Accurate Classification in Uncertainty Dataset Using Particle Swarm Optimization-trained Radial Basis Function

Nergz Sattar Mohammed[1,*], Hakem Said Beitollahi[1]
[1] Computer Science, Faculty of Science, Soran University, Soran, 44008, Kurdistan Region, Iraq
* Corresponding Author Email: nsm170h@cs.soran.edu.iq

Abstract

Data uncertainty can be generated for a variety of reasons, including measurement error, sampling error, environmental monitoring, sensor networks, and medical diagnostics. The process of mining information from emerging applications such as sensors or location-based services should be handled carefully to prevent erroneous outcomes. Several heuristic techniques and Machine Learning (ML) techniques have been used to classify data in the presence of uncertainty. This paper proposes a novel ML technique by combining the Radial Basis Function (RBF) network with the particle-swarm optimization algorithm. We begin by applying a significant level of uncertainty to a data set and then cleaning up training data and applying data normalization. Next, an RBF network is trained by the optimizer algorithm of particle swarm. Finally, we compare our proposed method with well-known ML techniques namely, k-nearest neighbor (k-NN), Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), and Extreme Gradient Boosting (Xgboost). Our model outperforms previous standard and well-known ML approaches as it has the lowest error rate according to error metrics. The results of the experiments show that our proposed method is better at classifying uncertain data than previous methods based on standard performance metrics.

Keyword: Uncertain data, Machine Learning, Radial Basis Function, Particle Swarm Optimization, IoT Applications.

DOI Number: 10.14704/nq.2022.20.9.NQ440019 **NeuroQuantology 2022; 20(9): 166-179**

Introduction

Uncertain data refers to information that is characterized by the presence of noise, which causes it to diverge from the values that were intended for it or those that were initially stored. In this day, one thing that sets data apart is the degree to which it is uncertain or not valid. There are many factors that can make data uncertain, like measurement error, sampling error, monitoring the environment, market research, sensor networks, and making a medical diagnosis [1]. Also, data uncertainty needs to be dealt with correctly or many data mining applications may lose a lot of their basic performance and efficiency.

Data classification of huge datasets is a critical issue in data mining. It is among the most important data mining techniques to use classifications. In data mining, you can figure out which examples of data belong to which groups or classes[2]. Furthermore, several heuristic and machine learning techniques have been used to classify data in the presence of uncertainty, including decision trees[3], Bayesian belief

networks (BBNs), Fuzzy Sets[4], Rule-Based Classification Algorithm (RBCA)[5], neural networks (NN)[6], the naïve Bayes method[7], and genetic models. In addition, investigators'interests in clustering of uncertain data have recently been brought together. This is limited by the necessity for methods that can deal with unclear data as well as the requirement to use clustering methods on certain types of data[8].

Different characteristics and structures of the Internet of Things (IoT) network create significant uncertainty in data. Data mining isn't as accurate as it could be because of how things change quickly, there aren't enough resources, and the IoT system is getting more complicated. In order to properly characterize the reasons for data uncertainty in IoT, as well as the ways of addressing it, the technology shouldn't be viewed as if it were autonomous[9]. The Internet of Things has a massive amount of uncertain data because of the sensors, measurements, and the data taken from the users is also uncertain data. That's why the IoT is one of the sources of uncertain data. In this work, we classify IoT data, especially healthcare data. The dataset used is a set of people's blood glucose levels (BGL) and their values for some of their most visible body parts. The given dataset is a list of people of different ages who either had diabetes or didn't have it, along with their blood glucose level readings and other basic information about their bodies, like body temperature, systolic blood pressure, heart rate, blood pressure, and so on.

This paper, in this way, treats uncertain data taken from the IoT. The uncertain data is classified using the Radial Basis Function (RBF) network. The RBF network is very efficient and accurate for uncertain data, and then we used the Particle Swarm Optimization (PSO) algorithm to better train the network, improve its performance in terms of precision and accuracy, and bring the network together as quickly as possible. Finally, according to measurements, our method outperforms mainstream and well-known machine learning algorithms. The tests also show that our suggested method finds uncertain data better than other methods, based on standard measures of performance.

The remainder of the paper is structured as follows: Section two describes reviews of some works on methods for classifying uncertain data. In section three, explain the proposed method. In section four, explain the experiment and its discussion. Finally, conclusions are stated in section five.

Related work

Uncertainty in data mining is a rising area of data mining attention. A number of other data mining techniques were also suggested by researchers.

