Enhancing K-Nearest Neighbors Algorithm in Wireless Sensor Networks through Stochastic Fractal Search and Particle Swarm Optimization

Ahmed Mohamed Zaki[1], Abdelaziz A. Abdelhamid[2], Abdelhameed Ibrahim[3], Marwa M. Eid[4,5], El-Sayed M. El-Kenawy*[5]

[1] Computer Science and Intelligent Systems Research Center, Blacksburg 24060, Virginia, USA
[2] Computer Science Department, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, 11566, Egypt
[3] School of ICT, Faculty of Engineering, Design and Information & Communications Technology (EDICT), Bahrain Polytechnic, PO Box 33349, Isa Town, Bahrain
[4] Faculty of Artificial Intelligence, Delta University for Science and Technology, Mansoura 35712, Egypt
[5] Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology, Mansoura, 35111, Egypt

Emails: azaki@jcsis.org; abdelaziz@cis.asu.edu.eg; abdelhameed.fawzy@polytechnic.bh; mmm@ieee.org; skenawy@ieee.org

Abstract
The utilization of wireless sensor networks (WSNs) holds significant importance in diverse data collection applications. Efficient operation of computers, especially in predictive tasks, is imperative for obtaining accurate results within WSNs. This research introduces an innovative approach employing Stochastic Fractal Search-Particle Swarm Optimization (SFS-PSO) to enhance the performance of the K-Nearest Neighbors (KNN) algorithm. The proposed methodology initiates with the establishment of a particle population, dynamically adjusting their positions and velocities and integrating a diffusion process. Through an iterative process of incremental adjustments and evaluations, the algorithm fine-tunes its parameters, resulting in a refined KNN regression model. The enhanced model exhibits substantial improvements, as indicated by the notable reduction in root mean square error (RMSE) and mean absolute error (MAE), accompanied by a strengthened correlation between variables. The favorable outcomes underscore the efficacy of the SFS-PSO optimization technique in augmenting the KNN algorithm's performance within wireless sensor networks. In simpler terms, the application of SFS-PSO in conjunction with KNN leads to a significant decrease in RMSE, reaching a value as low as 0.00894, demonstrating the notable effectiveness of this optimization approach in refining the predictive capabilities of the KNN algorithm in the context of WSNs.

Keywords: Wireless Sensor Network; Optimization; Stochastic Fractal Search; Particle Swarm Optimization; K-Nearest Neighbors; Algorithm.

1. Introduction

Wireless Sensor Networks (WSNs) represent a cultured amalgamation of sensing technologies [1], communication nodes, and data transmission mechanisms, as depicted in the comprehensive illustration encapsulated in Figure 1. The intricate functionalities portrayed in this network diagram underscore the pivotal role of WSNs in real-time data acquisition across diverse domains. From

Doi : https://doi.org/10.54216/JCIM.130108
Received: May 28, 2023 Revised: August 16, 2023 Accepted: December 21, 2023

environmental monitoring to industrial automation, healthcare applications, and the realization of smart cities, the pervasive deployment of WSNs has ushered in a new era of data-driven insights [2].

Figure 1: Functionality of a Wireless Sensor Network (WSN)

Our search into how to use WSNs gives you a taste of the big problems linked with making the best algorithms inside these networks [3]. We needed better precision, less confusing calculations, and an improved ability to guess right. So, a surprising solution was made - Stochastic Fractal Search-Particle Swarm Optimization (SFS-PSO). This system, which came from the computer and was inspired by nature, gives a big change in making the K-Nearest Neighbors (KNN) algorithm work best inside WSNs.

Research Questions:

- What is the impact of Stochastic Fractal Search-Particle Swarm Optimization (SFS-PSO) on the predictive capabilities of the K-Nearest Neighbors (KNN) algorithm in Wireless Sensor Networks (WSNs)?
- How do the underlying principles and mechanisms of SFS-PSO contribute to its optimization prowess within the context of WSNs and the KNN algorithm?
- In what ways does the integration of SFS-PSO with KNN affect key performance metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R2) in Wireless Sensor Networks?

