Journal of Autonomous Intelligence (2024) Volume 7 Issue 4
doi: 10.32629/jai.v7i4.1206

ORIGINAL RESEARCH ARTICLE

AdaBoost_wear: Adaboost model-based Python software for predicting the coefficient of friction of babbitt alloy

Mihail Kolev*

Bulgarian Academy of Sciences, Institute of Metal Science, Equipment and Technologies with Center for Hydro- and Aerodynamics "Acad. A. Balevski", Boulevard "Shipchenski Prohod" 67, 1574 Sofia, Bulgaria; mihail1kolev@ims.bas.bg

ABSTRACT

AdaBoost_wear is a Python software that implements the AdaBoost algorithm to predict the coefficient of friction (COF) of B83 babbitt alloy as a function of time. The software uses data from pin-on-disk tests with different loads to train and test the model. The software also provides performance metrics, such as R^2 score, mean squared error, and mean absolute error, to evaluate the accuracy of the predictions. The software also generates plots of the actual and predicted COF values, as well as histograms and boxplots of the COF distribution. The software is open source and released under the MIT license.

Keywords: machine learning; adaboost; coefficient of friction; babbitt alloy; python

ARTICLE INFO

Received: 1 September 2023
Accepted: 4 December 2023
Available online: 1 March 2024

1. Introduction

Babbitt alloys are a group of tin- or lead-based alloys that are widely used as bearing materials for various industrial applications, such as compressors, turbines, electric motors, gear drives, pumps, cement, steel, chemical, paper, petroleum, and coal mine equipment. They were invented by Isaac Babbitt in 1839 and have since been developed and improved to suit different operating conditions and requirements.

The performance and reliability of babbitt bearings depend on several factors, such as the alloy composition, the manufacturing process, the operating conditions, and the lubrication system. Among these factors, the coefficient of friction (COF) is an important parameter that affects the frictional losses, wear rate, and temperature rise of the bearings[1–3]. Therefore, it is desirable to predict the COF of babbitt alloys under different conditions and to optimize their design and selection. However, the COF of babbitt alloys is influenced by many complex and nonlinear factors, such as the load, speed, temperature, surface roughness, lubricant viscosity, and additives. Therefore, it is challenging to establish a simple and accurate mathematical model to describe the COF behavior of babbitt alloys.

Machine learning (ML) is a powerful tool that can accelerate materials discovery, design, characterization, and optimization by exploiting the vast amount of data generated from experiments, simulations, and literature. It has been effectively employed in various fields of science and engineering, where it can solve complex problems and generate novel solutions. It can also be used to model complex nonlinear relationships between input and output variables, such as the COF of babbitt alloy bearings and the experimental conditions[4–6].

Several ML models have been proposed to predict the COF of babbitt alloy bearings, such as artificial neural networks (ANNs)[2], support vector machines (SVMs)[7], random forests (RFs)[7,8], and k-nearest neighbors (KNNs)[7]. However, these models may suffer from overfitting, underfitting, or high computational complexity. AdaBoost is a ML technique that uses multiple simple learners (such as decision trees) to create a complex learner that can achieve high accuracy and generalization[9]. AdaBoost is an ensemble learning method that iteratively assigns weights to the training samples according to their prediction errors and updates the weights of the weak learners according to their performance. AdaBoost can handle noisy data, missing values, and outliers, and can also reduce the variance and bias of the model. The authors Yadav et al.[10] apply AdaBoost to accurately predict micro-hardness in a welding process based on process parameters. They also identify tool rotational speed as the most important factor affecting micro-hardness. According to Rajput et al.[11] AdaBoost was used to predict the ultimate tensile strength of aluminum metal matrix hybrid composites based on eleven input parameters. Adaboost was also used to predict the compressive strength properties of high calcium fly-ash-based geopolymer concrete and it showed higher accuracy, when compared with ANN model[12]. However, to the best of our knowledge, there is no existing software that implements the AdaBoost algorithm for the COF modeling of B83 babbitt alloys, which is a powerful and robust ML technique that can achieve high accuracy and generalization with low computational resources.

