Journal of Economics and Administrative Sciences
2024; 30(140), pp. 466-514

P-ISSN 2518-5764
E-ISSN 2227-703X

Journal of Economics and Administrative Sciences (JEAS)

Available online at http://jeasiq.uobaghdad.edu.iq
DOI: https://doi.org/10.33095/g4hems75

Estimation of the Regression Model Using M-Estimation Method and Artificial Neural Networks in the Presence of Outliers

Hussein Talib Jawad*
Department of Statistics,
College of Administration and Economics,
University of Baghdad,
Baghdad, Iraq.
hussein.taleb2101m@coadec.uobaghdad.edu.iq
***Corresponding author**

Rabab Abdul-Ridha Saleh
Department of Statistics,
College of Administration and Economics,
University of Baghdad,
Baghdad, Iraq.
rabab.saleh@coadec.uobaghdad.edu.iq

Received:9/10/2023 Accepted: 20/12/2023 Published Online First: 30 /4/ 2024

Abstract:

This study aimed to predict using regression models in the presence of outliers in the study data. The research delved into outliers, their detection, and model estimation through robust methods, represented by the M estimator and multilayer artificial neural networks. A comparison between these methods was conducted, and they were applied to real data representing a survey of private-sector power generators for 2021. Model evaluation was performed using Mean Squared Error (MSE) and the determination coefficient. The results indicated that the M-estimator with the Huber function outperformed its counterpart with the Tukey function. The best-performing architecture for the artificial neural network was ML-FF (5, 16, 32, 46, 128, 1) with the relu activation function. This network effectively handled extreme values and exhibited strong predictive capabilities. The choice of activation function and the number of hidden layers significantly impacted the neural network's performance, with the results showing the superiority of this artificial neural network over the robust estimators.

Paper type *Research paper*

Keywords: Regression, M-Estimation, Artificial Neural Networks, Outlier, Robust Regression, Activation Function.

*(Estimating a multiple regression model using some robust methods and artificial intelligence algorithms with application.)

1. Introduction:

Regression analysis aims to find a mathematical equation that relates the dependent variable to the independent variables. This model allows us to understand the nature of the relationship and identify the factors affecting it.

It also allows us to predict the effect of any independent variable on the dependent variable

Some of the objectives of regression analysis include:

- Describing the data in order to analyze the relationship between the variables.
- Estimating the model parameters for the variables that describe the data.
- Controlling the dependent variable's values by changing the explanatory variable's values.
- Making predictions after finding the estimates of the parameters for decision-making and future planning (Abu Shaer and Al-Sarraf, 2018).

These objectives have made regression analysis widely used in various scientific fields. Regression equations are typically estimated using the least squares and maximum likelihood methods due to their ease of estimation and computational efficiency. However, of outliers in the data render these methods inefficient for estimating and describing the relationships between variables. To address this issue, robust methods are often employed for model estimation in the presence of outliers. This research will be divided into four chapters. The first chapter represents the introduction along with the literature review. The second chapter includes the theoretical aspect of the methods that will be used in this research, followed by the application of these methods in the third chapter to obtain results from employing these methods. Subsequently, the fourth chapter will discuss the most important conclusions reached. Some **literature review**

Imam and Tarawneh (2014) used neural networks and multiple linear regression techniques to estimate the concentration of iron within a given material based on experimental data. Their findings indicated a superior performance of neural networks in comparison to the multiple linear regression model.

Jawad (2015) conducted a comparative analysis to evaluate the effectiveness of genetic algorithms and neural networks in estimating the median location for nonparametric multivariate regression models. The feedforward neural networks demonstrated superiority compared to the ART neural network, multilayer network, and the genetic algorithm proposed by the researcher..

Al-Athari and Al-Amleh (2016) compared the trimmed least squares method and the MM-Estimation method for estimating a linear regression model. This evaluation was performed through simulation techniques and the assessment of mean square error (MSE). The results of their investigation revealed the superior performance of the MM-Estimation when contrasted with the trimmed least squares (LTS) method.

Khalaf (2017) conducted a comparative analysis between artificial neural networks and non-parametric regression to estimate atomic radiation doses. The study revealed, through simulation, that the backpropagation network exhibited superior estimation capabilities compared to other networks. Furthermore, Gaussian kernel density estimation emerged as the optimal data smoothing method. In contrast, the method of least trimmed squares was identified as the most effective data smoothing technique for non-parametric approaches. In practical application, neural networks outperformed non-parametric methods.

AL Zirej and Hadi (2019) used neural networks to predict the penetration rate in oil reservoir formations and compared this approach with multiple regression analysis. Their results showed the superiority of the proposed neural network model over the multiple regression model.

Ali (2019) compared robust regression estimation methods, including the biweight and Tukey's bisquare robust method (M-estimation) and the S-estimation method. The results indicated that the M-estimation method was the most effective for estimation.

Hajjaj and Abdelqader (2020) used artificial neural networks and robust regression to predict birth rates in Egypt, with artificial neural networks proving superior to the robust regression approach.

P-ISSN 2518-5764
E-ISSN 2227-703X

Mohammed (2023) estimated a seemingly unrelated regression system using several robust estimators. He employed four robust methods: M-Estimation, S-Estimation, MM-Estimation, and fastSUR. The results indicated that the MM-Estimation method is the most effective among the techniques employed for handling outliers.

