International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.3 (2025) pp. 3759-3767

http://www.ijcesen.com

ISSN: 2149-9144

Research Article

The effect of some normalization methods on neural networks and robust methods with the presence of outliers

Rabab Abdulrida Saleh[1]*, Hussein Talib Jawad[2]

[1]Department of Statistics, College of Administration and Economics, University of Baghdad, Baghdad, Iraq
* **Corresponding Author Email:** rabab.saleh@coadec.uobaghdad.edu.iq - **ORCID:** 0009-0004-8147-217X

[2]Department of Statistics, College of Administration and Economics, University of Baghdad, Baghdad, Iraq
Email: hussein.taleb2101m@coadec,uobaghdad.edu.iq - **ORCID:** 0009-0004-8505-1495

Article Info:

DOI: 10.22399/ijcesen.1716
Received : 23 December 2024
Accepted : 05 April 2025

Keywords

Neural Networks
Recurrent Neural Network
Min-Max
Z-score
Robust Methods

Abstract:

This research aims to use some statistical transformations such as Z-Score and MIN_MAX to see how these transformations affect the performance of some robust methods and neural networks when there are outliers in the data and to compare the robust methods (LTS, MCD, and MM) and some neural networks including (RNN, GRU, and LSTM) with different activation functions represented by (Relu, Elu, and Selu).
The research sample included 3000 private sector electricity generators in Iraq for the year 2021 taken from the Central Statistical Organization.
The comparison was made using the mean square error (MSE) and using the statistical programs Python, R, and Excel.
The results showed that both normalization techniques significantly improved the performance of the models, especially Min-Max normalization is the best for all (robust methods and neural networks) and especially the superiority of neural networks RNN, especially with deeper structures, showing good performance across different activation functions. As for the robust methods, the MM method was consistently the best, giving the lowest mean square error across all normalization techniques.

1. Introduction

The problem of outliers is a common issue that arises for various reasons. The presence of these values in the data leads to the inability to obtain a good estimator for regression models or the ability to achieve good predictions in artificial neural networks. Therefore, we will use transformations to determine their effect on estimation in the presence of these values within the data. Typically, these transformations are used to standardize variable values and unify measurement units between variables. We will use (z-score and MIN_MAX) transformations and study their impact on outliers through the use of robust methods (LTS, MCD, MM) and neural networks (RNN, GRU, LSTM) with multiple layers.
In 2022, the researchers (Saleh and Salman) compared some artificial neural networks for graduate students, and a comparison was made between three types of neural networks (feedforward neural network (FFNN), backpropagation network (BPL), and recurrent neural network (RNN). The study concluded the lowest rate of false prediction was for the structure of the recurrent network.
In 2023, the researchers (Irshayyid and Saleh) Robust estimates for a three-parameter exponential to estimate the parameters using robust methods (Median-of-Means, Forward search, M-Estimation) [14],
while Jawad and Saleh (2024), Estimation of the Multiple Regression Model Using M Robust Methods, Artificial Intelligence Algorithms (Multilayer Feedforward Neural Network).
In this paper, we study the effect of data transformations Z-Score and Min-Max transformations on RNN, GRU, and LSTM architectures and robust methods. Using mean squared error (MSE) and coefficient of determination, we assess the impact of these techniques on survey data from private-sector energy generators in Iraq. The goal is to identify optimal of transformations and models to enhance robustness and predictive reliability in the presence of outliers [15].

2. Research Methodology

The study relied on clarifying some concepts and methods related to the research, in addition to clarifying the applied aspect based on realistic data on the electrical energy resulting from the use of generators in Iraq.

This research discusses the importance of using some transformations on the data to estimate models using neural networks and robust methods.

2.1 Normalization

Normalization is essential when dealing with variables that differ widely in their units and scale, which can result in unequal mean and standard deviation values across variables. Such discrepancies may cause certain variables to exert more influence on the model than others, leading to biased results where some variables dominate while others are marginalized [20]. Normalization addresses this issue by transforming variable values to a common scale, preserving the inherent differences in value ranges without distortion [17]

2.1.1 Min-Max

Min-Max normalization is a technique that applies a linear transformation to the original data, aiming to scale values proportionally within a defined range, typically between 0 and 1. This method adjusts each value based on the minimum and maximum values of the variable [11] as represented by the formula:

$$x_{new} = \frac{x-min(x)}{max(x)-min(x)} \qquad (1)$$

2.1.2 Z-score

Z-score normalization, also referred to as zero-mean normalization, standardizes data based on the mean and standard deviation of the original dataset. This technique transforms values so that the mean of the normalized data equals zero, and the standard deviation equals one [9,25], according to the formula

$$x_{new} = \frac{x-\bar{x}}{s_x} \qquad (2)$$

2.2 Outliers

Outliers are data points that deviate significantly from the majority of observations, often lying outside the expected range of variability or standard deviation for a given variable or population [3]

2.3 Robust regression

Robust regression is a valuable approach for analyzing data that contain influential outliers, as it builds models resilient to the distorting effects of such values. Traditional methods like ordinary least squares (OLS) assume that all data points follow a certain distribution, but outliers disrupt these assumptions, leading to skewed or biased predictions.