Ran Wang et al.[10], learned how to develop uncertainty reduction in the ELM-Tree based on large data sets. They used information entropy and ambiguous as uncertainty measurements in their decision tree (DT) nodes. When the available split gain ratios tend to fall below a predefined threshold, ELMs are added as leaf nodes in DT induction to fix the problem of over-partitioning in DT induction. Afterward, they utilized parallel computing for five

components of the ELM-Tree model in order to reduce the computational time for a large dataset. Their strategy was shown to have a significant impact on the amount of time it took to solve the problem. However, their approach does not allow for mixed-type properties to be used.

In mobile wireless communication, Qinghua Luo et al. [11]developed utilizing clustering of unknown data streams to estimate dynamic distances. They created a dynamic distance estimation approach based on uncertain intervals of data stream clustering, dubbed the Dynamic Distance Estimation approach utilizing Uncertain Data Stream Clustering (DDEUDSC). The statistical information from RSSI data was utilized to identify the RSSI-D mapping connection for interval data. After that, the data pattern was deemed to be made up of several consecutive cluster centers, and they used their uncertain RSSI data stream clustering technique to estimate the dynamic communication distance. The results of the experiments showed that the suggested strategy was a good way to improve the accuracy of RSSI-D estimates in RSSI data streams that were uncertain and changed over time.

Myriam Bounhas et al. [12]suggested naive probabilistic algorithms for numerical data that were imprecise as well as ambiguous. When there is inconsistency, they analyzed the performance of naïve probabilistic classifiers. Possibilistic analyzers were expanded to handle uncertainty in data representation by altering the numerical values. The Gaussian probabilistic distributions were encoded using the probability distribution of the possible distribution. Uncertainty was divided into two categories: the uncertainty associated with the training class set, which was constructed with a probability distribution above the class labels, and the uncertainty associated with indistinctness, which was only associated with the interval scale. Data on target class uncertainties was taken into account by an intuitionistic classification model that had been adjusted for use. They came up with an algorithm that used inaccurate attribute values. According to the results and conclusions of the paper, possibility classifiers did better in the trial than the current methods.

Kiran and Venugopal [13]advocated the utilization of decision trees to classify ambiguous data. A relevance-based boosting style technique was developed in their study to construct a modified LazyDT ensemble. To address the issue of LazyDT overfitting, the researchers suggested a distance-based pruning strategy. They found that if there were a lot of data tuples, their suggested approach was very successful and could also be utilized to create decision trees using the standard algorithm.

Chunquan Liang et al. [14]devised a quick DT learning method for uncertain data streams comprising positive and unlabeled values. In their research, they used positive and unlabeled samples to classify the data that was uncertain. Using the concept-adapting very fast decision tree (CVFDT) technique, they developed a new algorithm called puuCVFDT. The results of the analysis showed that the puuCVFDT algorithm can deal with idea drift in both

positive and unlabeled instructional strategies.

The classification performance of uncertain data was improved by considering numerous categories, as advocated by Lei Xu and Edward Hung[15]. The limitations of uncertain data categorization, where the data locations were unknown and the probability density functions (pdf) representation were examined in the research. In their study (SUMS), the uncertain K-means method and various subclasses were presented. Subclasses might be created if an item is regarded to belong to more than one class (SUMS). Additionally, they presented a constrained supervised UK-means to avoid overfitting by means of numerous subclasses (BSUMS). The testing findings showed that SUMS and BSUMS outperformed the standard method.

Biao Qin et al. [16]established a Bayesian classifier for data with uncertainty. In their technique, they used statistical and probabilistic concepts to deal with uncertainty. Their technique for calculating Bayes hypothesis conditional probabilities was devised. A Bayesian classification system was designed to handle ambiguous data based on this calculation of conditional probabilities. The experimental findings demonstrated that the A Bayesian classifier method may be a better way to sort uncertain data than a Naive Bayesian classifier (NBC) approach. It was also more reliable than the current extended Naive Bayesian technique.

Sadegh Aminifar[17]suggested that iterative KM is used in one IT2FS defuzzification approach. Iterative KM-type reduction may be a computational barrier for real-time IT2FSs. Several interval type-2 defuzzification algorithms lack a meaningful link between membership function uncertainties and system output owing to poorly specified variables associated with uncertainty. In this work, a novel strategy for IT2FS defuzzification is proposed by reconfiguring interval type-2 fuzzy sets and uncertainty. This research analyzes uncertainty avoidance and IT2FS output provided by centroid or bisection algorithms to places with reduced uncertainty. Uncertainty affects T1FS response. A prototype of IT2FS and a number of studies were done to show that the suggested approach to defuzzification and avoiding uncertainty works.

