach that allows every observation to participate and train the model.

## 2.5 Using occurrence vector approach

Section 2.4 provided good results at the expense of discarding information. The values in the occurrence vector obtained from PAELLA provided the measure by which a sample was assigned into two categories. A natural extension to this approach could be cutting the values of the occurrence vector into $N$ intervals (bins) and testing the models that use every possible combination of these categories as dataset. For a set of $N$ categories, $2^N$ different combinations are possible and, thus, the number of different models quickly grows way past sensible limits as $N$ increases. Table 2 shows the number of training executions needed to cover all possible combinations and the results obtained using the artificial dataset. Notice that for $N = 20$ we felt compelled to not run all the trainings but just a random search using genetic algorithms [5] which sufficed for the purpose of this experiment as the results confirmed. In this occasion, we contemplated the same experimental setup as in previous sections but a grid search of the best hyperparameters was not performed due to the high time cost that would suppose. Instead, we set the number of hidden neurons to 3, the learning rate to 0.001 and the momentum to 0.001 which proved to be successful in previous experiments.

The results shown in Table 2 suggests a trend: the higher the value of $N$, the narrower the bins and the better the results. When $N$ has the same size as the training samples, all possible combinations of samples would be used for training different MLP models. Unfortunately, with relatively low values of $N$ the approach becomes unfeasible due to its computational cost. Nevertheless, the trend claims for the participation of individual observations. As it is not possible to consider every combination of the categories in the values of the occurrence vector, a probabilistic context using these values as likelihoods seems promising.

## 2.6 Macro sampling

Building on the results of previous sections, a new approach based on probabilistic sampling is considered in this section. This might endow every observation the potential to participate probabilistically on the training with a probability based on the values of the occurence vector from PAELLA. The scheme proposed in this section fits a model to a dataset generated by sampling the primitive one using the likelihoods expressed by some function of the occurrence vector for drawing the samples. This procedure can be repeated multiple times and a representative of the experimental training will be chosen.

The scheme of this macro sampling experiment follows these steps:

1. A likelihood is assigned to each observation according to the frequency of outlierness displayed by this observation in the occurrence vector.

2. The dataset is sampled using these likelihoods in order to form a new dataset which is used to build a predictive model.

3. The predictive model is trained and its performance evaluated

4. Steps 2 through 3 are repeated for a predefined number of times. In our experiments, we perform 1000 iterations.

We considered different powers $p$ of the occurrence vector $v$ obtained from PAELLA as likelihood functions $v^p$. For the sampling strategy, we chose a random sampling with replacement. We set the size of the sampling equals to the number of samples in the training set and the vector of probability weights that gets the elements of the training set equals to the likelihood function. MLP models were trained following the experimental setup described in Section 2.5. The models that produced the minimum MSE error in the 1000 runs of the method were considered.

TABLE 3
Results from macro sampling

| Likelihood function | MSE |
|---|---|
| $v^1$ | 0.0785 |
| $v^2$ | 0.0634 |
| $v^3$ | 0.0488 |
| $v^4$ | 0.0487 |
| $v^5$ | 0.0314 |
| $v^{10}$ | 0.0241 |

Table 3 shows the results obtained with this method for different powers of the occurrence vector. These results suggest that a model with similar performance to the simple application of the PAELLA partitive cleaning approach (MSE equals 0.0200) can be obtained.

## 2.7 Micro sampling

As a consequence of the good results obtained in the previous section, a slightly different approach is considered in this section. It could be interesting to assess the performance of a different sampling strategy. Instead of sampling the primitive dataset before training the model the sampling can be applied every $k$ iterations, thus training with different datasets every time. To evaluate this method, we train a set of MLP models following the experimental setup of Section 2.4. Table 4 shows the results obtained using the artificial dataset described and used in previous sections. It can be seen that far for improving, the results are considerably worse and experimentation showed instability in the performance, especially using the momentum variant of the gradient descent training algorithms.

This variation on the sampling procedure failed to improve the results shown by the macro sampling approach. It is our belief that the poor results are due to instability caused by the training algorithm used.

## 2.8 Weighted regression via PAELLA

In this experiment, we trained a set of MLP models but this time without dismissing any samples of the training set and using weighted regression. Following this approach, each

TABLE 4
Results from micro sampling

| Likelihood function | MSE |
|---|---|
| $v^1$ | 0.2366 |
| $v^2$ | 0.2133 |
| $v^3$ | 0.1914 |
| $v^4$ | 0.2043 |
| $v^5$ | 0.2017 |
| $v^{10}$ | 0.3842 |

TABLE 5
Hyperparameters taken into account in the grid search performed for MLP weighted regression.

| Parameter | Values |
|---|---|
| hidden neurons | $[1, 2, \ldots, 9]$ |
| learning rate | $[0.01, 0.001, 0.05]$ |
| momentum | $[0.9, 0.09]$ |
| Nesterov | [True, False] |
| epochs | $[1000, 5000]$ |
| $p$ | $[1, 2, \ldots, 10]$ |

sample is assigned a weight, usually defined as a multiplying factor that modifies the influence of the residual in the global loss function. The likelihood functions, which are defined as the PAELLA occurrence vector powered to a given exponent $p$, were used as sample weights in order to optimize the MSE error during training of the MLP models. The neural networks were trained using the stochastic gradient descent (SGD) with Nesterov momentum [16] and early stopping technique. The optimal hyperparameters were chosen through an exhaustive grid search with the details shown in Table 5.

The best result yielded a MSE equals to 0.0202 that was obtained for $p = 10$. This performance is comparable to the one obtained with outlier removal, MSE equals to 0.0200. This approach displays the benefit of not discarding any observation a priori but getting similar results as with outlier removal. Therefore, it satisfies the assumptions made in this paper.