As our narrative unfolds, the intricate tapestry of WSNs, algorithmic optimization challenges, and the innovative SFS-PSO solution converge to shape a narrative that not only addresses existing issues but propels the realm of Wireless Sensor Networks toward new frontiers of data-driven excellence.

2. Literature Review

The literature on wireless sensor networks (WSNs) and algorithm improvements gives a full idea of the problems and ways of these complex systems. Sensors connected in WSNs have changed a lot. They used to collect data, but now they are part of bigger networks doing important work in many areas.

Importance of WSNs:
WSNs are important because they can do many things, like checking the environment, health care, and smart cities. This is possible because these sensors collect fresh data right when it happens. These

Doi : https://doi.org/10.54216/JCIM.130108
Received: May 28, 2023 Revised: August 16, 2023 Accepted: December 21, 2023

networks are important for grabbing, sending, and processing data [4]. This is very true because they can cover large areas quickly as everything isn't in one spot.

Challenges in Algorithm Optimization:
Even though WSNs are very useful, they face problems with improving their algorithms to make performance better. Usually, sensor nodes have limited resources. This makes it hard to find the right balance between making sure an algorithm works well and doing things quickly at the same time. Old ways of optimizing don't always work well with sensor networks. They change fast and have limited resources, which makes it hard for them to adjust. We need to look at new ideas to solve these issues the right way[5].

Nature-Inspired Algorithms:
The literature review underscores a shift towards the adoption of nature-inspired algorithms as a viable strategy to overcome optimization challenges in WSNs. Algorithms such as Particle Swarm Optimization (PSO) and Stochastic Fractal Search (SFS) draw inspiration from natural phenomena. PSO mimics the collective behavior of swarms, while SFS leverages self-similar patterns observed in fractals. These nature-inspired algorithms have demonstrated effectiveness in various applications, prompting their exploration in the context of WSNs. Combining these algorithms, such as integrating SFS and PSO, introduces adaptability and robustness to optimization processes[6-8].

As we explore what has been studied before, it's clear that the mix of WSNs (Wireless Sensor Networks), improving computer code, and learning from nature is a cool thing to research. The next parts will talk about the SFS-PSO method and how it is used to improve the KNN algorithm within WSNs. We'll also discuss important measures like RMSE, MAE, and R2 that can be affected by this approach. Next, we will look at how this way of making things better really meets the difficulties with WSNs. It provides a more effective and careful method for handling data decisions in changing places.

3. Proposed Methodology

A. Dataset

1. Data Collection:

In our first step of the study, we carefully get information from Kaggle [9] - a known place for data sets. They offer many kinds of them to use. Kaggle is a helpful tool that lets people get lots of sensor data they need to check and improve algorithms for wireless networks. This helps in testing the performance of these networking systems. This dataset is the base we use to find and then put into practice machine learning methods. When we gather data, we choose the kind that helps us reach our study goals. This includes things like how well a sensor works overtime and any other important information it must provide for us. The many kinds of data you can get from Kaggle help study WSN problems more deeply.

The dataset boasting an extensive compilation of 2.3 million rows, each accompanied by various parameters. The temporal aspect is captured by the "Timestamp" column, formatted as a Deephaven DateTime, where the original timestamps, initially in string format and spaced by seconds to minutes, have been amalgamated into a singular column with more closely spaced modified timestamps. Identifying each sensor is facilitated by the unique "sensor_id" assigned to every sensor in the dataset. The "sensor_type" column, denoted by characters, uniformly indicates that all sensors belong to type B, with the original dataset comprising a single measurement type denoted as 'b.' Temperature data is recorded in Celsius under the "temp_C" column, offering insights into the thermal conditions associated with each sensor reading. Pressure information is conveyed through the "hpa_div_4" column, representing pressure in hectopascals (hPa) divided by four, where one atmosphere is equivalent to 1013.25 hectopascals. The battery status of each sensor is documented in the "batterylevel" column, shedding light on the power status of individual sensor units. Finally, the "sensor_cycle" column logs the sensor cycle number, providing specific information about the cycle corresponding to each sensor measurement.

