Kurdistan Journal of Applied Research (KJAR)
Print-ISSN: 2411-7684 | Electronic-ISSN: 2411-7706

Website: Kjar.spu.edu.iq | Email: kjar@spu.edu.iq

A Novel Classification of Uncertain Stream Data using Ant Colony Optimization Based on Radial Basis Function

Tahsin Ali Mohammed Amin
Department of Database Technology
Technical College of Informatics
Sulaimani Polytechnic University
Sulaymaniyah, Iraq
tahsin.ali@spu.edu.iq

Sabah Robitan Mahmood
Department of Information Technology
Technical College of Informatics
Sulaimani Polytechnic University
Sulaymaniyah, Iraq
sabah.robitan@spu.edu.iq

Rebar Dara Mohammed
Department of Database Technology
Technical College of Informatics
Sulaimani Polytechnic University
Sulaymaniyah, Iraq
rebar.dara.m@spu.edu.iq

Pshtiwan Jabar Karim
Department of Computer Science
College of Science
University of Garmian
Kalar, Iraq
pshtiwan.jabar@garmian.edu.krd

Article Info

Volume 7 - Issue 2- December 2022

DOI: 10.24017/Science.2022.2.5

Article history:

Received: 12/09/2022
Accepted: 21/11/2022

Keywords:

Uncertainty data, Machine Learning, Radial Basis Function, Ant Colony Optimization (ACO). Internet of Things (IoT), Applications.

ABSTRACT

There are many potential sources of data uncertainty, such as imperfect measurement or sampling, intrusive environmental monitoring, unreliable sensor networks, and inaccurate medical diagnoses. To avoid unintended results, data mining from new applications like sensors and location-based services needs to be done with care. When attempting to classify data with a high degree of uncertainty, many researchers have turned to heuristic approaches and machine learning (ML) methods. We propose an entirely new ML method in this paper by fusing the Radial Basis Function (RBF) network based on ant colony optimization (ACO). After introducing a large amount of uncertainty into a dataset, we normalize the data and finish training on clean data. The ant colony optimization algorithm is then used to train a recurrent neural network. Finally, we evaluate our proposed method against some of the most popular ML methods, including a k-nearest neighbor, support vector machine, random forest, decision tree, logistic regression, and extreme gradient boosting (Xgboost). Error metrics show that our model significantly outperforms the gold standard and other popular

ML methods. Using industry-standard performance metrics, the results of our experiments show that our proposed method does a better job of classifying uncertain data than other methods.

1. INTRODUCTION

Information that contains noise and thus deviates from the values that were intended for it or those that were initially stored is known as uncertain data. The degree to which information is questionable or invalid today is a distinguishing feature. Measurement error, sampling error, conducting research on the environment, conducting market research, utilizing sensor networks, and making a medical diagnosis are all examples of activities that can introduce uncertainty into data [1].

In addition, many data mining applications risk losing a significant amount of fundamental performance and efficiency if data uncertainty is not properly handled. Classifying massive datasets is a major challenge in the field of data mining. Classifications are used in many different data mining methods. Data mining enables you to discover the categories to which particular data points belong [2]. Decision trees [3] are just one example of the many heuristics and machine-learning techniques that have been applied to the problem of classifying data under conditions of uncertainty. Such methods include BBNs, Fuzzy Sets [4], and other related approaches. The RBCA [5] Rule-Based Classification Algorithm, computer models of the brain called neural networks (NN) [6], the Naive Bayes technique [7], and genetic models.

In addition, investigators' interests in clustering uncertain data have recently been brought together. However, this is constrained by the need for methods that can handle ambiguous data, as well as the necessity of employing clustering techniques on specific data [8]. Significant data uncertainty is caused by the IoT network's unique characteristics and architectures. The fast pace of change, insufficient available resources, and the ever-increasing complexity of the Internet of Things all contribute to less-than-accurate data mining. The Internet of Things (IoT) shouldn't be viewed as if it were autonomous, in order to properly characterize the causes of data uncertainty in IoT and the ways of addressing it [9]. The sensors, measurements, and user data collected by the Internet of Things all introduce a great deal of uncertainty into the mix. Because of this, there is some evidence to suggest that data obtained from the IoT may be unreliable.

