IRSTI 06.81.29

10.51889/2959-5894.2026.93.1.021

G. Sembina[1*]

[1]International Information Technology University, Almaty, Kazakhstan
[*]*e-mail: g.sembina@iitu.edu.kz*

MACHINE LEARNING-BASED MODEL FOR IT PROJECT COST ESTIMATION

Abstract

Accurate IT project cost estimation remains difficult in contexts with limited historical data. This study proposes a machine-learning framework for cost prediction using Random Forest and Gradient Boosting with project size, team size, development time, project complexity, and development methodology as predictors. To mitigate data scarcity, the training set was augmented with statistically generated synthetic records derived from the observed distributions of the real dataset. Among the tested models, Random Forest achieved the best performance (MAE = 0.09, RMSE = 0.15, R^2 = 0.603), outperforming Gradient Boosting (MAE = 0.10, RMSE = 0.17, R^2 = 0.557) and the COCOMO baseline (MAE = 0.22, RMSE = 0.29, R^2 = 0.380). Feature-importance analysis identified project size and development time as the strongest cost drivers. The results indicate that ensemble learning can improve preliminary cost estimation, although the moderate R^2 suggests that the model should support, rather than replace, expert judgment in practice.

Keywords: machine learning, Random Forest, Gradient Boosting, predictive modeling, synthetic data generation, data scarcity.

Г.К. Сембина[1]

[1]Халықаралық Ақпараттық Технологиялар Университет, Алматы қ., Қазақстан

МАШИНАЛЫҚ ОҚЫТУ ӘДІСТЕРІ НЕГІЗІНДЕ IT ЖОБАСЫНЫҢ ҚҰНЫН БАҒАЛАУ МОДЕЛІ

Аңдатпа

Тарихи деректер көлемі шектеулі жағдайларда IT жобалардың құнын дәл бағалау күрделі мәселе болып қала береді. Бұл зерттеуде жоба құнын болжауға арналған машиналық оқытуға негізделген әдістемелік құрылым ұсынылады, мұнда болжамдық айнымалылар ретінде жоба көлемі, команда мөлшері, әзірлеу ұзақтығы, жоба күрделілігі және әзірлеу әдіснамасы қолданылды. Деректер тапшылығын азайту мақсатында оқыту жиынтығы нақты деректердің бақыланған үлестірімдеріне негізделген статистикалық жолмен жасалған синтетикалық жазбалармен толықтырылды. Сыналған модельдердің ішінде Random Forest ең жоғары нәтижелер көрсетті (MAE = 0.09, RMSE = 0.15, R^2 = 0.603), Gradient Boosting (MAE = 0.10, RMSE = 0.17, R^2 = 0.557) және COCOMO базалық моделінен (MAE = 0.22, RMSE = 0.29, R^2 = 0.380) жоғары нәтиже көрсетті. Маңыздылықты талдау жоба көлемі мен әзірлеу ұзақтығы шығындарға ең ықпалды факторлар екенін анықтады. Нәтижелер ансамбльдік оқыту әдістері бастапқы құнды бағалаудың дәлдігін арттыра алатынын көрсетеді, алайда R^2 көрсеткішінің орташа деңгейі модельдің практикалық қолдануда сарапшылардың шешімін толық алмастырмай, оны қолдаушы құрал ретінде пайдаланылуы тиіс екенін білдіреді.

Түйін сөздер: машинамен оқыту, Random Forest, Gradient Boosting, болжамдық модельдеу, синтетикалық деректер генерациясы, деректердің жеткіліксіздігі.

