Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 107 (2022) 528–533

www.elsevier.com/locate/procedia

55th CIRP Conference on Manufacturing Systems

Production System Efficiency Optimization Using Sensor Data, Machine Learning-based Simulation and Genetic Algorithms

Joao Henrique Cavalcanti[a*], Tibor Kovács[b], Andrea Kő[b]

[a]*Corvinus University of Budapest,The Doctoral School of Economics, Business and Informatics*
[b]*IT Institute, Corvinus University of Budapest, Fővám tér 8. 1093 Budapest, Hungary*

* Corresponding author Tel.: +36-70-515-9419; *E-mail address:* jh_gcc@hotmail.com

Abstract

In modern industries, there is a significant repository of sensor data, which contains a large amount of information. Unfortunately, this rich source of information is undervalued and underutilized, and its full potential is not fully exploited by modern day manufacturers. In the Industry 4.0 era, exploiting these powerful datasets is becoming critical for manufacturers' survival and competitiveness in the age of artificial intelligence. Cooperative and mutual efforts between academia and the industrial sector to take advantage of these rich datasets have the potential to reap extraordinary benefits for business, the economy and society. Applying the latest artificial intelligence methods could increase production efficiencies and reduce environmental impacts. In view of the availability of large amounts of sensor data and its lack of full utilization, this research proposes an artificial intelligence solution that combines data envelopment analysis (DEA), machine learning-based simulation and genetic algorithms to optimize the efficiency of production systems through recommendations of the optimal model settings. First, DEA is used to identify the efficient and inefficient states of a production system, this information is input to the second step to build a machine learning model that makes predictions through simulations and production efficiency scenarios. Then, a genetic algorithm proposes an optimal scenario with the corresponding settings. The main research contribution of this proposed solution is its unique combination of DEA with machine learning models and genetic algorithms.

Keywords: Industry 4.0; Machine Learning; Data Envelopment Analysis; Genetic Algorithims

1. Introduction

Production systems are dynamic and fast paced. Lean manufacturing, a production system philosophy, seeks to remove all types of production waste to consequently improve production efficiency. Most of the time, it is worth improving production systems' efficiency, even in the slightest way, as modern industries are in a continuous race for positive results. As science and technology advance, an increasing number of techniques are used in support of lean production systems. Industry 4.0, also called the fourth industrial revolution, encompasses a wide range of advanced technologies, such as artificial intelligence (AI), robotics, the internet of things and cloud computing, which are all changing production paradigms and business models. In short, Industry 4.0 automates industrial processes and integrates advanced technology with the workforce to increase production efficiency.

Given the abundance of sensor data generated by highly automated and interconnected production systems, it makes sense for industries, in their continuing drive for waste reduction, to utilize these rich data depositories as a resource. This study proposes an artificial intelligence solution that combines data envelopment analysis (DEA), machine learning (ML) and genetic algorithms (GA) to increase production efficiencies by finding optimal production settings and configurations. Hence, the proposed AI solution contributes to the extension of Industry 4.0 and lean manufacturing techniques for modern industries. This article has eight sections. Section 1 introduces the motivation and the background. Section 2 discusses the state-of-the-art of ML, GA and DEA. Section 3 provides an overview of the proposed

10.1016/j.procir.2022.05.020

solution and describes the model inputs, components and outputs in depth. Finally, Section 4 concludes this article with the benefits, limitations and potential future steps for the research. The suggested AI solution can be implemented in the industrial sector as an add-on decision support system for SCADA (supervisory control and data acquisition) systems.

2. Literature review

2.1. DEA and machine learning

DEA is a well-established technique to identify the efficiency frontier of multiple input–output systems. DEA was initially proposed to evaluate the activities of business entities; however, it can be extended to other problems where multiple input–output combinations can be observed with differing efficiencies. The technique is based on mathematical programming and assumes that there is no noise in the data. If the data have noise, then the approximation of the efficiency frontier may be overstated [1].