Robust regression methods address this by minimizing the impact of outliers, thus maintaining accuracy and reliability in the model estimates [18,23]

2.3.1 MM-Estimation

MM-Estimation is a robust regression technique aimed at minimizing the influence of outliers on the estimation of regression parameters. This method employs an S-estimator to achieve high efficiency and a significant breakdown point, defined as the proportion of outliers that can be present in the data before the estimates become unreliable [1,26].

The MM-estimation process begins by obtaining initial parameter values through the S- estimator. Subsequently, it calculates the weights assigned to each data point based on their residuals. The weighting function is given by

$$w_i(u_i) = \begin{cases} \left[1-\left(\frac{u_i}{c}\right)^2\right]^2, |u_i| \leq c \\ 0, |u_i| > c \end{cases} \qquad (3)$$

Where c = 4.685 and ui the standard deviation of the residuals, estimated using the Tukey objective function. This iterative approach enhances the robustness of the estimates by effectively down-weighting the influence of outlier values

2.3.2 Least Trimmed Squares (LTS)

Least Trimmed Squares (LTS) is a robust statistical technique used for estimating parameter vectors in linear regression models. It serves as an alternative to traditional regression methods by minimizing the sum of squared residuals while effectively mitigating the influence of outliers. The LTS estimator is formulated as follows:

$$\hat{\beta}_{(LTS)} = arg \min_{\beta} \sum_{i=1}^{h} r_i^{(2)} \qquad (4)$$

Where $ri^{(2)}$ represents squared residuals, and h is the trimming constant (the size of the partial samples), which falls within the range $\frac{n}{2} \leq h \leq n$. The remaining $n-h$ data points with the largest residual squares are trimmed [8].

To determine the value of h, the following equation is employed:

$$h = n(1-\alpha) + \alpha(p+1) \qquad (5)$$

Where α represents the percentage of outlier values, or trimming ratio when $h = n$ it
means the trimming ratio is zero, and the estimation is equal to the estimation of ordinary least
squares. α represents the trimming ratio, or the proportion of outliers in the dataset. If $h = n$, no trimming occurs, and the estimation aligns with that of ordinary least squares regression.
This methodology allows for more accurate parameter estimation in the presence of outlier values, enhancing the model's overall reliability.

2.3.3 Minimum Covariance Determinant (MCD) estimator

Minimum Covariance Determinant (MCD) estimator is a highly regarded robust statistical method for estimating multivariate location and scatter, particularly known for its resilience against outliers. This characteristic makes the MCD estimator invaluable for outlier detection and has led to its application across various domains, including finance, medicine, and chemistry. It is commonly employed in robust techniques such as multiple regression analysis, factor analysis, and principal component analysis [13].
The core concept of the MCD estimator involves identifying a subset of the data, referred to as the Minimum Covariance Determinant (MCD) subset, which possesses the smallest determinant. This subset is used to accurately estimate the center and scatter of the entire dataset, allowing for more reliable analysis in the presence of influential outliers [21]. The MCD approach enhances the robustness of statistical models, ensuring valid results even when the dataset contains outlier values.

2.4 Artificial Neural Network

Artificial neural network architectures have been developed from well-known models of the biological nervous system and the human brain itself. The processing units or computational components of the network are called artificial neurons, which are simplified models of biological neurons. These models are inspired by how electrical pulses are generated and propagated across the cell membrane. Typically, the outputs of the neural network are continuous variables and are nonlinear [33], The artificial neural network consists of [28]:
Input Layer, Hidden Layers, Output Layer.
The hidden layer and output layer may contain activation functions, which will be explained below.

2.4.1 Activation Functions

These are functions used in artificial neural networks to transform input signals into output signals, which serve as inputs to the next layer in the architecture. The activation function is applied to the output of the layer to obtain its outputs as inputs to the next layer [29].