Information Technology and Medical Data Classification are the focus of this work. Blood glucose levels (BGL) and values for various external body parts make up the dataset used. The provided data set includes the ages of individuals who did or did not have diabetes, as well as their blood glucose levels and other demographic and health-related data. Uncertain information gleaned from the Internet of Things is dealt with in this paper. We use a Radial Basis Function (RBF) network to categorize the hazy data. In order to train the RBF network, improve its performance in terms of precision and accuracy, and bring the network together as quickly as possible, we used the ant colony optimization algorithm, which is known for its efficiency and accuracy when dealing with uncertain data.

The primary intent of the Ant Colony Strategy (ACS) was to boost ACO's efficiency in solving the recommendation system problem [23]. Its layout is also inspired by a colony of real ants going out on a hunt for food. To sum up, experimental results show that our approach is superior to other, more well-known machine learning methods. Standard measures of performance also indicate that our proposed method outperforms alternatives when searching for uncertain data. As for the rest of the paper, the outline looks like this: The second part provides summaries of relevant literature on approaches to the classification of ambiguous data. Third, describe the strategy you will use to accomplish your goals. Submit your explanation of the experiment and its analysis in Section 4. In the final section, we draw some conclusions.

2. LITERATURE REVIEW

There has been a recent uptick in research on the topic of uncertainty in data mining. Other data mining methods were also proposed by the study's authors.

Based on massive amounts of data, Ran Wang et al. [10] were able to refine ELM Tree's method of reducing uncertainty. In the nodes of their decision tree (DT), they used information entropy and ambiguity to quantify the degree to which a given outcome was uncertain. Over-partitioning in DT induction is remedied by including ELMs as leaf nodes when the available split gain ratios tend to fall below a predefined threshold. To cut down on computation time for a massive dataset, they subsequently used parallel computing for five parts of the ELM-Tree model. The amount of time it took to find a solution was shown to be significantly affected by their strategy. However, they can't use mixed-type properties in their developed method for estimating dynamic distances in mobile wireless communication by clustering unknown data streams. They came up with a method called Dynamic Distance Estimation employing Uncertain Data Stream Clustering, which estimates distances between points in time based on the variability of data streams (DDEUDSC). The RSSI-D mapping connection for interval data was discovered using the statistical information provided by the RSSI data. Later, they assumed the data pattern consisted of a series of adjacent cluster hubs and estimated the dynamic communication distance using their uncertain RSSI data stream clustering method. The experimental results demonstrated that the proposed strategy was an effective means of enhancing the precision of RSSI-D estimates in uncertain and dynamic RSSI data streams. For vague and imprecise numerical data, Myriam Bounhas et al. [12] proposed naive probabilistic algorithms. They looked at how well naive probabilistic classifiers work when there is inconsistency. Expanded probabilistic analyzers can now adjust the numerical values used to represent data in the face of uncertainty.

A probability distribution was used to encode the underlying Gaussian probabilities. Both the uncertainty in the training class set, which was built with a probability distribution over the class labels and the uncertainty in the indistinctness of the interval scale were accounted for. The data on the target class uncertainties were used to fine-tune an intuitionist classification model.