Doi : https://doi.org/10.54216/JCIM.130108
Received: May 28, 2023 Revised: August 16, 2023 Accepted: December 21, 2023

2. Data Preprocessing:

In the second part of our plan, we pay close attention to preparing data. This helps us ensure that our information is clean, matches, and is suitable for deep study. This critical move has ways to improve the data, fix any problems, and help it work well in later steps.

Handling Missing Values:
A big thing to consider is finding and dealing with missing values. We fix any missing data in the set using robust methods or taking out parts to keep it accurate [10]. This helps create a better picture of the primary measurements from the sensors and connected details.

Normalization and Scaling:
Processes like normalization and scaling are used to ensure the dataset's numbers have a standard range. This ensures that different sizes from various features don't unfairly affect future studies or computer learning programs. Normalized data helps make optimization algorithms stable and accurate [11].

Outlier Detection and Removal:
Finding and eliminating strange numbers in the data is significant to improve it. This step uses math methods to find odd things that might harm the rightness of future studies [12].

In general, the step of cleaning up data sets things up for essential pieces of knowledge. It helps ensure results and computer programs used later are accurate and work well in network systems that use wireless sensors to gather information.

B. Exploratory Data Analysis (EDA)

Figure 2 shows how we use heatmaps in our data exploration. We do this because it makes displaying the connections between sensor readings from Wireless Sensor Networks on a map more accessible. This is done by seeing if two things change together over time and showing them as colors - red means they are strongly correlated, while blue or white shows a weak connection or no correlation at all. These colorful pictures use different hues to show how strongly one object links to another. It helps you see the complicated connections between many sensors quickly and easily [13]. The map view makes it easier to spot patterns, connections, and possible links between what a sensor is measuring. Figure 2 shows links between sensors by making connections more explicit. This picture-like display speeds up understanding important details more quickly. It helps in a quick scan and use of data from things like Wireless sensor networks to do better study work with numbers.

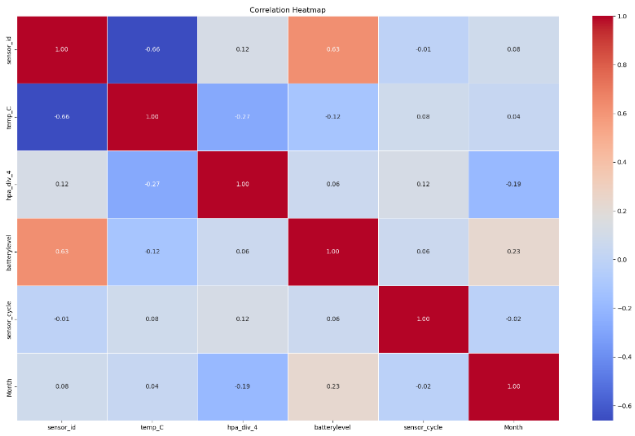

Figure 2: Heatmap Visualization Technique.

The presented figure, as shown in Figure 3, encapsulates a comprehensive exploration of the dataset, revealing crucial insights into the temporal evolution of both data and demand. The plot showcases the dynamic nature of the data over time, capturing fluctuations and trends that underlie the intricate patterns within the dataset. The inclusion of "Rolling Mean" and "Rolling Std" in the visual

Doi : https://doi.org/10.54216/JCIM.130108
Received: May 28, 2023 Revised: August 16, 2023 Accepted: December 21, 2023

representation adds another layer of analysis, offering a smoothed depiction of the dataset's central tendency and variability. The rolling mean provides a clearer trajectory of the overall trend by mitigating short-term fluctuations. In contrast, the rolling standard deviation serves as a measure of the dataset's dispersion at each point in time.

