IMPROVING THE RELIABILITY OF MACHINE LEARNING MODELS BY FILLING IN MISSING NAN VALUES IN MEDICAL DATASETS USING A GENETIC ALGORITHM

Sariyev Shokhrukh,
Samarkand State University
named after Sharof Rashidov,
assistant,
Samarkand, Uzbekistan.
sariyevshokhrukh@gmail.com

Abstract. This article proposes a genetic algorithm-based approach to optimize the filling of missing NaN values in a dataset. The focus is on selecting NaN values in the dataset directly corresponding to the results of the classification task. In the proposed method, each individual is represented as a chromosome in the form of a vector of all missing values. The search space is bounded by the given intervals for numerical attributes, and by the set of appropriate categories for categorical attributes. The accuracy indicator of the Random Forest ensemble model was used as the fitness function in the genetic algorithm.

Keywords. Genetic algorithm, ML,AI, Random Forest, KNN.

INTRODUCTION

Nowadays, artificial intelligence models are being applied to all fields. As a result, the reliability of the dataset is important for the results of machine learning models to be stable. When preparing a dataset, it is important to fill in missing values (NaN) that are considered relevant. There are classic methods for filling in missing values. However, the application of these methods depends on the characteristics of the data set. Classical methods can be used for data sets related to regression problems, but classical methods for discrete and categorical data sets in data sets related to classification problems lead to poor results for machine learning models. Therefore, a method for filling in missing NaN values in a dataset using a genetic algorithm was proposed for multi-class classification problems using genetic evolutionary approach algorithms. To implement this approach, the diabetes dataset column was used from a current topic in the medical field. Diabetes mellitus is one of the most common chronic diseases in modern medicine, and its early detection and identification of individuals at risk are important in increasing the efficiency of the healthcare system. Factors that influence the development of diabetes are multifaceted and include clinical and functional indicators such as high blood pressure, general health, frequent urination, gender, age, slow wound healing, sedentary lifestyle, family history of diabetes, excessive thirst, sweating, body mass index, decreased vision, and difficulty walking or climbing stairs. Machine learning models built on these features allow for early diagnosis of diabetes, segmentation of risk groups, and planning of individual preventive measures. In practical settings, missing NaN values are observed in data sets collected from physicians, laboratories, and electronic health information systems. Empty cells appear in the clinical chart as a result of certain indicators not being measured, filling out the wrong questionnaire, or not being entered into the database for technical reasons. If such records are completely excluded, the number of samples will be drastically reduced, reducing the training set for machine learning models and reducing generalization ability. Therefore, the issue of filling in the gaps in modern literature is considered a separate methodological step. Multiple imputation methods, such as Multiple Imputation by Chained Equations, and approaches based on chained regression models are widely used in the scientific literature as they allow for flexible imputation of different types of variables[1-2]. However, while nearest neighbor-based methods such as KNN-imputation have also been shown to be effective in clinical datasets, they often preserve local structure but fail to adequately represent the deeper

relationships that make up the global terrain of the data and are sensitive to the choice of the appropriate k parameter. Classical statistical and regression-based imputation schemes also run the risk of distorting the covariance structure, artificially reducing variance, or introducing bias; this can negatively impact the accuracy and fairness of the resulting model, particularly in clinical classification problems such as diabetes. In recent years, a number of developments have been proposed to fill in missing data using approaches based on evolutionary computing methods, in particular genetic algorithmsJuan Carlos and co-authors have shown that it is effective to impute blank responses in a questionnaire using a genetic algorithm with different fitness functions[3-4]. Lobato et al. developed an evolutionary imputation method for pattern classification that is adapted to mixed-type numerical and categorical attributes and reported superior results compared to classical imputation methods by directly incorporating classification accuracy into the fitness function [5]. The main advantage of genetic algorithms is that they perform a global search in a high-dimensional and zero-gap solution space and reduce the risk of getting "stuck" in local minima. In the context of the imputation task, each individual can be viewed as a vector of imputed values for all NaN cells in the dataset; the fitness function is the cross-validation accuracy, macro-F1 score, or AUC values of the diabetes classification model. The process of filling in missing values in the study is to determine the exact maximum for defining the diabetes classes into healthy, prediabetic, and diabetic, rather than a final setup - a simple statistical evaluation. In addition, based on a genetic algorithm, it is possible to preserve the data distribution by introducing penalty terms that control whether the statistical indicators such as mean, variance, and covariance of the imputed values are close to the initial data structure[6-7].