Proposed Method

In this section, we have to get into significant detail on how we propose to implement our approach. In a hierarchical way, many classifiers process the data one at a time after it has been prepared. Feature processing was accomplished by our novel proposed method. Uncertain data was classified using a particle swarm optimization approach trained on the Radial Basis Function (RBF) network. Lastly, we compare our proposed method to the well-known k-nearest neighbor (k-NN), Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), Extreme Gradient Boosting (Xgboost), and RBF with the SVM. The following subsection provides an overview of the proposed method and these different classifiers.

A. Radial Basis Function Network

*e*ISSN 1303-5150

The Radial Basis Function (RBF) network is a popular choice for solving function approximation issues when it comes to artificial neural networks (ANN). The RBF network is different from other neural networks (NN) because it can get close to a universal approximation and learn at a faster rate [18]. As Figure1 displays, RBF networks are feed-forward neural networks with three layers: an input layer, a hidden layer, and an output layer. The input layer is in charge of assigning a neuron to each input characteristic and sending those features unchanged to the buried layer. The hidden layer makes a nonlinear connection between the input space and the output space, which is a space of interest with more dimensions. Lastly, an output layer is responsible for creating a weighted linear sum [19]. In RBF networks, Gaussian functions are often used to make activation functions (AFs) because this is what has always been done.The output layer uses a sigmoid activation, which is typical for classification tasks like the ones we've been doing. When the RBF networks are trained, the structure of the network is chosen through trial and error to select the network's topology. There are two phases that are involved in determining network parameters:at the beginning of the first pha se, the k-means clustering technique can be used to find out where the nodes of mass centroid are that make up the hidden layer [20]. During the second stage, a linear regression model is used to figure out how much weight each connection should have.

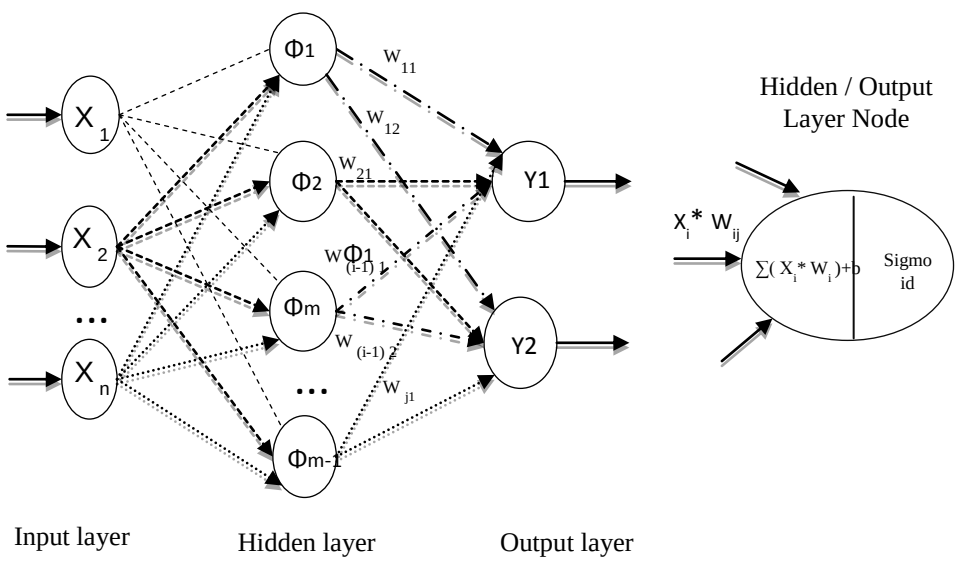


Figure 1: Radial Basis Function (RBF) network architecture.

B. Training RBF network using particle swarm optimization

Particle Swarm Optimization (PSO)was introduced by Eberhart andKennedy[21]in 1995. This is a population-based stochastic method that was inspired by the smart group behavior of some animals, like flocks of birds or schools of fish [22]. The PSO method works by randomly initiating a swarm in the search space, which then attracts the particles to seek out the places in space that have the highest fitness *(p_{best})*. The optimized function determines the adaptive value of each particle. The other particle swarm they use is an average known as (g_{best}). There will be an ideal path for each particle to follow in the search space based on its velocity and distance traveled. The speed of each particle will define both its direction and its distance, and the particles will travel in a manner that best for particle searching inside the region being searched.As illustrated in equation 1, 2

$$v_{(t+1)} = w_t v_t + c_1 \, r_1 \left(p_t^b - x_t\right) + c_2 \, r_2 \left(p^g - x_t\right) \quad (1)$$

$$x_{(t+1)} = x_t + v_{(t+1)} \quad (2)$$

Where t is the number of times the algorithm is run, $w_t$is the weight of each neuron, and r_1 and r_2 are random numbers between 0 and 1, c_1 and c_2 are correction factors, $p_t^b$is the p_{best} position, $p^g$is the g_{best} position, and $v_t$and $x_t$are velocity and position at iteration t , respectively.