Research [2] aims to establish a linkage between the DEA method and the ML algorithm by measuring and predicting the DEA efficiency performance of Chinese manufacturing companies in 2016. In these scenarios, ML is integrated into DEA with the intent of predicting the efficiency of the manufacturers, but the process does not involve any optimization or sensor data. Moreover, [3] has a more general approach, mixing DEA and machine learning to evaluate the financing efficiency of 39 listed agricultural companies in China from 2013 to 2017. The influencing factors that have a higher weight on the ML model are explored and discussed. Similarly, [4] uses a hybrid model of DEA and a decision tree algorithm to evaluate the efficiency of ports. Combining DEA and ML has also been used for computational optimization. In manuscript [5], a hybrid DEA-Adaboost model is proposed to simplify the supplier selection process. The results show that the approach reduced the time consumption and computational complexity.

Combinations of DEA and ML has been widely documented in the literature for different purposes. However, DEA calculations of manufacturing systems via production sensor data for production efficiency optimization have not yet been fully explored by researchers.

2.2. Machine learning and genetic algorithms

There are many scenarios when ML can be combined with GA; ML can speed up GA when used as a fitness function, or GA can be used to optimize the tuning of ML model hyperparameters. Another underexplored use case is the usage of ML as a fitness function when it is not possible to have a fitness function equation.

The literature on the combined usage of ML and GA is large and diverse. [6, 7] combine GA with ML to select the optimal feature subset. Similarly, [8] combines the two methods to find the optimal hyperparameters for classification systems and improved model performances. [9] compares the two methodologies, one is based on deep Q-learning and another is based on genetic algorithms. Furthermore, [10,11] combine GA and ML to explore the advantages in the acceleration of searches. This approach is similar to our proposal to some extent, as it uses ML as a fitness function for GA; however, the reasons are different, as their focus is on speeding up GA by mixing ML with it. In these two research papers, fitness functions are possible to calculate but the challenge is in the calculation expense of repeatedly doing it. The difference is that our research combines ML and GA because it is not possible to calculate GA chromosome fitness without it and not for the purpose of accelerating the computational time. In short, no research combining ML with the GA's fitness function for the purpose of optimizing production systems was found.

2.3. Combination machine learning, DEA and genetic algorithm

Although combining ML with GA, or DEA with ML, has been widely discussed in the literature, we did not find any research combining all three. In addition to the combination of these three methods, this research is novel in applying this approach to production efficiency optimizations. The combination of the three methods can open new paths of research and help enlarge the usage of hybrid artificial intelligence models in Industry 4.0. Therefore, the main scientific contribution of this solution is the unique combination of DEA with ML models and GA. In addition, this study opens the possibility for applying these combinations in different industries for optimization problems.

3. Model overview

The proposed AI solution reads a dataset of sensor signals and input parameters by the user, processes this information, and generates recommendations of optimal settings and configurations that aim to maximize production efficiency. The model has 3 main components, DEA, ML and GA, which, when combined, work as a powerful AI solution capable of predicting the relative production efficiencies of different settings and configurations. Moreover, GA component uses these predictions to search for different settings and configurations using the user's inputs, and selects the optimal ones. The system is outlined at a high level in Fig. 1.

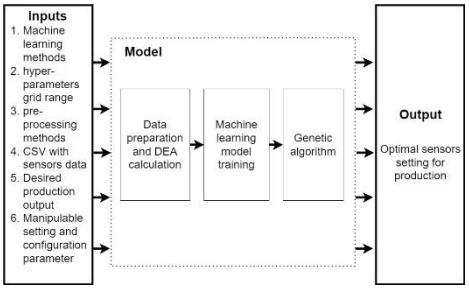

Figure 1 AI solution macroprocess

The Inputs phase is the first step of the system, containing the initial settings prior to the model's calculations. There are 6 inputs required by the AI solution:

- Machine learning methods
- Hyperparameters grid range

- Pre-processing methods
- CSV with sensors data
- Desired production output
- Configuration and settings parameters that the user can select and adjust.