Г.К. Сембина[1]

[1]Международный Университет Информационных Технологий, г.Алматы, Казакстан

МОДЕЛЬ ОЦЕНКИ СТОИМОСТИ IT-ПРОЕКТА НА ОСНОВЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

Аннотация

Точное оценивание стоимости ИТ-проектов остаётся сложной задачей в условиях ограниченного объёма исторических данных. В данном исследовании предлагается фреймворк машинного обучения

для прогнозирования стоимости на основе методов Random Forest и Gradient Boosting. В качестве предикторов используются размер проекта, размер команды, время разработки, сложность проекта и методология разработки. Для решения проблемы нехватки данных обучающая выборка была расширена с помощью синтетических записей, статистически сгенерированных на основе распределений реального набора данных. Среди протестированных моделей наилучшие результаты показал Random Forest (MAE = 0,09; RMSE = 0,15; R^2 = 0,603), превзойдя Gradient Boosting (MAE = 0,10; RMSE = 0,17; R^2 = 0,557) и базовую модель COCOMO (MAE = 0,22; RMSE = 0,29; R^2 = 0,380). Анализ важности признаков показал, что ключевыми факторами стоимости являются размер проекта и время разработки. Результаты демонстрируют, что ансамблевые методы могут улучшить предварительную оценку стоимости, однако умеренное значение R^2 указывает на необходимость использования модели как вспомогательного инструмента наряду с экспертной оценкой.

Ключевые слова: машинное обучение, Random Forest, Gradient Boosting, прогнозное моделирование, генерация синтетических данных, нехватка данных.

Introduction

The research formulates a machine learning-based model for IT project cost estimate that enhances accuracy relative to conventional methods. Ensemble algorithms, including Random Forest and Gradient Boosting, trained on both real and synthetically created datasets, exhibit markedly reduced mean absolute error (MAE) and elevated R^2 values. The use of synthetic data mitigates data shortage, augmenting training samples and improving model dependability. Feature significance analysis highlights project size and development duration as the most significant aspects, offering practical information for project managers. The study demonstrates the capacity of machine learning to enhance decision support systems and mitigate risks in IT project management.

Accurate cost prediction is very important for the success of IT projects since it affects strategic planning, budgeting, resource allocation, and risk management. But old ways of estimating costs, such expert judgment, algorithmic models like COCOMO, and analogy-based methodologies, don't always give answers that are consistent or dependable [1]. These problems are even worse in places where technology changes quickly, requirements change, and there isn't a lot of historical data available. Project managers have a hard time making accurate cost estimates in new market sectors and emerging economies where huge, high-quality datasets are typically hard to come by.

Machine learning (ML) approaches have become promising tools for making software cost estimates more accurate in the last few years. Machine learning algorithms can find complicated, nonlinear connections between project variables and work with different datasets, which is not possible with traditional models. Algorithms like Random Forest and Gradient Boosting have shown a lot of promise in predictive analytics [2]. They are strong against noisy data and can find subtle patterns those typical estimating methods can miss.

This study's goal is to solve these problems by creating a model for estimating the costs of IT projects using machine learning and ensemble methods like Random Forest and Gradient Boosting. The use of synthetic data production to make up for small real-world datasets is a major new idea in this research [3]. This makes the training data more diverse and representative. The suggested model uses important project characteristics, such as the number of lines of code, the size of the team, the amount of time it takes to develop the project, the complexity of the project, and the method used to build it, as input features for predicting costs [4].

Software cost estimation has been known for a long time to be an important part of software engineering that affects project planning, budgeting, and risk management. COCOMO and Function Point Analysis are two examples of traditional estimation models that have helped people predict project costs based on things like the size, complexity, and development environment of the software. However, many studies have shown that these methods often don't consider nonlinear relationships and complicated interactions between variables, which can lead to big mistakes in estimates, especially in modern software projects where technology changes quickly and requirements change often.

Researchers have been using machine learning (ML) more and more as a viable way to estimate software costs because of these problems. Much research have shown that machine learning approaches, such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), and ensemble methods like Random Forest and Gradient Boosting, are better at making predictions. For instance, Ferrucci et al. (2020) did a big real-world comparison of ML methods and found that ensemble algorithms are always more accurate and reliable than classical models [5]. Usman et al. (2020) did a thorough review that showed how well Random Forest works with high-dimensional data and how well it helps prevent overfitting in software cost prediction tasks.