2.Material and Methods

In this chapter, we looked at outliers to understand them and determine how to count them. Next, we defined robust regression and delved into the M-Estimator. The chapter then discussed artificial neural networks and their components and explained the multilayer feed forward neural network.

2.1 Outliers

An outlier is defined as a value in a dataset that is unusual or disparate, disrupting various statistical measures, such as the mean, median, standard deviation, and others. These outlier values significantly affect the analytical results, and the degree of impact is directly proportional to the number of outlier values (Kamal and Khalil, 2021).

The causes of outlier occurrence (Khalil and Mohammed, 2012):

- Calculation errors
- Reading errors
- Recording errors

As a result, outlier values represent observations that exhibit large deviations from the central tendency of the sample, resulting in significant inaccuracies and biases when compared to the remaining observations. This ultimately reduces estimation effectiveness (Al-Azzawi and Mohammed, 2022).

To detect the presence of outliers (Hassan and Reda, 2011):

One of the methods for detecting outliers is the Box-Whisker plot, also known as the Box Plot. This method relies on the interquartile range (IQR), a measure of variations based on the second quartile range, as illustrated in figure 1.

$$IQR = Q_3 - Q_1 \qquad (1)$$

Q_1 represents the first quartile, and Q_3 represents the third quartile (Hassan and Reda, 2011).
Detection occurs under either of the following conditions (Hassan and Reda, 2011):

$$\begin{cases} y_i > Q_3 + (1.5IQR) \\ y_i < Q_1 - (1.5IQR) \end{cases} \qquad (2)$$

It is considered an outlier within this range.

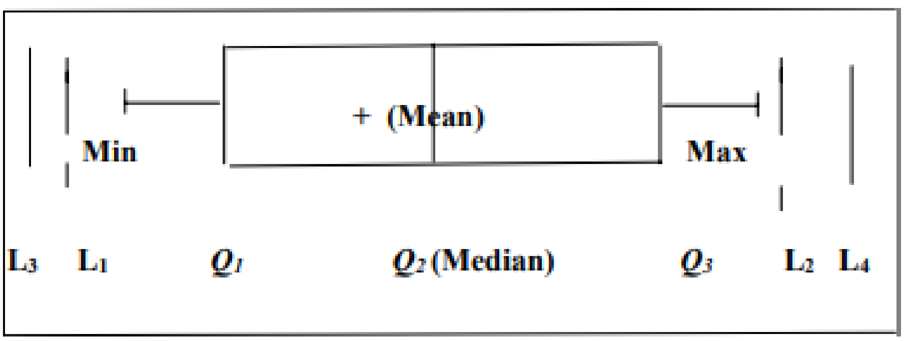

Figure 1: Components of Box-Whisker plot (Hassan and Reda, 2011).

2.2 Robust Regression

It is considered an alternative method for estimating parameters in regression models, especially in the presence of outliers that hinder the efficiency of estimating parameters in the traditional regression model. Such situations occur when the basic assumptions necessary for the model and the study data do not align, reducing the effectiveness of traditional methods, such as ordinary least squares (Hassan and Reda, 2011). Thus, the importance of using robust methods in regression becomes clear, because they show minimal exposure to the influence of outliers (Al-Hajjaj and Al-Qadar, 2020).And one of these robust methods is:

2.2.1M-Estimator

This approach belongs to the robust regression technique category and is considered one of the most prominent methods in this field. Represented by the symbol (M), it addresses the challenge presented by outliers by replacing the squared residuals with the loss function ρ. Despite this adjustment, the primary goal of the estimation method remains focused on minimizing the estimator as much as possible (Arshid and Saleh, 2022).

In this method, the objective is to minimize the quantity (Mahdi, 2010):

$$\sum_{i=1}^{n} \rho(Y_i - X_i' \beta) \qquad (3)$$

Where ρ is a symmetric and convex function designed to achieve scale invariance.

$$\sum_{i=1}^{n} \rho\left(\frac{Y_i - X_i' \beta}{\hat{\sigma}}\right) \qquad (4)$$

Taking the derivative concerning the parameter vector β and setting it equal to zero.

$$\sum_{i=1}^{n} \Psi\left(\frac{Y_i - X_i' \beta}{\hat{\sigma}}\right) X_{sj} = 0 \qquad (5)$$

After rewriting equation 5, it becomes:

$$\sum_{i=1}^{n} \frac{W_i X_{ij} (Y_i - X_i' \beta)}{\hat{\sigma}} = 0 \qquad (6)$$

Here, W_i is a weight function calculated using the equation:

$$W_i = \frac{\Psi\left(\frac{Y_i - X_i' \beta_h}{\hat{\sigma}}\right)}{\left(\frac{Y_i - X_i' \beta}{\hat{\sigma}}\right)} \qquad (7)$$

Where β_h represents initial parameters and σˆ is the scale parameter calculated using the equation (Bahez and Rasheed, 2022).

$$\hat{\sigma} = \frac{MAD}{0.6745} = \frac{median|u_i - median(u_i)|}{0.6745} \qquad (8)$$

By solving equation 6 using the method of least squares, the parameter estimates are obtained:

$$\hat{\beta} = (X'WX)^{-1} X'WY \qquad (9)$$

Where W is a diagonal weight matrix with W_i as its diagonal elements.

