P-LIME: PSO-based Local Interpretable Model-Agnostic Explanations Approach for More Reliable AI Explanations

Khaled Bechoua, Ahmed Taki Eddine DIB, Hichem Haouassi, and Hichem Rahab

Original scientific article

***Abstract*—The rapidly expanding fields of artificial intelligence and machine learning see growing use of intelligent models for predictive tasks and decision support in areas like healthcare, autonomous transportation, and finance. However, the absence of transparency in these models makes them difficult for end-users to understand, limiting their trust and adoption. To address this challenge, techniques such as Local Interpretable Model-agnostic Explanations (LIME) have been developed to provide local model-agnostic explanations independently of the model's internal structure. Despite its effectiveness, LIME suffers from limitations in the random generation of perturbed instances, which can lead to unstable and low-quality explanations. To handle these drawbacks, this work introduces a PSO-based Local Interpretable Model-Agnostic Explanations (P-LIME) approach. P-LIME leverages the Particle Swarm Optimization (PSO) metaheuristic to intelligently generate perturbed instances, thereby improving the quality and stability of the explanations. Experimental results demonstrate that the proposed approach significantly outperforms the original LIME method, offering a more reliable and interpretable framework for understanding complex artificial intelligence models. This advancement contributes to the broader goal of enhancing transparency and trust in artificial intelligence systems.**

***Index terms*—Explainable artificial intelligence, interpretable machine learning, Local Interpretable Model-agnostic Explanations (LIME), Particle Swarm Optimization, Model-agnostic Explanations, metaheuristics, Trust in artificial intelligence.**

I. Introduction

With artificial intelligence (AI) and machine learning continuing to advance rapidly, intelligent models are increasingly being used for prediction and decision-making support across a broad range of fields such as healthcare, autonomous transportation, finance, and other domains [1]. However, as AI technologies advance, models like Support Vector Machines (SVM), Random Forest, and Deep Neural Networks (DNN) are commonly termed 'black-box models' [2]. While they deliver high performance, their lack of transparency makes them challenging for end-users to comprehend. This presents major problems, particularly in contexts where automated decisions require clear justification.

To address the need for model interpretability, several approaches have been developed to make machine learning models more understandable. Among these techniques is Local Interpretable Model-agnostic Explanations (LIME) [3], introduced by Ribeiro, Guestrin, and Singh in 2016. The purpose of LIME is to generate local, model-agnostic explanations for classification models without requiring access to their internal structure. The approach works by first creating a set of Perturbed Instances (PIs) near the input instance. A simplified model for interpretation purposes, such as linear regression, is then trained on these PIs. This simpler model serves to locally approximate the original model's decision logic within that localized region [4]. The process enables the determination of each feature's influence on the prediction, thereby making the model's behavior more understandable.

Although LIME provides a solution for interpretability, it has a limitation in its method of generating PIs. Specifically, PIs are created randomly in the vicinity of the instance requiring explanation. However, this random approach can result in low-quality PIs, which reduces the efficiency of the generated linear regression model and, consequently, compromises the stability of the local explanations.

Swarm intelligence refers to the collective behavior displayed by decentralized and self-organizing systems, often modeled after natural groups like ant colonies and bird flocks. In these systems, simple agents interact locally with each other and their environment, resulting in the formulation of complex global patterns and problem-solving capabilities. A key principle of swarm intelligence is the trade-off between exploration and exploitation, which enables the efficient search of large solution spaces while refining promising solutions to achieve optimal results. This concept has been widely applied in optimization algorithms, such as Particle Swarm Optimization (PSO) [5], Rat Swarm Optimizer (RSO) [6], and crow search algorithm [7], which mimic the behavior of swarms to solve complex problems in engineering, robotics, and AI. Swarm intelligence (SI) offers a robust and flexible framework for tackling challenges in dynamic and uncertain environments.

Manuscript received May 20, 2025; revised June 11, 2025. Date of publication July 15, 2025. Date of current version July 15, 2025. The associate editor prof. Damir Krstinić has been coordinating the review of this manuscript and approved it for publication

K. Bechoua and A. T. Eddine DIB are with the LIRE laboratory, Software and Information Systems Technologies (TLSI) department, Faculty of New Information and Communication Technologies, Constantine 2 University – Abdelhamid Mehri, Algeria (e-mails: {khaled.bechoua, ahmed.dib}@univ-constantine2.dz).

H. Haouassi and H. Rahab are with the ICOSI Lab, Computer Science Department, Faculty of Science and Technology, Abbas Laghrour University, Khenchela, Algeria (e-mails: {haouassi.hichem, rahab.hichem}@univ-khenchela.dz).

Digital Object Identifier (DOI): 10.24138/jcomss-2025-0070

While SI provides a powerful framework for solving complex optimization problems using decentralized and self-organized behavior [5], its application in high-stakes domains raises concerns about explainability and trust. In fields such as healthcare and finance, where decision-making impacts human lives and regulatory compliance, it is crucial to balance performance with interpretability. Many machine learning models (e.g., deep neural networks and SVM) often operate as "black boxes" because of their mathematical nature. Therefore, integrating swarm intelligence algorithms with explainable AI (XAI) techniques could help bridge the gap between high-performance and model interpretability, which presents a significant research challenge.

To enhance the quality of explanations generated by LIME, we introduce an approach that leverages the Particle Swarm Optimization (PSO) metaheuristic to intelligently generate PIs instead of relying on random generation. PSO is a stochastic optimization method based on population dynamics and is driven by the collective behavior of animal swarms, such as birds and bees searching for food [5]. PSO utilizes simple interactions between individuals to guide the search process effectively.

On the other hand, the random generation of PIs in LIME risks producing data far from the instance to be explained. However, the application of PSO enables the search for PIs close to the sample to be explained, thereby enhancing the quality of explanations. Then, in this work, a PSO-based Local Interpretable Model-Agnostic Explanation (P-LIME) is proposed.

The main contributions of this work are the following:

- Adaptive perturbation method: Instead of the traditional random method, the PSO optimization algorithm searches for optimal PIs in LIME that improve the quality of local model interpretability.
- Enhancing stability: By using the optimizing PIs set, the P-LIME reduces variance in explanations.
- Enhancing fidelity: The PSO search space exploration generates instances that better reflect the data distribution, leading to improved explanation fidelity.