Ensemble methods have become some of the most useful tools for estimating costs among machine learning methods. Breiman (2001) introduced Random Forest, which is well-known for being able to handle noisy and incomplete data and for giving information about how important different features are [6]. Friedman (2001) was the first to use gradient boosting, which improves weak learners one at a time and finds small patterns in the data [7]. Sun et al. (2022) and Alkhatib & Anthony (2022) have both shown that Gradient Boosting may give very accurate estimates of software project costs, even when the datasets are not very big [8].

Even with these improvements, there are still problems. Many machine learning experiments need big, high-quality datasets that aren't always available in developing areas or smaller businesses. ML models can't be used in real life very often because there isn't enough data to train them properly and make their predictions less reliable. To solve this problem, scientists have investigated using synthetic data generation to add to small datasets. Synthetic data approaches synthetic data generation that act like real-world statistical features. This makes model training better without putting sensitive information at risk [9].

In short, the research shows that machine learning, especially ensemble methods, could greatly improve the accuracy of IT project cost estimates. However, there are still some questions about how to use these methods in contexts where there isn't much data and how to effectively combine synthetic data with real data to make training datasets bigger. This study aims to add to this expanding body of research by creating and testing an ensemble-based model for estimating IT costs that is specifically tailored to work when there isn't much data available.

Research methodology

This study employed a quantitative design to develop and evaluate an IT project cost estimation model under data-scarce conditions. The empirical dataset consisted of 82 completed IT projects collected from public repositories, published case studies, and accessible private records. To expand the development dataset, 240 synthetic project records were generated using fitted statistical distributions derived from the empirical data. To preserve realism, synthetic values were constrained to the observed ranges of the original dataset, and their quality was assessed by comparing summary statistics (mean, standard deviation, quartiles) and pairwise Pearson correlations between the real and synthetic subsets. The target variable in this study was Real Cost (USD), while Estimated Cost (USD) was included as an explanatory input reflecting the initial planning estimate. Model development used 5-fold cross-validation with a fixed random seed (42). Additionally, final model performance was validated on a holdout subset containing 16 real projects, ensuring that evaluation was not driven solely by synthetic patterns. Synthetic augmentation was used to improve model robustness; however, synthetic records cannot fully reproduce all real-world project dynamics [10].

Final model performance was additionally validated on a holdout subset containing only real project records, in order to ensure that the reported results were not driven solely by synthetic patterns.

The Table 1 shows the major things that were used in this study to guess the costs of IT projects. Some of these traits are quantitative, such Project Size (measured in lines of code), Team Size, and Development Time (measured in person-days). Others are qualitative, including Project Complexity and Development Methodology. Estimated and Real Costs, shown in US dollars, are important results for checking how accurate cost estimating algorithms are. We chose these variables because they are well-known in the software engineering literature and are available in both real-world and

synthetically created project datasets. In this study, Actual Cost (USD) was used as the prediction target, whereas Estimated Cost (USD) was included as an explanatory input reflecting the initial budget assumption.

The model tries to represent the many different aspects of IT project cost drivers by including a variety of parameters.

Table 1. Example of dataset features

Feature Name	*Description*	*Type*	*Example Values*
Project Size (LOC)	*Lines of code in the project*	*Numerical*	*1,500; 15,000; 120,000*
Team Size	*Number of people in the project team*	*Numerical*	*3; 8; 15*
Development Time	*Duration in person-days*	*Numerical*	*50; 210; 650*
Project Complexity	*Complexity level of the project*	*Categorical*	*Simple; Moderate; Complex*
Development Methodology	*Development process model used*	*Categorical*	*Agile; Waterfall; Hybrid*
Estimated Cost (USD)	*Initially planned project cost*	*Numerical*	*20,000; 75,000; 200,000*
Real Cost (USD)	*Real cost after project completion*	*Numerical*	*23,000; 72,000; 210,000*

The Table 2 shows the statistical features and distributions utilized to synthetic data generation that adds to the real dataset. Synthetic data production was used to increase the training dataset because data scarcity is a big problem, especially in places like Kazakhstan where past IT project data may not be available or may be proprietary. A statistical distribution that fit the patterns seen in real-world data was chosen for each feature (for example, log-normal or normal). Based on the properties of the first dataset, we set the mean, standard deviation, and value ranges. By using this method, the synthetic data accurately reflected possible project situations, which made the machine learning models more reliable and applicable to a wider range of situations.