The robust method M relies on several functions, including (Zaman and Bulut, 2018):

1) Huber function:

$$\Psi(u) = \begin{cases} u & , |u| < h \\ h \, sgn(u) & , |u| \geq h \end{cases} \qquad (10)$$

Where h is a constant that takes the values 1.5, 1.7, and 2.08

2) Tukey function:

$$\Psi(u) = \begin{cases} u(1-u^2)^{\,2} & ,|u| < c \\ 0 & ,|u| \geq c \end{cases} \qquad (11)$$

Where c is a constant represented by the values 4.685 and 6.

2.3 Artificial Neural Networks

Artificial neural networks have gained great recognition in modern data processing methodologies owing to their efficacy and objectivity. Its origins go back to 1943 when Warren McCulloch and Walter Pitts introduced the foundational model of artificial neural networks (Salman and Salah, 2022).

The basic concept underlying artificial neural networks is to emulate data, to create a model of said data for purposes of comprehensive analysis, classification, prediction, or other forms of processing, all accomplished without the need for a predefined model for the given dataset (Jabbar and Mohammed, 2020).

2.3.1Components of Artificial Neural Networks:

1) Input Layer: This layer receives information or signals from the external environment. (Silva et al ,2017).
2) Hidden Layers: These layers consist of neural cells extracting patterns associated with the process or system. These layers perform most of the internal processing of the network (Silva et al, 2017).
3) Output Layer: This layer also consists of neurons and produces and delivers the final network outputs. These outputs result from the processing performed by the neurons in the previous layer (Silva et al, 2017).

2.3.2Activation Functions:

These functions is to transform input values according to the type of function used, based on a scale of the output values. They transform the output from the hidden layers or the output layer. Some of these functions are:

1) Sigmoid Function: This function transforms outputs to specific values between (0, 1). It is referred to as the binary activation function. It is one of the most widely used functions in backpropagation networks. The equation is as follows (Sharma et al,2020) :

$$f(x) = \frac{1}{1+e^{-x}} \qquad (12)$$

2) Tanh Function: This function is similar to the sigmoid function, but symmetrical around the origin. The output of this function ranges between (-1, 1). The equation is (Sharma et al,2020):

$$f(x) = Tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \qquad (13)$$

3) Relu Function (Rectified Linear Unit): Applying this function addresses the vanishing gradient problem in sigmoid and tanh functions. The equation is:

$$f(x) = \begin{cases} x & if\ x > 0 \\ 0 & if\ x \leq 0 \end{cases} \qquad (14)$$

The relu equation states that if the input from the sum is greater than zero, the output is the same value; Otherwise, if it is less than or equal to zero, the output is zero (Bai, 2022).

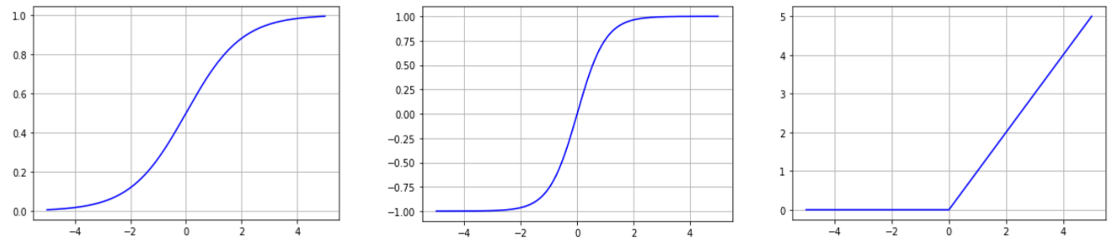

Figure 2: Activation functions, (a) sigmoid, (b) tanh, (c) relu

1.1.1Multilayer Feedforward Neural Network:

A feedforward neural network is a multi-layered network characterized by the strict forward propagation of input signals. Each layer includes units that directly receive inputs from units in the preceding layer and transmit their outputs to units in the subsequent layer. Notably, no connections are created within the same layer. This sequential process persists until reaching the output layer, which maintains a connection with the external environment. The structural arrangement is explained as follows (Amorim et al., 2022):

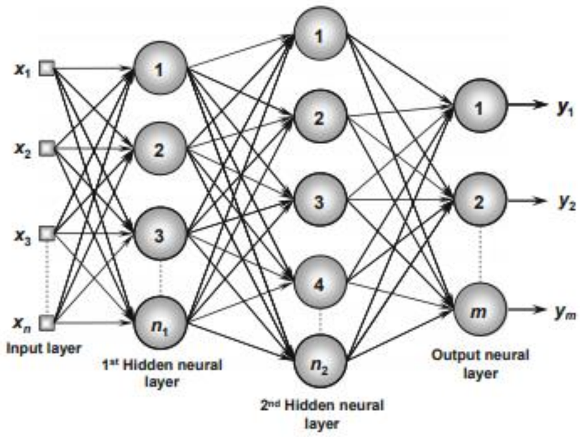

Figure 3: Multilayer Feedforward Neural Network (Silva et al, 2017).

Likewise, the error back propagation algorithm is applied as a training algorithm for the network with any number of layers (Amorim et al., 2022).

The error back propagation algorithm:

One of the most widely used algorithms, it finds its application in training fully connected, feedforward, multi-layered, and nonlinear neural networks. This algorithm can be seen as a generalization of the error-correcting training method, operating in two main stages (Al-Tha'labi and Omran, 2016).