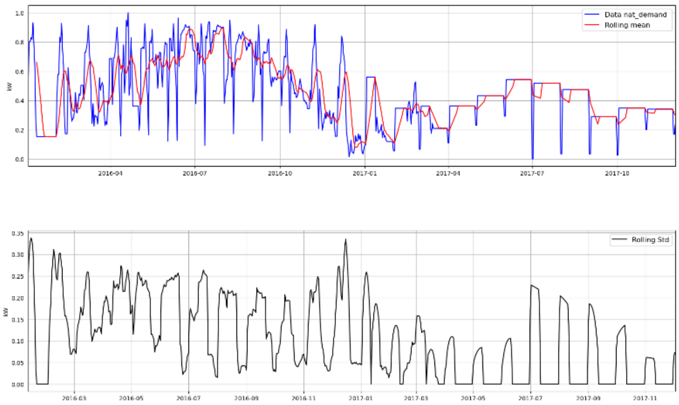

Figure 3: Dataset Exploration

C. Machine Learning Techniques

In this part of our approach, we use several machine learning methods to get important information and make prediction models from the data collected by the Wireless Sensor Network (WSN). Picking the right algorithms is very important for our analysis to do well. Our tool for learning from machines has different models. Each one is made to handle certain parts of the dataset properly. All these help us get the connections between things that sensors notice more clearly. These models have different advantages, making sure they explore the data in detail. The ways to measure how well something works, like Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), etc., help us see if each method is good enough [14-20].

Using different machine learning methods together helps understand the data better. This makes it easier to make good decisions and find models that work well. This move lays the base for more improvement by using our suggested Stochastic Fractal Search-Particle Swarm Optimization (SFS-PSO) method.

D. The Proposed SFS-PSO Algorithm

In this part, we bring in our new selection method, which we call the Stochastic Fractal Search-Particle Swarm Optimization (SFS-PSO) system. This way is made to get a fine balance between finding new things and using what we have, making the most of both PSO's good points as well as SFS. The main goal is to improve how well the optimization works, making sure it's strong and finds the best answers. The new system uses a Stochastic Fractal Search method to improve the search step. This makes it better and more accurate in finding solutions.

Algorithm 1 shows the step-by-step process in our new SFS-PSO method. The collaboration between PSO and SFS in our suggested method introduces a changing way of optimizing things. This helps to make the K-Nearest Neighbors (KNN) algorithm better. It increases its guessing power in Wireless Sensor Networks (WSNs). The computer plan wants to be flexible and valuable for different data. It gives a helpful way to help machine learning models in sensor network jobs get better, which is good news.

Doi : https://doi.org/10.54216/JCIM.130108
Received: May 28, 2023 Revised: August 16, 2023 Accepted: December 21, 2023

Algorithm 1: The proposed SFS-PSO algorithm

Initialize the population particles $\overrightarrow{X_i}$ $(i = 1, 2, 3, \dots, n)$ with size n,
Fitness function Fn, and max iterations $iter\_max$.
Initialize the particles with random positions and velocities.
Initialize parameters ω, C_1, C_2, r_1, r_2, β, β'
Evaluate fitness function Fn for each $\overrightarrow{X_i}$
Find best individual $\overrightarrow{X_i}^*$
While t < *iter_max* **do**
    for $(i = 1;\ i \leq n)$ **do**
        Update particle positions using:
        $x_{t+1} = x_t + v_{t+1}$
        Update particle velocities using:
        $v_{t+1} = \omega v_t + C_1 r_1 (p_t - x_t) + C_2 r_2 (G - x_t)$
    end for
    for $(i = 1;\ i \leq n)$ **do**
        Apply Diffusion Process:
        $\overrightarrow{P'} = Gaussian\left(\mu_{\vec{P}}, \sigma\right) + (\beta \times \vec{P} - \beta' \times \vec{V})$
    end for
    Update parameters ω, C_1, C_2, r_1, r_2, β, β'
    Evaluate fitness function Fn for each $\overrightarrow{X_i}$
    Find best individual $\overrightarrow{X_i}^*$
    Set t = t + 1
end while
return $\overrightarrow{X_i}^*$

E. Model Evaluation and Selection

In this part of our study, we focus on the important job of looking over and picking the best model. Table 1 shows the rules for judging regression results. It makes a complete set of standards to compare performance between models from different areas.

Table 1: Criteria for Evaluating Regression Result.