2.4.1.1 ReLU (Rectified Linear Unit)

The ReLU function represents an almost linear function, which preserves the properties of linear models, making it easy to optimize [24], the function is defined as follows: -

$$f(x) = \begin{cases} x \; if \; x > 0 \\ 0 \; if \; x \le 0 \end{cases} \qquad (6)$$

2.4.1.2 ELU (Exponential Linear Unit)

ELU is a type of activation function that can alleviate the vanishing gradient problem by using a definition for positive values and improving learning properties [24], the function can be defined as follows:

$$f(x) = \begin{cases} x \; if \; x > 0 \\ \alpha(e^x - 1) \; if \; x \le 0 \end{cases} \qquad (7)$$

2.4.1.3 SELU (Scaled Exponential Linear Unit)

This function was developed to be faster in operation than the ELU function and can overcome the problem of the negative region present in the ReLU function, thus avoiding the gradient vanishing problem. In this way, SELU works in both positive and negative regions. The function can be defined as follows [10,19]:

$$f(x) = \lambda \begin{cases} x \; if \; x > 0 \\ \alpha(e^x - 1) \; if \; x \le 0 \end{cases} \qquad (8)$$

Where α and λ are constant values, usually approaching

$$\alpha \cong 1.6732 \,, \lambda \cong 1.0507 \qquad (9)$$

2.4.2 RNN (Recurrent Neural Network)

RNN is a type of artificial neural network that can process sequential data, forming a dynamic system where cell outputs depend not only on inputs but also on the previous outputs of the cell (hidden state) [5]

RNN cell is similar to regular feed forward neural networks, the input vector representing the current input xt is multiplied by a weight matrix and then passed through a nonlinear activation function to calculate the values for a layer of hidden units. These hidden units are then used to calculate corresponding outputs, and the main difference from the feed forward network lies in the recurrent connection, this connection increases the computational inputs to the hidden layer by the value of the hidden layer from the previous iteration, which is considered a form of memory [16,27] The outputs of an RNN cell are calculated using the following equations [30]:

$$h_t = tanh(x_t W_h + h_{t-1} u_h + b_h) \qquad (10)$$

$$y_t = W_y h_t + b_y \qquad (11)$$

Where x_t represents the cell inputs and y_t represents the network outputs. $W_q,\ u_q$ represent the weights of the cell, and b represents the bias. $h_t$represents the cell outputs in the current iteration [31]

2.4.3 Long Short-Term Memory (LSTM)

The LSTM model is a powerful recurrent network system designed to overcome the problem of vanishing gradients that arise when learning long-term dependencies. These cells are themselves recurrent networks

A vanilla LSTM unit consists of an input gate, output gate, forget gate and cell state. The forget gate allows the network to reset its state. The cell remembers values over random periods, and the three gates regulate the flow of information associated with the cell [12].

These gates in the LSTM cell enable it to maintain a more stable error that can be backpropagated over time steps and layers, allowing recurrent networks to continue learning over many time steps. These gates work together to learn and store information relevant to both long and short-term sequences [4] The cell states in LSTM are calculated through the following equations [22]:

$$\begin{aligned}
f_t &= \sigma\left(x_t W_f + h_{t-1} u_f + b_f\right) \\
i_t &= \sigma(x_t W_i + h_{t-1} u_i + b_i) \\
o_t &= \sigma(x_t W_o + h_{t-1} u_o + b_o) \qquad (12) \\
\tilde{C}_t &= tanh(x_t W_C + h_{t-1} u_C + b_C) \\
C_t &= \sigma\left(f_t + h_{t-1} i_t \times \tilde{C}_t\right) \\
h_t &= tanh(\tilde{C}_t) \times o_t
\end{aligned}$$

W_q , u_q contain the weights for inputs and recurrent connections, where i represents the input gate, o represents the output gate, f represents the forget gate or memory gate, C_t represents the current cell state, and $\tilde{C}_t$ is the new candidate value for the cell state. σ represents the sigmoid function

2.4.4 GRU (Gated Recurrent Unit)

GRU is a type of artificial recurrent neural network first introduced by [7]. It is similar to LSTM but with fewer parameters and also has gates like LSTM, which control the flow of information within the cell [2]

The formulation of GRU can be given by the following equations:

$$\begin{aligned}
r_t &= \sigma(x_t W_r + h_{t-1} u_r + b_r) \\
z_t &= \sigma(x_t W_z + h_{t-1} u_z + b_z) \\
\tilde{h}_t &= tanh(x_t W_h + u_h(r_t h_{t-1}) + b_f) \qquad (13) \\
h_t &= z_t \times h_{t-1} + (1 - z_t) \times \tilde{h}_t
\end{aligned}$$

Where r_t represents the reset gate, z_t represents the update gate, $\tilde{h}_t$ represents the candidate state, and h_t represents the outputs. x_t represents the inputs, and W_q , U_q represent the weights for each gate, and b_q represents the bias for each gate.

2.5 Comparison Metrics

2.5.1 Mean Squared Error (MSE)

It is the square of the difference between the estimated values and the true values, also known as the mean square deviations [6,32].

It can be expressed by the following formula

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} \qquad (14)$$

3. Discussion of Result

In this aspect of the study, practical application will be carried out on real data, which is the data of power generators in Iraq. The number of observations was 3000, with one dependent variable, which is the actual operating capacity. The independent variables are as follows:

$\underline{X_1}$: Number of people using the power generator.