This work is organized as follows: Section II goes on to related work on LIME-based methods and explainable artificial intelligence. The theoretical background of LIME and Particle Swarm Optimization (PSO) is presented in Section III. Section IV presents the proposed P-LIME approach, along with its architecture and implementation. Section V describes the experimental setup, datasets, and evaluation measures. Section VI addresses the outcomes and contrasts with the original LIME technique. Section VII concludes the work and lists prospective study areas at last.

II. Related Work

Today, the explainability of an AI model (i.e., understanding how a decision is made) is a primary metric as much as the model's performance [8]. A good compromise between explainability and performance is increasingly necessary, as it guarantees that AI models remain as effective and transparent as possible. Achieving this compromise is key for making AI models more understandable by end users. Several explanation methods have been introduced in the literature, such as Local Interpretable Model-Agnostic Explanations [3], SHapley Additive exPlanations (SHAP) [9], and Local Rule Based Explanations (LORE) [10].

LIME is the most well-known explanation technique; it creates a set of PIs around a candidate sample x, evaluates the model's responses, and then constructs a simpler interpretable model, generally linear regression, allowing for a local approximation of the decision boundary and an explanation of the prediction. The original LIME method [3] creates PIs around x by applying random changes to feature values, which allows the model behavior to be locally approximated by a simple linear function. However, this random perturbation approach can result in significant instability of explanations and reduce accuracy, as it does not account for the true data distribution or the actual significance of features in a given prediction.

To overcome these limitations, various extensions of LIME have been proposed. Influence-based LIME (ILIME) [11] enhances the generation of PIs by applying influence functions that weight each sample according to its actual effect on the target instance. This improvement results in more stable explanations. Deterministic LIME (DLIME) [12] adopts a clustering-based approach to select representative samples, reducing the variance introduced by random perturbations and providing a better representation of the model's local regions. The Autoencoder LIME (ALIME) method [13] incorporates adversarial sample generation to generate realistic adversarial samples, which takes into account the real data distribution to create more relevant perturbations. LIME SUPervised Partitioning (LIME-SUP) [14] is designed for image classification models; this method divides images into superpixels and randomly masks some of these segments to generate PIs, ensuring better visual stability. K-LIME [15] incorporates kernel functions to weight PI according to their similarity to the target instance, leading to more accurate and stable local approximation.

GraphLIME [16] adopts LIME to graph neural network explanation by modifying node relationships to generate PIs adapted to relational data, ensuring highly accurate explanation but at a high computational cost. Other variations of LIME, such as LIMEtree [17] and LIME-C [18], contribute by using decision trees on perturbed samples or by adapting the perturbation to categorical variables through controlled modifications, respectively. These methods offer varying levels of stability and precision, based on the application domain and the characteristics of the dataset being used.

While these methods have significant improvements, they still apply heuristic or data-driven perturbation methods that may not necessarily optimize the selection of PIs. This results in potential instability or poor representation of the neighborhood decision boundary. To overcome these limitations, we propose in this work the P-LIME, an extension of LIME that uses the PSO metaheuristic to intelligently generate PIs. PSO optimizes the set of PIs to enhance the quality of the local explanation model while maintaining the stability in the explanation.

III. BACKGROUND

In this section, we present the theoretical foundations relevant to our proposed method. We begin by introducing the LIME technique, which serves as the base framework for model-agnostic explanations. Then, we provide an overview of particle swarm optimization (PSO), the metaheuristic algorithm we employ to improve the perturbation generation process.

A.Local Interpretable Model-Agnostic Explanations (LIME)

LIME represents a widely used approach to generate local explanations for black-box classification models. It functions by creating a simpler, interpretable model (typically linear) that locally mimics the behavior of the complex model around a specific instance prediction. The procedure followed by LIME begins by generating random samples close to the instance being explained. The samples are then passed through the black-box model to obtain their predicted outputs, forming a new dataset for further analysis. In the third step, each sample is then assigned a weight that reflects its importance, according to its Euclidean distance to the original instance, imitating its importance in the dataset. Finally, LIME trains the linear surrogate model on the weighted dataset and identifies the most significant features influencing the prediction of the instance under analysis [3].

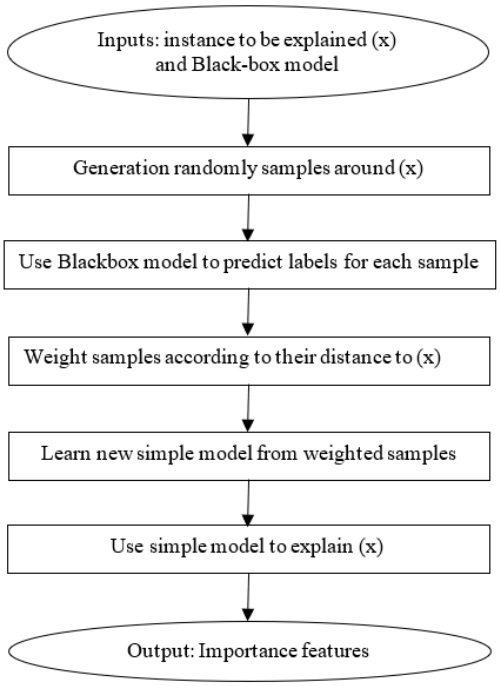

Fig.1. Flowchart of LIME

B.Particle Swarm Optimization (PSO)

The ability of birds to fly synchronously within a flock demonstrates a social behavior that inspired researchers Kennedy and Eberhardt to develop a simple and powerful algorithm for optimization tasks called Particle Swarm Optimization (PSO) [5, 20]. In a PSO, particles are positioned in a multidimensional space, and their locations are considered potential solutions for improvement. Each particle is evaluated by the objective function, also known as the fitness function. The fitness function plays an important role in guiding the PSO algorithm toward the optimal solution by measuring the quality of the solution at each step until the desired quality is achieved. The particle moves through this space influenced by its best local location and the best global location of the swarm particles. This simple behavior of particles contributes to discovering optimal solutions for the multidimensional search space.

The steps of the PSO algorithm and its mathematical modeling are presented as follows:

Step 1: PSO initialization: PSO starts by assigning random initial positions to a population of particles within the defined search space, each assigned a random initial speed.

The population (particles) of PSO can be mathematically represented using a matrix as presented in Equation (1).

$$X_{i,d} = lb_d + r * (ub_d - lb_d) \qquad (1)$$

Where $X_{i,d}$ is the value of the d^{th} variable in the i^{th} particle, r is a random number in the interval [0,1], lb and ub are the lower bound and upper bound of the d^{th} variable, respectively.

Step 2: Fitness assessment: Use the fitness function to evaluate each particle's current position. This determines the best local for individual particles and the best global for the entire swarm.