Once the initial inputs are ready, the model starts to act. The first step is data preparation and a DEA calculation using historical data. During this phase, outliers are removed, and the relative production efficiencies are calculated. The second step is to train a machine learning model capable of predicting the relative production efficiency given a set of sensor values, settings and configurations. Finally, the GA simulates different settings and configurations and selects the optimal one for the desired production output, using the ML model as the fitness function to calculate relative production efficiencies. As a result of all the steps, the optimal settings and configuration can be used as parameters for achieving the desired production output.

3.1. Model inputs

3.1.1. Machine learning methods

Relative production efficiency is a continuous data type; on the other hand, sensor data and configuration parameters can be a mix of binary and numerical data. There are three ML models recommended for this type of data: linear regression (LN), regression trees (RT) and neural networks (NN), which are the ML model types capable of achieving such results [12].

This research proposes the use of elastic net (LN method), gradient boosting (RT method) and support vector machine (RT method), which are well-known, industry standard ML methods used to predict continuous data based on continuous/binary dependent variables [13, 14, 15]. For more flexibility, the user is able to choose which one of those methods to use, or multiple methods can be selected, and the best fit will be selected by the AI component. It is worth noting that the calculation time for fitting the model is directly proportional to the number of models included. Neural network-based methods are not included because they can significantly increase the training time [16].

3.1.2. Hyperparameter grid range

Hyperparameter tuning is one of the most crucial phases of ML model training. The most common practice for tuning hyperparameters considers the selection of an ad hoc range of common hyperparameter values and empirically searches for the best fit [17]. For the proposed research, a fixed range of the most common values of hyperparameters can be used, or this range of values can be set manually by the user. However, large ranges of values can lead to long training times.

3.1.3. Pre-processing methods

Data pre-processing is an essential step of ML modelling. Pre-processing delivers higher quality data before ML training and consequently improves the prediction performance. Min-Max Scaler, Standard Scaler, Max-Abs Scaler, Robust Scaler, Quantile Transformer (Normal), Quantile Transformer (Uniform) and Power Transformer are conventional pre-processing techniques. Current practices for selecting pre-processing methods typically involve trying variations of these methods and selecting the most accurate one [18]. Hence, similar to the previous inputs, the suggested AI solution should provide options to the user to include all or some of those pre-processing methods. The best fit from the selected pre-processing methods will be used in the ML model. Similar to the previous input, the training time increases as more possible pre-processing methods are selected.

3.1.4. Sensor data of various historical operations output

A CSV dataset containing sensor signals is the primary source of data for the ML model. This dataset should include continuous data such as infeed speed, pressure, temperature, and the binary signals of the configuration parameters indicating the operation status of a feature/input. Other sensor data may fall outside of the control of the operations and can be collected by external sensors or public data sources (e.g., weather forecasts). However, one of the major problems in ML is the selection of the most representative features by taking into account the possible disadvantages that can exist. A large number of features or signals used as dependant variables, noise data and irrelevant data should be avoided and not included in the model's dataset [19].

3.1.5. Desired production output

The sensor data, that indicate the production output, should be specified since it will be used for DEA and used further on as a dependent variable in the machine learning model. Moreover, the desired production output should be stated since it represents what output needs to be achieved at the maximum possible production efficiency. Desired production outputs might include values that originate from production plans or contractual obligations as well as production forecasts. This stated value should be appended to the column of the dataset containing the actual production output so that each line containing sensor data will also have a value of the production output. The last line of the dataset should contain the current desired production output, together with the latest sensor measured values.

3.1.6. Manipulable settings and configuration parameters

Production systems have multiple signals, and some of these data can be manipulated by adjusting production input settings, or choosing from different configuration parameters. Some of these settings or configuration parameters cannot be, or can only be, changed within boundaries for various reasons. Safety, product specifications and environmental awareness are some examples of why some settings or parameters are immutable.

In the proposed solution, the user should determine which of those values can be manipulated, and those pointed out by the user will be the ones that the AI model will try to optimize.

3.2. Data preparation and DEA

Data preparation and DEA calculation are the first components of the AI solution. During this step, two predefined inputs are used from the historical data (CSV): the recorded values of the manipulable input settings, which were set by the

user, and the recorded values of the sensor data. These sensor data are either production output values or production input values that were achieved as a result of the manipulable input settings.