Evolutionary algorithms are diverse.A generalized evolutionary technique is something like the Genetic Algorithm (GA), PSO, or the ant colony algorithm. Compared to the GA and other evolutionary algorithms, the following are some of the benefits of the PSO method: (a) the method is easy to understand and use. (b) The algorithm doesn't require a lot of processing power. (c) Because the method is so good at computing, it can be put into place very quickly. A PSO is used in this work to improve the RBF network connection weights and threshold values.The PSO algorithm is used to better train the network, improve its performance in terms of precision and accuracy, and bring the network together as quickly as possible. To be more clear, (1) PSO can locate the global optimal solution with increased rates of success. (2) PSO improves performance (precision and accuracy). (3) PSO results in a quicker rate of convergence, making it more efficient.

In our RBF network, the weights, centers, and spread width of the network's training approach are parameters of a hidden layer should ensure that this is properly calibrated. The weight between the input and the hidden layer is always one. Initially, the weights, centers, and spread width were random for each neuron. Taking the Mean Squared Error (MSE) of all the neurons, which neuron has the lowest MES will be the best neuron and have the best solution. The remaining particle swarms take the average of their weights and centers.This action will keep happening until they reach the stop condition you set or until all the neurons have the lowest MSE.In RBF networks, Gaussian functions are often used to make activation functions

(AFs) because this is what has always been done. The output layer uses a sigmoid activation, which is typical for classification tasks like the ones we've been doing.

The neurons of a hidden layer utilize a Gaussian function as an RBF network. As shown in equation 3.

$$\varphi_j\ (\mathrm{X}) = e^{\left(-\frac{\|\ \mathrm{X} - Xc_j\ \|}{2\sigma_j}\right)} \qquad (3)$$

Where Xc_j ,σ_j , and φ_j (X)are, in order, its well-pointed center, its spread width, and the response of the j^{th}hidden neuron to input X. In this formula, the expression $\|\ \mathrm{X}\ - Xc_j\|$ shows the Euclidean distance between the parts of the input vector X and the center of the GaussianXc_j . This distance can be found in the following way:

$$\|\ \mathrm{X}\ - Xc_j\| \ = \sqrt{\sum_{i=1}^{n}\left(x_i\ - Xc_j\right)} \ (4)$$

Where X = [x_1, x_2,x_3,...., x_n], x_i is feature i of the input layer, and n is the number of attributes in the input layer.

When making RBF networks, it is important to make sure that the nonlinear function of the hidden layer neurons covers all important parts of the input vector space.

The result values of the neurons in the hidden layer are multiplied by the weights associated with each neuron before being sent on to the neurons in the output layer. Each output neuron does an addition of the weighted values. The final um is run through the sigmoid activation function to decide whether the data coming in is uncertain or certain. Equation 5 displays the output function.

$$f(X) = \emptyset\left(\sum_{j=1}^{m}(w_j\ \times\ \varphi_j(\mathrm{X}))\right) \ \ (5)$$

In the equation 5, the symbol ∅ is the sigmoid activation function. The RBF network is used in the proposed method because it is quick in detecting data uncertainty. The RBF network is a wise decision.As Figure 2 illustrated PSO in our RBF network training.