This paper presents AdaBoost_wear, a Python software that implements the AdaBoost algorithm to predict the COF of a B83 babbitt alloy bearing material as a function of time. AdaBoost has several advantages over other ML algorithms, such as being robust to noise and outliers, being adaptive to changing data distributions, being able to handle high-dimensional data, being easy to implement and interpret, and being able to achieve high accuracy[13]. The algorithm is suitable for modeling the COF behavior of babbitt alloys, as it can capture the complex and nonlinear relationship between the time and COF values. It can also handle the variability and uncertainty of the COF values, as well as the different data sets for different materials and loads. AdaBoost_wear models the COF of three materials from a B83 babbitt alloy, each subjected to a pin-on-disk test with different loads (40 N, 50 N, and 60 N). The model uses the average COF calculated from the friction force and time data in the files. Performance metrics, such as coefficient of determination (R^2), root mean squared error (RMSE), mean squared error (MSE), and mean absolute error (MAE), are provided to evaluate the accuracy of the predictions. Furthermore, plots of the actual and predicted COF values, as well as histograms and boxplots of the COF distribution, are generated.

2. Software description

AdaBoost_wear is a Python software that models the COF of three materials from a B83 babbitt alloy, each subjected to a pin-on-disk test with different loads (40 N, 50 N, and 60 N).

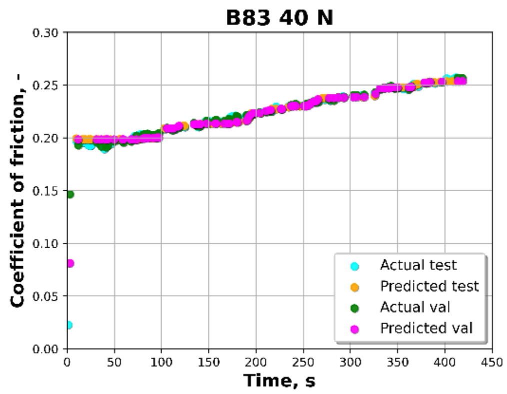
2.1. Main features and functionalities

The software performs the following main features and functionalities:

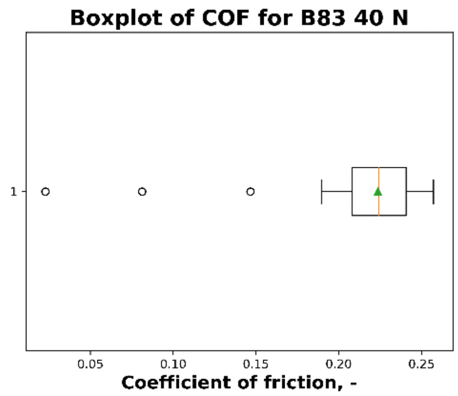
- It reads the experimental data of time and friction force from XLSX files for each material and load combination.
- It calculates the COF values by dividing the friction force by the normal load and computes the average COF values for each material and load combination.
- It calculates and saves some descriptive statistics (mean, median, standard deviation, minimum, and maximum) of the COF values for each material and load combination to TXT files. **Table 1** shows the

descriptive statistics of the average COF of B83 babbitt alloy under different loads (40 N, 50 N, and 60 N).