MAIN PART

In this study, missing NaN values in a clinical dataset of diabetes diagnosis were used to improve the results of machine learning models under existing conditions. Nowadays, artificial intelligence models are entering all sectors, which leads to increased development and efficiency of employees' working hours. In this study, one of the current topical issues in the medical field is to prepare a data set for diagnosing diabetes. In this research, as with data in all fields, the medical dataset also has some missing values. This research paper proposes modern methods such as genetic algorithms, due to the fact that traditional methods of filling in missing values lead to errors in filling in nonlinear values and categorical values. The genetic algorithm is a powerful evolutionary model that encompasses the entire domain of interest [8-9]. For this reason, it is the most effective method for filling in missing values. The disadvantages of the mean, median, mode, KNN, and other methods over classical methods are that they lead to standard deviations when filling in discrete and categorical values [10-11]. As a result, the results of machine learning models deteriorate. The genetic algorithm method proposed in this study is a highly efficient solution because it simultaneously optimizes the results of artificial intelligence models by maximizing accuracy or F1-score or minimizing the loss function [12-14].

Table 1. Characteristics of the diabetes data set

High_BP – High blood pressure (hypertension)	Sedentary – Sedentary / low physical activity lifestyle
Family_Diabetes – Family history of diabetes mellitus	Gen_health_1_5– Self-reported general health status (1 to 5 scale)
Freq_urination – Frequency of urination (polyuria)	Excess_thirst – Excessive thirst (polydipsia)
Gender – Patient gender (male / female)	Numbness – Numbness/tingling in hands and feet
Age – Age (years)	Vision_loss – Decreased vision / vision impairment
Slow_healing – Slow healing of wounds and cuts	BMI – Body Mass Index (kg/m²)
Difficulty_walking_stairs – Difficulty walking or climbing stairs	

In Table 1 above, the output target column, which lists the characteristics of diabetes, is a multi-

class classification, corresponding to the values 0: healthy, 1: prediabetes, and 2: diabetes. In the dataset used in this study, missing values are listed in Figure 1 by column value, and there are no NaN values in the remaining columns.

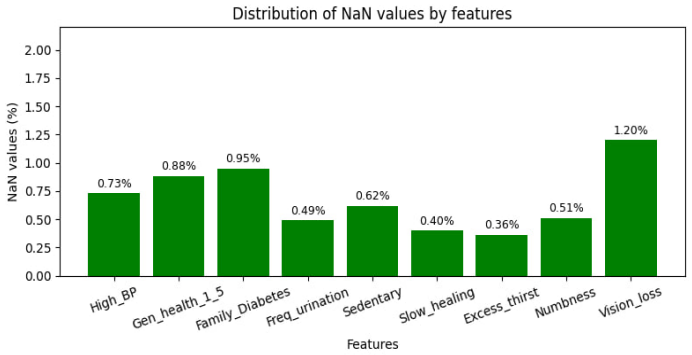

Figure 1. Missing values in the diabetes dataset.

In this Figure 1, a genetic algorithm model was developed to eliminate NaN values in missing columns. In the developed model, missing values were eliminated by maximizing the accuracy value of the Random Forest ensemble model as the objective function, i.e., the fitness function. The data set used in this study consists of 12024 real-world data sets obtained from the Samarkand Regional Endocrinology Hospital.

GENETIC ALGORITHM MATHEMATICAL MODEL

The initial data set is implemented through the input feature matrix in the following formula (1).

$$X=\left(x_{ij}\right)\in R^{m\times t},\ y=\left(y_1,...y_m\right) \tag{1}$$

Here $m-$ number of rows, $t-$ number of columns, $x_{ij}-$ diabetes feature vector and y_i class result.