Forward Propagation Phase:

Each node in the input layer receives its input signal and sends it to nodes in the hidden layer. This process is repeated layer by layer until the output layers is reached. Assuming Figure 4 shows the neurons, neuron j receives input signals from a group of neurons in the left layer. The local field $v_j(n)$ at the neuron j associated with the activation function is calculated as follows:

$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n)$ (15)

In this equation, $y_i(n)$ denotes the output values from the previous layer, whether it is an input or hidden layer. It is also considered the input value for the current layer at iteration (n). $v_j(n)$ Is the result of multiplying the layer's weights by its inputs at iteration (n). (m) Represents the total number of inputs (excluding bias) applied to neuron j, and the interconnection weight w_{j0} (corresponding to the constant input $y_0 = 1$) denotes the bias used in neuron j.
Hence, $y_i(n)$ emerges as the output of neuron j at iteration n as follows:

$y_j(n) = \varphi_j(v_j(n))$ (16)

Where $\varphi_j(v_j(n))$represents the outcome of the activation function applied to the layer's outputs.

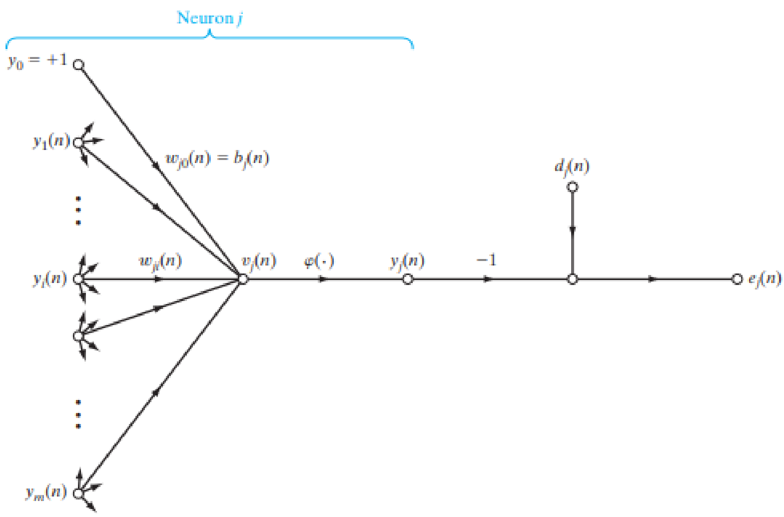

Figure 4: shows the signals in the neural network at neuron j in the layer (Haykin, 2009).

Back propagation phase (Haykin, 2009):
After completing the forward propagation phase, the error and loss functions are calculated using the following equations:

$e_j(n) = d_j(n) - y_j(n)$ (17)

$\varepsilon_j(n) = \frac{1}{2} e_j^2(n)$ (18)

$d_j(n)$ is the input value (dependent variable), $e_j(n)$ is the error value at iteration n, and $\varepsilon_j(n)$ represents the cost at iteration n.

$\varepsilon(n) = \sum_{j \in C} \varepsilon_j(n)$

$= \frac{1}{2} \sum_{j \in C} e_j^2(n)$ (19)

$\varepsilon_{av}(N) = \frac{1}{N} \sum_{n=1}^{N} \varepsilon(n)$

$= \frac{1}{2N} \sum_{n=1}^{N} \sum_{j \in C} e_j^2(n)$ (20)

The error back propagation algorithm applies corrections $\Delta w_{ji}(n)$ to the interconnection weight $w_{ji}(n)$, which is proportional to the partial derivative $\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)}$. Using the chain rule for differentiation and integration, this gradient can be expressed as:

$$\frac{\partial\varepsilon(n)}{\partial w_{ji}(n)} = \frac{\partial\varepsilon(n)}{\partial e_j(n)}\frac{\partial e_j(n)}{\partial y_j(n)}\frac{\partial y_j(n)}{\partial v_j(n)}\frac{\partial v_j(n)}{\partial w_{ji}(n)} \qquad (21)$$

Applying the above derivation leads to:

$$\frac{\partial\varepsilon(n)}{\partial w_{ji}(n)} = -e_j(n)\varphi_j'(v_j(n))y_i(n) \qquad (22)$$

The correction $\Delta w_{ji}(n)$applied to $w_{ji}(n)$is defined by the equation:

$$\Delta w_{ji}(n) = -\eta\frac{\partial\varepsilon(n)}{\partial w_{ji}(n)} \qquad (23)$$

Substituting equation 22 into equation 23 yields

$$\Delta w_{ji}(n) = \eta\delta_j(n)\, y_i(n) \qquad (24)$$

Where the local gradient $\delta_j(n)$is defined as:

$$\delta_j(n) = \frac{\partial\varepsilon(n)}{\partial v_j(n)}$$
$$= e_j(n)\varphi_j'(v_j(n)) \qquad (25)$$

From equations 24 and 25, it is clear that the main factor affecting the calculation of the weight correction $\Delta w_{ji}(n)$ is the error $e_j(n)$ in the output neuron j. In the second case, where neuron j is located in a hidden layer, the development (modification) of the backpropagation algorithm is completed. Assuming the scenario shown in Figure 4, where neuron j is situated in a hidden layer within the network, equation 25 allows the local gradient $\delta_j(n)$for the hidden neuron j as follows:

$$\delta_j(n) = \frac{\partial\varepsilon(n)}{\partial y_j(n)}\frac{\partial y_j(n)}{\partial v_j(n)}$$
$$= -\frac{\partial\varepsilon(n)}{\partial y_j(n)}\varphi_j'(v_j(n)) \qquad (26)$$

Referring to figure 5, we observe:

$$\varepsilon(n) = \frac{1}{2}\sum_{k\in C} e_k^2\,(n) \qquad (27)$$

Where k is an output neuron, thus, equation 27 is similar to equation 19, with the index k instead of j denoting output neurons.