Step 3: Update position: Calculate the particle's new velocity using the following equation:

$$\mathrm{v_{i,d}} = \mathrm{v_{i,d}} + c_1 r_1 \left(\mathrm{p\_best_{i,d}} - \mathrm{x_{i,d}}\right) + c_2 r_2 \left(\mathrm{g\_best_{d}} - \mathrm{x_{i,d}}\right) \quad (2)$$

Where $v_{i,d}$ is the i^{th} particle's velocity in the $\mathrm{d^{th}}$ dimension, c_1 and c_2 are the cognitive and social coefficients. $\mathrm{r_1}$ and $\mathrm{r_2}$ are independent random values sampled from [0, 1], $p\_best_{i,d}$ represents the best position of the $\mathrm{i^{th}}$ particle, and g_best_d represents the global best position of all the swarm. The positions are then updated using the following equation.

$$\mathrm{x_{i,d}} = \mathrm{x_{i,d}} + \mathrm{v_{i,d}} \qquad (3)$$

where $x_{i,d}$ represents the particle's position in d^{th} dimension.

Step 4: The process iterates by re-evaluating each particle's fitness and adjusting their positions and velocities until a termination criterion is satisfied, such as reaching a predefined iteration limit or achieving a solution with sufficient accuracy.

IV. PROPOSED P-LIME APPROACH

The proposed P-LIME introduced an intelligent method for generating PIs in the vicinity of the instance to be explained. P-LIME aims to improve LIME by boosting the stability and fidelity of the interpretation generated for black-box models.

The P-LIME framework is designed with two main steps, each performing a PI set and the other for simple model learning to provide interpretable decisions. Figure 2 presents a summary of this framework. P-LIME focuses on enhancing the

perturbation process to more accurately mimic how the black-box model functions locally around the interpreted instance.

The process of the proposed approach begins with the input of the instance to be explained, denoted as (x), along with the black-box model and number of PIs to be generated. PIs are then intelligently generated from the instance (x) using the PSO algorithm. Next, these PIs are labeled using the black-box model to form a new dataset. Each instance in this dataset is weighted according to its relevance to the instance (x). After that, a surrogate model, typically a linear regression model, is trained on the new dataset to provide interpretable decisions. Figure 2 illustrates the flowchart of the proposed approach.

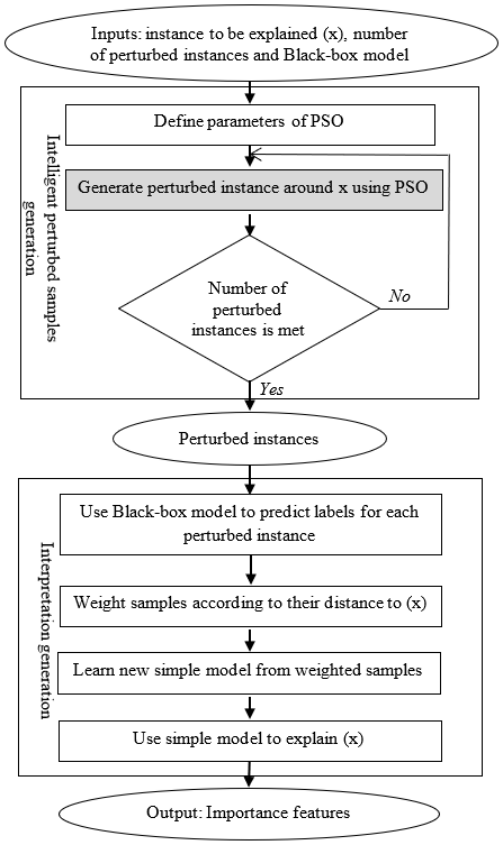

Fig. 2. Flowchart of P-LIME approach

A.PSO-Based Perturbed Instance Generation

The PI generation process in the proposed P-LIME can be formulated as an optimization problem designed to produce a set of PIs in the vicinity of the instance to be explained that effectively approximate the local decision boundary of the black-box model.

Consider an original instance x and a black-box model. The goal is to produce a PIs set X, which preserves the local characteristics of x while ensuring sufficient diversity to capture the model's behavior. The optimization problem can be defined as minimizing the distance between the PIs and the original instance x, subject to constraints on the diversity of the produced instances. Each PI is generated using the PSO optimization algorithm as presented in Figure 2.

In the following, we explain the PSO-based PIs generation process, where P is the population (i.e., set of PIs from x) and T is the maximum number of iterations.

A.1.Parameters Initialization

The process of PSO-based PIs generation begins with the PSO parameters initialization: number of particles (P), number of iterations (T), inertia weight (w), cognitive coefficients **(**c_1), and social coefficients (c_2).

A.2.Particle's Position Representation

Every particle p is associated with a position within the search space, which signifies a candidate solution. Additionally, each particle has a velocity that determines its position updates over time. A position is represented as in Figure 3.

We tend to represent the particle by an n bit vector of real numbers. Each bit represents a feature in the dataset, and the vector values represent a PI (a candidate solution). The position vector is initialized randomly by a PI from the original sample x.

Let x = (0.5, 0.3, 0.1, 0.7, 0.8, 0.1) be a sample, and let p = (0.4, 0.3, 0.1, 0.8, 0.7, 0.2) be a PIs. The position of p is represented as a vector, as shown in Figure 3.

f1	f2	f3	f4	f5	f6
0.4	0.3	0.1	0.8	0.7	0.2

Fig. 3. Particle position encoding

A.3.Fitness Function

The fitness function evaluates the quality of the PI corresponding to each position. The optimal solution (i.e., the best PI) is found at the position that achieves the best value of the fitness function. Equation 4 defines the used fitness function.

$$\text{Fitness (p)} = \sqrt{\sum_{i=1}^{n} (x[i] - p[i])^2} \qquad (4)$$

where $x[i]$ and $p[i]$ represent the values of the i^{th} bit in the position vectors of the original instance and PIs, respectively.

A.4.Updating Positions

Each particle's position in space is updated using Equations (2) and (3).

V. EXPERIMENTS

This section describes the experimental protocol used to evaluate the effectiveness of the proposed P-LIME approach. We begin by presenting the datasets employed in the evaluation, followed by the machine learning models used to generate

predictions. Then, we outline the evaluation metrics adopted to assess explanation quality in terms of fidelity and stability.

