Correlation and Regression in Computer Science: Differences, Applications, and Result Interpretation

Ivan Trenchev [1,2], Radoslav Mavrevski [2], Miglena Trencheva [2], Tereza Trencheva [1]

[1] *University of Library Studies and Information Technologies, Sofia, Bulgaria*
[2] *South-West University "Neofit Rilski", Blagoevgrad, Bulgaria*

Abstract **– Understanding the distinction between correlation and regression is essential for interpreting statistical relationships in computer science experimental data. Although both methods explore associations between variables, they serve different purposes and offer distinct insights. Correlation measures how strongly two variables are related and whether this relationship is positive or negative, without implying causation. Regression, on the other hand, models predictive relationships, allowing one variable to be estimated based on another. Despite their widespread use, these concepts are often confused in computer science research and data analytics, leading to misinterpretation of results. This paper clarifies the conceptual and practical differences between correlation and regression, outlines when each method is appropriate, and demonstrates their application through computer science–related examples.**

Keywords **– Correlation, regression, computer science statistics, predictive modeling, variable relationships.**

DOI: 10.18421/TEM153-14
https://doi.org/10.18421/TEM153-14

Corresponding author: Radoslav Mavrevski,
South-West University "Neofit Rilski",
Blagoevgrad, Bulgaria
Email: mavrevski@swu.bg

Received: 04 November 2025.
Revised: 20 April 2026.
Accepted: 27 April 2026.
Published: 27 August 2026.

1. Introduction

This section introduces the fundamental concepts of regression analysis, outlines its historical development, and explains its relevance and application in computer science, particularly in relation to variable relationships and predictive modeling.

In 1886, Sir Francis Galton first introduced the concept of regression in statistics. While studying the inheritance of physical traits, particularly height, Galton observed that children of exceptionally tall or short parents tended to exhibit heights closer to the population average. This phenomenon, initially called "regression toward mediocrity" by Galton, is now known as regression toward the mean [1]. This foundational work laid the groundwork for modern regression analysis, which is now widely applied in computer science for predictive modeling, algorithm performance evaluation, and system optimization.

Regression analysis in modern statistics examines how a dependent variable is influenced by one or more independent (explanatory) variables [2]. According to [3], regression analysis is used to predict the average value of a dependent variable from the values of explanatory variables . It is crucial to note that a strong statistical relationship does not imply necessarily causation. The establishing causality requires theoretical justification beyond statistical evidence.

In any regression framework, independent variables are those that can often be controlled or manipulated, such as input size, while dependent variables reflect outcomes, like algorithm execution time. Experimental research involves altering the independent variable to examine its impact on the dependent variable. In observational or non-experimental studies, variables are observed without direct control, and the assignment of dependent and independent roles is more interpretive, for example, analyzing the relationship between CPU usage and memory consumption across different software applications.

There are two broad types of variable relationships: **deterministic** and **stochastic**. In deterministic relationships, a specific value of the independent variable leads to a predictable and fixed outcome (assuming all other conditions are strictly controlled and constant), such as calculating the execution time of an algorithm given a fixed input size and system configuration [4], [5]. In contrast, stochastic relationships involve uncertainty; the same value of the independent variable may correspond to different outcomes due to the influence of additional, unmeasured factors [6]. For instance, in real-world environments, the runtime of a software application may vary even if the input size and system configuration remain constant, due to factors like background processes, memory fragmentation, or network latency.

This variability is typically visualized using a scatter plot. In stochastic relationships, data points form a cloud around a central trend line, rather than aligning perfectly along a deterministic curve. The distance of the data points from the regression line represents the degree of association between the variables.

An important consideration is that some observed relationships, while statistically present, may be spurious or illegitimate [7], [8]. For example, page load time and memory usage may both correlate with the number of active users on a web application. While the relationship between user load and each of these variables is logical, a direct correlation between page load time and memory usage may be misleading, as both are influenced by a third variable – the overall system load.

In the field of computer science, understanding the distinction between correlation and regression is vital. Analysts often encounter scenarios where two performance metrics are correlated, yet such a relationship does not automatically indicate causality. In the context of computer performance analytics, correlation quantifies the degree and direction of association between two variables, while regression examines the predictive relationship of one variable on another [9].