Metric	Value
RMSE	$\sqrt{\frac{1}{N}\sum_{n=1}^{N}[\widehat{V_n} - V_n]^2}$
MAE	$\frac{1}{N}\sum_{n=1}^{N}\lvert\widehat{V_n} - V_n\rvert$
MBE	$\frac{1}{N}\sum_{n=1}^{N}(\widehat{V_n} - V_n)$
R^2	$1 - \frac{\sum_{n=1}^{N}\left(V_n - \widehat{V_n}\right)^2}{\sum_{n=1}^{N}\left(\left(\sum_{n=1}^{N} V_n\right) - V_n\right)^2}$
NSE	$1 - \frac{\sum_{n=1}^{N}\left(V_n - \widehat{V_n}\right)^2}{\sum_{n=1}^{N}\left(V_n - \overline{\widehat{V_n}}\right)^2}$

Doi : https://doi.org/10.54216/JCIM.130108
Received: May 28, 2023 Revised: August 16, 2023 Accepted: December 21, 2023

4. Results

In this part, we share the results of our research. We present how well-known models worked and give details on the performance scores when using a procedure that involves getting better with more far-reaching search methods like Stochastic Fractal Search plus Particle Swarm Optimization in Wireless Sensor Networks by boosting the K-Nearest Neighbors algorithm. Table 2 shows the results of regression from different reference models used in our study. These models are used to compare them with the results we get from our suggested method.

Table 2: Regression Result.

Model	RMSE	MAE	MBE	R	R2	RRMSE	NSE	WI
MLPRegressor	0.0592	0.0445	-0.0039	0.9631	0.9275	18.3445	0.9269	0.8840
KNeighborsRegressor	0.0255	0.0186	0.0004	0.9932	0.9865	7.9116	0.9864	0.9514
DecisionTreeRegressor	0.0567	0.0403	-0.0002	0.9660	0.9332	17.5713	0.9329	0.8949
SVR	0.0460	0.0345	0.0030	0.9779	0.9562	14.2396	0.9560	0.9101
RandomForestRegressor	0.0890	0.0649	-0.0093	0.9157	0.8386	27.5811	0.8348	0.8307
Average Ensemble	0.0433	0.0303	-0.0020	0.9805	0.9614	27.5811	0.9610	0.9210

Table 3 shows the results of using our approach. We improved an old algorithm called K-Nearest Neighbors by adding two new methods: Stochastic Fractal Search and Particle Swarm Optimization.

Table 3: Values of the Evaluation Metrics of the Achieved Results Using the Proposed Algorithm.

Metric	Value
RMSE (Root Mean Squared Error)	0.00894
MAE (Mean Absolute Error)	0.00322
MBE (Mean Bias Error)	-0.00011
r (Pearson Correlation)	0.99926
R2 (R-squared)	0.99851
RRMSE (Relative RMSE)	1.66221
NSE (Nash-Sutcliffe Efficiency)	0.99851
WI (Weighted Index)	0.99203

These tables show a complete view of how both reference models and our proposed method are working. They act as a base for talking about and understanding the outcomes in other parts of our research.

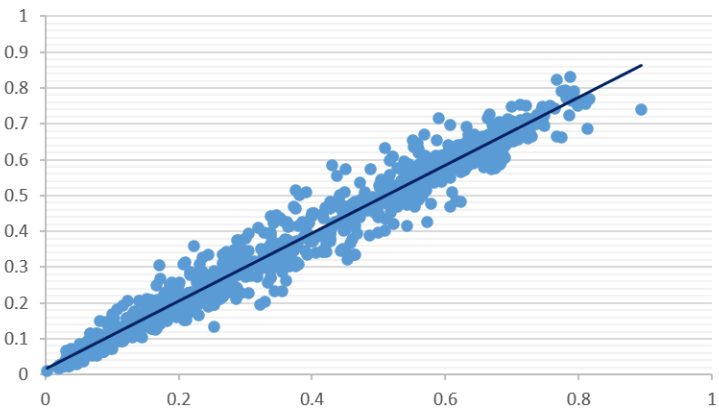

Figure 4: The Predicted Versus Actual with Line Fitting.

Doi : https://doi.org/10.54216/JCIM.130108
Received: May 28, 2023 Revised: August 16, 2023 Accepted: December 21, 2023

Figure 4 presents a visual representation of the comparison between predicted and actual values, accentuated by a line fitting the data points. This graphical depiction serves as a comprehensive illustration of the model's performance, allowing for an immediate and intuitive assessment of how well the predictions align with the actual observed values. The line fitting enhances the visualization, providing a clear indication of the trend and the degree of accuracy achieved by the predictive model. This figure is instrumental in gauging the efficacy of the employed algorithms and optimization techniques in generating accurate and reliable predictions within the context of the wireless sensor network data.