They devised a method that relied on misleading attribute values. Results and discussions in the paper indicate that possibility classifiers outperformed state-of-the-art approaches in the experiment. To better categorize uncertain information, decision trees were recommended by Kiran and Venugopal [13]. In their research, they implemented a relevance-based boosting-style technique to build an improved LazyDT ensemble. Researchers recommended a distance-based pruning strategy to deal with LazyDT overfitting. They discovered that their proposed method was highly effective, and could also be used to create decision trees using the conventional algorithm if there were numerous data tuples. For unlabeled and positive values in uncertain data streams, Chunquan Liang et al. [14] developed a fast DT learning method. They classified the ambiguous data in their study by comparing it to positive and unlabeled samples. They created a novel algorithm called puuCVFDT by modifying the concept-adapting very fast decision tree (CVFDT) method. Data demonstrated that the puuCVFDT algorithm can handle concept drift in both rewarded and unrewarded pedagogical approaches. Lei Xu and Edward Hung [15] argued that taking into account numerous classes would improve the classification performance of uncertain data. The study looked at the drawbacks of using probability density functions (PDF) to represent data when the data's locations were unknown. They introduced the uncertain K-means technique and several different subclasses in their research (SUMS).

If a thing is considered to fit into multiple categories, then subclasses can be made for it (SUMS). Also, to prevent overfitting via numerous subclasses, they introduced constrained supervised UK means (BSUMS). The results of the comparison tests indicated that both SUMS and BSUMS were superior to the gold standard. For unreliable data, Biao Qin et al. [16] developed a Bayesian classifier. They incorporated statistical and probabilistic thinking into their method in order to account for improbability.

They developed a method to determine conditional probabilities based on the Bayes hypothesis. In order to deal with uncertain information, the Bayesian classification scheme was developed. The results of the experiments showed that compared to the Naive Bayesian classifier (NBC), the A Bayesian classifier method performed better when sorting uncertain data. This method was also more trustworthy than the state-of-the-art extended Naive Bayesian approach. One method of defuzzifying IT2FS that makes use of iterative KM was proposed by Sadegh Aminifar [17]. For real-time IT2FSs, the computational cost of iterative KM-type reduction could be a significant bottleneck. Due to poorly specified variables associated with uncertainty, many interval type-2 defuzzification algorithms fail to provide a meaningful connection between membership function uncertainties and system output. In this paper, we propose a new method for defuzzifying IT2FS by rearranging uncertainty and interval type-2 fuzzy sets.

In this study, we examine the use of centroid or bisection algorithms to produce IT2FS results at locations with lower uncertainty, with the goal of avoiding uncertainty wherever possible. The T1FS response is influenced by uncertainty. Numerous studies and a working prototype of IT2FS demonstrate the efficacy of the proposed strategy for defuzzification and reduction of uncertainty.

Table 1: Analysis of properties of methods

Authors	Problem	Method
Qinghua Luo et al. [11]	Uncertain Data Stream Clustering	(DDEUDSC)
Myriam Bounhas et al. [12]	Uncertain Numerical Data Classification	Naive Probabilistic Algorithms
Kiran and Venugopal [13]	Classification of Uncertain Data	Decision Tree
Chunquan Liang et al. [14]	Uncertain Data Stream Classification	puuCVFDT
Lei Xu and Edward Hung [15]	Uncertain Data Classification	Uncertain K-means (UK-means)
Biao Qin et al. [16]	Classification for Uncertain Data	Bayesian Classifier

3. Proposed Method

After the data has been prepared, it is processed by multiple classifiers in a hierarchical fashion. We used our novel approach to successfully process features. Ant colony optimization based on a Radial Basis Function (RBF) network was used to categorize data with some uncertainty. We conclude by contrasting our proposed approach with some well-known alternatives, including the K-nearest neighbor (KNN), the Support Vector Machine, the Random Forest, the Decision Tree, the Logistic Regression, the Extreme Gradient Boosting, and the RBF with the support vector machine (SVM). The following subsection provides an overview of the proposed method and these different classifiers.

3.1. Radial Basis Function Network

When it comes to artificial neural networks, the Radial Basis Function (RBF) network is often used to address function approximation issues in artificial neural networks (ANN). The RBF network stands out among neural networks (NNs) because of its near-universal approximation and rapid learning speed [18]. Fig. 1 depicts a basic RBF network architecture, which consists of an input layer, a hidden layer, and an output layer as part of a feed-forward neural network. Each input feature is given a neuron in the input layer, and the layer sends the features unmodified to the hidden layer. The hidden layer creates a nonlinear map between the input space and the desired higher-dimensional output space. When everything is said and done, it's the job of an output layer to generate a weighted linear sum [19]. Traditionally, activation functions (AFS) in RBF networks have been constructed using Gaussian functions. The sigmoid activation is used in the output layer, which is typical for classification tasks. When RBF networks are trained, their topologies are determined by a process of trial and error.