Overall, the increasing availability of large-scale data in computer science highlights the growing importance of statistical methods such as regression and correlation analysis. These techniques are widely applied in areas such as performance evaluation, machine learning, and system optimization, where understanding relationships between variables is essential for effective decision-making. This reinforces the need for accurate modeling and careful interpretation of statistical results in modern computational research.

2. Material and Methods

This section presents the methodological framework used in the study, including linear regression, multiple regression, and correlation analysis.

2.1. Linear Regression

Linear regression provides a mathematical framework to describe, interpret, or predict the values of a dependent variable based on one or more independent variables [10]. The linear regression model is expressed mathematically as:

$$Y = \alpha + \beta X + \epsilon \quad (1)$$

where Y represents the dependent variable, X the independent variable, α and β are constants, and ε is the error term. Linear regression is a particular form of regression where the relationship between variables is assumed to be straight-line.

For example, an algorithm's execution time may be estimated based on the size of its input dataset:

$$Y = \alpha + \beta 1X1 + \varepsilon \quad (2)$$

where Y represents the execution time, X_1 is the input size, and ε represents the error.

2.2. Multiple Regression

In multiple regression analysis, the outcome variable is modeled as a function of several predictor variables. This is useful when multiple factors influence the dependent variable simultaneous [11]. For example, one might predict the execution time of an algorithm based not only on the input size but also on the number of concurrent processes and the memory available. In general, multiple regression may be expressed by the following equation:

$$Y = \alpha + \beta 1X1 + \beta 2X2 + \cdots + \beta nXn + \varepsilon \quad (3)$$

where Y corresponds to the response variable of the model (execution time), X_1, X_2, …, X_n are independent variables (e.g., input size, number of concurrent processes, available memory), and ε is the error term. In cases with categorical variables (e.g., gender or presence/absence of a condition), dummy variables are often used.

The parameters α and β are commonly estimated using **the least squares approach**, which seeks to reduce the squared differences between the observed values Y_i and the predicted values Y'_i:

$$\sum(Yi - (\alpha + \beta 1X1i + \beta 2X2i + \cdots + \beta nXni))^2 \quad (4)$$

An important challenge in multiple regression analysis is multicollinearity, occurring when the explanatory variables show strong interdependence.

This situation complicates the identification of the individual influence of each predictor on the response variable and may produce increased standard errors as well as less reliable coefficient estimates.

Another common issue is *overfitting*, which arises when a regression model captures not only the true relationships but also the random noise in the data. Overfitted models tend to perform poorly on new datasets, limiting their practical usefulness for prediction.

2.3. Correlation Analysis

Correlation analysis is used to determine how closely two variables move together and to identify the direction of their association. For instance, the correlation between the size of an input dataset *(X)* and the execution time of an algorithm *(Y)* may be investigated. The coefficient r, known as Pearson's correlation, is commonly used to quantify the degree of a linear type of relationship linking two variables [12]. To compute Pearson's correlation coefficient, the formula below is used:

$$r = \frac{[\Sigma(X_i-\bar{X})(Y_i-\bar{Y})]}{\sqrt{[\Sigma(X_i-\bar{X})^2\Sigma(Y_i-\bar{Y})^2]}} \quad (5)$$

where X_i *and* Y_i represent the individual observations of variables *X* and *Y*, and $\bar{X}$ and $\bar{Y}$ are their respective means.

Values of $\boldsymbol{r}$ indicate the type of linear correlation: $\boldsymbol{r}$ = 1 corresponds to a perfect positive correlation, $\boldsymbol{r}$ = -1 to a perfect negative correlation, and $\boldsymbol{r}$ = 0 shows the absence of any linear correlation between the variables. If $\boldsymbol{r}$ = 0, there is no linear correlation between the variables. The correlation coefficient can indicate whether two variables increase or decrease together but does not imply causality.

Pearson's correlation coefficient is suitable for assessing linear relationships between quantitative variables with normal distributions. When dealing with ordinal data or when variables show a monotonic relationship instead of a linear one, nonparametric techniques like Spearman's correlation coefficient are recommended.

2.4. Comparison of Correlation and Regression

Correlation evaluates the extent of association between two variables while not specifying the direction of causality.

Both variables in correlation analysis are treated as stochastic and interchangeable. Regression analysis treats a single variable as the response (dependent) variable, with all other variables functioning as explanatory (independent) variables.