Deriving equation 27 concerning the function $y_j(n)$yields:

$$\frac{\partial\varepsilon(n)}{\partial y_j(n)} = \sum_k e_k\frac{\partial e_k(n)}{\partial y_j(n)} \qquad (28)$$

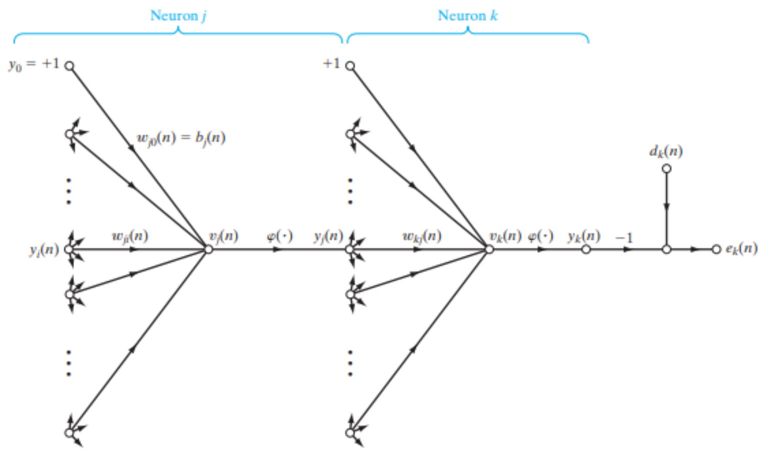

Figure 5: represents the signals within the network at neuron j in the case when it is located in a hidden layer (Haykin, 2009).

Following the chain rule for partial differentiation $\frac{\partial \varepsilon(n)}{\partial y_j(n)}$ and a rearrangement of equation 28, we arrive at the equivalent form:

$$\frac{\partial \varepsilon(n)}{\partial y_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} \qquad (29)$$

However, by examining figure 5, we can observe:

$$e_k(n) = d_k(n) - y_k(n)$$
$$e_k(n) = d_k(n) - \varphi_j(v_j(n)) \qquad (30)$$

Where neuron k is an output neuron, thus:

$$\frac{\partial e_k(n)}{\partial v_k(n)} = -\varphi_k'(v_k(n)) \qquad (31)$$

Similarly, for neuron k, as depicted in Figure 5, the local field produced by:

$$v_k(n) = \sum_{j=0}^{m} w_{kj}(n) y_j(n) \qquad (32)$$

Where m is the total number of inputs (excluding the bias) applied to neuron k. Once again, the weight $w_{k0}(n)$is equivalent to the bias $b_k(n)$ in neuron k and corresponds to an input of 1. Deriving equation 32 concerning $y_j(n)$ leads to:

$$\frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n) \qquad (33)$$

By using equations 31 and 33 in equation 29, we obtain the required partial derivative:

$$\frac{\partial \varepsilon(n)}{\partial y_j(n)} = -\sum_k e_k(n) \varphi_k'(v_k(n))\, w_{kj}(n)$$
$$= -\sum_k \delta_k(n)\, w_{kj}(n) \qquad (34)$$

In the second line, we use the definition of the local gradient $\delta_k(n)$ given in equation 25, using the index k instead of j. Finally, by replacing equation 34 with equation 26, we obtain the reverse propagation algorithm for the local gradient $\delta_j(n)$, described as:

$$\delta_j(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n)\, w_{kj}(n) \qquad (35)$$

Where neuron j is hidden.

P-ISSN 2518-5764
E-ISSN 2227-703X

3. Discussion of Results :

The research sample is described in this chapter, and the necessary tests for the multiple regression model are conducted. After that, the methods mentioned in the second chapter were applied, and they were compared. The Python programming language was used to implement the techniques described in the previous chapter, and the statistical software SPSS was employed to extract the test results and describe the sample.

3.1 Sample Description :

The sample consists of one dependent variable and four explanatory variables, and includes 1250 observations. These data are extracted from a survey conducted on private-sector power generators in the year 2021 and were subsequently published in 2023 by the Industrial Statistics Directorate, an entity associated with the Central Statistical Organization under the purview of the Iraqi Ministry of Planning.