There are two stages to establishing a network's parameters: in the first stage, the K-means clustering technique can be used to pinpoint the hidden layer's nodes of mass centroid [20]. In the second phase, we use a linear regression model to determine the relative importance of each connection.

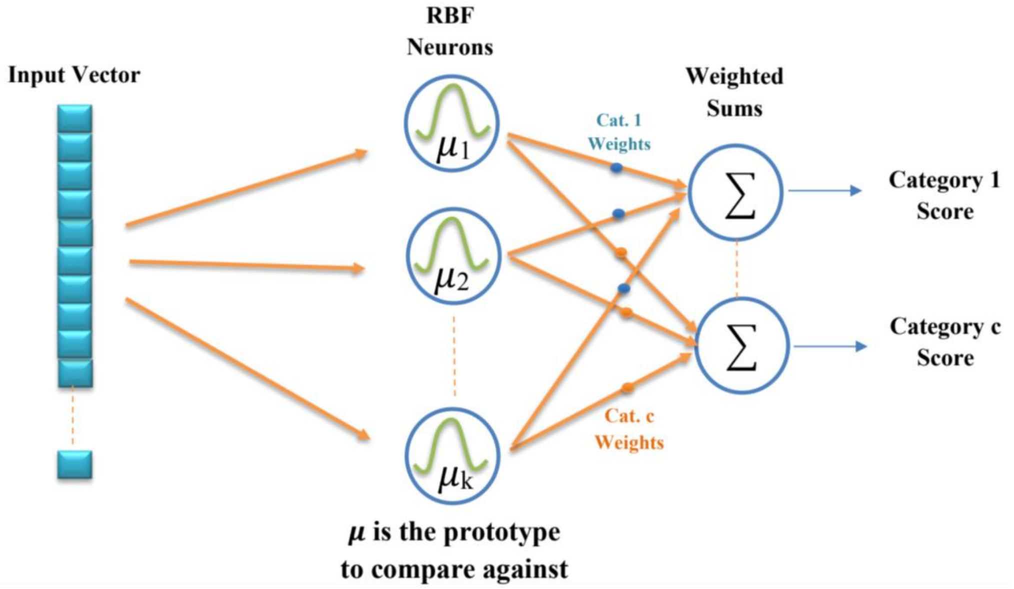


Figure 1: The RBF network architecture is based on the Radial Basis Function.

3.2. Training RBF network using Ant Colony Optimization

It is possible to obtain approximations to complex optimization problems by employing Ant Colony Optimization (ACO), a population-based meta-heuristic. In ACO, artificial ants construct a solution to a combinatorial optimization problem by moving along a construction graph that is both directed and fully connected. First, each instantiated decision variable X_i=v^j_i, X is leading to a total inhibitory input, V^j_i is an input vector of the corners called a solution component and is denoted by C_{ij}. C stands for the complete set of components for any given solution. After that, we define the building graph G_C(V, E) by linking the components C to the vertex set *V* or the edge set *E*, respectively[23].

Each c_{ij} has an associated value of τ_{ij} that represents the pheromone trail it contributes to. (Remember that the pheromone values are generally proportional to the number of iterations of the algorithm, t:τ_{ij}=$\tau_{ij}(t)$.) Different components of the solution can have their probabilities modeled using pheromone values. At various points in the search process, the ACO algorithm uses and updates pheromone values[23].

The ants iteratively create a solution by traversing the building graph from vertices to vertex along its edges, using the data provided by the pheromone values. In addition, the ants leave behind a trace amount $\Delta\tau$ of pheromone on the parts they visit, either at the vertices or along the edges. The efficacy of the solution may determine how much pheromone is released. Subsequent ants use the pheromone data to narrow their search to more fruitful areas[23].