Missing values in the initial data set are determined by (2) below.

$$U=\left(u_{ij}\right)\in\{0,1\}^{m\times t},\ u_{ij}=\begin{cases}1,\ x_{ij}\ \text{if known}\\ 0,\ x_{ij}\ \text{NaN}\end{cases} \tag{2}$$

In this dataset, columns containing NaN values are extracted using the following (3).

$$\Omega=\left\{(i,j):u_{ij}=1\right\},\ \Omega^s=\left\{(i,j):u_{ij}=0\right\} \tag{3}$$

Here $\Omega-$ NaN non-existent elements and Ω^s- NaN is the set of elements that exist. From here in the dataset $u=\left|\Omega^s\right|$ represents the total number of missing elements. Ω^s is defined by (4) below, introducing a correspondence connecting the indices in the set through a single-dimensional index.

$$\varphi:\{1,2,3...,u\}\to\Omega^s,\ \varphi(k)=\left(i_k,j_k\right) \tag{4}$$

Missing values are initially considered as a vector of parameters determined by a genetic algorithm (4).

$$o=\left(o_1,o_2,...,o_u\right) \tag{4}$$

The general scope for filling in NaN values in the research dataset is defined by (5) below.

$$C_o=T_1\times T_2\times\cdots\times T_u \tag{5}$$

Here, each o_k in the matrix $\varphi(k)=\left(i_k,j_k\right)$ The value corresponding to and $x_{i_kj_k}$ Specifies the appropriate value to replace NaN for the element. The data matrix filled with data without NaN values is implemented using the following formula (6).

$$\bar{X}(o)=\left(\bar{x}_{ij}(o)\right)\in R^{m\times t}$$

$$\bar{x}_{ij}(o)=\begin{cases}x_{ij},\ (i,j)\in\Omega\\ o_k,\ (i,j)\in\Omega^s\end{cases} \tag{6}$$

Here $\bar{X}(o)-$ NaN is a data matrix filled with no values. Each missing column in the dataset has its own domain, depending on the type of character that corresponds to the value in that column. $\varphi(k)=(i,j)$ if j_k if continuous, then $o_k\in T_k,\ T_k=[a_{j_k},b_{j_k}]$ is carried out through. In this a_{j_k} and b_{j_k} represents the maximum and minimum values given in advance. If j_k if there is a category then $o_k\in T_k,\ T_k=\left\{\vartheta_{j_k,1},\vartheta_{j_k,2},...,\vartheta_{j_k,K_{jk}}\right\}$ here $\vartheta_{j_k,l}$, j_k is done through a set of categories for. To prevent the dataset from going over the limit when filling in NaN

values, each gene o_k is stored in the following field (7).

$$\psi_{D_k}(l)=\begin{cases}\min\{\max\{l,a_{jk}\},b_{jk}\}, & T_k=[a_{jk},b_{jk}],\\ \arg\min\limits_{s\in T_k}\operatorname{dis}(l,s), & T_k \text{ categorical}\end{cases} \quad (7)$$

The initial prepared population $\bar{X}(o)$ diabetes dataset is trained using a Random Forest model to provide accuracy to the fitness functionTo evaluate the Random Forest model and obtain the classification result, the fitness function of the genetic algorithm is calculated using formula (8).

$$\begin{cases} w(o)=\arg\min\limits_{w} U(w;\bar{X}(o),y)\\ \hat{y}_i(o)=f_{w(o)}(\bar{x}_i(o))\end{cases} \quad (8)$$

Here represents the loss function. $U-$ determined by minimization. The model takes each observation vector $\bar{x}_i(o)$ as input and $\hat{y}_i(o)$, $i-$ returns the class prediction for the sample. The accuracy evaluation function of the Random Forest model as a fitness function when filling in the missing NaN values of the dataset with a genetic algorithm is calculated by formula (9).

$$F(o)=\frac{1}{K}\sum_{k=1}^{K}Accuracy_k(o) \quad (9)$$

The population of the genetic algorithm is implemented by the following formula (10).

$$P(d)=\left\{o^{(1)}(d),o^{(2)}(d),...,o^{(N)}(d)\right\}\subset C_o \quad (10)$$

Here $d-$ In the iteration, the population of the genetic algorithm is implemented with $P(d)$. $N-$ population size, $o^{(i)}(d)$, $i-$ the chromosome of the individual (vector of values for NaN cells) and each $o^{(i)}(d)$ search spacea C_o belongs to, and for continuous values $\left[a_j;b_j\right]$, for categorical columns $\left\{g_j\right\}$ taken from the collection. $P(d)-$ The population set of solutions to be evaluated in an iteration. The roulette method is implemented by (11) to determine the probability of each solution being selected as a father and mother, respectively, based on its fitness function.

$$p(o^{(i)}(d))=\frac{F(o^{(i)}(d))}{\sum_{j=1}^{M}F(o^{(j)}(d))} \quad (11)$$

Here $F(o^{(i)}(d)),$ $i-$ the accuracy of the individual and $p(o^{(i)}(d))\in(0;1)$.