$\underline{X_2}$: Number of amperes.

$\underline{X_3}$: Amount of water used for cooling, measured in cubic meters.

$\underline{X_4}$: Number of daily operating hours in summer.

$\underline{X_5}$: Design capacity of the generator, measured in kilovolts.

$\underline{Y}$: The power used in the station, measured in kilovolts.

Python and R were used to obtain the results of the comparison between the methods mentioned above, while Excel was used to create the figures

First, Outliers value in the data will be detected, by observing the Figure (1), it was confirmed that there were outliers in the data used in the research.

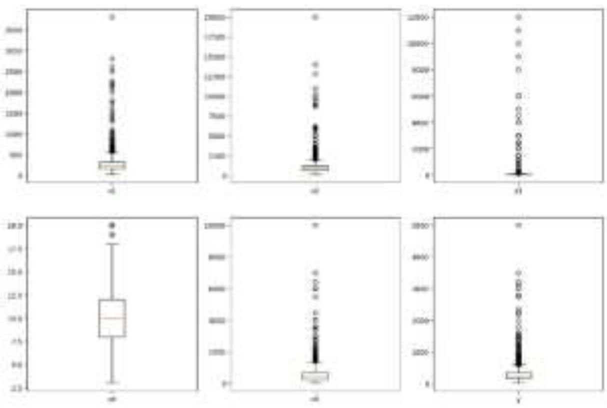

Figure 1. *Box Plot of Research Variables*

Second, several normalization methods were used on the data, including Min-Max, and Z-score. Both Robust methods and Neural Networks were applied before and after data normalization. The tables below show the results of the methods

Table 1. *MSE for the robust methods (LTS, MM, MCD)*

Method	No Normalization	Min Max	Z – Score
LTS	14602.25	0.000589	0.21874
MM	**13824.41**	**0.000557**	**0.207088**
MCD	20587.78	0.00082	0.304335

From Table 1, Firstly the Least Trimmed Squares (LTS) method demonstrates a Mean Squared Error (MSE) of 14,602.25 without any normalization. With Min-Max normalization, the MSE significantly decreases to 0.000589, This suggests that Min-Max normalization greatly enhances the model's precision. when Z-score normalization is applied, the MSE increases to 0.21874

Secondly for the MM method, the MSE is 13,824.41 without normalization, indicating it performs slightly better than LTS in terms of model fit under the same conditions. When Min-Max normalization is applied, the MSE decreases to 0.000557, with Z-score normalization, the MSE rises to 0.207088, the model still retains a good level of explanation.

And thirdly Minimum Covariance Determinant Method (MCD): The Minimum Covariance Determinant (MCD) method exhibits the highest MSE of 20,587.78 among the methods without normalization, indicating the poorest model performance. When Min-Max normalization is applied, there is decrease in MSE to 0.00082 with Z-score normalization, the MSE increases to 0.304335. The Figure 2 shows the results of the MSE values for the robust methods. From Table 2 at the first layer and with the architecture (5,16,32,1) and The Relu activation function , we find that the recurrent neural network (RNN) is the best when No Normalization is applied, Min-Max and Z-Score are 8818.166, 0.000362, 0.09232 respectively, then GRU and the worst is the Long Short-Term Memory (LSTM) neural network. And we find that the performance of all networks is better when use normalization Min-Max from use normalization Z-score.

Table 2. *MSE for (RNN, LSTM, GRU) in (5,16,32,1) Relu*

Method	No Normalization	Min-Max	Z-Score
RNN (	**8818.166**	**0.000362**	**0.09232**
LSTM	10430.51	0.000437	0.128695
GRU	10360.36	0.000427	0.114054

Table 3. *MSE for (RNN, LSTM, GRU) in (5,16,32,46,128,1) Relu*

Method	No Normalization	Min Max	Z – Score
RNN	**8019.848**	**0.000319**	**0.01971**
LSTM	9195.427	0.000509	0.116505
GRU	9364.466	0.000407	0.0683

From Table 3 at the second layer and with the architecture (5,16,32,64,128,1) and activation function Relu, we find that the recurrent neural network (RNN) is the best when No Normalization applied, Min-Max and Z-Score are 8019.848, *0.000319*, *0.01971* respectively, then GRU and the worst is the LSTM neural network.. And we find that the performance of all networks is better when use normalization Min-Max from use normalization Z-score.