A group of *m* ant colonies create solutions by combining elements from a finite set of available solution components, $C = \{c_{ij}\}, |i = 1, \dots, n, |j = 1, \dots, |D_i|$ The renovation of a solution begins with a non-empty partial solution S^p=∅. After that, a global optimal element is added to the existing partial solution S^p from the set of feasible neighbors at each step of the design process *N (Sp)* ⊆ *C*. Designing and building a solution can be thought of as a route along a construction graph G_C (*V*, *E*). Given a partial solution S^p, $G_{C,}$ the solution construction mechanism identifies the set $N(S^p)$ that can be taken as a solution.

Selecting an element from among those available to complete a solution $N(S^p)$ is performed in a probabilistic fashion during each stage of building. Various ACO variants have slightly different guidelines for the stochastic selection of possible solutions. The ant system [23] has the most well-known rule.

$$F(c\_ij \mid s\wedge p\,) = (\tau\_ij\wedge\alpha \cdot \eta\_ij\wedge\beta)/(\textstyle\sum\_(c\_il \in N(s\wedge p\,))\,\tau\_il\wedge\alpha \cdot \eta\_il\wedge\beta\,), \forall c\_ij \in N(s\wedge p\,) \qquad (1)$$

where τ_{ij} is the pheromone value associated with the component C_{ij} and η_{ij} is the heuristic value. Further, α and β parameters whose values, in the positive real parameters, evaluate the weight given to pheromones as opposed to heuristics.

The purpose of the update to the pheromone values is to boost the pheromone values linked to good solutions and decrease the pheromone values related to bad ones. To do this, you either *i* evaporate all the pheromones, lowering their value, or pheromone value denoted by (*ii*) increase the pheromone to emphasize the ability with a selected group of desirable solutions M_{upd} [23].

$$\tau\_ij \leftarrow (1-\rho)\cdot\tau\_ij + \rho\cdot\textstyle\sum\_(s \in V\_upd\ c\_ij \in s)M(U) \quad (2)$$

where V_{upd} represents the pool of updated solutions, $\rho \in (0,1]$ is measured by a metric known as the evaporation rate, and $F{:}S \rightarrow R^{+}_{0}$ has the form is a function where [23], M: is regularly called the fitness function and U is the chosen set of good solutions.

$$f(s) < f(s') \Rightarrow F(s) \geq F(s'), \forall s \neq s' \in V \qquad (3)$$

F(·) referred to as the fitness function.

The IB- update principle is one illustration of a pheromone optimization method that is employed quite frequently in practice; this principle is an instance that stands for *iteration-best.*

$$M_{upd} \leftarrow arg \max_{s \in S_{iter}} F(s) \qquad (4)$$

In this study, we apply ACO to the RBF network to optimize the weights of the connections and the thresholds. Training the network with the ACO algorithm helps to accelerate convergence, increase accuracy, and improve precision. ACO has a higher success rate than other methods in locating the globally optimal solution (1). That's because (2) ACO boosts efficiency (precision and accuracy). ACO is more effective because (3) it achieves a higher rate of convergence.

The weights, centers, and spread width of our RBF network's training approach are parameters of a hidden layer that should guarantee this is calibrated accurately. Between the input and the hidden layer, the weight is always 1. Each neuron's initial weights, centers, and spread width were completely arbitrary. The best neuron is the one with the lowest Mean Squared Error (MSE), calculated by comparing all the neurons' MSEs. When it comes to the remaining particle swarms, they take a mean center of gravity and mass. This process will continue until either the stop condition you specify is met or all of the neurons have the same minimum squared error (MSE). Traditionally, activation functions (AF) for RBF networks have been derived from Gaussian functions. The sigmoid activation is used in the output layer, which is typical for classification tasks. A hidden layer's neurons use a Gaussian function as a recurrent basis function (RBF) network. Equation 5 shows that this is the case.