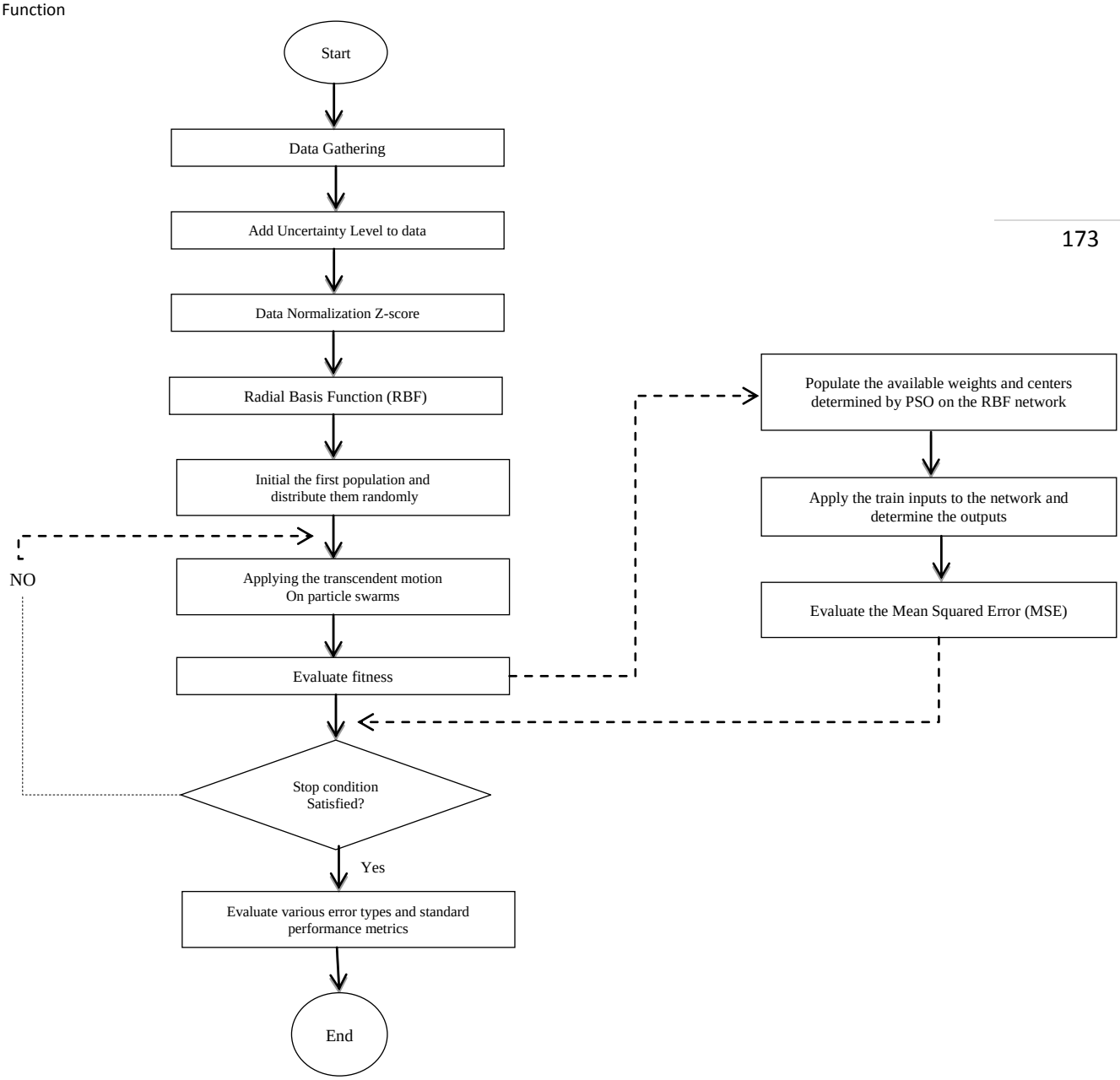


Figure **2**: The flowchart of the proposed technique.

C. Model training by Supervised Learning

K–Nearest Neighbor (KNN): One of the most basic classification techniques is KNN.It saves a copy of every sample and us es a similarity metric to classify the.This similarity metric is generally a distance in the feature space. For calculating distances, Euclidean and Minkowski are the two most commonly used distance measurements.

The Random Forest (RF): Classifier takes into account a number of different decision trees that are constructed concurrently. The development of these trees begins with the selection of a subset of characteristics and samples chosen at random from the initial training data. The final forecast will be based on the category

that receives the most votes from each tree. In this piece, we have determined that there should be a total of 10 decision trees.

Support Vector Machine (SVM): Classification problems with large dimensions are well-suited to the SVM's practical approach to solving them. Classification in SVM is accomplished by the use of a line or a hyperplane. It looks for a hyperplane with the greatest separation between it and the closest data for each category (margin). With a larger margin, a reduced generalization error is possible.

Decision Tree (DT): Is a classification issue that has received a lot of attention in the fields of data mining and artificial intelligence. By asking a series of questions about the things that the data pieces have in common, a decision tree can be used to sort the data. A DT is a hierarchical tree structure that is similar to a flow chart. It consists of three fundamental characteristics: decision nodes connected to features,edges and
branches that match up to the various viabl e attribute values.

Logistic Regression (LR): Is a statistical modeling method that is used in cases when the result is binary. This method is useful because it can be used to predict binary outcomes even when the independent variables are either continuous or discrete.

Extreme Gradient Boosting (Xgboost): is a modular and extremely accurate implementation of gradient boosting, which raises the level of computational power for massively increased tree techniques. It was made mainly to improve the performance of machine learning models and speed up the rate at which computations can be done.

Radial Basis Function (RBF) SVM: Is a technique used for determining the degree of similarity (distance) between two samples. This variant of the SVM makes use of the RBF kernel to do the calculation.