Table 4. *MSE for (RNN, LSTM, GRU) in (5,16,32,1) Elu*

Method	No Normalization	Min-Max	Z-Score
RNN	**8770.89**	**0.000439**	0.127438
LSTM	10631.54	0.00046	0.122037
GRU	10150.01	0.000433	**0.115523**

From Table 4 at the first layer and with the architecture (5,16,32,1) and activation function Elu, we find that the recurrent neural network (RNN) is the best when No Normalization , Min-Max applied are 8770.89, **0.000439** respectively, then GRU in Z-Score is the best 0.115523 and the worst is the LSTM neural network. And we find that the performance of all networks is better when use normalization Min-Max from use normalization Z-score.

Table 5. *MSE for (RNN, LSTM, GRU) in (5,16,32,46,128,1) Elu*

Method	No Normalization	Min-Max	Z-Score
RNN	**7378.68**	**0.000449**	0.119734
LSTM	9867.532	0.000452	0.114292
GRU	9781.464	0.000451	**0.10542**

From Table 5 at the second layer and with the architecture (5,16,32,64,128,1) and activation function Eelu, we find that the recurrent neural network (RNN) is the best when No Normalization, Min-Max applied are 7378.68, *0.000449* respectively, then GRU in Z-Score is the best 0.10542 and the worst is the LSTM neural network. And we find that the performance of all networks is better when use normalization Min-Max from use normalization Z-score

Table 6. *MSE for (RNN, LSTM, GRU) in (5,16,32,1) Selu*

Method	No Normalization	Min-Max	Z-Score
RNN	**9200.705**	0.000430	0.127435
LSTM	10356.83	**0.000425**	0.114607
GRU	9342.068	0.000449	**0.111234**

From Table 6 at the first layer and with the architecture (5,16,32,1) and activation function Selu, we find that the recurrent neural network (RNN) is the best when No Normalization applied 9200.705, the best in LSTM neural network in Min Max is 0.000425 is 0.000425 , then GRU in Z-Score is the best 0.111243 And we find that the performance of all networks is better when use normalization Min-Max from use normalization Z-score

Table 7. *MSE for (RNN, LSTM, GRU) in (5,16,32,46,128,1) Selu*

Method	No Normalization	Min-Max	Z-Score
RNN	6650.176	0.000451	0.125067
LSTM	9674.384	0.00047	0.104169
GRU	**4684.58**	**0.000408**	**0.092007**

From Table 7 at the second layer and with the architecture (5,16,32,64,128,1) and activation function Selu, we find that the GRU is the best when No Normalization, Min-Max and Z-Score are 4684.58, 0.000408,0.092007 respectively, And we find that the performance of all networks is better when use normalization Min-Max from use normalization Z-score