Table 1: Variables description

Variable	Variable representation
X1	Number of subscribers at the station
X2	Quantity of water used for cooling (cubic meters)
X3	Operating hours in the summer season
X4	Generator-designed capacity (kV units)
Y	Station utilized capacity (kV units)

Table 2: Descriptive statistics of the variables

	y	**x1**	**x2**	**x3**	**x4**
count	1250	1250	1250	1250	1250
mean	275.4288	257.6272	73.7272	10.9672	519.52
std	140.4153	142.5748	424.5714	3.036693	346.0633
min	20	15	0	3	20
25%	188	160	6	9	250
50%	250	225	20	10	450
75%	350	310	45	12	700
max	800	1070	10000	20	3600

The sample was divided into two parts: the first part was allocated for model training, representing 80% of the total sample size, and the remaining part was allocated for model testing, representing 20% of the total sample size. Before starting the application and ascertaining any issues related to the regression model with the dependent variable, the Durbin-Watson test was administered to assess the presence of autocorrelation problems. The test findings indicated an absence of autocorrelation problems, as the test statistic registered a value of 2.054, which is near two, signifying minimal autocorrelation. Furthermore, with the Kolmogorov-Smirnov test for normal distribution, shows that the data does not follow a normal distribution. The test resulted in a value of 0.123, with a significance level of (8.29E-41), which is less than 0.05, and using the Variance Inflation Factor (VIF) test to detect multicollinearity issues in the model, it is evident that there is no problem. This is because all VIF values are less than five, as illustrated in Table (3).

Table 3: VIF Values for Independent Variables in the Model

Independent Variable	VIF
X1	1.369
X2	1.000
X3	1.08
X4	1.46

3.2 M-Estimator

The M-Estimator method was applied using the Huber and Tukey functions by using python programming language, and the results were extracted and interpreted as follows.

3.2.1M- Estimator (Huber)

Table 4: The result of M-Estimator using the Huber function

	prams	**Se**	**t_value**	**p_value**
$\boldsymbol{\beta_0}$	49.74416	9.097573	5.46785	4.56E-08
$\boldsymbol{\beta_1}$	0.513755	0.018887	27.2016	6.2E-163
$\boldsymbol{\beta_2}$	0.009401	0.00539	1.74407	0.081147
$\boldsymbol{\beta_3}$	-0.12175	0.751089	-0.1621	0.871227
$\boldsymbol{\beta_4}$	0.172553	0.007684	22.45606	1.1E-111

From the previous table, we observe the t-test values for the model coefficients to evaluate the variables' significance in influencing the model. It is clear that variables x2 and x3 do not significantly affect the dependent variable, as their p-values exceed 0.05. Conversely, variables x1 and x4 significantly impact the dependent variable. Furthermore, when applying the Huber function for M-estimation, the coefficient of determination (R-squared) equals 0.672538. This indicates that the model utilizing the Huber function explains approximately 67% of the variance in the dependent variable. The mean squared error for the estimators in this model is 6433.258.

3.2.2M- Estimator (Tukey)

Table 5: The result of M-Estimator using the tukey function

	prams	**Se**	**t_value**	**p_value**
$\boldsymbol{\beta_0}$	50.41273	9.034456	5.580052	2.4E-08
$\boldsymbol{\beta_1}$	0.512625	0.018756	27.33136	1.8E-164
$\boldsymbol{\beta_2}$	0.003874	0.005353	0.723746	0.469221
$\boldsymbol{\beta_3}$	-0.00834	0.745878	-0.01118	0.991081
$\boldsymbol{\beta_4}$	0.165768	0.007631	21.72377	1.2E-104

From the above table, we observe the t-test values for model coefficients to ascertain the variables that significantly impact the model. It is evident that, within this model, variables x2 and x3 lack a substantial effect on the dependent variable (response variable) with p-values of 0.469221 and 0.991081, respectively. These p-values exceed the significance level of 0.05. In contrast, variables x1 and x4 notably impact the dependent variable. Moreover, utilizing the M-Estimator with the Tukey function yields a coefficient of determination (R-squared) of 0.669581, signifying strong explanatory power, explaining around 67% of the variance in the dependent variable. The mean squared error for the estimators in this model is 6491.341.

P-ISSN 2518-5764
E-ISSN 2227-703X

3.3 Artificial Neural Networks :

Artificial neural networks with a feedforward multilayer architecture will be used. The procedure begins with the initial definition of the neural network's structure, followed by the model selection process, which involves evaluating models using Mean Squared Error (MSE).

3.3.1 Architecture of Artificial Neural Networks:

1)Input layer with four neurons, where each neuron represents a descriptive variable.
2)There are one to four hidden layers, with varying numbers of neurons in each layer compared to the subsequent layer.
3)The same activation function, relu, sigmoid, or tanh, is used for all layers in each neural network.
4)Output layer containing a single neuron with no activation function.
5)Each neuron, whether in a hidden or output layer, is associated with a bias weight representing a constant threshold within the neuron before entering the activation function.

3.3.2 Evaluation of Artificial Neural Network Models:

Table 6: The Evaluation of Artificial Neural Network Models

Activation Function	**Train data**		**Test data**	
	MSE	R^2	**MSE**	R^2
ML-FF(4,16, 1)				
relu	5916.963	0.698818	5365.244	0.730506
sigmoid	9876.772	0.497258	10609.46	0.467091
tanh	9730.613	0.504697	10262.46	0.484521
ML-FF(4,16,32,1)				
relu	5595.957	0.715158	5338.434	0.731853
sigmoid	9184.017	0.53252	8961.771	0.549854
tanh	9444.229	0.519275	9610.343	0.517277
ML-FF(4,16,32,46,1)				
relu	5053.547	0.742767	5483.955	0.724543
sigmoid	6960.26	0.645713	6986.538	0.649069
tanh	10648.08	0.457997	10797.91	0.457626
ML-FF(4,16,32,46,128,1)				
relu	4856.608	0.752791	5337.586	0.731895
sigmoid	7218.665	0.632559	7143.15	0.641203
tanh	6913.12	0.648112	6476.307	0.674698

From Table (6), it is clear that the artificial neural network with four hidden layers, denoted as ML-FF(4,16,32,46,128,1), and using the relu activation function, shows the lowest mean squared error (MSE) among all the employed networks. Furthermore, the test results in the neural networks closely align with the training data results, indicating the proficiency of neural networks in estimation without merely memorizing the training data.