In the next step, a two-point crossover operator was used to maintain genetic diversity. Pair for combination purposes $(p_1;p_2)$ probably from p_s Crossing is performed with. Crossing is performed in the sequence of execution by the following formula (12).

$$\begin{cases}(S_1,S_2)=crossover(p_1,p_2),\\ S_1=(p_1[1:k_1];p_2[k_1+1,k_2],p_1[k_2+1;u]),\\ S_2=(p_2[1:k_1],p_1[k_1+1:k_2],p_2[k_2+1:u])\end{cases} \quad (12)$$

Here k_1 and k_2 intersection points, S_1 and S_2 chromosomes of each resulting generation. A set of children resulting from crossing and mutation $S(d)$ combined with populations, $R(d)=P(d)\cup S(d)$ an expanded set of candidate solutions is generated. After that, the best according to the principle of elitism E individuals are kept unchanged. As a result $M-E$ The positions will be filled based on the ranking. This strategy prevents the disappearance of the best solutions in one direction[15-16]. The remaining spaces on the second side allow you to try new combinations. This yield is determined by the following formula (13) for populations.

$$P(d+1)=\underbrace{TopE(R(d),E;F)}_{\text{elitlar}}\cup\underbrace{Top(R(d)\setminus Elite,M-E;F)}_{\text{qolganlar}} \quad (13)$$

Here

- $d\in M$, genetic algorithm iteration index;
- $P(d)=\{o^1(d),o^2(d),...o^M(d)\}$, current population size M, each $o^i(d)-$ chromosome;

- $S(d)-$ a newly formed set of children: Solutions resulting from selection, crossing over, and mutation;
- $R(d)-$ expanded candidate pool;
- $F-$ The fitness function measures how good a solution is;
- $E-$ number of elites (whole number: $1 \le E < M$) an indicator of how many of the best individuals are preserved without modification;
- $TopE(A,E;F)-$ set A highest in fitness out of E operator that takes elements;
- $Elite = TopE(R(d),E;F),$ elite group;

In this study, the best candidate solutions were extracted by using the elitism operator. In the operator of elitism E The individual is kept unchanged and automatically transferred to the next population. If there is monotonicity in elitism, the following fitness function (14) does not deteriorate[17-18].

$$\max_{o \in P(d+1)} F(o) \ge \max_{o \in P(d)} F(o) \quad (14)$$

All the rest $M-E$ places $R(d) \setminus \text{Elite}$ fills in the fitness rating from the set. *Stopping condition criterion.* To control convergence in a genetic algorithm, the algorithm stops when the condition in formula (15) below is met.

$$F_{\max}(d) = \max_{o \in P(d)} F(o) \quad (15)$$

Here $D-$ upper limit on the number of iterations.

Final solution. After the genetic algorithm stops, the solution population with the highest fitness is selected as the solution (16) [19-20].

$$o^* = \arg\max_{o \in P(d^*)} F(o), \quad \bar{X}_{fin} = \bar{X}(o^*) \quad (16)$$

Here $\bar{X}_{fin}-$ The final result is a dataset filled with no NaN values.

EXPERIMENTAL RESULTS

The results obtained from the Random Forest model on the experimental results on the diabetes dataset conducted in this study are presented in Table 1 below.

TABLE 1. ACCURACY, PRECISION, RECALL, AND F1-SCORE RESULTS FOR ENSEMBLE METHOD

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.9034	0.9042	0.9038	0.9040

Figure 1 shows the multi-class ROC curves for the Random Forest model after imputation with a genetic algorithm for filling in missing NaN values in the diabetes dataset. The AUC values for the healthy, prediabetic, and diabetic classes indicate the high discrimination of the model.