TABLE 6
Methods and best MSE errors obtained per method.

| Method | MSE | Remaks |
|---|---|---|
| raw MLP | 0.2799 | PAELLA was not used in any way |
| outlier filtered MLP | **0.0200** | traditional PAELLA with threshold 99% |
| binning the occurrence vector | 0.0277 | for 20 bins |
| probabilistic macro sampling | 0.0241 | for $v^{10}$ |
| probabilistic micro sampling | 0.1914 | for $v^3$ |
| MLP weighted regression | **0.0202** | for $v^1 0$ |

### 2.9 Summary of methods and results

In this section, we collect the results obtained with the different methods presented in this paper. It can be seen that using a simple technique such as MLP weighted regression can successfully be employed via PAELLA for weighted regression modeling without dismissing samples of the dataset.

## 3 Conclusions

The experiments reported in this paper have confirmed the usefulness of taking advantage of the information contained in the occurrence vector provided by the PAELLA algorithm. They also show that there is plenty of room for strategies aiming at lessening the impact of unreliable samples, such as outliers, while extracting information from the whole dataset.

In those scenarios suffering from difficulties in obtaining datasets from experimentation, the approach proposed in this paper represents a feasible solution with the benefits of not reducing the sample size by an outlier removal procedure.

The results reported confirm that outliers do not necessarily have to be discarded as their effect can be effectively lessened. In particular, the PAELLA powered weighted regression based on likelihood functions using the occurrence vector is capable of outperforming outlier removal approaches.

# Funding

This work was supported by the ‘Ministerio de Economía, Industria y Competitividad’ [DPI2016-79960-C3-2-P].

# Acknowledgements

We would like to also express our gratitude to Castilla y León Supercomputing Center whose cooperation allowed us to run around one million neural network trainings for the experiments reported on this paper.

# References

[1] Atkinson, A. C., Riani, M., and Torti, F. (2016). Robust methods for heteroskedastic regression. *Computational Statistics & Data Analysis*, 104:209 – 222.

[2] Bataineh, M. and Marler, T. (2017). Neural network for regression problems with reduced training sets. *Neural Networks*, 95:1 – 9.

[3] Dasu, T. and Johnson, T. (2003). *Exploratory Data Mining and Data Cleaning*. Wiley.

[4] de A. Lima Neto, E. and de A.T. de Carvalho, F. (2018). An exponential-type kernel robust regression model for interval-valued variables. *Information Sciences*, 454-455:419 – 442.

[5] García, A. B., Díaz, A. M., Meré, J. O., and Nicieza, C. G. (1996). Generalization of the influence function method in mining subsidence. *International Journal Of Surface Mining And Reclamation*, 10(4):195–202.

[6] Gonzalez-Marcos, A., Alba-Elias, F., Castejon-Limas, M., and Ordieres-Mere, J. (2011). Development of neural network-based models to predict mechanical properties of hot dip galvanised steel coils. *International Journal of Data Mining, Modelling and Management*, 3(4):389–405.

[7] Hussein, R., Elgendi, M., Wang, Z. J., and Ward, R. K. (2018). Robust detection of epileptic seizures based on l1-penalized robust regression of eeg signals. *Expert Systems with Applications*, 104:153 – 167.

[8] Kronberger, G., Kommenda, M., Lughofer, E., Saminger-Platz, S., Promberger, A., Nickel, F., Winkler, S., and Affenzeller, M. (2018). Using robust generalized fuzzy modeling and enhanced symbolic regression to model tribological systems. *Applied Soft Computing*, 69:610 – 624.

[9] Limas, M. C., Meré, J. B. O., Ascacibar, F. J. M. D. P., and González, E. P. V. (2004). Outlier detection and data cleaning in multivariate non-normal samples: The PAELLA algorithm. *Data Mining and Knowledge Discovery*.

[10] López, J. and Maldonado, S. (2018). Robust twin support vector regression via second-order cone programming. *Knowledge-Based Systems*, 152:83 – 93.

[11] Menéndez, C., Ordieres, J., and Ortega, F. (1996). Importance of information preprocessing in the improvement of neural network results. *Expert Systems*, 13(2):95–103.

[12] Ordieres, J., López, L., Bello, A., and Garcia, A. (2003). Intelligent methods helping the design of a manufacturing system for die extrusion rubbers. *International Journal of Computer Integrated Manufacturing*, 16(3):173–180.

[13] Ordieres-Meré, J., Martínez-de Pisón-Ascacibar, F., González-Marcos, A., and Ortiz-Marcos, I. (2010). Comparison of models created for the prediction of the mechanical properties of galvanized steel coils. *Journal of Intelligent manufacturing*, 21(4):403–421.

[14] Patan, K. (2018). Two stage neural network modelling for robust model predictive control. *ISA Transactions*, 72:56 – 65.

[15] Pernía-Espinoza, A. V., Ordieres-Meré, J. B., Martínez-de Pisón, F. J., and González-Marcos, A. (2005). TAO-robust backpropagation learning algorithm. *Neural Networks*, 18(2):191–204.

[16] Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145 – 151.

[17] Tao, J., Zhou, D., and Zhu, B. (2018). Robust latent regression with discriminative regularization by leveraging auxiliary knowledge. *Neural Networks*, 101:79 – 93.

[18] Vo, G. D. and Park, C. (2018). Robust regression for image binarization under heavy noise and nonuniform background. *Pattern Recognition*, 81:224 – 239.

[19] Walczak, B. (1996). Neural networks with robust backpropagation learning algorithm. *Analytica Chimica Acta*, 322(1):21–29.

[20] Xu, Q., Deng, K., Jiang, C., Sun, F., and Huang, X. (2017). Composite quantile regression neural network with applications. *Expert Systems with Applications*, 76:129 – 139.