A.Data Sets

We evaluated the proposed P-LIME approach using six publicly available datasets retrieved from the UCI Machine Learning Repository [19]. These datasets span a range of binary and multiclass classification tasks and include both categorical and continuous features. Table I summarizes the number of classes, features, and instances for each dataset. The following is a brief description of each:

- Breast Cancer Wisconsin Diagnostic Dataset: This dataset consists of 569 instances, each representing characteristics of cell nuclei from digitized images of breast masses. It includes 30 numerical features and aims to classify tumors as benign or malignant [12].
- German Credit Dataset: This dataset contains 1000 instances with 20 attributes representing personal and financial information. The goal is to classify loan applicants as creditworthy or not, based on historical repayment behavior [10].
- COMPAS Dataset: The COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) dataset contains 7214 records used to assess the risk of re-offending. It includes 15 features that help predict a defendant's likelihood of committing a future offense [10].
- Adult Income Dataset: Also known as the "Census Income" dataset, it includes 48,842 records with 14 demographic features. The classification task is to predict whether a person's annual income exceeds $50K [10].
- Hepatitis Dataset: This medical dataset consists of 155 instances with 20 features describing various clinical attributes of patients. It aims to predict patient survival following a hepatitis diagnosis [12].
- Hypothyroid Dataset: This dataset comprises 7200 records with 21 features. It is a multiclass classification task intended to detect different types of thyroid disorders based on test results [12].

These datasets were selected due to their diversity in data types, sample sizes, and application domains. They serve as a robust benchmark to assess the performance and generalizability of the proposed P-LIME approach compared to the original LIME method.

B.Used Machine Learning Algorithms

B.1.Random Forest (RF)

RF is among the most powerful machine learning algorithms, introduced by Leo Breiman in 2001 [21]. It consists of multiple decision trees. Key strengths of this algorithm are its high accuracy and effectiveness in handling high-dimensional datasets. The decision trees inside the random forest are trained on distinct sets of randomly chosen attributes to reduce the model's susceptibility to the training data [22]. In addition, it identified the important features in decision-making, which makes it an interpretable model [23]. It is used in classification and prediction [24] and has been successfully applied in several areas, including social media analysis, environmental monitoring, medical diagnostics, and the neuroimaging field [25, 26].

Algorithm1 : PSO for PIs generation

Input : Instance to explain x, maximum iteration T, number of particles P, inertia weight (w), cognitive coefficients (c_1), social coefficients (c_2) and feature values.

Output : Global best position Gbest

```
For each particle P Do // Initialize randomly position and velocity of p
    P[i]. current_position ← random_value _within(feature values)
    P[i]. current_velocity ← random_value _within(feature values)
    P[i].p_best ←P[i].position;
    P[i].p_best_value← fitness(particle[i].position) using Equation (4)
End for
Gbest←particle with best p_best value for all particles
Gbest_value← Gbest's p_best value
For i=1 to T do
    For each particle p
        Update P[i]. current_velocity using Equation (2)
        Update  P[i]. current_position using Equation (3)
        P[i].fitness←fitness(particle[i].current_position) using Equation (4)
        if P[i].fitness > P[i].p_best_value then
            P[i].p_best ← P[i]. current_position
            P[i].p_best_value ← P[i].fitness
        End if
    End for
    if P[i].fitness > Gbest_value then
        Gbest← P[i]. current_position;
        Gbest_value← P[i].fitness;
    End if
End for
return Gbest.
```

TABLE I
USED DATASETS DESCRIPTIONS

Dataset	Classes	Features	Instances
Breast Cancer	2	30	569
German	2	20	1000
COMPAS	2	15	7214
Adult	2	14	48842
Hepatitis	2	20	155
Hypothyroid	3	21	7200

B.2.Artificial Neural Network (ANN)

ANNs are considered important algorithms for designing machine learning models, as they simulate the biological neural network that generates human intelligence. The artificial neural network has three interconnected layers, namely the input layer, hidden layers, and output layer, which is concerned with the outputs of the network, whether they are classification or prediction [26, 27]. ANNs are used in many fields [28], such as image recognition, fraud detection, natural language processing, and speech recognition.

B.3.Support Vector Machine (SVM)

The SMV algorithm offers a powerful technique for analyzing complex, non-linear data with many features and few

samples. It is used in supervised machine learning (regression, classification). This algorithm is based on finding the maximum distance (margin) between samples of different classes [29].

C.Evaluation Metrics

The quality of the proposed method's performance is evaluated in terms of two metrics: fidelity and stability.

C.1.Fidelity

Fidelity metric refers to the degree to which an interpretable model accurately represents the predictions or decisions produced by a complex black-box model. High fidelity signifies that the explanation model reflects the original black-box model, enhancing their reliability and trustworthiness. However, there is often a trade-off between fidelity and interpretability, as more complex models may provide highly accurate predictions but are harder to interpret, while simpler models may offer easier explanations but with reduced accuracy. Therefore, finding equilibrium between interpretability and fidelity is key to ensuring that explanations are both understandable and faithful to the model's true behavior [10].

C.2.Stability

In XAI, stability refers to an essential property that guarantees the fidelity of explanations offered to users. It reflects a method's ability to produce coherent and reproducible results when faced with similar or identical inputs. An explanatory method is considered stable if, when it receives the same input data (or slightly perturbed inputs) with similar predictions, it generates explanations with minimal variations [12].

VI. Experimental Results

This section is devoted to presenting experiments evaluating the proposed P-LIME method's performance against the original LIME when applied to machine learning models trained on six different datasets. More specifically, we aim to answer these two key questions: (1) Can the proposed P-LIME effectively generate PIs? (2) Does the generated set of PIs enhance the quality of explanation?

A.PSO Parameters Tuning

The performance of metaheuristic algorithms is influenced by multiple factors, such as the selected parameters: *P, T, w,* c_1, and c_2. Therefore, in the first step of our experiments, we focus on parameter tuning of PSO in the proposed P-LIME approach.

The parameters *w*, c_1, and c_2 are set as defined in the original paper of PSO [5], while the parameters *P* and *T* are studied. In this work we evaluate combinations of three values of *P* (10, 50, 100) and three values of *T* (10, 50, 100) on the benchmark dataset German Credit and the Random Forest model, based on its balanced complexity, moderate dimensionality, and frequent use in explainability benchmarking tasks. As illustrated in Table II. For each combination, the fidelity of P-LIME is measured to assess the quality of the generated explanation.