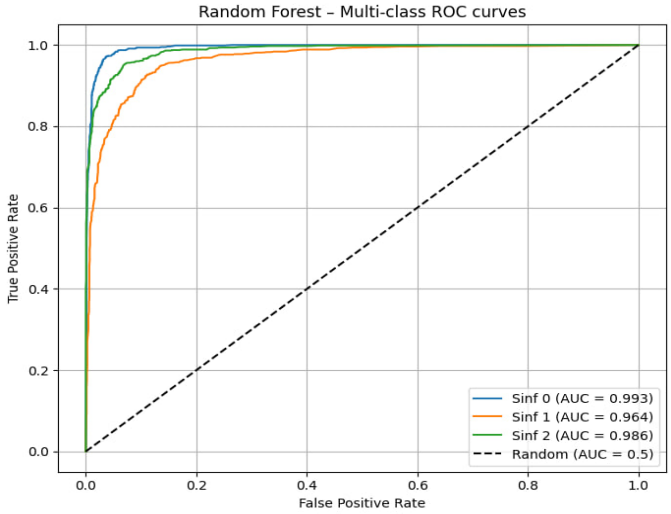

Figure 1 Multi-class ROC curves in the Random Forest model.

Conclusion

In this study, a new approach based on genetic algorithms was proposed to fill in missing NaN values in a dataset for multi-class classification of diabetes diagnosis. 0 – healthy, 1 – prediabetes, 2 – was applied to a real clinical dataset consisting of diabetes classes. This dataset contains 12024 records, Samarkand Regional Endocrinology Hospital, and features such as high blood pressure, frequent urination, thirst, general health, decreased vision, and difficulty walking, preventing NaN values from significantly degrading machine learning results[21-22]. The analysis of the distortion of the covariance structure and the

introduction of bias into the final model in the conditions of mixed-type attributes and nonlinear dependencies of the classical methods based on mean, mode, median, and regression was carried out. In the proposed genetic algorithm, the vector of imputed values for NaN columns and rows was taken as a chromosome and used as the accuracy fitness function of the Random Forest ensemble model. The results showed that the Random Forest model trained with GA-imputation showed higher accuracy and F1-score compared to classical methods, especially improved the correct identification of prediabetes and diabetes classes, justifying its use for risk group segmentation and optimization of early diagnosis.

Reference

1. Van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in R. Journal of Statistical Software, 45(3), 1–67. https://doi.org/10.18637/jss.v045.i03

2. N. Fayzullo, S. Sariyev and Y. Sherzodjon, "Analyzing the Effectiveness of Ensemble Methods in Solving Multi-Class Classification Problems," 2025 International Russian Smart Industry Conference (SmartIndustryCon), Sochi, Russian Federation, 2025, pp. 788-793, doi: 10.1109/SmartIndustryCon65166.2025.10986248

3. Juan Carlos Figueroa-García, Roman Neruda, German Hernandez–Pérez, A genetic algorithm for multivariate missing data imputation, Information Sciences, Volume 619, 2023, Pages 947-967, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2022.11.037.

4. K. Laxmikant, R. Bhuvaneswari and B. Natarajan, "An Efficient Approach to Detect Diabetes using XGBoost Classifier," 2023 Winter Summit on Smart Computing and Networks (WiSSCoN), Chennai, India, 2023, pp. 1-8, doi: 10.1109/WiSSCoN56857.2023.10133854.

5. Lobato, Fábio & Araújo, Igor & Tadaiesky, Vincent & Santana, Ádamo. (2015). An Evolutionary Missing Data Imputation Method for Pattern Classification. 10.1145/2739482.2768451.

6. S. Sariyev, I. Yalgoshev and M. Nuriddinova, "Traditional Methods and Modern Approaches Based on Ensemble Algorithms for Decision-Making in Diagnostics Using Medical Data," 2025 International Russian Automation Conference (RusAutoCon), Sochi, Russian Federation, 2025, pp. 971-975, doi: 10.1109/RusAutoCon65989.2025.11177423.

7. S. Sariyev, G. Negmatova and A. Sayidkulov, "Preparation Datasets Based on Artificial Intelligence Models and Classification Algorithms," 2025 International Russian Automation Conference (RusAutoCon), Sochi, Russian Federation, 2025, pp. 260-266, https://doi.org/10.1109/RusAutoCon65989.2025.11177339

8. Qiang Long, Changzhi Wu, Tingwen Huang, Xiangyu Wang,A genetic algorithm for unconstrained multi-objective optimization,Swarm and Evolutionary Computation,Volume 22,2015,Pages 1-14,ISSN 2210-6502,https://doi.org/10.1016/j.swevo.2015.01.002.