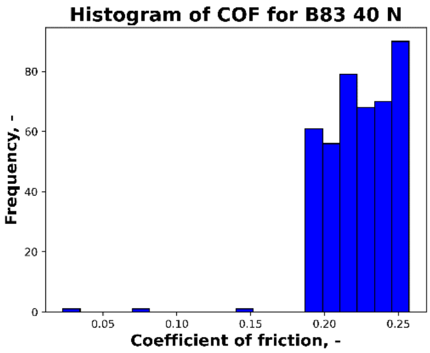
- It saves the average COF values and time values for each material and load combination to XLSX files, and merges them into one DataFrame for further analysis.
- It splits the data into training, validation, and testing sets with an 80/20 ratio for training/testing and a 75/25 ratio for training/validation.
- It performs grid search with cross-validation to find the best hyperparameters for an AdaBoostRegressor model, which is a ML algorithm that combines multiple weak learners (in this case, decision trees) to create a strong learner that can predict the COF values based on the time values. The model uses the grid search with cross-validation to find the best values for the tuning hyperparameters that suit the data and the problem. The hyperparameters that are used in this model are as follow: 'n_estimators': [50, 100, 200]; 'learning_rate': [0.1, 0.5, 1.0]; 'loss': ['exponential']; 'base_estimator': [DecisionTreeRegressor (max_depth = 3)].
- It evaluates the performance of the AdaBoostRegressor model on the test and validation sets using different metrics (R^2 score, RMSE, MSE, and MAE) and saves them to TXT files. **Table 2** shows the performance indicators for AdaBoost model of test and validation sets for each material and load combination.
- It plots the actual and predicted COF values as a function of time for each material and load combination, and saves them to PNG files. **Figures 1a**, **2a**, and **3a** show the plots of the actual vs predicted COF values (test and validation) for B83 babbitt alloy under loads of 40 N, 50 N, and 60 N respectively.
- It plots a histogram and a boxplot of the COF values for each material and load combination, and saves them to PNG files. **Figures 1b**, **2b**, and **3b** show the boxplots of the COF values for B83 babbitt alloy under loads of 40 N, 50 N, and 60 N respectively. **Figures 1c**, **2c**, and **3c** show the histograms of the COF distribution for B83 babbitt alloy under loads of 40 N, 50 N, and 60 N respectively.
- It plots a bar plot of the mean COF values with error bars for each material and load combination, and saves it to a PNG file. **Figure 4** shows the bar plot of the mean COF values with error bars for B83 babbitt alloy under loads of 40 N, 50 N, and 60 N.



(a)

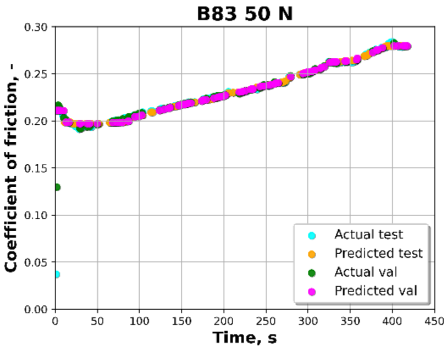


(b)

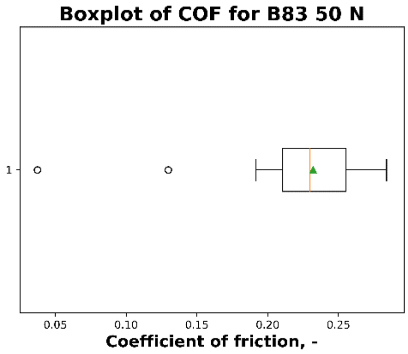


(c)

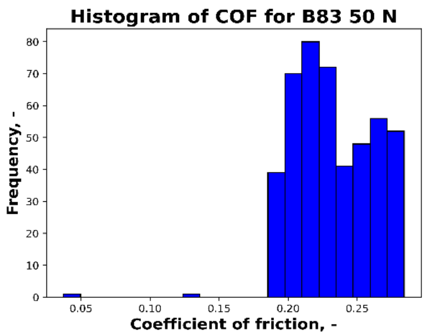
Figure 1. Graphical representation of the B83 babbitt alloy material subjected to wear at a load of 40 N: **(a)** plot of the actual vs predicted COF (test and validation); **(b)** boxplot of the COF values; **(c)** histogram of the COF distribution.



(a)

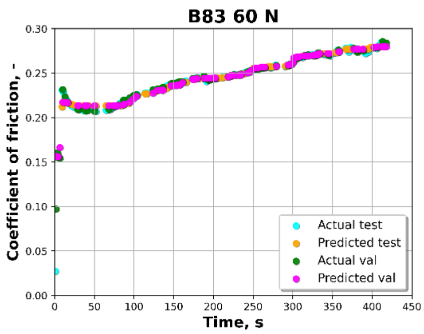


(b)