$$\varphi_j(X) = e^{\left(-\frac{\|X-Xc_j\|}{2\sigma_j}\right)} \qquad (5)$$

Where , , and φj (X)are, in order, its well-pointed center, its spread width, and the response of the jth hidden neuron to input X. In this formula, the expression ‖ X − Xcj ‖ shows the Euclidean distance between the parts of the input vector X and the center of the Gaussian . This distance can be found in the following way:

$$\| X - Xc\_j \| = \surd(\textstyle\sum\_(i = 1)\hat{}n\,(x\_i - Xc\_j\,)\,) \qquad (6)$$

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

Making sure the nonlinear function of the hidden layer neurons encompasses critical regions of the input vector space is crucial when developing RBF networks.

Before sending it on to the neurons in the output layer, the result values of the neurons in the hidden layer are multiplied by the weights associated with each neuron. Each neuron in the output circuit adds the weighted values. To determine whether the data is uncertain or certain, the sigmoid activation function is applied to the final neuron. The result function is shown in Equation 7.

$$f(X) = \emptyset(\textstyle\sum\_(j = 1)\hat{}m\,(w\_j \times \varphi\_j\ (X))) \qquad (7)$$

The sigmoid activation function is represented by the symbol ∅ in Eq. 7. RBF networks are used in the proposed method due to their speed and accuracy in identifying data uncertainty. It was a good idea to go with the RBF network. Training our RBF network with ACO, as shown in Fig. 2.