At this phase, two main processes can be performed in parallel, as seen in Fig. 2. On the left branch of Fig. 2, CSV data are used to perform DEA and calculate the productive efficiency of the decision-making units (DMUs, the different valid configurations and combinations of input–output values), which will then pass through an outlier removal performed by the interquartile range (IQR) method. Outlier removal is an important pre-processing method for machine learning models. The usage of such a method can significantly improve the accuracy of the ML model [20, 21]. In contrast, keeping outliers in the dataset increases the error variance and reduces the power of the statistical tests; therefore, outlier removal is recommended for most ML models [22]. The pre-processed sensor dataset with an attached relative productive efficiency is the result of the left branch. This new dataset will be later used on the ML model training.

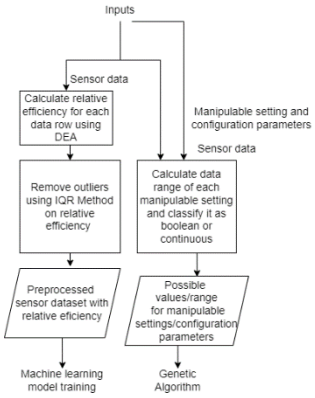

Figure 2. Data preparation and DEA flowchart

On the right branch of Fig. 2, information given by the user of what manipulable input settings were used, together with the CSV dataset containing all sensor data, are used to determine the data range and data type of the manipulable input settings. Considering that not all input settings can be manipulated, determining which inputs can be manipulated is necessary for the production efficiency optimization. There can be many reasons why some inputs cannot be manipulated. Input variables could be measuring external variables that are outside of the control of the operations (e.g., external temperature) or variables that, for specific reasons, are not flexible for changes. Safety parameters, product specification and environmental awareness are some examples of types of production system concerns that can make an input value unchangeable.

Data types are needed to classify the manipulable input data into binary or continuous types. The data types and a possible range of values of sensor data are later used to generate an initial population and mutate offspring during the GA phase. The scalability of the DEA method in handling a large portion of datasets extracted from the manufacturing sensors is limited by the number of variables the LP solver is allowed to handle (e.g., the Python package (PyDEA)), using the PuLP package and IBM ILOG CPLEX as a solver is limited to 10000 variables as an academic licence). Data will be stratified and downsampled to meet these limitations.

3.3. Machine learning model training

Machine learning model training is the next step for the proposed AI solution after DEA and data preparation. At this phase, the previously treated dataset will be used as input to find the optimal ML model capable of predicting the production efficiency given a set of sensor data configurations. However, splitting the data into training and test datasets should be done before the training starts. Splitting datasets is a common practice in ML algorithms. A train-test split separates a portion of data to be used exclusively for validation, producing more robust and unbiased models [23, 24]. 60/40, 70/30, 75/25 and 80/20 are common proportions used for the train-test split. The proposed train-test split proportion for this AI system is 70/30 since it is commonly used in many previous studies, but other common values could also be implemented [25, 26].

As mentioned before, the user can select multiple ML methods, possible hyperparameter values and pre-processing methods. All these parameters will be validated, and their scores will be checked. The best model will be selected through an exhaustive search method called a grid search. Grid search is a hyperparameter optimization method that simply makes a complete search over a given subset of the hyperparameter space. This subset of parameters has a well-known possible range of values for each ML model, and the process of choosing the optimal one is an empirical process; therefore, grid searching through these possible values is a common practice. Fig. 3 describes the machine learning model training steps via a process flowchart.