A key feature of regression is its ability to determine the "best-fit" line, which estimates the response variable *Y* based on the explanatory variable *X*, whereas correlation measures the degree of association between the two variables, regardless of which one is considered the cause (Figure 1).

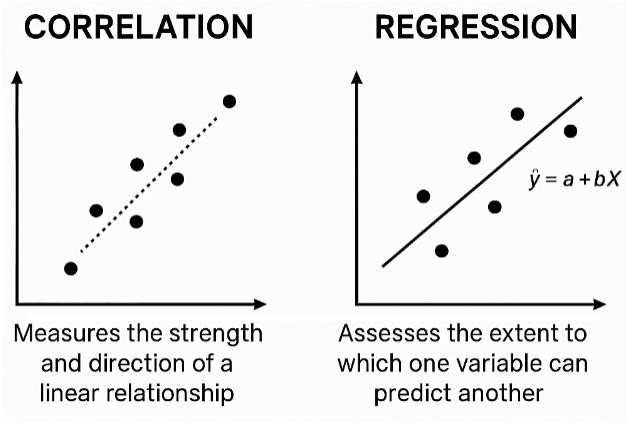

Figure 1. Differences between correlation and regression

3. Results

This section presents the results of the regression and correlation analyses performed on simulated datasets to evaluate the relationship between input size, system load, and algorithm execution time.

3.1. Linear Regression with a Single Predictor

Using GraphPad Prism 3.0, a simple linear regression analysis was performed to examine the effect of input data size on the execution time of the algorithm (hypothetical algorithm) in 20 simulated trials [7], [8], [13], [14], [15]. The data were generated for illustrative purposes.

Results from the analysis showed a positive relationship between the two variables that was statistically significant (Table 1).

Table 1. Input size and execution time for 20 simulated test cases

Test Case	Input Size (X)	Execution Time (Y)
1	30	47
2	32	50
3	35	55
4	37	57
5	38	59
6	40	62
7	41	63
8	43	66
9	44	68
10	46	70
11	48	72
12	50	75
13	52	78
14	54	81
15	56	83
16	58	86
17	60	88
18	62	91
19	64	93
20	66	96

Findings indicated a **significant positive** linear correlation between the two variables. The estimated regression equation was:

$$\boldsymbol{Execution\ Time}\ (\boldsymbol{Y}) = \mathbf{7.984} + \mathbf{1.348} * \boldsymbol{Input\ Size}\ (\boldsymbol{X})$$

The slope coefficient was **1.348 ± 0.0131**, with a 95% confidence interval extending from **1.321 to 1.376**, suggesting that for every one-unit increase in input size, the execution time increased on average by approximately 1.35 units. The y-intercept was **7.984 ± 0.642**, with a confidence interval of 95% from **6.635 to 9.333**.

The regression model achieved $\mathbf{R^2 = 0.9983}$, showing that **99.83%** of execution time variance was explained by the input dataset size. The standard error of the estimate (***Sy.x***) was **0.6241**, reflecting a low level of unexplained variation.

Statistical testing indicated that the regression model was significant, $\mathbf{F(1, 18) = 10{,}580}$, $\mathbf{p < 0.0001}$, indicating the slope of the regression line differed significantly from zero.

The graphical representation of the regression model, including data points and the fitted regression line, is shown in Figure 2.

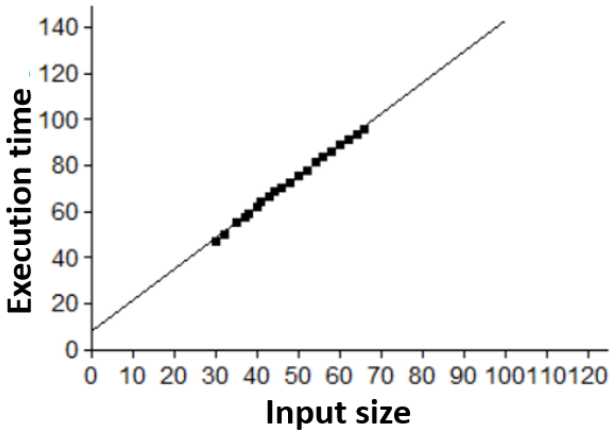

Figure 2. Linear regression plot of execution time

3.2. Regression with Several Independent Variables

A regression analysis with multiple predictors using Origin 9.0 software was performed to assess the combined effect of input size (X_1) and number of concurrent processes (X_2, i.e., tasks running simultaneously on the same system) on algorithm execution time (Y) across 20 simulated test cases. The data for the number of concurrent processes were also simulated for demonstration purposes (Table 2).