9. V. Venugopal, T.T. Narendran,A genetic algorithm approach to the machine-component grouping problem with multiple objectives,Computers & ndustrial Engineering,Volume 22, Issue 4,1992,Pages 469-480,ISSN 0360-8352,https://doi.org/10.1016/0360-8352(92)90022-C.

10. Yang Yuan, Jianqiang Du, Yanchen Zhu, Jigen Luo, Qiang Huang,Research on the application of dynamic weighted KNN with preprocessing based on a normal distribution in metabolomics data imputation,Computational Biology and Chemistry,Volume 121,2026,108804,ISSN 1476-9271, https://doi.org/10.1016/j.compbiolchem.2025.108804.

11. Rubul Kumar Bania, Anindya Halder,R-Ensembler: A greedy rough set based ensemble attribute selection algorithm with kNN imputation for classification of medical data,Computer Methods and Programs in Biomedicine,Volume 184,2020,105122,ISSN 0169-2607, https://doi.org/10.1016/j.cmpb.2019.105122.

12. R. Hephzibah, A. H. Christinal, R. Jayanthi, D. A. Chandy and C. Bajaj, "A Novel Ensemble Classifier Framework for Accurate Fetal Heart Rate Classification," 2023 4th International Conference on Signal Processing and Communication (ICSPC), Coimbatore, India, 2023, pp. 321-324, doi: 10.1109/ICSPC57692.2023.10125713.

13. S. Akhatkulov, I. Yalgoshev and J. Haydarov, "Different Warmup and Annealing Strategies for ANN Models to Predict Air Quality Index," 2025 International Russian Automation Conference (RusAutoCon), Sochi, Russian Federation, 2025, pp. 491-496, doi: 10.1109/RusAutoCon65989.2025.11177428.

14. N. F. Makhmadiyarovich and Y. Sherzodjon, “Methods of increasing data reliability based on distributed and parallel technologies based on blockchain,” Artificial Intelligence, Blockchain, Computing and Security Volume 2. eBook ISBN: 9781032684994, pages 637 – 642, January 2023.

15. A. Axatov, M. Nurmamatov, F. Nazarov, and Sh. Sariyev, “Genetic algorithm application technology in multi-parameter optimization problems,” AIP Conf. Proc., vol. 3244, art. no. 030025, 2024, doi: 10.1063/5.0242074

16. D. E. Goldberg and J. H. Holland, “Genetic algorithms and machine learning,” Machine Learning, vol. 3, pp. 95–99, 1988, doi: 10.1023/A:1022602019183.

17. S. E. Haupt, “Introduction to genetic algorithms,” in Artificial Intelligence Methods in the Environmental Sciences, S. E. Haupt, A. Pasini, and C. Marzban, Eds. Dordrecht, The Netherlands: Springer, 2009, pp. 103–125, doi: 10.1007/978-1-4020-9119-3_5.

18. S. Katoch, S. S. Chauhan, and V. Kumar, “A review on genetic algorithm: past, present, and future,” Multimedia Tools and Applications, vol. 80, no. 5, pp. 8091–8126, 2021, doi: 10.1007/s11042-020-10139-6.

19. T. Alam, S. Qamar, A. Dixit, and M. Benaida, “Genetic algorithm: reviews, implementations and applications,” International Journal of Engineering Pedagogy (iJEP), vol. 10, no. 6, pp. 57–77, 2020, doi: 10.3991/ijep.v10i6.14567

20. A. Hassanat et al., “Choosing mutation and crossover ratios for genetic algorithms—A review with a new dynamic approach,” Information, vol. 10, no. 12, art. 390, 2019, doi: 10.3390/info10120390

21. Sokhobiddin Akhatkulov, Islom Yalgoshev; Air quality prediction based on machine learning techniques. AIP Conf. Proc. 15 September 2025; 3356 (1): 030005. https://doi.org/10.1063/5.0296121

22. S. Akhatkulov, I. Yalgoshev and Z. Urinboyev, "Vehicle CO2 Emission Prediction Using Deep Learning and Ensemble Machine Learning Methods," 2025 International Russian Automation Conference (RusAutoCon), Sochi, Russian Federation, 2025, pp. 819-824, doi: 10.1109/RusAutoCon65989.2025.11177377.