Additionally, the MSE values in the previous table reveal that networks using the sigmoid and tanh activation functions struggle to handle outliers effectively. In contrast, networks using the relu activation function can effectively manage outliers in the data, bypassing all the robust methods.

Moreover, from Table (6), it is clear that MSE values decrease as the number of layers in artificial neural networks with the relu activation function increases. Conversely, in networks with the sigmoid activation function, MSE values decrease initially and then experience a slight increase in the fourth layer. For networks with the Tanh activation function, MSE values decrease in the second layer, increase in the third layer, and decrease significantly in the fourth layer.

3.4Comparison between Robust Estimator M and Artificial Neural Networks:

Based on the previous results of both methods, it is clear that the M-estimator using the Huber function outperforms the M-estimator using the Tukey function. Moreover, the most effective artificial neural network architecture is the ML-FF(4,16,32,46,128,1) with the relu activation function, which also outperforms the M-estimator. These findings highlight the capability of some artificial neural networks to handle outlier values, as shown in the following table.

Table 7: Comparison between Robust Estimator M and Artificial Neural Networks

model	**train**		**test**	
	MSE	R^2	**MSE**	R^2
M huber	12060.77	0.81682	10205.44	0.822188
M tukey	13678.46	0.79225	12077.81	0.789565
ML-FF(4,16,32,46,128,1)	8497.73	0.870936	9346.371	0.837156

4. Conclusion:

- The results indicated that the M-Estimator using the Huber function outperformed the M-Estimator using the Tukey function.
- The most effective artificial neural network architecture was identified as ML-FF(4,16,32,46,128,1) with the relu activation function. This neural network model demonstrated superior outlier capabilities and showed predictive performance.
- The analysis confirmed the significant impact of the number of hidden layers and the choice of activation function on artificial neural network performance. Networks using the relu activation function showed enhanced resilience to outliers and provided more accurate predictions, while networks using sigmoid and tanh activation functions struggled to manage outlier values effectively.
- The results of this study underscore the significance of choosing methods and parameters wisely when analyzing real-world data. Using the Huber function and well-structured artificial neural networks, the M-Estimator can provide valuable insights and accurate predictions for complex datasets.

Authors Declaration:

Conflicts of Interest: None

-We Hereby Confirm That All The Figures and Tables In The Manuscript Are Mine and Ours. Besides, The Figures and Images, Which are Not Mine, Have Been Permitted Republication and Attached to The Manuscript.

- Ethical Clearance: The Research Was Approved By The Local Ethical Committee in The University.

P-ISSN 2518-5764
E-ISSN 2227-703X

References:

1.Abu Shaer, M. J., and Al-Sarraf, N. M. 2018. Linear Regression: Insights from Basics to Mastery. Abdul Salam Printing, First Edition.

2.Al-Athari,F.,Al-Amleh,M.. 2016.A Comparison between Least Trimmed of Squares and MM Estimation in Linear Regression Using Simulation Technique," in Proceedings of the International Arab Conference on Mathematics and Computations, Zarqa University, Zarqa, Jordan

3. Al-Azzawi, A. A., and Mohammed, L. A. 2022, Robust Estimation of Partial Regression Model using Wavelet Threshold, Journal of Economic and Administrative Sciences, 28(133) , pp. 97-113.

4.Ali, S. H. 2019. Selecting the Best Estimation of Robust Regression Model Parameters with Practical Application. Master's Thesis, University of Karbala.

5.Al-Tha'labi, S. H., and Omran, K. M. 2016, Determining the Best Model for Predicting Electricity Consumption in the Southern Region, Journal of Economic and Administrative Sciences, 22(90), pp. 437-457.

6.AL Zirej,Z. , Hadi, H., 2020, An Artificial Neural Network for Predicting Rate of Penetration in AL Khasib Formation ,*Iraqi Journal of Science*, 61(5), pp: 1051-1062

7.Amorim, L., Cavalcanti, G., and Cruz, R. 2022. The choice of scaling technique matters for classification performance. *Applied Soft Computing*, 133, pp 1-37.

8.Arshid, A. J., and Saleh, R. A. 2022, Robust Estimations for Single-parameter Asymmetrical Regression Model, *Journal of Economic and Administrative Sciences*, 28(134), pp.147-159.

9.Bahez, Z., K., Rasheed, H., A., 2022, Estimation of a Multiple Linear Regression Model Using Some Robust Methods, *Mathematical Statistician and Engineering Applications*, 71(4), pp 4944-4954

10. Bai, Y. 2022. RELU-Function and Derived Function Review. SHS Web Conf., Vol 144

11. Hajjaj, A. W., and Abdelqader, M. 2020. Using artificial intelligence and robust regression to predict birth rates in Egypt. *Scientific Journal of the Colleges of Commerce - Al-Azhar University*, 24, pp. 600-637.