Data pre-processing

In this section, concentrate on data preprocessing. The dataset used is a set of people's blood glucose levels (BGL) and their values for some of their most visible body parts. The given dataset is a list of people of different ages who either had diabetes or didn't have it, along with their blood glucose level readings and other basic information about their bodies, like body temperature, systolic blood pressure, heart rate, blood pressure, and so on. The primary objective of the data collection is to get an understanding of the impact that a person's BGL has on their body. The dataset contains ten features and 16969 records, andit requires some preliminary preprocessing before it can be used. The procedures outlined below were used to prepare a standard dataset for subsequent processing.

1. Using the probability to tuple numeric features in the range [0, 0.2].
2. For the categorical feature, add the blood type attribute to the exit dataset. It is worth mentioning that the entire column consists of eight values of blood groups such as (A+, B+, O+, AB+, A-, B-, O-, and AB-).

3. Normalize the data of numeric features by using the z-score function.
4. Using Euclidean distance to pick the probability of any blood type for the attribute added to the dataset.

Experimental and discussion

This section focuses on the performance evaluation of the proposed research methodology. The dataset used is a set of people's BGL and their values for some of their most visible body parts. The dataset has 16969 entries with numbers that have been labelled and structured. The Python machine learning libraries were used to construct the model. The Numpy, Pandas, and Sklearn libraries are included in the Python packages.The uncertain data is classified using the Radial Basis Function (RBF) network.

The RBF network is very efficient and accurate for uncertain data, and then we used the Particle Swarm Optimization (PSO) algorithm to better train the network, improve its performance in terms of precision and accuracy, and bring the network together as quickly as possible. Then we used seven traditional classification ML models, such as KNN, SVM, RF, LR, DT, Xgboost, and RBF SVM, to compare the results of our proposed method and those algorithms. Furthermore, we split the dataset into two sections, which are the train_set and test_set. The train_set includes 70% of the data, and the test_set holds 30%. The data in the train_set will be used to train the machine before it is put to the test on the test_set. The uncertain data is classified using the RBF network with the PSO approach.

To figure out how well the proposed method works, we use the following measurements and metrics to evaluate the performance of the model: **Confusion Matrix:** is a table displaying the types of correct forecasts and incorrect forecasts.

Table 1: Confusion Matrix.

Predictions	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Accuracy: The number of true positives and true negative values is divided by the number of all predictions.

$$\text{Accuracy} = \frac{\text{Tp}+\text{Tn}}{\text{Tp}+\text{Tn}+\text{Fp}+\text{Fn}} \quad (6)$$

In the present study, the outcome of RBF + PSO was 98%. It boasts the best success rate in comparison to other ML approaches. The proposed method is 4.4 more accurate on average than all of the traditional machine learning-based methods. This is a clear sign that the proposed model is better at classifying uncertain data than traditional ML methods. Figure 3 compares the accuracy of the proposed method to that of traditional ML techniques.

Precision: the number of true positives is divided by the number of true positives and false predictions. It is a measurement of how accurate a classifier is. A high rate of false positives is a sign of low precision.

$$\mathbf{Precision} = \frac{\mathbf{Tp}}{\mathbf{Tp+Fp}} \quad (7)$$

As shown in Table 2, the findings of our work, the precision rate of our proposed method is higher than other ML models in terms of performance. The result of RBF+PSO is 97%. Compared to classic ML techniques that have been used before, our new method improves precision by an average of 4.1%.

Recall: The number of true positives is divided by the number of positive values and false negative values in the test data. It is also known as the True Positive Rate (TPR). It is a measure of how comprehensive a classifier is. A significant number of false negatives suggest a low recall.

$$\mathbf{Recall} = \frac{\mathbf{Tp}}{\mathbf{Tp+Fn}} \quad (8)$$

In this work, the recall rate for RBF + PSO was 95%. According to other ML algorithms, this is the greatest rate. The recall of several ML methods is shown in Figure3. Compared to ML techniques that have been used before, our novel method improves recallby an average of 4.4%.

F-Measure: the sum of a measurement's recall and precision weights.