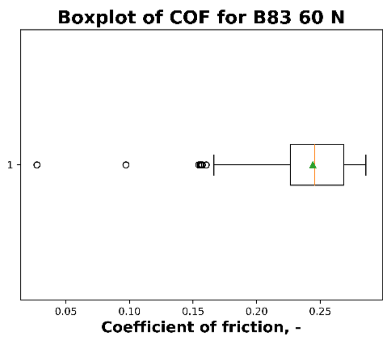


(c)

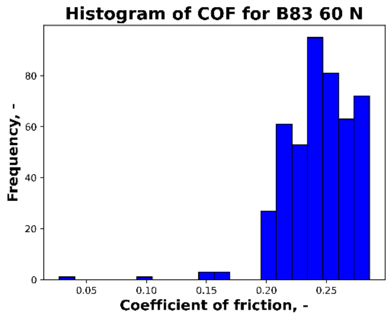
Figure 2. Graphical representation of the B83 babbitt alloy material subjected to wear at a load of 50 N: **(a)** plot of the actual vs predicted COF (test and validation); **(b)** boxplot of the COF values; **(c)** histogram of the COF distribution.



(a)



(b)



(c)

Figure 3. Graphical representation of the B83 babbitt alloy material subjected to wear at a load of 60 N: **(a)** plot of the actual vs predicted COF (test and validation); **(b)** boxplot of the COF values; **(c)** histogram of the COF distribution.

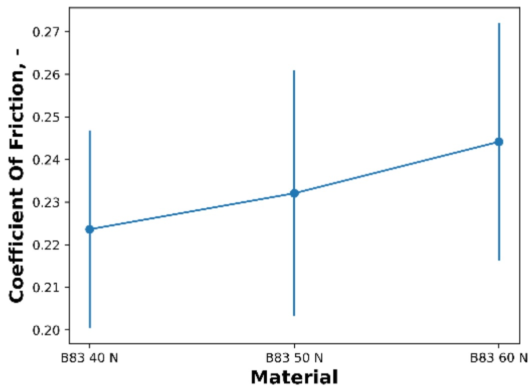


Figure 4. Bar plot of the mean COF values with error bars for B83 babbitt alloy under loads of 40 N, 50 N, and 60 N.

Table 1. Descriptive statistics of the average COF of B83 babbitt alloy under different loads (40 N, 50 N, and 60 N).

Load	Mean	Median	Standard deviation	Minimum	Maximum
40 N	0.2236	0.2241	0.0232	0.0228	0.2572
50 N	0.2321	0.2299	0.0288	0.0373	0.2840
60 N	0.2442	0.2456	0.0278	0.0272	0.2858

Table 2. Performance indicators for AdaBoost model of test and validation sets.

Set	R^2	MAE	RMSE	MSE
B83 babbitt alloy under load of 40 N				
Test	0.9471	0.0030	0.0069	0.0000
Validation	0.8622	0.0027	0.0075	0.0001
B83 babbitt alloy under load of 50 N				
Test	0.7454	0.0033	0.0182	0.0003
Validation	0.8995	0.0022	0.0087	0.0001
B83 babbitt alloy under load of 60 N				
Test	0.8306	0.0037	0.0139	0.0002
Validation	0.9497	0.0027	0.0069	0.0000

2.2. Installation and usage

The software is written in Python 3.9.13 and uses several libraries such as pandas, numpy, sklearn, matplotlib, and patheffects. The software is open source and available on GitHub at https://github.com/mihail-

15/AdaBoost_wear. The software is also verified and certified for computational reproducibility by CodeOcean[14]. The software can be accessed on CodeOcean at https://codeocean.com/capsule/0143927/tree/v1.

The software's installation and usage instructions are provided in detail in the README file on GitHub. The software requires Python 3.9 or higher and the following libraries: pandas 2.0.2 or higher, numpy 1.24.3 or higher, sklearn 1.2.2 or higher, matplotlib 3.7.1 or higher. The software can be installed by cloning or downloading the GitHub repository. The software can be used by running the AdaBoost_wear.py file in a Python interpreter or an IDE such as VS Code or PyCharm. The software will read the data from the XLSX files in the data folder, perform the analysis, save the results to TXT files in the results folder, and save the plots to PNG files in the plots folder.