Table 2. Input size, number of concurrent processes, and execution time for 20 simulated test cases

Test Case	Input Size (X_1)	Number of Concurrent Processes (X_2)	Execution Time (Y)
1	30	5	47
2	32	5	50
3	35	6	55
4	37	5	57
5	38	5	59
6	40	6	62
7	41	6	63
8	43	6	66
9	44	6	68
10	46	6	70
11	48	6	72
12	50	6	75
13	52	6	78
14	54	7	81
15	56	7	83
16	58	7	86
17	60	7	88
18	62	8	91
19	64	8	93
20	66	8	96

The model yielded the following regression equation:

$$\textit{Execution Time (Y)} = -5.90 + 0.735 * \textit{Input Size } (X_1) + 0.126 \times \textit{Number of Concurrent Processes } (X_2)$$

Statistical testing showed that the model was significant overall, $\mathbf{F(2, 17)} = 6295.66$, $\mathbf{p} < 0.001$, showing that the model accurately predicts execution time. Approximately **99.85%** of execution time variation was accounted for by the model (the adjusted coefficient of determination $\mathbf{R^2} = 0.9985$), suggesting an excellent fit.

The coefficient for input size was statistically significant ($\boldsymbol{\beta} = 0.735$, $\mathbf{SE} = 0.0175$, $\boldsymbol{p} < 0.001$), indicating a strong and reliable contribution to the model.

However, the effect of the number of concurrent processes was not significant ($\boldsymbol{\beta} = 0.126$, $\mathbf{SE} = 0.262$, $\boldsymbol{p} > 0.05$), suggesting that in this dataset, the number of concurrent processes did not meaningfully predict execution time after accounting for input size. These results emphasize that while both variables are conceptually relevant, input size alone accounted for the majority of the explained variance in execution time within this model. Figure 3 shows the residual plots from the multiple regression analysis for the two independent variables. While the residuals for input size (Figure 3 (a)) appear randomly scattered around zero, the residuals for number of concurrent processes (Figure 3 (b)) exhibit greater variability, suggesting a weaker influence of this predictor in the model.

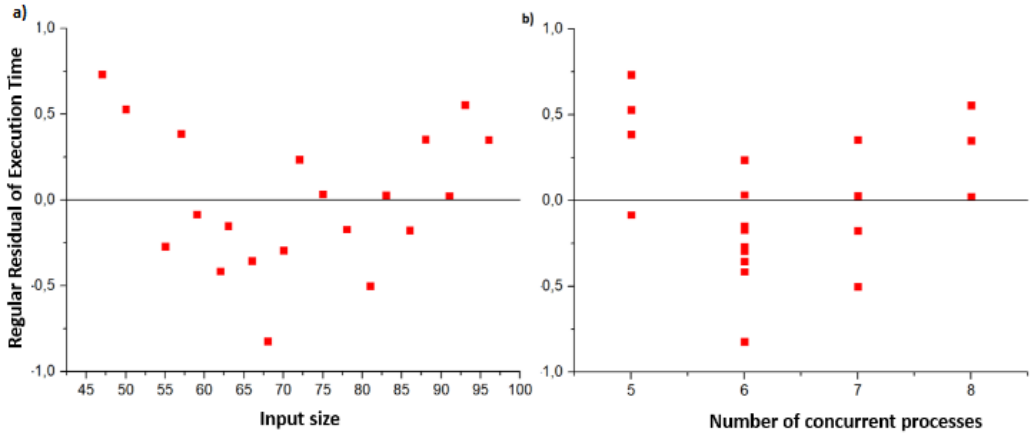

Figure 3. Residual plots from the multiple regression model: (a) residuals versus input size, (b) residuals versus number of concurrent processes

3.3. Results of Correlation Analysis

The relationships between input size (X_1), number of concurrent processes (X_2), and algorithm execution time (Y) across 20 simulated test cases were analyzed using Pearson correlation coefficients. Results indicated a very strong positive correlation between input size and execution time ($\mathbf{r} = \mathbf{0.999}$, $\mathbf{p} < 0.001$). Execution time increased in strong positive association with the number of concurrent processes ($r = 0.926$, $p < 0.001$).