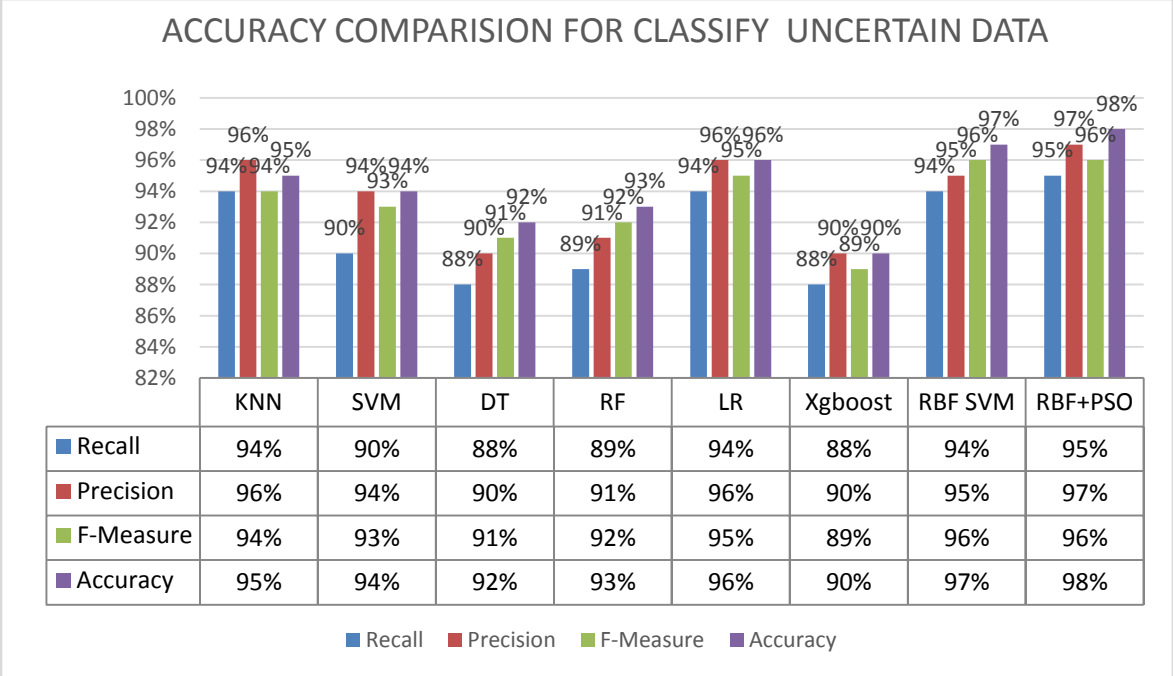
$$\mathbf{F-Measure} = \mathbf{2\ x}\ \frac{\mathbf{Precision\ x\ Recall}}{\mathbf{Precision+\ Recall}} \quad (9)$$

As demonstrated in Table 2, the research findings, our suggested technique, RBF+PSO, has a 96% success rate. It has the highest F-score percentage of other ML algorithms.Train the RBF network using the PSO algorithm to identify uncertain and noisy data. After distributing data to the training and testing sets, we implemented our proposed method and different approaches of ML, as provisory mentioned, to get results and compared which ones were the most effective. As Table 2 shows, the experimental comparison results between our proposed method and traditional ML techniques.

Table 2: Metrics Comparison for Classify Uncertain Data.

Algorithms	Recall	Precision	F-Measure	Accuracy
KNN (K=3)	0.94	0.96	0.94	0.95
SVM	0.90	0.94	0.93	0.94
DT	0.88	0.90	0.91	0.92
RF	0.89	0.91	0.92	0.93
LR	0.94	0.96	0.95	0.96
Xgboost	0.88	0.90	0.89	0.90
RBF SVM	0.94	0.95	0.96	0.97
RBF + PSO	0.95	0.97	0.96	0.98

Finally, we conducted the RBF network using the PSO optimization algorithm. According to measurements, our method outperforms mainstream and well-known machine learning algorithms. The tests also show that our suggested method finds uncertain data better than other methods. Compared to ML techniques that have been used before, our novel method improves accuracy by an average of 4.4%, based on standard measures of performance. As explained in Figure 3.



	KNN	SVM	DT	RF	LR	Xgboost	RBF SVM	RBF+PSO
Recall	94%	90%	88%	89%	94%	88%	94%	95%
Precision	96%	94%	90%	91%	96%	90%	95%	97%
F-Measure	94%	93%	91%	92%	95%	89%	96%	96%
Accuracy	95%	94%	92%	93%	96%	90%	97%	98%

Figure 3: Metric Comparison for Classify Uncertain Data.

Conclusion

This paper proposed a technique based on machine learning to categorize uncertain data. The proposed technique is a hybrid model of a Radial Basis Function (RBF) network and the Particle Swarm Optimization (PSO) algorithm. The RBF network is used for uncertain data classification, while PSO is used to train the RBF network. Several tests were done to look at the proposed method and compare it to the k-NN classifier, the SVM, the DT, the RF, the LR, the Xgboost, and the RBF SVM. In particular, the simulation's results were: accuracy = 98%, precision = 97%, recall = 95%, and F-Measure = 96%. This is a clear sign that the proposed model is better at classifying uncertain data than standard ML methods. The proposed method is 4.4% more accurate on average than all of the ML-based methods.