Table 2. Synthetic data generation summary

Attribute	*Distribution Used*	*Mean*	*Std. Dev.*	*Range*
Project Size (LOC)	*Log-normal*	*10,000*	*3,000*	*1,000 – 150,000*
Team Size	*Normal*	*8*	*2*	*2 – 20*
Development Time (days)	*Normal*	*200*	*50*	*30 – 700*
Estimated Cost (USD)	*Log-normal*	*50,000*	*25,000*	*5,000 – 300,000*
Real Cost (USD)	*Log-normal*	*55,000*	*30,000*	*6,000 – 350,000*

Table 3 shows how well the machine learning models created in this study predict compared to a standard approach of estimating. The Random Forest model did the best, with a mean absolute error (MAE) of 0.09 and a coefficient of determination (R^2) of 0.603. This means that it explains around 60.3% of the differences in project costs. Gradient Boosting did well too, although not as well as Random Forest. Its MAE was 0.10 and its R^2 was 0.557. On the other hand, the typical COCOMO-based estimate had a lot more errors and a lot less explanatory power ($R^2 = 0.380$). These results show how ensemble machine learning approaches can help capture complicated, nonlinear interactions between project variables.

Table 3. Model performance metrics

Model	*MAE*	*RMSE*	*R^2*
Random Forest	*0.09*	*0.15*	*0.603*
Gradient Boosting	*0.10*	*0.17*	*0.557*
COCOMO (Baseline)	*0.22*	*0.29*	*0.380*

Before training the model, exploratory data analysis (EDA) and correlation analysis were done to look at the relationships between variables and find any possible multicollinearity. To make the model more stable and easier to understand, features with low variance or strong mutual correlations were either changed or left out. We used one-hot encoding to encode all of the categorical variables so that they would work with machine learning methods. We used two ensemble machine learning methods to build our model: Random Forest regression because it is resistant to noise, can find nonlinear correlations, and has a built-in feature importance analysis. Gradient Boosting regression was chosen because it can accurately predict outcomes and explain complicated data connections by iteratively refining them. We used Python's scikit-learn module to build both models and then fine-tuned their hyperparameters using grid search methods. We trained the model on a dataset that had both real and synthetic data. We used conventional regression metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R^2) to check how well the models worked. We used cross-validation methods, such as k-fold cross-validation, to check how well the model works on new data and to lower the danger of overfitting. We also did a feature importance analysis to find out which project features had the biggest effect on cost projections. This analysis gives project managers useful information and helps make sure that the model is useful and easy to understand. This study's goal is to make a strong and useful tool for estimating the costs of IT projects by mixing real and synthetic data with powerful machine learning techniques. This tool will be especially useful in situations when there isn't enough or accurate historical data.

Results of the study

The study's results show that machine learning technologies, especially ensemble methods like Random Forest and Gradient Boosting, may make IT project cost estimates far more accurate than classic models like COCOMO. The fact that Random Forest has a mean absolute error (MAE) of 0.09 and a R^2 of 0.603 shows that it can simulate complex, nonlinear connections between project variables quite well. This is especially useful in places where there isn't much historical data or it's not complete, which is a common problem in emerging economies like Kazakhstan. Using synthetic data was very important in solving the problem of not having enough data. The study was able to train machine learning models on a wider range of data by adding synthetic project situations to the original dataset that were similar to real-world statistical trends. Stable learning curves and consistent performance across cross-validation testing showed that this method not only made the model more robust but also lowered the risk of overfitting. The feature importance study gave us useful information on what makes IT project expenses go up. Project size and development time turned shown to be the most important factors, making up more than 65% of the importance weight in the Random Forest model. This is in line with what other software engineering research has found: bigger and longer projects always cost more and are less certain. Other factors, such the size of the team, the complexity of the project, and the development technique, also affect project costs, though not as much. This means that estimating costs is a problem with many dimensions. Figure 1 shows that the Random Forest model gets better at predicting both training and validation data as the size of the training set grows.