Table II
Combinations Of Parameters P And T

Parameter combination	Fidelity (%)
P=10; T=10	88.35
P=10; T=50	89.21
P=10; T=100	89.80
P=50; T=10	89.69
P=50; T=50	87.91
P=50; T=100	87.23
P=100; T=10	88.46
P=100; T=50	86.83
P=100; T=100	87.36

From Table II, the fidelity of P-LIME varies depending on the chosen values of P and T. The highest fidelity (89.80%) is obtained in the case of P=10 and T=100, suggesting that a smaller number of particles with a higher number of iterations leads to better results. However, in the cases that P=100 or P=100 and T=50 or T=100, the fidelity drops, indicating that increasing the number of particles may introduce unnecessary diversity in the population, leading to less stable perturbations and lower fidelity. Therefore, for the subsequent experiments, the selected parameters are P=10 and T=100.

B.Evaluation Of Generated PIs

The quality of PIs generated by XAI methods is a key factor in enhancing the model's interpretability [11, 12]. The perturbation method should maintain a balance between diversity and proximity to the original instance, which improves the relevance of generated perturbations. Local fidelity relies heavily on how "close" these instances are to the original data point, as explanations derived from distant samples risk capturing non-local, and thus less relevant, model behavior.

In this subsection, we evaluate the PIs set generated by P-LIME in comparison to the DLIME method and the original LIME method. Specifically, we evaluate the average distance between the PIs and the original instance across different datasets. A lower average distance indicates that the generated instances remain closer to the original instance, potentially leading to more stable explanations. Table III reports the average Euclidean distance between the original instance and the generated perturbations across six benchmark datasets for LIME, DLIME, and P-LIME.

P-LIME consistently yielded the shortest average distances across all datasets, underscoring its capacity to concentrate explanation creation to the immediate vicinity of the instance under analysis. In the German dataset, P-LIME attained an average distance of 2127.49, whereas LIME recorded 4283.78 and DLIME 5017.87. In the Adult dataset, the average distance decreased to 271.10 with P-LIME, in contrast to 718.15 with DLIME and a significantly higher 1344.55 for LIME. These findings indicate that P-LIME's optimization mechanism

effectively focuses the perturbation space in a more significant and interpretable locality.

Although P-LIME beats DLIME, in certain datasets it often generates more localized perturbations than LIME. For example, the average PI distance in the Breast Cancer dataset for DLIME is 424.91, far less than the 948.30 noted for LIME. This is most likely the result of DLIME's more context-aware sampling strategy by using clustering methods to limit perturbation generating inside a given cluster. But as Hypothyroid shows, where LIME yields somewhat less average distances (182.48) than DLIME (257.76), this benefit is not consistent across all datasets.

Reflecting the unstructured character of its random sampling technique, LIME has the worst average distances among the three techniques in practically all datasets. Sometimes the created perturbations span more than three to five times the distance generated by P-LIME. For example, P-LIME's average PI distance is just 4.63, while LIME's in the COMPAS dataset is 56.38. As seen in earlier parts of the evaluation, these far-off disturbances impair the representativeness of the local linear surrogate model and help to produce weaker integrity and stability.

Table III highlights P-LIME's fundamental advantage: it may create perturbations that are rather local to the instance under explanation. This ensures that interpretability maintains its connection to the original input's decision context, thereby enhancing the quality of the explanation. Furthermore, it is noteworthy that there is variation in distances between datasets, which implies that the perturbation behavior of data distribution and feature scaling has an effect. Still, P-LIME maintains the lowest perturbation distances in every scenario and adapts effectively over datasets, hence strengthening its generalizability and resilience.

TABLE III
AVERAGE DISTANCE OF PIS GENERATED BY LIME AND P-LIME.

Datasets	Average distance		
	DLIME	LIME	P-LIME
Breast Cancer	426.9059	948.3038	265.9545
German	5017.8666	4283.7817	2127.4907
COMPAS	36.8496	56.3828	4.6250
Adult	718.1529	1344.5476	271.1038
Hypothyroid	257.7629	182.4844	108.4367
Hepatitis	54.3574	63.2894	47.7756

C.P-LIME Evaluation and Comparison

To evaluate the efficacy of the suggested P-LIME method against the original LIME, six datasets (Breast Cancer, German, COMPAS, Adult, Hypothyroid, and Hepatitis) and three machine learning algorithms (Random Forest, Neural Networks, and SVM) have been used to learn models to be explained. The performance of the explanation methods is measured using two metrics: stability and fidelity. The experimental result is presented in Tables IV and V.

Explainability depends critically on stability, that is, the repeatability of explanations over several runs for the same input. When used often in the same instance. Table IVpresents the result in terms of the stability metric achieved by P-LIME, DLIME, and LIME.

The results presented in Table IV demonstrate that P-LIME achieves perfect stability (100%) in all cases. The remarkable stability of P-LIME is mainly due to the implementation of PSO for producing neighborhood instances. P-LIME, in contrast to LIME and DLIME, incorporates randomness control by establishing a fixed random seed utilized in PSO. This guarantees that the stochastic elements of the algorithm, specifically the particle movements inside the search space, maintain a consistent course upon each repetition of the explanation for the identical instance. Consequently, the neighborhood generation process is entirely replicable, and the explanations remain consistent throughout iterations.

In contrast, LIME depends on uncontrolled random sampling, resulting in considerable diversity in its interpretations. Table IV indicates that LIME's stability scores are significantly lower, varying from 16.67% to 72.73%, contingent upon the dataset and classifier utilized. This instability arises from LIME's stochastic creation of perturbations, which can fluctuate significantly with each execution, resulting in different local models and explanations.

While DLIME also achieves perfect stability (100%), this is mostly due to its deterministic clustering technique, which generates neighborhood samples within preset data clusters. However, this deterministic structure limits its flexibility and reactivity to the unique properties of each given instance. Unlike P-LIME, which employs PSO to dynamically generate instance-specific neighborhoods through guided exploration, DLIME applies the same static clustering bounds regardless of local decision complexity. As a result, DLIME may overlook slight differences in the model's behavior around certain cases. In contrast, P-LIME combines adaptive sampling with controlled randomness, enabling it to tune the neighborhood to each situation while preserving full stability throughout repeated runs.

Fidelity refers to the extent to which an explanation model accurately mimics the predictions of the underlying black-box model within a local neighborhood of the instance being explained. High fidelity implies that the surrogate model’s output closely matches that of the original classifier, thereby making its explanations more trustworthy and representative of the actual decision boundary. Table V presents a comparative analysis of the fidelity achieved by LIME, DLIME, and P-LIME across various datasets and classifiers.