References

[1] Agrawal, Rakesh, T. Imielinski and A. Swami, *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING,* vol. 5, pp. 5(6): 914-925., 1993.

[2] V. Choudhary and J. Pranita, "Classification: A Decision Tree for Uncertain Data Using CDF, International Journal of Engineering Research and Applications," pp. 3(1):

1501-1506., 2013.

[3] S. Tsang, K. Ben, Y. Y. Kevin, H. Wai-Shing and D. L. Sau, "Decision Trees for Uncertain Data, IEEE Transactions On Knowledge And Data Engineering.," pp. 23(1):64-78, 2011.

[4] Suresh, V. G. and S. ,. O., "classification of uncertain data using fuzzy neural networks, world of computer science and information networks, world of computer science and informationTechnology Journal.," p. 1(4), 2011.

[5] Q. Biao, Y. Xia, Prabhakar and Y. Tu, "A Rule-Based Classification Algorithm for Uncertain Data, in proceedings of IEEE International Conference on.," 2009.

[6] J. Ge and X. Yuni, "UNN: A Neural Network for uncertain data classification," pp. 6118: 449-460., 2010.

[7] J. Ren, D. L. Sau, C. Xianlu, K. Ben, C. Reynold and C. David, "Naive Bayes Classification of Uncertain Data, in proceedings of ninth IEEE International Conference on," 2009.

[8] W. Ngai, K. B., K. C. C., C. R. and Y. Chau, "Efficient clustering of uncertain data,Springer, Heidelberg," pp. 4065: 436-445., 2006.

[9] S. Singh, M. Chris, P. Sunil, S. Rahul and H. Susanne, "Indexing Uncertain Categorical Data, In Proc.of ICDE," pp. 616-625., 2007.

[10] M. Mehta and A. Rakesh, "SLIQ- A Fast Scalable Classifier for Data Mining, In 5th Intl. Conf. on Extending Database Technology.," 1996.

[11] L. Qinghua, X. Yan, J. Li and Y. Peng, "DDEUDSC: A Dynamic Distance Estimation using Uncertain Data Stream Clustering in mobile wireless sensor networks, Journal on Measurement," pp. 55: 423-433, 2014 .

[12] M. Bounhas, G. Mohammad, P. Henri, S. Mathieu and M. Khaled, "Naive possibilistic classifiers for imprecise or uncertain numerical data, Journal on fuzzy sets and systems," pp. 239: 137-156, 2014.

[13] S. Kiran, R. M. Venugopal and P. N. Reddy, "Classification of uncertain data using decision trees, International Journal of Advanced Research in Computer Science and Software Engineering," pp. 3(10): 40-46, 2013.

[14] L. Chunquan, Y. Zhang, P. Shi and Z. Hu, "Learning very fast decision tree from uncertain data streams with positive and unlabeled samples," *Journal on Information Sciences* , pp. 213: 50-67, 2012.

[15] Xu, Lei and H. Edward, "Improving classification accuracy on uncertain data by considering multiple subclasses, Journal on Neuro Computing," pp. 145: 98-107 , 2014.

[16] B. Qina, X. Yuni, W. Shan and D. Xiaoyong, "A novel Bayesian classification for uncertain data, Journal on Knowledge-based System," pp. 24(8): 1151-1158., 2011.

[17] A. Sadegh, "Uncertainty Avoider Interval Type II Defuzzification Method,Mathematical Problems in Engineering," 2020.

[18] C. S. K. Dash, A. K. Behera, S. Dehuri and S.-B. Cho, "Radial basis function neural networks: a topical state-of-the-art," p. 6:33–63, 2016.

[19] R. Wang, D. Li and K. Miao, "Optimized Radial Basis Function Neural Network Based Intelligent Control Algorithm of Unmanned Surface Vehicles," 2020.

[20] K. Ivan, H. Marta, V. Jan, K. Jan and L. Ondrej, "Radial Basis Function Neural Network-Based Modeling of the Dynamic Thermo-Mechanical Response and Damping Behavior of Thermoplastic Elastomer Systems ,polymers," 2019.

[21] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *in Proceedings of the IEEE International Conference on Neural Networks,*, 1995.

[22] E. Assareh, M. ,. Behrang, M. R. Assari and A. Ghanbarzadeh, "Application of PSO (particle swarm optimization) and GA (genetic algorithm) techniques on demand estimation of oil in Iran," p. 5223–5229, 2010.