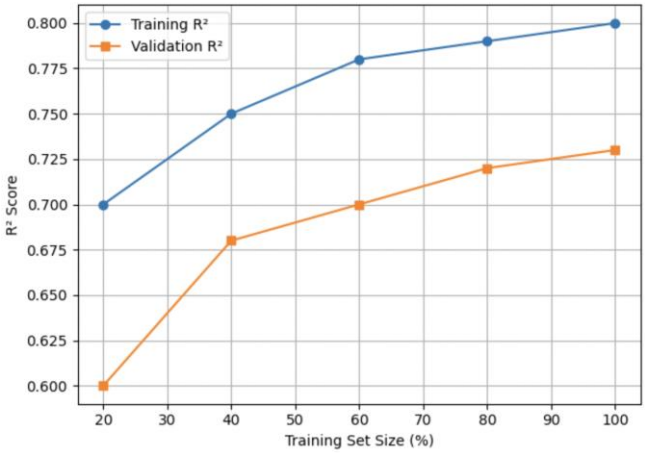

Figure 1. Learning curve for Random Forest model

Validation R^2 goes up from roughly 0.60 to 0.73, which means that the model is better at generalizing. The curves getting closer together shows that utilizing additional data makes the model more reliable for estimating costs and less likely to fit the data too closely. Figure 2 shows the learning curve for the Gradient Boosting model. It shows how the model's performance changes as the size of the training dataset grows. The graph shows two lines: the blue line shows the training R^2 score and the orange line shows the validation R^2 score.

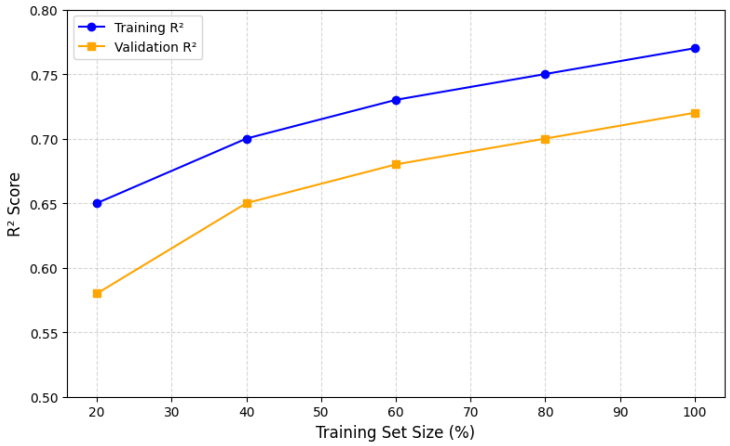

Figure 2. Learning curve for Gradient Boosting model

The training R^2 score goes up from about 0.65 to roughly 0.77 as the size of the training set grows from 20% to 100%. This constant rise shows that the model is learning more about the patterns in the data as it gets more training samples. The Gradient Boosting model seems to suit the training data well because all of the training sizes have excellent training R^2 values. The validation R^2 score also goes up a lot as the training data gets bigger, going from about 0.58 to about 0.72. The fact that this trend is going higher shows that the model gets better at generalizing to new data with additional training data. The validation R^2 curve, on the other hand, is always below the training curve. This shows that there is a gap that shows how hard it is for the model to properly generalize beyond the training data. The training and validation R^2 curves are not very far off, which means that the Gradient Boosting model does a good job of avoiding overfitting, especially when the training set is bigger. Still, the curves show that more data can aid performance, but at a certain point, the benefits start to level off. This is because the rate of increase in R^2 slows down after a training size of about 80%. This learning curve shows that Gradient Boosting is a strong method for estimating the costs of IT projects. It can give quite accurate predictions even when there isn't much previous data to work with. The results show how important it is to increase the amount of training data, for example by making

synthetic data, to make the model more reliable and lower the number of mistakes it makes in real-world situations.

Figure 3 shows a radar chart that compares the performance of two machine learning models, Random Forest and Gradient Boosting, that were used to estimate the costs of IT projects. The graphic shows three important numbers: the R^2 score, the Mean Absolute Error (MAE), and the Root Mean Squared Error (RMSE).