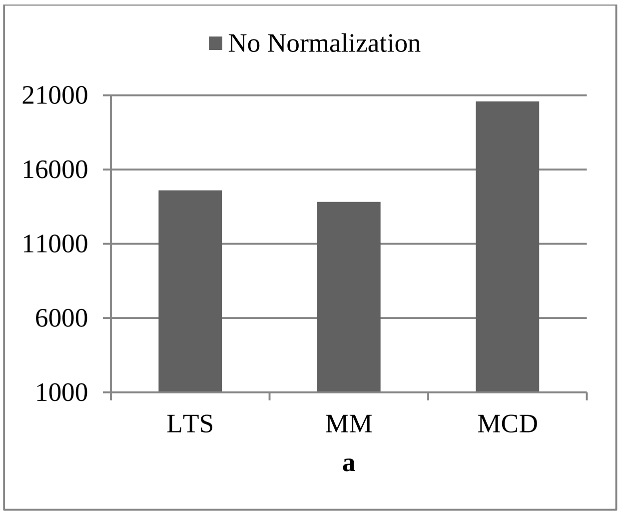

Figure 2. *MSE for Robust regression methods no-normalization*

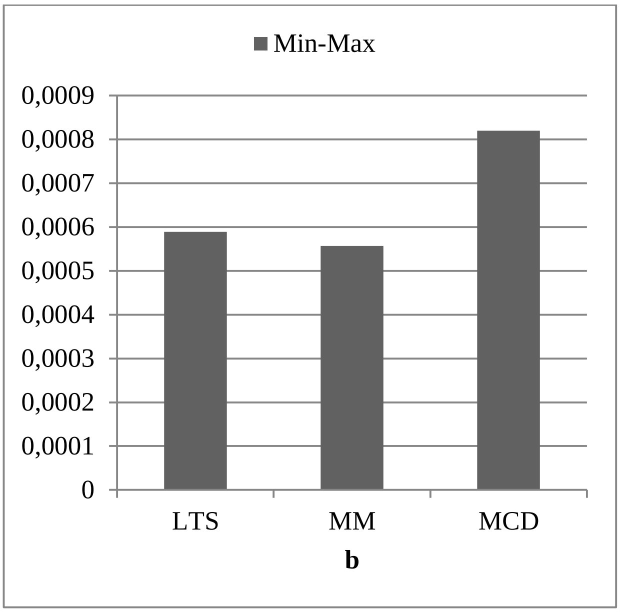

Figure 3. *MSE for Robust regression methods Min-Max*

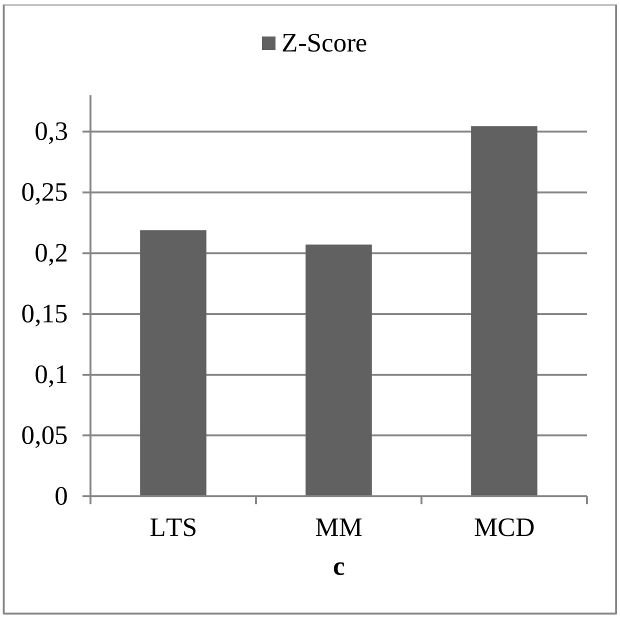

Figure 4. *MSE for Robust regression Z-score*

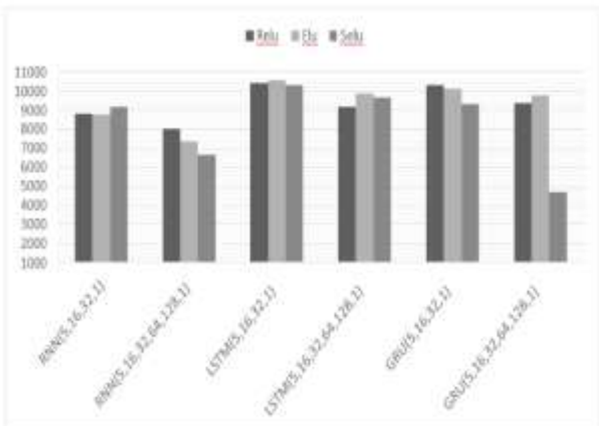

Figure 5. *MSE for all Artificial Neural Network in no-normalization*

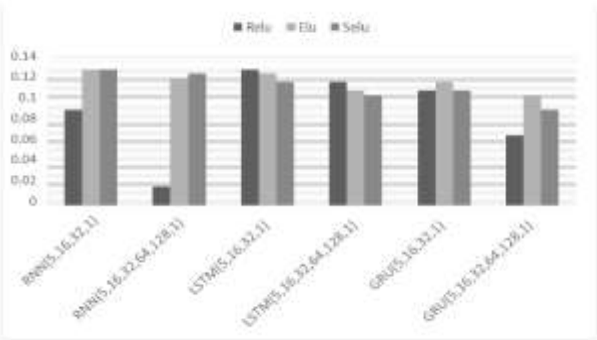

Figure 6. *MSE for all Artificial Neural Network in Min-Max*

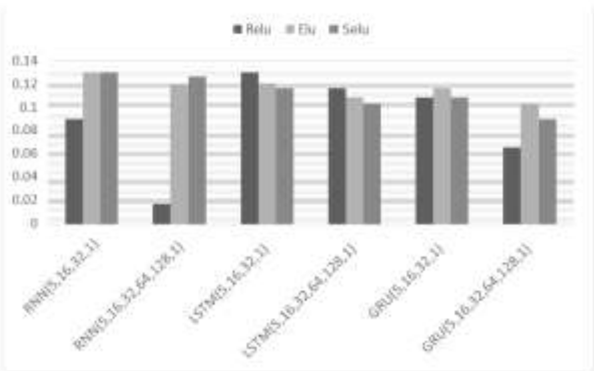

Figure 7. *MSE for all Artificial Neural Network in Z-score*

The figures (2,3,4) shows the results of the MSE values for the robust methods. We find a summary of the best methods, in no-normalization, Min-Max and Z-Score methods. the MM method, is the best.
In the figure (5) we find a summary of the best methods , in no-normalization methods GRU neural network (5,16,32,64,128,1) with the selu activation function achieves, is the best.
In the figure (6) we find a summary of the best methods When using the Min-Max transformation, the RNN neural network (5,16,32,64,128,1) with the Relu activation function is the best.
In the figure (7) we find a summary of the best methods When using the Z-score transformation, the RNN neural network (5,16,32,64,128,1) with the Relu activation function is the best.