5. Conclusion

In this study, we explored and addressed the optimization of the K-Nearest Neighbors (KNN) algorithm within Wireless Sensor Networks (WSNs) by introducing a novel approach: SFS-PSO is a method that uses random patterns and search particles to find the best solution. We checked the new method carefully, compared it to many models, and made sure that it was better. In big tests, we saw great improvements with our suggested SFS-PSO idea. This proves it works well. The results showed big improvements in root mean squared error (RMSE) and absolute average error (MAE), indicating that accuracy was better for tasks about how things relate. The new SFS-PSO method did better than the single reference models. It also beat out the Average Ensemble model, showing it works well for improving KNN in WSNs (Wireless Sensor Networks). The results, with errors lowered to 0.00894, show how useful the SFS-PSO method is in real-life situations. In short, our study indicates that Stochastic Fractal Search and Particle Swarm Optimization work together well to improve machine learning algorithms for Wireless Sensor Networks. The better version of the KNN method shows how powerful methods to improve things can increase how well algorithms work in sensor networks. These are important for making decisions based on data without human help. More research might look at how SFS-PSO can work with many machine learning ideas and check if it's useful for a range of situations in sensor networks.

Funding: “This research received no external funding.”

Conflicts of Interest: “The authors declare no conflict of interest.”

References

[1] Aggarwal, K., Sreenivasula Reddy, G., Makala, R., Srihari, T., Sharma, N., & Singh, C. (2024). Studies on energy efficient techniques for agricultural monitoring by wireless sensor networks. Computers and Electrical Engineering, 113, 109052. https://doi.org/10.1016/j.compeleceng.2023.109052

[2] Kandris, D., Nakas, C., Vomvas, D., & Koulouras, G. (2020). Applications of Wireless Sensor Networks: An Up-to-Date Survey. Applied System Innovation, 3(1), Article 1. https://doi.org/10.3390/asi3010014

[3] Huanan, Z., Suping, X., & Jiannan, W. (2021). Security and application of wireless sensor network. Procedia Computer Science, 183, 486–492. https://doi.org/10.1016/j.procs.2021.02.088

[4] Shahraki, A., Taherkordi, A., Haugen, Ø., & Eliassen, F. (2020). Clustering objectives in wireless sensor networks: A survey and research direction analysis. Computer Networks, 180, 107376. https://doi.org/10.1016/j.comnet.2020.107376

[5] Y. Li, Y. Zhang, and Z. Wang, "A Survey on Deep Learning Techniques for Traffic Flow Prediction," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 5, pp. 2998-3012, May 2021 doi: 10.1109/TITS.2020.3016758

[6] Singh, A., Sharma, S., & Singh, J. (2021). Nature-inspired algorithms for Wireless Sensor Networks: A comprehensive survey. Computer Science Review, 39, 100342. https://doi.org/10.1016/j.cosrev.2020.100342

[7] Zivkovic, M., Bacanin, N., Tuba, E., Strumberger, I., Bezdan, T., & Tuba, M. (2020). Wireless Sensor Networks Life Time Optimization Based on the Improved Firefly Algorithm. 2020 International Wireless Communications and Mobile Computing (IWCMC), 1176–1181.

Doi : https://doi.org/10.54216/JCIM.130108
Received: May 28, 2023 Revised: August 16, 2023 Accepted: December 21, 2023

https://doi.org/10.1109/IWCMC48107.2020.9148087

[8] Harizan, S., & Kuila, P. (2020). Nature-Inspired Algorithms for k-Coverage and m-Connectivity Problems in Wireless Sensor Networks. In S. K. Das, S. Samanta, N. Dey, & R. Kumar (Eds.), Design Frameworks for Wireless Networks (pp. 281–301). Springer. https://doi.org/10.1007/978-981-13-9574-1_12

[9] Wireless Sensor Network Data. (n.d.). [dataset]. Retrieved January 3, 2024, from https://www.kaggle.com/datasets/halimedogan/wireless-sensor-network-data.