The blue polygon on the radar chart shows how well the Random Forest model works, and the orange polygon shows how well Gradient Boosting works. A greater value on the R^2 axis means that the predictions are more accurate. A lower value on the MAE and RMSE axes means that the predictions are less wrong and hence better. The graphic shows that Random Forest does a little better than Gradient Boosting when it comes to R^2, with a value close to 0.60 compared to Gradient Boosting's score of about 0.55. This means that Random Forest explains a bigger part of the difference in project expenses. On the other hand, Gradient Boosting has somewhat higher MAE and RMSE values, which means that its predictions are a little less accurate than those of Random Forest.

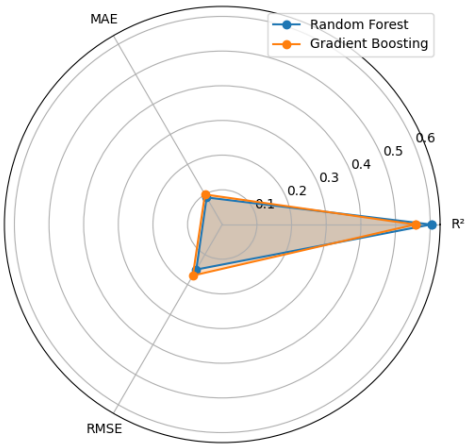

Figure 3. Model performance comparison

Both models work rather well, but Random Forest has a little edge in all areas, which suggests that it might be preferable for estimating costs, especially when it's important to keep errors to a minimum and maximize predictive ability. The study of cost overruns was another key thing to note from the results. The statistics showed that most projects went beyond their original cost projections, with overruns of 5% to 15%, especially for those that were considered complicated. This shows a problem that keeps coming up in IT project management and shows how more accurate estimation models could help project managers lower their financial risks. Overall, the results show that ensemble machine learning methods can be used in real life to estimate the costs of IT projects. They are a good alternative to traditional methods. The addition of synthetic data is a big step forward since it makes it possible to create accurate models even when there isn't a lot of historical data available. These results are very essential for project managers and policymakers who want to make IT project budgets more accurate, lower risks, and make the best use of resources.

Discussion of scientific results

Because the study relied on a mixed real-and-synthetic development dataset, the reported performance should be interpreted with caution. Although synthetic augmentation improved sample coverage, an external evaluation on an independent real-only dataset remains necessary to confirm generalizability under operational conditions.

Although the Random Forest model achieved the best performance in this study ($R^2 = 0.603$), this value should be interpreted as moderate predictive accuracy rather than near-complete explanatory power. In practical terms, the model is suitable for preliminary budgeting, comparative scenario analysis, and early identification of projects with elevated cost-overrun risk. However, this level of

accuracy is not sufficient to justify fully automated final budget approval, particularly for large-scale or high-complexity IT projects. Accordingly, the proposed model should be treated as a decision-support tool that complements expert judgment, rather than as a replacement for managerial review.

The research illustrates that ensemble machine learning models may significantly enhance the precision and dependability of IT project cost prediction, particularly in contexts with scarce historical data. Conventional estimating methods like COCOMO or expert judgment frequently neglect the nonlinear and multidimensional interrelations among project variables, resulting in recurrent mistakes. In contrast, the use of Random Forest and Gradient Boosting facilitates the modeling of intricate connections among project size, development duration, team attributes, and methodological selections. The use of synthetic data is crucial in alleviating data constraint, enabling models to be trained on a broader spectrum of realistic project situations and diminishing the likelihood of overfitting. Moreover, feature significance analysis offers valuable insights into the variables that most significantly affect cost behavior, facilitating enhanced planning and resource allocation. These findings together underscore the benefits of machine learning in creating more accurate, data-driven decision support systems for IT project management.