3. Software outcomes

AdaBoost_wear has been used to model the COF behavior of B83 babbitt bearing alloy under different loads, which is important for understanding the wear mechanisms and optimizing the performance of sliding bearings. The software has the potential to contribute to scientific research by providing an accurate and efficient way of predicting the COF values based on experimental data, which can help researchers design better tribological systems.

The software has demonstrated its impact by producing high-quality results that are consistent with the experimental data and the literature. The software has achieved high R^2 scores (above 0.8) for both test and validation sets, indicating a good fit between the actual and predicted COF values. The software has also produced low values of mean absolute error (below 0.004), root mean squared error (below 0.02), and mean squared error (below 0.0004) for both test and validation sets, indicating a high accuracy of the predictions. The software has also generated informative plots that show the relationship between time and COF, as well as the distribution and variability of the COF values for each material and load combination.

The software has the possibility to demonstrate its reusability by being applicable to different materials and loads, as well as different types of data. The software can be easily adapted to model the COF behavior of other materials or loads by changing the input files or parameters. The software can also be used to model other types of data that have a similar structure or pattern, such as temperature, pressure, or velocity. The software can also be extended or modified to incorporate other features or functionalities, such as different ML algorithms, different performance metrics, or different visualization techniques.

The software has also demonstrated its relevance by being based on AdaBoost algorithm, which is a widely used ML technique for regression and classification problems. The model used by the software is similar to the ones applied to various domains and challenges, such as predicting micro-hardness in a welding process based on process parameters[10], diagnosis of motor bearing faults[15], predicting ultimate tensile strength of aluminium metal matrix hybrid composites[11], detecting faces in images captured by embedded systems[16], diagnosing breast cancer[17], and filtering spam sms[18]. AdaBoost algorithm has been shown to have several advantages over other ML techniques, such as being robust to noise and outliers, being adaptive to changing data distributions, being able to handle high-dimensional data, being easy to implement and interpret, and being able to achieve high accuracy with low computational cost.

It is open source, licensed under MIT License and available as a reproducible capsule at CodeOcean[14] and in a GitHub repository[19]. The AdaBoost_wear has been used for study and research in the department of High-porosity Metallic Materials part of IMSETHC-BAS and has been the subject of discussions with colleagues at the institute.

4. Future investigation and challenges

AdaBoost_wear has shown its potential and usefulness for modeling the COF behavior of B83 babbitt alloy under different loads, as well as comparing it with other materials. However, there are still some aspects that can be improved or explored in the future, such as:

- Extending the software to model the COF behavior of other types of tribological systems, such as lubricated contacts, rolling contacts, or abrasive wear.
- Incorporating other factors that may affect the COF behavior, such as temperature, humidity, surface roughness, or wear debris.
- Evaluating the robustness and generalizability of the software to different data sets, such as synthetic data, noisy data, or incomplete data.
- Developing a user-friendly graphical user interface (GUI) for the software that can allow users to interact with the software more easily and intuitively.

These future investigations and challenges can provide new insights and opportunities for advancing the research in tribology and machine learning, as well as improving the quality and impact of the software.

Funding

This research was funded by Bulgarian National Science Fund, Project № КП-06-Н57/20 “Fabrication of new type of self-lubricating antifriction metal matrix composite materials with improved mechanical and tribological properties”.

Acknowledgments

We acknowledge the support of the European Regional Development Fund within the OP Science and Education for Smart Growth 2014–2020, Project CoE National Center of Mechatronics and Clean Technologies, BG05M2OP001-1.001-0008.

Conflict of interest

The authors declare no conflict of interest.