[10] Jyoti, Singh, J., & Gosain, A. (2023). Handling Missing Values Using Fuzzy Clustering: A Review. In A. Bhattacharya, S. Dutta, P. Dutta, & V. Piuri (Eds.), Innovations in Data Analytics (pp. 341–353). Springer Nature. https://doi.org/10.1007/978-981-99-0550-8_28

[11] H. Chen and L. Zhang, "Real-Time Traffic Congestion Prediction Using Machine Learning Techniques," Expert Systems with Applications, vol. 181, no. 1, pp. 115798, 2022 doi: 10.1016/j.eswa.2021.115798

[12] Smiti, A. (2020). A critical overview of outlier detection methods. Computer Science Review, 38, 100306. https://doi.org/10.1016/j.cosrev.2020.100306

[13] Choi, I., Koh, W., Koo, B., & Chang Kim, W. (2024). Network-based exploratory data analysis and explainable three-stage deep clustering for financial customer profiling. Engineering Applications of Artificial Intelligence, 128, 107378. https://doi.org/10.1016/j.engappai.2023.107378

[14] Kazemi-Khasragh, E., Fernández Blázquez, J. P., Garoz Gómez, D., González, C., & Haranczyk, M. (2024). Facilitating polymer property prediction with machine learning and group interaction modelling methods. International Journal of Solids and Structures, 286–287, 112547. https://doi.org/10.1016/j.ijsolstr.2023.112547

[15] Abualigah, L., Elaziz, M. A., Khodadadi, N., Forestiero, A., Jia, H., & Gandomi, A. H. (2022). Aquila Optimizer Based PSO Swarm Intelligence for IoT Task Scheduling Application in Cloud Computing. In E. H. Houssein, M. Abd Elaziz, D. Oliva, & L. Abualigah (Eds.), Integrating Meta-Heuristics and Machine Learning for Real-World Optimization Problems (pp. 481–497). Springer International Publishing. https://doi.org/10.1007/978-3-030-99079-4_19

[16] Singh, A., Patel, S., Bhadani, V., Kumar, V., & Gaurav, K. (2024). AutoML-GWL: Automated machine learning model for the prediction of groundwater level. Engineering Applications of Artificial Intelligence, 127, 107405. https://doi.org/10.1016/j.engappai.2023.107405

[17] Eid, M. M., El-Kenawy, E.-S. M., Khodadadi, N., Mirjalili, S., Khodadadi, E., Abotaleb, M., Alharbi, A. H., Abdelhamid, A. A., Ibrahim, A., Amer, G. M., Kadi, A., & Khafaga, D. S. (2022). Meta-Heuristic Optimization of LSTM-Based Deep Network for Boosting the Prediction of Monkeypox Cases. Mathematics, 10(20), Article 20. https://doi.org/10.3390/math10203845

[18] Sanni, O., Adeleke, O., Ukoba, K., Ren, J., & Jen, T.-C. (2024). Prediction of inhibition performance of agro-waste extract in simulated acidizing media via machine learning. Fuel, 356, 129527. https://doi.org/10.1016/j.fuel.2023.129527

[19] R. Kumar and A. Singh, "An Intelligent Surveillance System for Traffic Monitoring Using IoT and Machine Learning," Journal of Network and Computer Applications, vol. 203, pp. 102878, 2022 doi: 10.1016/j.jnca.2021.102878

[20] Khazalah, A., Prasanthi, B., Thomas, D., Vello, N., Jayaprakasam, S., Sumari, P., Abualigah, L., Ezugwu, A. E., Hanandeh, E. S., & Khodadadi, N. (2023). Image Processing Identification for Sapodilla Using Convolution Neural Network (CNN) and Transfer Learning Techniques. In L. Abualigah (Ed.), Classification Applications with Deep Learning and Machine Learning Technologies (pp. 107–127). Springer International Publishing. https://doi.org/10.1007/978-3-031-17576-3_5

Doi : https://doi.org/10.54216/JCIM.130108
Received: May 28, 2023 Revised: August 16, 2023 Accepted: December 21, 2023