Analysis indicates that both input size and number of concurrent processes are closely related to execution time (Figure 4). Although the correlation shows a strong relationship between number of concurrent processes and execution time, the multiple regression analysis indicates that when controlling for input size, the effect of number of concurrent processes is no longer significant. This suggests that input size is a more important predictor of execution time.

	Input Size (X1)	*Number of Concurrent Processes (X2)*	*Execution Time (Y)*
Input Size (X1)	1		
Number of Concurrent Processes (X2)	0,926486315	1	
Execution Time (Y)	0,999316447	0,925497723	1

Figure 4. Pearson correlation matrix between input size - number of concurrent processes and execution time

4. Discussion

This section interprets the results obtained from the regression and correlation analyses, discusses their implications, and highlights important methodological considerations such as multicollinearity, overfitting, and the distinction between correlation and causality.

Linear regression and multiple regression, combined with correlation analysis, are powerful tools for understanding relationships between variables. Linear regression provides a straightforward approach to predicting a dependent variable from independent variables, whereas multiple regression enables the analysis of several predictors simultaneously and their overall impact [2], [13], [16].

However, multiple regression models must be applied carefully, as they are susceptible to problems such as multicollinearity and overfitting [17], [18], [19]. Multicollinearity occurs when independent variables have a strong correlation, potentially distorting the estimated coefficients and masking the unique effect of each predictor. For example, when predicting algorithm execution time, including both input size and estimated number of operations as independent variables may introduce multicollinearity, as these variables are often strongly correlated [17].

Overfitting occurs when a model becomes overly complex, capturing random noise instead of the true underlying pattern, which limits its performance on new data. Such issues are particularly relevant in computer science applications, where datasets may be small and independent variables are often interrelated. In a model that predicts algorithm execution time based on multiple factors, overfitting may occur if a large number of predictor variables are used given the number of test cases. This may cause the model to capture random noise instead of the actual patterns, which lowers its predictive accuracy.

Previous studies have highlighted the impact of overfitting and the importance of applying appropriate techniques to improve model generalizability. Penalized regression methods, such as LASSO, have been shown to effectively reduce overfitting while maintaining model interpretability [19].

Correlation, on the other hand, does not imply causality but shows whether two variables change together. A high correlation should not be interpreted as evidence of a causal relationship [20], [21].

The results obtained in this study further demonstrate that input size is the dominant factor influencing algorithm execution time, as evidenced by both regression and correlation analyses.

In contrast, the number of concurrent processes showed limited statistical significance in the multiple regression model, suggesting that its effect is context-dependent and may be overshadowed by stronger predictors. This highlights the importance of carefully selecting explanatory variables in performance modeling tasks in computer science [6], [11].

These findings are particularly relevant for software performance analysis and system optimization, where identifying the most influential factors can support more efficient resource allocation and algorithm design.

In addition, regression and correlation techniques provide a foundation for understanding performance variability in computational systems. However, their effectiveness depends on proper model specification and data quality, which can significantly influence the reliability of results. This is particularly important in real-world applications, where noise and external factors may affect system behavior.

5. Conclusion

This section summarizes the main findings of the study and highlights the key conclusions regarding the use of regression and correlation analysis in understanding relationships between variables in computer science applications.

Linear regression, multiple regression, and correlation analysis are essential techniques for investigating relationships between variables in various fields, including computer science, software engineering, and data analysis. While regression can provide predictive insights, correlation helps identify the strength of associations without assuming causality. Proper application of these statistical methods can lead to valuable insights into variable dependencies and enhance prediction accuracy. Different statistical software packages (GraphPad Prism, Origin, and Excel) were utilized for various analyses in this study to leverage their specific strengths and ensure robustness of the results.

The findings of this study emphasize the importance of correctly interpreting statistical relationships in computer science applications. In particular, the distinction between correlation and regression is crucial, as a strong correlation does not necessarily indicate predictive capability or causal relationships. This has important implications for fields such as performance analysis, algorithm optimization, and system design, where incorrect interpretation may lead to misleading conclusions.

Despite the valuable insights provided, this study has certain limitations. The analyses were based on simulated datasets, which may not fully reflect the complexity of real-world computational environments.

Additionally, the relatively small sample size may limit the generalizability of the findings.