Results in the Table V amply demonstrate P-LIME's constant performance in terms of fidelity above LIME and DLIME. For

TABLE IV
STABILITY OF P-LIME VS LIME VS DLIME

Dataset	Random Forest			Artificial Neural Networks			SVM		
	LIME	DLIME	P-LIME	LIME	DLIME	P-LIME	LIME	DLIME	P-LIME
Breast Cancer	40.00%	100%	100%	25.00%	100%	100%	72.73%	100%	100%
German	61.54%	100%	100%	57.14%	100%	100%	72.73%	100%	100%
COMPAS	72.73%	100%	100%	81.82%	100%	100%	54.55%	100%	100%
Adult	66.67%	100%	100%	58.33%	100%	100%	16.67	100%	100%
Hypothyroid	21.74%	100%	100%	40.00%	100%	100%	27.78%	100%	100%
Hepatitis	69.23%	100%	100%	72.73%	100%	100%	81.82%	100%	100%

TABLE V
FIDELITY OF P-LIME VS. LIME VS DLIME

Dataset	Random Forest			Artificial Neural Networks			SVM		
	LIME	DLIME	P-LIME	LIME	DLIME	P-LIME	LIME	DLIME	P-LIME
Breast Cancer	77.50%	75.36%	98.52%	76.38%	79.50%	86.06%	80.36%	76.60%	85.14%
German	91.76%	93.32%	99.60%	74.68%	73.43%	91.66%	96.80%	95.89%	99.96%
COMPAS	92.90%	83.30%	100%	83.56%	83.30%	99.98%	90.14%	84.32%	99.84%
Adult	96.10%	96.12%	99.22%	94.66%	93.33%	99.60%	97.98%	96.22%	99.84%
hypothyroid	99.70%	99.70%	100%	88.38%	96.85%	99.98%	87.80%	97.14%	99.88%
Hepatitis	95.92%	80.00%	100%	86.60%	90.00%	100%	100%	100%	100%

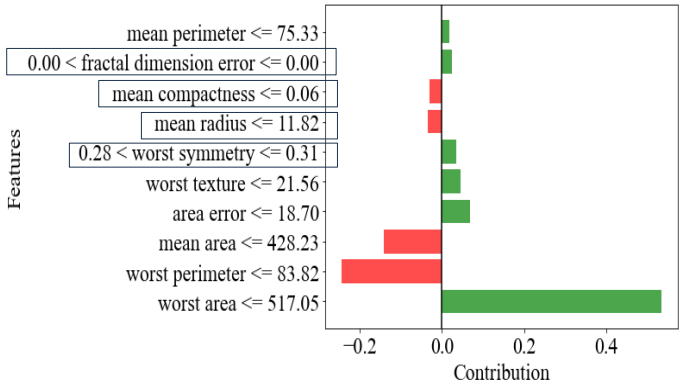

Fig. 4. Iteration 1: Explanations generated with LIME

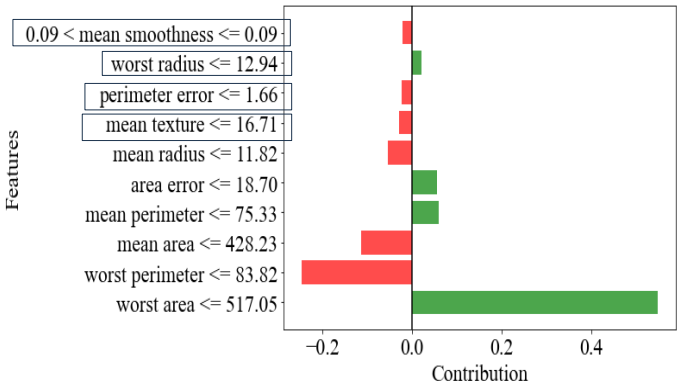

Fig. 5. Iteration 2: Explanations generated with LIME

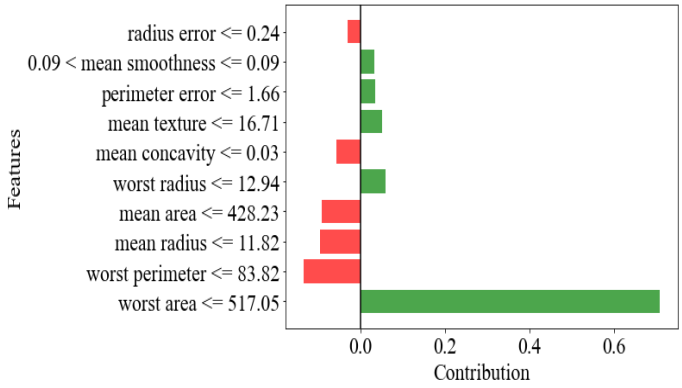

Fig. 6. Iteration 1: Explanations generated with P-LIME

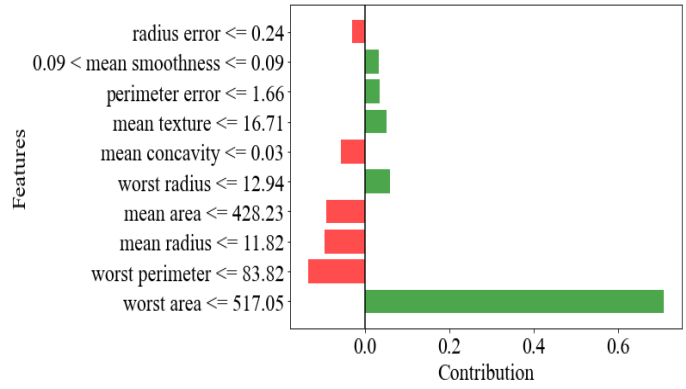

Fig. 7. Iteration 2: Explanations generated with P-LIME

Fig. 8. Jaccard distances for 10 generated explanations for the artificial neural network on the Breast Cancer dataset (LIME).

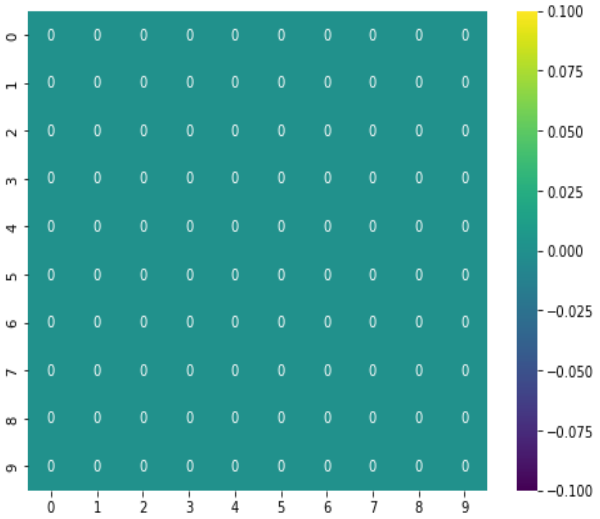

Fig. 9. Jaccard distances for 10 generated explanations for the artificial neural network on the Breast Cancer dataset (P-LIME).

every dataset and for every classification method random forest, artificial neural networks, and support vector machines P-LIME produces the best fidelity scores. For the Breast Cancer dataset, for instance, P-LIME achieves 98.52% fidelity with Random Forests, much exceeding LIME (77.50%) and DLIME (75.36%). Likewise, P-LIME clearly shows its robustness on the German dataset by reaching 99.60%, 91.66%, and 99.96% fidelity with the three respective classifiers.