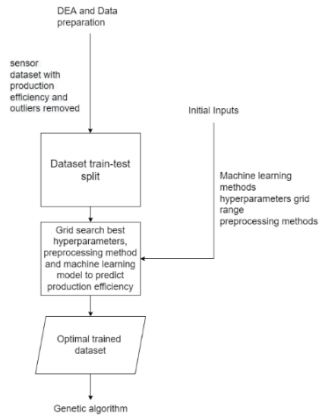

Figure 3. Machine learning model training flowchart

Raising the number of possible values within the common optimal range can significantly increase the training time; hence, this input set should be chosen carefully by the user. Genetic algorithms have been intensively discussed as an alternative for grid search, and many studies have obtained good results with this approach [27]. However, this AI solution uses grid search, using GA to find the optimal production settings by using ML as a fitness function to predict a given efficiency; hence, the computational resources focus on that. The GA phase will be described in more detail in the next section. Another consideration worth mentioning is that this AI solution extends the grid search to the chosen ML and the pre-processing methods mentioned in Section 3.1. Hence, the model's hyperparameters are not the only consideration as possible values for the model optimization. Extending grid

search is a common technique and is supported by the Python library scikit-learn [28].

[29] suggests the employment of R-squared as the standard statistical measure to evaluate regression analyses in any scientific area, and this research complies with it. Therefore, R-squared is the chosen scoring method to grid search the optimal model.

3.4. Genetic Algorithm

The final component of the proposed AI system is the genetic algorithm. At this phase, the GA simulates different input settings and selects the optimal one using the ML model that was previously trained as the GA fitness function. The fitness function is part of the GA and is responsible for evaluating the quality of each potential solution. A flowchart of this phase is described in Fig. 4.

Manipulable input setting ranges and types are inputs from the previous data preparation phase, and they will be used to randomly generate an initial population for the algorithm. Afterwards, the trained ML model calculated in the previous step comes to action as the fitness function of the GA. Therefore, the initial randomly generated setup for sensors has their predicted efficiency calculated by the ML model, and every prediction done by the model should consider the desired production output as one of the dependent variables. This will result in a subset of different production settings and their respective efficiencies, which will be divided between worst and best settings. Moreover, the worst and best settings are then selected for crossover. There are other selection/crossover methods, such as selecting only the strongest and the crossover themselves; however, the best-worst selection criteria leads to better performance [30]. The genetic algorithm will be controlled by constraint boundaries for the parameters, so it will not generate infeasible solutions.

Afterwards, the best crossover with the worst settings occurs, creating offspring who have a mix of settings inherited from their parents. Similar to what happens in nature, the generated offspring are randomly affected by mutations. Mutations are one of the key engines of genetic algorithms since they maintain the genetic diversity from one population to the next population [31]. In this research, mutating means that some of the offspring genes inherited from their parents can suffer from random mutations at a 10% rate, and if a mutation occurs, 10% of the chromosome's genes are randomly mutated. The mutation rate in GA must be a low value because high mutation rates lead to a primitive random search and no convergence; on the other hand, too low mutation rates can lead to local maxima. Based on previous research, the mentioned rates are considered good candidates for GA tuning [32]. If the local maxima problem persists, the user can change the mutation rate to higher values, or if no convergence is found, mutation rates can be decreased.

After mutation, the previously trained ML model comes to action again, and the new population passes through the same process as the initial population. Hence, production efficiency is predicted by the ML model for each chromosome of the newly generated population. Finally, the GA checks if any termination criteria have been reached; if yes, the optimal solution is given. Otherwise, the algorithm returns to the selection phase where the best/worst chromosomes are chosen, later matched with crossover, and eventually mutated. The process repeats until at least one of the termination criteria is met.

There are three termination criteria, a timeout of 10 minutes, 100% efficient chromosomes and efficiency convergence. A ten-minute timeout is needed because production environments are fast paced and dynamic; therefore, waiting too long for a decision can make the decision useless and inapplicable. Moreover, sensor data can change significantly during the calculation time, causing prediction errors. However, finding an optimal solution with 100% relative production efficiency leads to termination, indicating that the genetic algorithm succeeded. There may be cases when 100% relative efficiency cannot be reached, and the iteration of GA does not give a better efficiency than the ones stored from previous generations. For that case, the algorithm should also be terminated because of convergence. When termination is reached, the user receives a recommendation of the manipulable input settings and configuration parameters, and the production system can be set to achieve optimal efficiency.