Future research may focus on applying these statistical methods to real-world datasets, incorporating additional variables that influence algorithm performance, and exploring more advanced modeling techniques, such as non-linear regression models, machine learning approaches, and robust regression techniques, to improve predictive accuracy and robustness.

Acknowledgements

This research was funded by the Bulgarian National Science Fund under the project "The Rise of Synthetic Reality: AI, Motion, and the Transformation of Human Perception", Contract No. KП-06-H95/13.

References:

[1]. Galton, F. (1886). Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, *15*, 246-263.

[2]. Mavrevski, R. (2014). Selection and comparison of regression models: estimation of torque-angle relationships. *Comptes Rendus De L Academie Bulgare Des Sciences*, *67*(10), 1345-1354.

[3]. Gujarati, D. N. (2004). *Basic Econometrics* (4th ed.). McGraw-Hill.

[4]. Madni, S. H. H., et al. (2017). Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment. *PloS one*, *12*(5), e0176321. Doi: 10.1371/journal.pone.0176321.

[5]. Ray, S., & De Sarkar, A. (2012). Execution analysis of load balancing algorithms in cloud computing environment. *International Journal on Cloud Computing: Services and Architecture (IJCCSA)*, *2*(5), 1-13.

[6]. Bessa, M. A., et al. (2017). A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality. *Computer Methods in Applied Mechanics and Engineering*, *320*, 633-667.

[7]. Ferezliev, A., Mavrevski. R., & Delkov A. (2018). Correlation between average and dominant height of middle-aged Douglas fir plantations in the north-west rhodopes. *Silva Balcanica*, 19(2), 13-26.

[8]. Serdar, C. C., et al. (2021). Sample size, power and effect size revisited: simplified and practical approaches in pre-clinical, clinical and laboratory studies. *Biochemia medica*, *31*(1), 27-53. Doi: 10.11613/BM.2021.010502.

[9]. Maszczyk, A., et al. (2014). Application of neural and regression models in sports results prediction. *Procedia-Social and Behavioral Sciences*, *117*, 482-487.

[10]. Wang, J.-L., Chiou, J.-M., & Müller, H.-G. (2015). *Review of functional data analysis*. arXiv.

[11]. Chicco, D., Warrens, M. J., & 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*, e623. Doi: 10.7717/peerj-cs.623

[12]. McLaren, S. J., et al. (2018). The relationships between internal and external measures of training load and intensity in team sports: a meta-analysis. *Sports medicine*, *48*(3), 641-658. Doi: 10.1007/s40279-017-0830-z.

[13]. Helmreich, J. E. (2016). Regression modeling strategies with applications to linear models, logistic and ordinal regression and survival analysis (2nd ed.). *Journal of Statistical Software*, *70*(2), 1–3. Doi: 10.18637/jss.v070.b02.

[14]. Mavrevski, R., et al. (2018). Approaches to modeling of biological experimental data with GraphPad Prism software. *WSEAS Transactions on Systems and Control*, *13*, 242–247.

[15]. Kuhn, M. (2008). Building predictive models in R using the caret package. *Journal of statistical software*, *28*, 1-26. Doi: 10.18637/JSS.V028.I05

[16]. Mekanik, F., et al. (2013). Multiple regression and Artificial Neural Network for long-term rainfall forecasting using large scale climate modes. *Journal of Hydrology*, *503*, 11-21. Doi: 10.1016/J.JHYDROL.2013.08.035

[17]. Kim, J. H. (2019). Multicollinearity and misleading statistical results. *Korean journal of anesthesiology*, *72*(6), 558-569.

[18]. Cai, J., et al. (2020). Prediction and analysis of net ecosystem carbon exchange based on gradient boosting regression and random forest. *Applied energy*, *262*, 114566. Doi: 10.1016/j.apenergy.2020.114566

[19]. Jonas, R., & Cook, J. (2018). LASSO regression. *British Journal of Surgery*, *105*(10). Doi: 10.1002/bjs.10895

[20]. Pearl, J. (2009). *Causal inference in statistics: An overview*. Statistics Surveys, 3, 96–146. Doi: 10.1214/09-SS057

[21]. Hernán, M. A., Wang, W., & Leaf, D. E. (2022). Target trial emulation: a framework for causal inference from observational data. *Jama*, *328*(24), 2446-2447. Doi: 10.1001/jama.2022.21383