Although DLIME occasionally beats LIME, its performance is clearly less consistent than that of P-LIME. For the Hypothyroid dataset with artificial neural networks, for example, DLIME achieves 96.85% fidelity while LIME achieves 88.38%. In other situations, such as the German dataset with SVMs, DLIME's fidelity (95.89%) lags rather behind LIME (96.80%). This discrepancy implies that occasionally the clustering-based local sampling of DLIME can improve approximation but may also reduce integrity when clusters do not fit the real decision boundary.

LIME yields moderate fidelity scores across datasets, validating its efficacy as a baseline approach. Nevertheless, it is evidently surpassed by both DLIME and particularly P-LIME in the majority of scenarios. In certain high-fidelity contexts, such as the adult dataset utilizing Random Forests (96.10%) or SVMs (97.98%), LIME demonstrates satisfactory performance. Nonetheless, the improvements presented by P-LIME are substantial: 99.22% and 99.84%, respectively. These findings underscore that although LIME is beneficial, its dependence on random sampling of perturbations constrains its capacity to accurately model the local behavior of intricate classifiers.

The persistent fidelity improvements realized by P-LIME result directly from its optimization-driven perturbation strategy, designed to produce more informative and locally pertinent samples. P-LIME employs PSO as an optimization technique to strategically direct sample generation, in contrast to LIME's uniform sampling and DLIME's cluster-based selection. This yields a more accurate representation of the actual decision function in the local vicinity, evidenced by its nearly flawless fidelity in datasets like COMPAS (100%, 99.98%, 99.84%) and Hepatitis (100% across all classifiers).

Figures 4, 5, 6, and 7 present a comparative visualization of the local feature contributions generated by the two explanation methods, LIME and P-LIME, over two different iterations, where the x-axis represents the contribution of each feature to the model prediction and the y-axis lists the input features.

Figures 4 and 5 illustrate the explanations generated by LIME during the first and second iterations, respectively. These charts reveal a high variance in both selected features and their contribution values. When comparing the explanations across the two iterations, we notice that LIME often selects different features or changes the contribution of similar features, which shows instability in explanation. In contrast, Figures 6 and 7 show the explanations produced by P-LIME for the same two iterations. Here the selected features and their contribution remain almost identical, indicating high stability of P-LIME. This comparison clearly highlights that P-LIME offers stronger stability across iterations than LIME. The instability observed in LIME is primarily due to its random perturbation strategy, whereas P-LIME leverages PSO to intelligently generate PIs, leading to more reliable and reproducible explanations.

To evaluate the dissimilarity between explanations generated over 10 iterations, we used the Jaccard coefficient (*Jdistance*), calculated using Equation (5). If the Jaccard coefficient is equal to 1, this indicates that the two explanations (i.e., sets of selected features) are identical, meaning the explanation is stable. Conversely, a coefficient equal to 0 indicates that the two explanations are completely different, indicating instability.

$$Jdistance = 1 - j(e_i - e_j) \qquad (5)$$

where e_i and e_j represent two sets of explanations produced at iterations i and j, respectively.

The Jaccard coefficient results of explanations generated over 10 iterations are presented in Figures 8 and 9. The figures show the Jdistance between the explanation for an input sample in each iteration i and the explanation in other iterations j, where

$i,j \in [0,9]$. A zero value indicates no variation between the explanations in iterations *i* and *j*, while a non-zero value represents the degree of dissimilarity between them. In Figure 8, we observe that the Jdistance values are zero for P-LIME, indicating that the explanations generated by P-LIME are stable across all iterations. In contrast, Figure 9 shows that the Jdistance for LIME contains significant values, highlighting that the explanations produced by LIME are unstable and vary across iterations.

VII. Conclusion

This paper introduced P-LIME as an enhancement to the standard LIME technique. We aimed to tackle common issues with LIME, namely the stability of its explanations and how faithfully they represent the underlying black-box model's local behavior. The key difference lies in how perturbed instances (PIs) are generated: where LIME uses random sampling, P-LIME employs the Particle Swarm Optimization (PSO) algorithm for a more guided approach. This allows the local surrogate model to better learn the nuances of the black-box model's decision boundary in the area of interest, ultimately yielding more stable and trustworthy explanations. By controlling the random values in the PSO algorithm, P-LIME generates the same PIs for a given input instance, thereby ensuring the reproducibility of explanations across different executions.

The experimental evaluations conducted on several benchmark datasets, including binary and multi-class classification problems, demonstrate that P-LIME consistently produces more stable and higher fidelity explanations than LIME across various machine learning models. As future work, we aim to further explore the integration of other metaheuristic algorithms for perturbation generation and extend the applicability of our framework to non-tabular data such as images and text, where explainability remains a challenge.

References

[1] V. Hassija, V. Chamola, A. Mahapatra, A. Singal, D. Goel, K. Huang, S. Scardapane, I. Spinelli, M. Mahmud, and A. Hussain, "Interpreting black-box models: a review on explainable artificial intelligence," *Cognitive Computation*, vol. 16, pp. 45-74, 2024. doi: 10.1007/s12559-023-10179-8

[2] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint* arXiv:1702.08608, 2017. doi: 10.48550/arXiv.1702.08608

[3] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?" Explaining the predictions of any classifier," *in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135-1144. doi: 10.1145/2939672.293977

[4] D. Maulud and A. M. Abdulazeez, "A review on linear regression comprehensive in machine learning," *Journal of applied science and technology trends*, vol. 1, pp. 140-147, 2020. doi:10.38094/jastt1457

[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," *in Proceedings of ICNN'95-international conference on neural networks*, 1995, pp. 1942-1948.doi: 10.1109/ICNN.1995.488968