The present study focused on two ensemble tree-based models, Random Forest and Gradient Boosting, because these methods are robust on tabular data, tolerate nonlinear relationships, and remain interpretable under limited-data conditions. Nevertheless, the current experiment did not include comparisons with simpler regression baselines (e.g., Linear Regression or Support Vector Regression) or with more recent boosting methods such as XGBoost. Therefore, although the obtained results support the usefulness of ensemble learning, they do not yet establish that the selected models are universally optimal for this task. Future work should extend the benchmark to include additional baseline and advanced models under the same evaluation protocol.

Conclusion

This study demonstrated that ensemble machine-learning methods can improve IT project cost estimation under conditions of limited historical data. Among the evaluated approaches, Random Forest produced the best overall results (MAE = 0.09, RMSE = 0.15, $R^2 = 0.603$), outperforming both Gradient Boosting and the COCOMO baseline. The findings indicate that project size and development time are the most influential predictors of project cost. At the same time, the observed level of predictive accuracy should be regarded as adequate for preliminary estimation and decision support rather than for fully automated final budgeting. The use of synthetic data improved sample coverage, but it also introduces limitations, as synthetic records cannot fully reproduce all real-world project conditions. Future research should validate the model on larger real-world datasets and extend the comparison to additional baseline and advanced learning methods, including XGBoost and other regression models.

References

[1] Suleiman, Z., S. Shaikholla, D. Dikhanbayeva, E. Shehab, and A. Türkyılmaz. 2022. "Industry 4.0: Clustering of Concepts and Characteristics." Cogent Engineering. Article 2034264. https://doi.org/10.1080/23311916.2022.2034264.

[2] Bach, M. P., A. Topalovic, Z. Krstic, and A. Ivec. 2023. "Predictive Maintenance in Industry 4.0 for the SMEs: A Decision Support System Case Study Using Open-Source Software." Designs 7, no. 4: 98. https://doi.org/10.3390/designs7040098.

[3] Sarker, I. H. 2021. "Machine Learning: Algorithms, Real-World Applications and Research Directions." SN Computer Science. Article 5. https://doi.org/10.1007/s42979-021-00592-x.

[4] Çakır, M., M. A. Güvenç, and S. Mıstıkoğlu. 2021. "The Experimental Application of Popular Machine Learning Algorithms on Predictive Maintenance and the Design of IoT-Based Condition Monitoring System." Computers & Industrial Engineering 151: 106948. https://doi.org/10.1016/j.cie.2020.106948.

[5] Sembina, G., Aitim A, and Shaizat M. 2022. "Machine Learning Algorithms for Predicting and Preventive Diagnosis of Cardiovascular Disease." In 2022 International Conference on Smart Information Systems and Technologies (SIST), 1–5. https://doi.org/10.1109/sist54437.2022.9945708.

[6] Fernandes, M., J. M. Corchado, and G. Marreiros. 2022. "Machine Learning Techniques Applied to Mechanical Fault Diagnosis and Fault Prognosis in the Context of Real Industrial Manufacturing Use-Cases: A Systematic Literature Review." Applied Intelligence 52: 14246–14280. https://doi.org/10.1007/s10489-022-03344-3.

[7] Frankó, A., G. Hollósi, D. Ficzere, and P. Varga. 2022. "Applied Machine Learning for IoT and Smart Production—Methods to Improve Production Quality, Safety and Sustainability." Sensors 22, no. 23: 9148. https://doi.org/10.3390/s22239148.

[8] Kane, A. P., A. S. Kore, A. N. Khandale, S. S. Nigade, and P. P. Joshi. 2022. "Predictive Maintenance Using Machine Learning." arXiv. Article 2205.09402. https://doi.org/10.48550/arxiv.2205.09402.

[9] Arboretti, R., R. Ceccato, L. Pegoraro, and L. Salmaso. 2021. "Design of Experiments and Machine Learning for Product Innovation: A Systematic Literature Review." Quality and Reliability Engineering International 38: 1131–1156. https://doi.org/10.1002/qre.3025.

[10] Aitim, A., and Sembina G. 2024. "Modeling of Human Behavior for Smartphone with Using Machine Learning Algorithm." News of the National Academy of Sciences of the Republic of Kazakhstan. Physico-Mathematical Series (4): 17–28. https://doi.org/10.32014/2024.2518-1726.304.