4. Conclusion

The results of the practical application yielded several important conclusions regarding the effectiveness of robust methods and neural networks in managing data variability and outlier impacts.
Both normalization techniques significantly improve the performance of the models, especially the normalization of Min-Max is the best for all (Robust methods and neural networks).
For the Robust methods, the MM method was consistently the best, giving the lowest mean square error across all normalization techniques, indicating that it is the most robust method for this dataset, followed by the LTS method. The worst was the MCD method, which is less effective in dealing with outliers compared to LTS and MM.
As for the neural networks, the results indicate that RNN models, especially with deeper architectures, show superior performance across different activation functions and normalization techniques, especially with the Relu and Elu activation functions. It is worth noting that the Selu activation function in the GRU model with depth shows great performance, confirming its effectiveness the accuracy of the model. The worst performance of the networks was for the LSTM neural network. In general, the choice of model architecture, activation function, and normalization technique plays a crucial role in improving performance in artificial neural networks.
The findings underscore the critical role of normalization in enhancing the performance of neural networks while reaffirming the robustness of certain statistical methods in the presence of outliers.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.

- **Author contributions:** The authors declare that they have equal right on this paper.

- **Funding information:** The authors declare that there is no funding to be acknowledged.

- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Alma, Ö. (2011). Comparison of Robust Regression Methods in Linear Regression. *International Journal of Contemporary Mathematical Sciences*. 6(9);409-421. https://avesis.deu.edu.tr/dosya?id=23aefcae-6b05-402d-89e8-ff6a8da2fc2b

[2] ArunKumar, K. E., Kalaga, D. V., Kumar, C. M., Kawaji, M., & Brenza, T. M. (2022). Comparative analysis of Gated Recurrent Units (GRU), long Short-Term memory (LSTM) cells, autoregressive

integrated moving average (ARIMA), seasonal autoregressive integrated moving average (SARIMA) for forecasting COVID-19 trends. *Alexandria Engineering Journal*. 61;7585-7603. https://doi.org/10.1016/j.aej.2022.01.011

[3] Begashaw, G. B., & Yohannes, Y. B. (2020). Review of Outlier Detection and Identifying Using Robust Regression Model. *International Journal of Systems Science and Applied Mathematics*. 5(1);4-11. http://dx.doi.org/10.11648/j.ijssam.20200501.12

[4] Bouktif, S., Fiaz, A., Ouni, A., & Serhani, M. A. (2018). Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches. *Energies*. 11(1636);1-20. https://doi.org/10.3390/en11071636

[5] Chen, G. (2018). A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation. *arXiv*. https://arxiv.org/pdf/1610.02583.pdf

[6] Chicco, D., Warrens, M., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*. 7(3);e623. http://dx.doi.org/10.7717/peerj-cs.623

[7] Cho, K., Merriënboer, B. V., & Bahdanau, D. (2014). On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. *arXiv:1409.1259v2*. https://doi.org/10.3115/v1/W14-4012

[8] Čížek, P., & Víšek, J. Á. (2000). Least trimmed squares. *SFB373 Discussion Paper*. No. 2000,53. https://hdl.handle.net/10419/62211

[9] Fan, C., Chen, M., Wang, X., Wang, J., & Huang, B. (2021). A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data. *Frontiers in Energy Research*. 9;652801. https://doi.org/10.3389/fenrg.2021.652801

[10] Feng, J., & Lu, S. (2019). Performance Analysis of Various Activation Functions in Artificial Neural Networks. *Journal of Physics: Conference Series*. 1237;022030. https://doi.org/10.1088/1742-6596/1237/2/022030

[11] Henderi, Wahyuningsih, T., & Rahwanto, E. (2021). Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer. *International Journal of Informatics and Information System*. 4(1);13-20. http://dx.doi.org/10.47738/ijiis.v4i1.73

[12] Houdt, G. V., Mosquera, C., & Napoles, G. (2020). A Review on the Long Short-Term Memory Model. *Artificial Intelligence Review*. 53;5929-5955. https://doi.org/10.1007/s10462-020-09838-1

[13] Hubert, M., Debruyne, M., & Rousseeuw, P. J. (2017). Minimum Covariance Determinant and Extensions. *WIREs Computational Statistics*. 2017;wics.1421. https://doi.org/10.1002/wics.1421

[14] Irshayyid, A. J., & Saleh, R. A. (2023). Robust estimates for a three-parameter exponential regression model. *Nonlinear Analysis and Applications*. 14(1);2799-2808. http://dx.doi.org/10.22075/ijnaa.2023.29395.4148

[15] Jawad, H. T., & Saleh, R. (2024). Estimation of the Regression Model Using M-Estimation Method and Artificial Neural Networks in the Presence of Outliers. *Journal of Economics and Administrative Sciences*. 30(140);688-716. https://doi.org/10.33095/g4hems75

[16] Jurafsky, D., & Martin, J. H. (2024). Speech and Language Processing. *Third Edition draft*. https://web.stanford.edu/~jurafsky/slp3/ed3bookfeb3_2024.pdf

[17] Kappal, S. (2019). Data Normalization using Median & Median Absolute Deviation (MMAD) based Z-Score for Robust Predictions vs. Min – Max Normalization. *London Journal of Research in Science: Natural and Formal*. 19(4);39-44.