[6] H. Rahab, H. Haouassi, M. E. H. Souidi, A. Bakhouche, R. Mahdaoui, and M. Bekhouche, "A modified binary rat swarm optimization algorithm for feature selection in Arabic sentiment analysis," *Arabian Journal for Science and Engineering*, vol. 48, pp. 10125-10152, 2023. doi: 10.1007/s13369-022-07466-1

[7] M. Ledmi, H. Moumen, A. Siam, H. Haouassi, and N. Azizi, "A discrete crow search algorithm for mining quantitative association rules," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 12, pp. 101-124, 2021. doi: 10.4018/IJSIR.2021100106

[8] D. Minh, H. X. Wang, Y. F. Li, and T. N. Nguyen, "Explainable artificial intelligence: a comprehensive review," *Artificial Intelligence Review*, pp. 1-66, 2022. doi: 10.1007/s10462-021-10088-y

[9] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[10] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti, "Local rule-based explanations of black box decision systems," *arXiv preprint* arXiv:1805.10820, 2018. doi: 10.48550/arXiv.1805.10820

[11] R. ElShawi, Y. Sherif, M. Al-Mallah, and S. Sakr, "ILIME: local and global interpretable model-agnostic explainer of black-box decision," in *Advances In Databases And Information Systems: 23rd European Conference, ADBIS 2019, Bled, Slovenia, September 8–11, 2019, Proceedings 23*, 2019, pp. 53-68. doi: 10.1007/978-3-030-28730-6_4

[12] M. R. Zafar and N. M. Khan, "DLIME: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems," *arXiv preprint* arXiv:1906.10263, 2019. doi: 10.48550/arXiv.1906.10263

[13] S. M. Shankaranarayana and D. Runje, "ALIME: Autoencoder based approach for local interpretability," in *Intelligent Data Engineering and Automated Learning–IDEAL 2019: 20th International Conference, Manchester, UK, November 14–16, 2019, Proceedings, Part I 20*, 2019, pp. 454-463. doi: 10.1007/978-3-030-33607-3_49

[14] L. Hu, J. Chen, V. N. Nair, and A. Sudjianto, "Locally interpretable models and effects based on supervised partitioning (LIME-SUP)," *arXiv preprint* arXiv:1806.00663, 2018. doi: 10.48550/arXiv.1806.00663

[15] P. Hall, N. Gill, M. Kurka, and W. Phan, "Machine learning interpretability with h2o driverless ai," *H2O*. ai, 2017.

[16] Q. Huang, M. Yamada, Y. Tian, D. Singh, and Y. Chang, "Graphlime: Local interpretable model explanations for graph neural networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, pp. 6968-6972, 2022. doi: 10.1109/TKDE.2022.3187455.

[17] K. Sokol and P. Flach, "Limetree: Interactively customisable explanations based on local surrogate multi-output regression trees," arXiv, 2020. doi: https://arxiv.org/abs/2005.01427v1

[18] T. Ito, K. Ochiai, and Y. Fukazawa, "C-lime: A consistency-oriented lime for time-series health-risk predictions," in *Knowledge Management and Acquisition for Intelligent Systems: 17th Pacific Rim Knowledge Acquisition Workshop, PKAW 2020, Yokohama, Japan, January 7–8, 2021, Proceedings 17*, 2021, pp. 58-69. doi: 10.1007/978-3-030-69886-7_5

[19] D. Dua and C. Graff, *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Science, 2019. [Online]. Available: http://archive.ics.uci.edu/ml

[20] M. Jain, V. Saihjpal, N. Singh, and S. B. Singh, "An overview of variants and advancements of PSO algorithm," *Applied Sciences*, vol. 12, p. 8392, 2022. doi: 10.3390/app12178392

[21] G. Biau, "Analysis of a random forests model," *The Journal of Machine Learning Research*, vol. 13, pp. 1063-1095, 2012.

[22] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5-32, 2001. doi: 10.1023/A:1010933404324

[23] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern recognition letters*, vol. 31, pp. 2225-2236, 2010. doi: 10.1016/j.patrec.2010.03.014

[24] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, pp. 197-227, 2016. doi: https://doi.org/10.1007/s11749-016-0481-7

[25] E. Scornet, G. Biau, and J.-P. Vert, "Consistency of random forests," 2015. doi: 10.1214/15-AOS1321

[26] W. S. Alaloul and A. H. Qureshi, "Data processing using artificial neural networks," in *Dynamic data assimilation-beating the uncertainties, ed: IntechOpen*, 2020. doi: 10.5772/intechopen.91935

[27] A. Sayal, J. Jha, V. Gupta, A. Gupta, O. Gupta, and M. Memoria, "Neural networks and machine learning," in *2023 IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*, 2023, pp. 58-63. doi: 10.1109/ICCCMLA58983.2023.10346612.

[28] H. Jiang, F. Qin, J. Cao, Y. Peng, and Y. Shao, "Recurrent neural network from adder's perspective: Carry-lookahead RNN," *Neural Networks*, vol. 144, pp. 297-306, 2021. doi: 10.1016/j.neunet.2021.08.032

[29] R. G. Brereton and G. R. Lloyd, "Support vector machines for classification and regression," *Analyst*, vol. 135, pp. 230-267, 2010. doi: 10.1039/B918972F

Khaled Bechoua received his Master's degree in Computer Science from Khenchela University in 2016, and his Engineer degree in Computer Science from Khenchela University in 2009. He is currently a PhD student in LIRE laboratory, Constantine 2 University – Abdelhamid Mehri. His research interests are focused on the explainable Artificial intelligence.

Ahmed Taki Eddine Dib is an Associate Professor and Vice-Rector at Abdelhamid Mehri– Constantine 2 University, Algeria. His research focuses on context-aware systems, multi-agent systems, and bigraphical modeling, with applications in cloud data security and ambient systems. He has published extensively in international journals and conferences and actively contributes to academic committees and doctoral supervision.

Hichem Haouassi received his Ph.D. degree in computer science from the University of Batna, Algeria in 2012, and now he is a full professor in Abbas Laghrour University, Khenchela, Algeria. His main research interests are the Artificial intelligence, Data mining, Metaheuristics and swarm-based optimization, Feature selection, and classification.

Hichem Rahab is currently working as an Associate Professor in the Department of Computer Science at Abbes Laghrour University of Khenchela, Algeria. He obtained his PhD in Computer Science from the University of Constantine 2 in 2020 and his Masters degree in Computer Science from Batna University, Algeria, in 2012. His research interests are focused on Natural Language Processing (NLP) and include machine learning, Misinformation detection and sentiment analysis.