12. Hassan, T. A., and Reda, M. S. 2011. Using Robust Regression to Find the Suitable Model for Representing Weather Data in Erbil City during the Period (1998-2010). *Journal of Management and Economics*, 89, pp.195-213.

13. Haykin, S. 2009. Neural Networks and Learning Machines. Pearson Education, Third Edition.

14. Imam,R.,Tarawneh,B., 2014,Regression versus Artificial Neural Networks: Predicting Pile Setupfrom Empirical Data,KSCE *Journal of Civil Engineering* , 18,pp 1018-1027

15. Jabbar, A.T., Mohammed, N. J., 2020, Comparison between Linear Time Series and Data Mining Techniques for Predicting Air Pollution in Baghdad City, *Al-Rafidain University College Journal of Science*, 46, pp. 220-238.

16. Jawad, F., A., 2015, A Comparison between Genetic Algorithms and Neural Networks in Estimating the Mediator Location for Nonparametric Multivariate Regression Models, Doctoral Dissertation, University of Baghdad

17. Kamal, G. I., Khalil, S. I. 2021. Comparison between Ordinary Least Squares (OLS) and Two-Stage Robust Weighted Least Squares (TSRWLS) in Estimating Parameters of Multiple Linear Regression Model in the Presence of Heteroscedasticity and Outliers in Response Variable. *Journal of Rafidain University College of Science*, 50, pp. 75-86.

18. Khalaf, K., I., 2017. A Comparison between Artificial Neural Networks and Nonparametric Regression in Estimating Atomic Radiation Doses. Master's thesis, University of Baghdad.

19. Khalil, M. I., and Mohammed, L. A. 2012, Robust Estimates in Cluster Analysis with Practical Application in the Field of Administrative and Financial Corruption, *Journal of Economic and Administrative Sciences*, 18(69), pp. 278-302.

P-ISSN 2518-5764
E-ISSN 2227-703X

20. Mahdi, D. I. 2010. Study of Robust M Estimation for Linear and Nonlinear Regression Models. Diyala Journal, Issue 46.
21. Mohammed, D. T. 2023. Estimation of a Seemingly Unrelated Regression Equation System Using Some Robust Estimators. Master's thesis, University of Baghdad.
22. Salman, M., J., Salah, R. A., 2022, Comparison of some artificial neural networks for graduate students, *Periodicals of Engineering and Natural Sciences Original Research*, 10, (3), pp 187-196
23. Sharma, S., Sharma, S., and Athaiya, A. 2020. Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology*, 4, pp. 310-316.
24. Silva, I., Liboni, L., Spatti, D., Alves, S., and Flauzino, R. 2017. Artificial Neural Networks: A Practical Course. Springer.
25. Zaman, T., Bulut, H., 2018, Modified ratio estimators using robust regression methods, Communications in Statistics – Theory and Methods, DOI:10.1080/03610926.2018.1441419

Journal of Economics and Administrative Sciences
2024; 30(140), pp. 466-514
P-ISSN 2518-5764
E-ISSN 2227-703X

تقدير انموذج الانحدار باستعمال طريقة M-estimation والشبكات العصبية الاصطناعية في ظل وجود القيم الشاذة

حسين طالب جواد
جامعة بغداد / كلية الإدارة والاقتصاد/ قسم الاحصاء
hussein.taleb2101m@coadec.uobaghdad.edu.iq

رباب عبد الرضا صالح
جامعة بغداد / كلية الإدارة والاقتصاد/ قسم الاحصاء
rabab.saleh@coadec.uobaghdad.edu.iq

Received:9/10/2023 Accepted: 20/12/2023 Published Online First: 30 /4/ 2024

مستخلص البحث:

هدفت الدراسة الى التنبؤ باستعمال نموذج الانحدار في ظل وجود القيم الشاذة في بيانات الدراسة اذ تم التطرق في هذا البحث الى مفهوم القيم الشاذة والكشف عنها وتقدير الانموذج من خلال الطرائق الحصينة المتمثلة بمقدر m والشبكات العصبية الاصطناعية متعددة الطبقات والمقارنة بينهما وتم تطبيق الطرائق على بيانات حقيقية متمثلة ببيانات مسح مولدات القدرة الكهربائية للقطاع الخاص لسنة 2021 وتم تقييم النماذج باستخدام متوسط الخطأ التربيعي (MSE) ومعامل التحديد. أظهرت النتائج أن المقدر M مع دالة Huber يتفوق على نظيره مع دالة Tukey. كانت بنية الشبكة العصبية الاصطناعية الأفضل أداءً هي (5،16،32،46،128،1) MLP-FF مع دالة تنشيط relu. تعاملت هذه الشبكة بشكل فعال مع القيم المتطرفة وأظهرت قدرات تنبؤية قوية. اذ أثر اختيار وظيفة التنشيط وعدد الطبقات المخفية بشكل كبير على أداء الشبكة العصبية اذ أظهرت النتائج تفوق هذه الشبكة العصبية الاصطناعية على المقدرات الحصينة

نوع البحث: ورقة بحثية.

الكلمات الرئيسة: الانحدار، مقدرM, الشبكات العصبية الاصطناعية، القيمة الشاذة، الانحدار الحصين، دالة التنشيط.

*بحث مستل من رسالة ماجستير