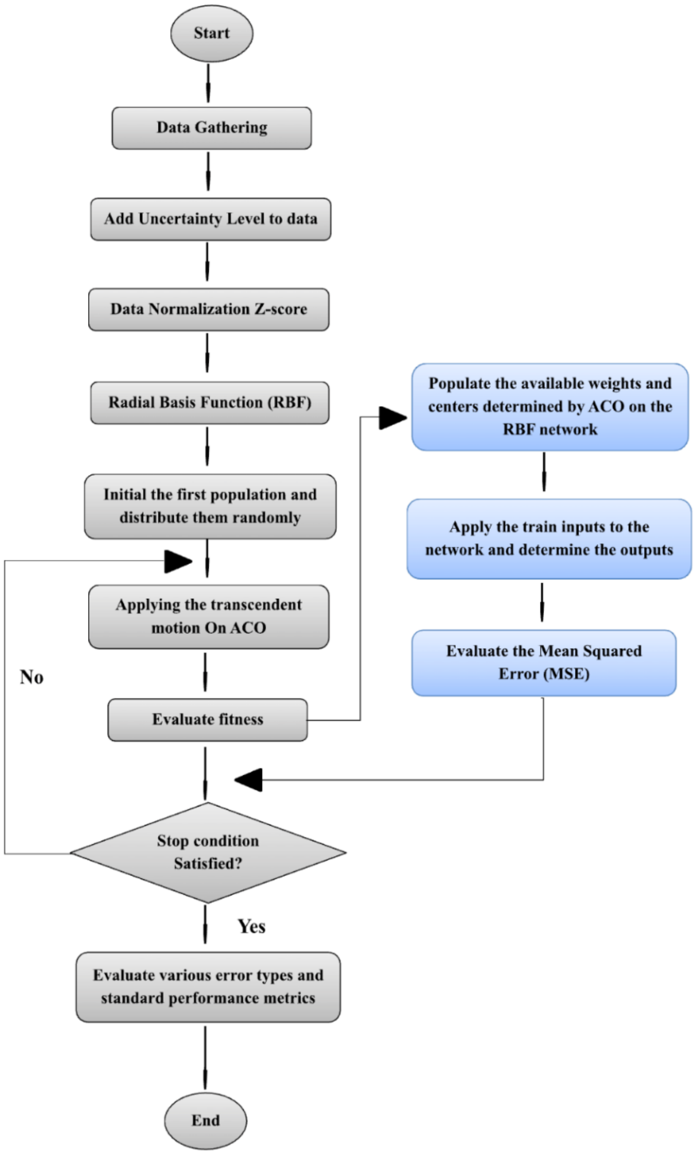

Start
Data Gathering
Add Uncertainty Level to data
Data Normalization Z-score
Radial Basis Function (RBF)
Initial the first population and distribute them randomly
Applying the transcendent motion On ACO
Evaluate fitness
Stop condition Satisfied?
No
Yes
Evaluate various error types and standard performance metrics
End
Populate the available weights and centers determined by ACO on the RBF network
Apply the train inputs to the network and determine the outputs
Evaluate the Mean Squared Error (MSE)

Figure 2: The flowchart of our novel technique

3.3. Model training by Supervised Learning

One of the earliest and most fundamental classification methods was K-Nearest Neighbor (KNN). It makes a record of each sample and then sorts them according to their similarities. Commonly, this measure of similarity takes the form of a distance in the feature space. Both the Euclidean and Minkowski distance systems are widely used for determining physical distances.

When classifying data, the Random Forest (RF): Classifier considers the outcomes of multiple independently developed decision trees. These trees are built by first selecting a random subset of features and samples from the original training data. The most popular answer from each tree will be used to make the final prediction. For the purposes of this article, we have settled on a total of ten decision trees.

The Support Vector Machine (SVM) is a useful tool for solving classification problems of high dimensionality. In support vector machines, classification is achieved by drawing a line or a hyperplane between two points. This method seeks out a hyperplane that is farthest from the closest data in each category (margin). A smaller margin of error in drawing broad conclusions is possible.

Data mining and AI researchers have paid a lot of attention to the classification problem known as the decision tree (DT). A decision tree can organize data by asking questions to determine which characteristics each data point shares. DTs are similar to flowcharts in that they are hierarchical tree diagrams. It has three primary characteristics: decision nodes, features, and edges/branches that correspond to the different possible attribute values.

When the outcome can be represented by only two possible values, statisticians turn to a technique called logistic regression (LR). It is helpful because it can be used to predict binary outcomes regardless of whether the independent variables are continuous or discrete.

Improve the computational power of massively increased tree methods with Extreme Gradient Boosting (Xgboost), a modular and extremely accurate implementation of gradient boosting. For the most part, it was developed to hasten the processing speed and enhance the efficiency of machine learning models.

Radial Basis Function (RBF) SVM: This is a technique used for determining the degree of similarity (distance) between two samples. This iteration of support vector machines employs the RBF kernel to perform the necessary calculations.

3.4. Data pre-processing

In this section, concentrate on data preprocessing. Blood glucose levels (BGL) and values for various external body parts make up the dataset used. The provided data set includes the ages of individuals who did or did not have diabetes, as well as their blood glucose levels and other demographic and health-related data. 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 10 features and, 16969 records, and it requires some preliminary preprocessing before it can be used. The procedures outlined below were used to prepare a standard dataset for subsequent processing.

1. Tuple numerical features in the [0, 0.2] range using the probability.
2. Include the blood-type attribute in the final dataset as the categorical feature. There are eight possible blood group values in this column: (A+, B+, O+, AB+, A-, B-, O-, and AB-).
3. It is recommended to use the z-score function to normalize the numerical feature data.
4. In order to select the probability of any blood type for the new attribute added to the dataset, we will use the Euclidean distance.

4. RESULTS AND DISCUSSION

Analysis of how well the suggested research approach works is the topic of the current subsection. The dataset consists of people's body-geometric-length measurements and their corresponding values. The dataset contains, 16969 labeled and organized numerical entries. The model was built using Python's ML libraries. Python distributions come with supplementary libraries like NumPy, Pandas, and Sklearn. We use a Radial Basis Function (RBF) network to categorize the hazy data, and our experiments are implemented by Python 3.9 on Google Colaboratory pro with 2 x vCPU, Nvidia k80 GPU, and 12 GB RAM.