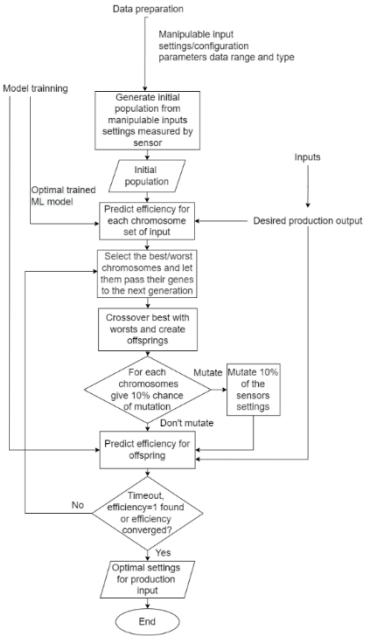

Figure 4. Genetic algorithm flowchart.

Conclusion

Although combining DEA-ML and ML-GA has been widely discussed in the literature, we did not find studies of a combination of the three methods in the literature. In this research, a combination of DEA, GA and ML was proposed and described, creating a hybrid artificial intelligence solution capable of optimizing production efficiency through manufacturing sensor data. Implementing such an AI solution can reduce production costs by optimizing resource usage and delivering economic and environmental benefits.

This new approach opens up space for different research directions, indicating to the scientific community that many problems may be resolved by such an AI mix. Furthermore, the implementation of this newly proposed solution in different

manufacturing fields can extend the capabilities of Industry 4.0.

Successful implementations require some new considerations. Limitations on calculation times and the dynamics of a changing production systems environment must be observed for each implementation case. Balancing the benefits of having a complex model with the cost of the calculation time should be adjusted. Decoupling GA from DEA-ML might be needed in cases where the DEA-ML calculation time is not suitable for the production system type. In that case, DEA-ML can be calculated at a different time frequency than GA; however, this may lead to inaccurate predictions, as some newer data are not included in the ML model. On the other hand, reducing the dataset to include only recent sensor signals can significantly improve the calculation time but may impact the prediction performance.

Implementation of the suggested AI solution is currently in the prototype phase, and further research regarding the impact of such a solution in a real production environment is ongoing. The suggested AI solution will be first implemented in the energy generation industry as an add-on decision support system for the SCADA production system. In the future, other SCADA-based industries can be explored.

Acknowledgements

The present publication is the outcome of the project "From Talent to Young Researcher project aimed at activities supporting the research career model in higher education", Identifier EFOP-3.6.3-VEKOP-16-2017-00007 co-supported by European Union, Hungary and the European Social Fund.

References

[1] Charnes, Abraham, William W. Cooper, and Edwardo Rhodes. "Measuring the efficiency of decision making units." European journal of operational research 2.6 (1978): 429-444.

[2] Salehi, V., B. Veitch, and M. Musharraf. "Measuring and improving adaptive capacity in resilient systems by means of an integrated DEA-Machine learning approach." Applied ergonomics 82 (2020): 102975.

[3] Lixia Liu, Xueli Zhan, "Analysis of Financing Efficiency of Chinese Agricultural Listed Companies Based on Machine Learning", Complexity, vol. 2019, Article ID 9190273, 11 pages, 2019. https://doi.org/10.1155/2019/9190273

[4] Hong, Han-Kook, Byung-hak Leem, and Sam-Moon Kim. "Using a Hybrid Model of DEA and Decision Tree Algorithm C5. 0 to Evaluate the Efficiency of Ports." The Journal of the Korea Contents Association 19.7 (2019): 99-109.

[5] Cheng, Yijun, et al. "A hybrid DEA-adaboost model in supplier selection for fuzzy variable and multiple objectives." IFAC-PapersOnLine 50.1 (2017): 12255-12260.

[6] Ko, Taehoon, et al. "Machine learning-based anomaly detection via integration of manufacturing, inspection and after-sales service data." Industrial Management & Data Systems (2017).

[7] Badnjević, Almir, et al. "Evidence-based clinical engineering: machine learning algorithms for prediction of defibrillator performance." Biomedical Signal Processing and Control 54 (2019): 101629.