References

1. Kolev M, Drenchev L, Stanev L. Tribological Characterization of Aluminum/Babbitt Composites and Their Application to Sliding Bearing. Archives of Foundry Engineering 2020, 20: 31-36.
2. Kankar PK, Sharma SC, Harsha SP. Fault diagnosis of ball bearings using machine learning methods. Expert Systems with Applications. 2011, 38(3): 1876-1886. doi: 10.1016/j.eswa.2010.07.119
3. Schwarz S, Grillenberger H, Graf-Goller O, et al. Using Machine Learning Methods for Predicting Cage Performance Criteria in an Angular Contact Ball Bearing. Lubricants. 2022, 10(2): 25. doi: 10.3390/lubricants10020025
4. Aydin F, Durgut R, Mustu M, et al. Prediction of wear performance of ZK60 / CeO2 composites using machine learning models. Tribology International. 2023, 177: 107945. doi: 10.1016/j.triboint.2022.107945
5. Tan H, Xie S, Liu R, et al. Bearing fault identification based on stacking modified composite multiscale dispersion entropy and optimised support vector machine. Measurement. 2021, 186: 110180. doi: 10.1016/j.measurement.2021.110180
6. He Z, Shi T, Xuan J, et al. Research on tool wear prediction based on temperature signals and deep learning. Wear. 2021, 478-479: 203902. doi: 10.1016/j.wear.2021.203902
7. Hasan MS, Kordijazi A, Rohatgi PK, et al. Triboinformatic modeling of dry friction and wear of aluminum base alloys using machine learning algorithms. Tribology International. 2021, 161: 107065. doi: 10.1016/j.triboint.2021.107065
8. Kolev M. COF-RF-Tool: A Python software for predicting the coefficient of friction of open-cell AlSi10Mg-SiC composites using Random Forest model. Software Impacts. 2023, 17: 100520. doi: 10.1016/j.simpa.2023.100520

9. Liu H. Data Mining and Processing for Train Unmanned Driving Systems. In: Unmanned Driving Systems for Smart Trains. Elsevier, 2021. pp. 211-252.
10. Yadav MK, Arora K, Kumar S, et al. Micro-hardness evaluation of the bobbin tool-friction stir welded AA6063 using regression-based machine learning. Materials Letters. 2023, 349: 134751. doi: 10.1016/j.matlet.2023.134751
11. Rajput R, Raut A, Gangi Setti S. Prediction of mechanical properties of aluminium metal matrix hybrid composites synthesized using Stir casting process by Machine learning. Materials Today: Proceedings. 2022, 59: 1735-1742. doi: 10.1016/j.matpr.2022.04.316
12. Ahmad A, Ahmad W, Chaiyasarn K, et al. Prediction of Geopolymer Concrete Compressive Strength Using Novel Machine Learning Algorithms. Polymers. 2021, 13(19): 3389. doi: 10.3390/polym13193389
13. Dev B, Rahman MA, Islam MdJ, et al. Properties prediction of composites based on machine learning models: A focus on statistical index approaches. Materials Today Communications. 2024, 38: 107659. doi: 10.1016/j.mtcomm.2023.107659
14. Kolev M. AdaBoost_wear: A Python software using AdaBoost model for predicting the coefficient of friction of B83 babbitt alloy. Available online: https://doi.org/10.24433/CO.8613877.v1 (accessed on 28 June 2023).
15. Jiang J, Liu Y, Xu C, et al. Research on Motor Bearing Fault Diagnosis Based on the AdaBoost Algorithm and the Ensemble Learning with Bayesian Optimization in the Industrial Internet of Things. Soni M, ed. Security and Communication Networks. 2022, 2022: 1-12. doi: 10.1155/2022/4569954
16. Yang M, Crenshaw J, Augustine B, et al. AdaBoost-based face detection for embedded systems. Computer Vision and Image Understanding. 2010, 114(11): 1116-1125. doi: 10.1016/j.cviu.2010.03.010
17. Zheng J, Lin D, Gao Z, et al. Deep Learning Assisted Efficient AdaBoost Algorithm for Breast Cancer Detection and Early Diagnosis. IEEE Access. 2020, 8: 96946-96954. doi: 10.1109/access.2020.2993536
18. Zhang X, Xiong G, Hu Y, et al. A Method of SMS Spam Filtering Based on AdaBoost Algorithm. In: Proceedings of the 2016 12th World Congress on Intelligent Control and Automation (WCICA); June 2016; pp. 2328-2332.
19. Kolev M. AdaBoost_wear: AdaBoost_wear Is a Python Software That Implements the AdaBoost Algorithm to Predict the Coefficient of Friction (COF) of B83 Babbitt Alloy as a Function of Time; Github.