[18] Bahez, Z. K., & Rasheed, H. A. (2022). Comparing Some of Robust the Non-Parametric Methods for Semi-Parametric Regression Models Estimation. *Journal of Economics and Administrative Sciences*. 28(132);105-117. https://doi.org/10.33095/jeas.v28i132.2275

[19] Kılıçarslan, S., Adem, K., & Çelik, M. (2021). An overview of the activation functions used in deep learning algorithms. *Journal of New Results in Science*. 10(3);75-88. https://doi.org/10.54187/jnrs.1011739

[20] Li, C. (2019). Preprocessing Methods and Pipelines of Data Mining: An Overview. *Machine Learning*. https://doi.org/10.48550/arXiv.1906.08510

[21] Mahdi, M. H., & Hussein, S. M. (2023). Estimating the Population Mean in Stratified Random Sampling Using Combined Regression with the Presence of Outliers. *Journal of Economics and Administrative Sciences*. 29(136);70-80.

[22] Mateus, B. C., Mendes, M., Farinha, J. T., Assis, R., & Cardoso, A. M. (2021). Comparing LSTM and GRU Models to Predict the Condition of a Pulp Paper Press. *Energies*. 14(6958). https://doi.org/10.3390/en14216958

[23] Nugrahani, I., Susanti, Y., & Qona'ah, N. (2021). Modeling of Rice Production in Indonesia Using Robust Regression with The Method of Moments (MM) Estimation. *Basic and Applied Science Conference (BASC) 2021*. NST Proceedings;79-87. https://doi.org/10.11594/nstp.2021.1111

[24] Nwankpa, C. E., Ijomah, W., Gachagan, A., & Marshall, S. (2021). Activation functions: comparison of trends in practice and research for deep learning. *2nd International Conference on Computational Sciences and Technology*. 124-133. https://doi.org/10.48550/arXiv.1811.03378

[25] Panigrahi, S., & Behera, H. S. (2013). Effect of Normalization Techniques on Univariate Time Series Forecasting using Evolutionary Higher Order Neural Network. *International Journal of Engineering and Advanced Technology*. 3(2);280-285.

[26] Rahayu, D. A., Nursholihah, U. F., & Suryaputra, G. (2023). Comparasion of The M, MM and S Estimator in Robust Regression Analysis on Indonesian Literacy Index Data 2018. *EKSAKTA Journal of Sciences and Data Analysis*. 4(1);11-22. https://doi.org/10.20885/EKSAKTA.vol4.iss1.art2

[27] Saleh, R. A., & Salman, M. J. (2022). Comparison of some artificial neural networks for graduate students. *Periodicals of Engineering and Natural Sciences Original Research*. 10(3);187-196. https://doi.org/10.21533/pen.v10i3.304

[28] Sharma, S., Sharma, S., & Athaiya, A. (2020). Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology*. 4;310-316. https://doi.org/10.33564/IJEAST.2020.v04i12.054

[29] Shewalkar, A., Nyavanandi, D., & Ludwig, S. A. (2019). Performance Evaluation of Deep neural networks Applied to Speech Recognition: Rnn, LSTM and GRU. *Journal of Artificial Intelligence and Soft Computing Research*. 9(4);235-245. https://doi.org/10.2478/jaiscr-2019-0006

[30] Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R. (2023). A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU. *arXiv:2305.17473*. https://doi.org/10.48550/arXiv.2305.17473

[31] Silva, I. N., Spatti, D. H., Flauzino, R. A., Liboni, L. H., & Alves, S. F. (2017). Artificial Neural Networks A Practical Course. *Springer*. https://doi.org/10.1007/978-3-319-43162-8

[32] Tatachar, A. V. (2021). Comparative Assessment of Regression Models Based On Model Evaluation Metrics. *International Research Journal of Engineering and Technology*. 8(9);853-860.

[33] Zarzycki, K., & Ławryńczuk, M. (2021). LSTM and GRU Neural Networks as Models of Dynamical Processes Used in Predictive Control: A Comparison of Models Developed for Two Chemical Reactors. *Sensors*. 21(5625). https://doi.org/10.3390/s21165625