[8] Di Noia, Antonio, et al. "Supervised machine learning techniques and genetic optimization for occupational diseases risk prediction." Soft Computing 24.6 (2020): 4393-4406.

[9] Miralles-Pechuán, Luis, et al. "A methodology based on deep q-learning/genetic algorithms for optimizing covid-19 pandemic government actions." Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2020.

[10] Jennings, Paul C., et al. "Genetic algorithms for computational materials discovery accelerated by machine learning." npj Computational Materials 5.1 (2019): 1-6.

[11] Guo, Peng, Wenming Cheng, and Yi Wang. "Hybrid evolutionary algorithm with extreme machine learning fitness function evaluation for two-stage capacitated facility location problems." Expert Systems with Applications 71 (2017): 57-68.

[12] Knijnenburg, Theo A., et al. "Logic models to predict continuous outputs based on binary inputs with an application to personalized cancer therapy." Scientific reports 6.1 (2016): 1-14.

[13] Touzani, Samir, Jessica Granderson, and Samuel Fernandes. "Gradient boosting machine for modeling the energy consumption of commercial buildings." Energy and Buildings 158 (2018): 1533-1543.

[14] Mokhtari, Soroush, William Navidi, and Michael Mooney. "White-box regression (elastic net) modeling of earth pressure balance shield machine advance rate." Automation in Construction 115 (2020): 103208.

[15] Laref, Rachid, et al. "On the optimization of the support vector machine regression hyperparameters setting for gas sensors array applications." Chemometrics and Intelligent Laboratory Systems 184 (2019): 22-27.

[16] Nabipour, Mojtaba, et al. "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis." IEEE Access 8 (2020): 150199-150212.

[17] Li, Hao, et al. "Rethinking the hyperparameters for fine-tuning." arXiv preprint arXiv:2002.11770 (2020).

[18] Misra, Puneet, and Arun Singh Yadav. "Impact of Preprocessing Methods on Healthcare Predictions." Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE). 2019.

[19] Moldovan, Dorin, et al. "Machine learning for sensor-based manufacturing processes." 2017 13th IEEE international conference on intelligent computer communication and processing (ICCP). IEEE, 2017.

[20] Nnamoko, Nonso, and Ioannis Korkontzelos. "Efficient treatment of outliers and class imbalance for diabetes prediction." Artificial Intelligence in Medicine 104 (2020): 101815.

[21] Perez, Husein, and Joseph HM Tah. "Improving the accuracy of convolutional neural networks by identifying and removing outlier images in datasets using t-SNE." Mathematics 8.5 (2020): 662.

[22] Brownlee, Jason. Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python. Machine Learning Mastery, 2020.

[23] Bai, Yu, et al. "How Important is the Train-Validation Split in Meta-Learning?." International Conference on Machine Learning. PMLR, 2021.

[24] Vabalas, Andrius, et al. "Machine learning algorithm validation with a limited sample size." PloS one 14.11 (2019): e0224365.

[25] Kebonye, Ndiye M. "Exploring the novel support points-based split method on a soil dataset." Measurement 186 (2021): 110131.

[26] Chuang, Kangway V., and Michael J. Keiser. "Comment on "Predicting reaction performance in C–N cross-coupling using machine learning"." Science 362.6416 (2018).

[27] Liashchynskyi, Petro, and Pavlo Liashchynskyi. "Grid search, random search, genetic algorithm: a big comparison for NAS." arXiv preprint arXiv:1912.06059 (2019).

[28] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[29] Chicco D, Warrens MJ, 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 https://doi.org/10.7717/peerj-cs.623

[30] Hussain, Abid, Yousaf Shad Muhammad, and Muhammad Nauman Sajid. "Performance evaluation of best–worst selection criteria for genetic algorithm." Math Comput Sci 2.6 (2017): 89-97.

[31] Katoch, Sourabh, Sumit Singh Chauhan, and Vijay Kumar. "A review on genetic algorithm: past, present, and future." Multimedia Tools and Applications 80.5 (2021): 8091-8126.

[32] Mirjalili, Seyedali. "Genetic algorithm." Evolutionary algorithms and neural networks. Springer, Cham, 2019. 43-55