Since the RBF network performs so well when presented with ambiguous information, we trained it using the Ant Colony Optimization (ACO) algorithm to increase its accuracy and efficiency. Next, we compared the efficacy of our proposed strategy to that of seven well-established classification ML models, including K-Nearest neighbor (KNN), Support vector machine (SVM), Radial function (RF), Logistic regression (LR), decision tree (DT), Xgboost, and radial basis function (RBF) with SVM. More specifically, we create a train set and a test set out of the dataset. Seventy percent of the data is in the train set, while only 30 percent is in the test set. The machine will be trained using the data in the training set, and then put to the test using the data in the test set. The RBF network coupled with the ACO technique is used to categorize unreliable data.

We are using the following success metrics to evaluate the efficacy of the proposed technique: The Confusion Matrix is a table that lists both the correct and incorrect predictions.

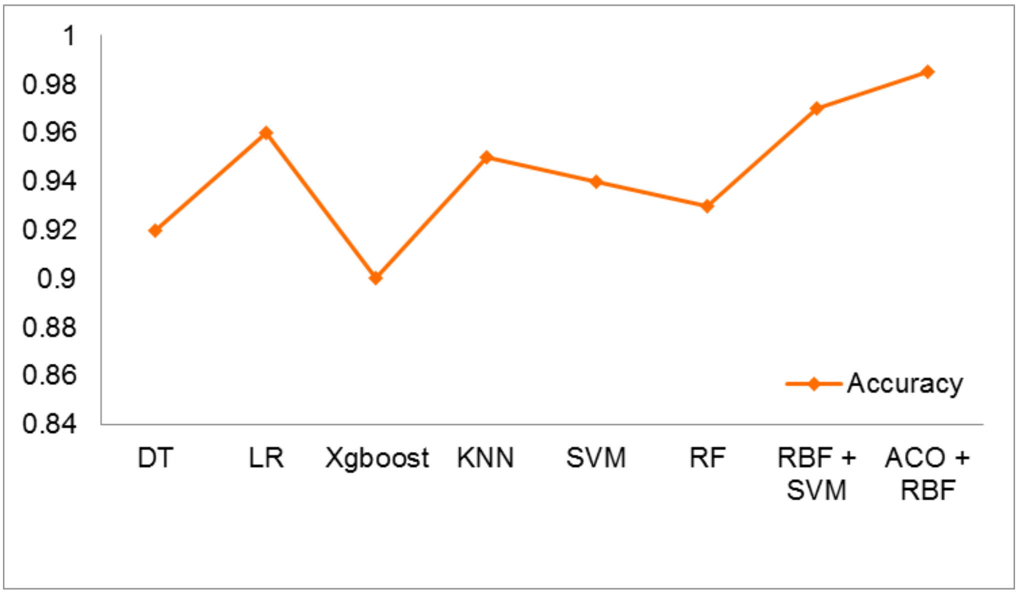
Table 1: Error Matrix

Predicted	**Positive**	**Negative**
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Accuracy: It is calculated by dividing the total amount of predictions by the amount of true positive and true negative values.

$$Accuracy \ = \frac{Tp+Tn}{Tp+Tn+Fp+Fn} \qquad (8)$$

The success evaluation of ACO + RBF was 98.5% in this investigation. When compared to other ML techniques, its success rate is unparalleled. On average, the proposed algorithm is 4.9 times more precise than the state-of-the-art machine learning-based methods. This demonstrates that the proposed model outperforms conventional ML techniques when it comes to classifying data with a high degree of uncertainty. The performance of the proposed method is compared to that of conventional machine learning (ML) algorithms in Figure 3.



Figur 3: The proposed novelty's accuracy in comparison to other algorithms

The accuracy is calculated by dividing the number of correct predictions by the sum of the correct and incorrect ones. It is a metric used to assess the reliability of a classifier. If there are a lot of "yes" results that are actually "no," then the accuracy is low. As shown in figure (4):

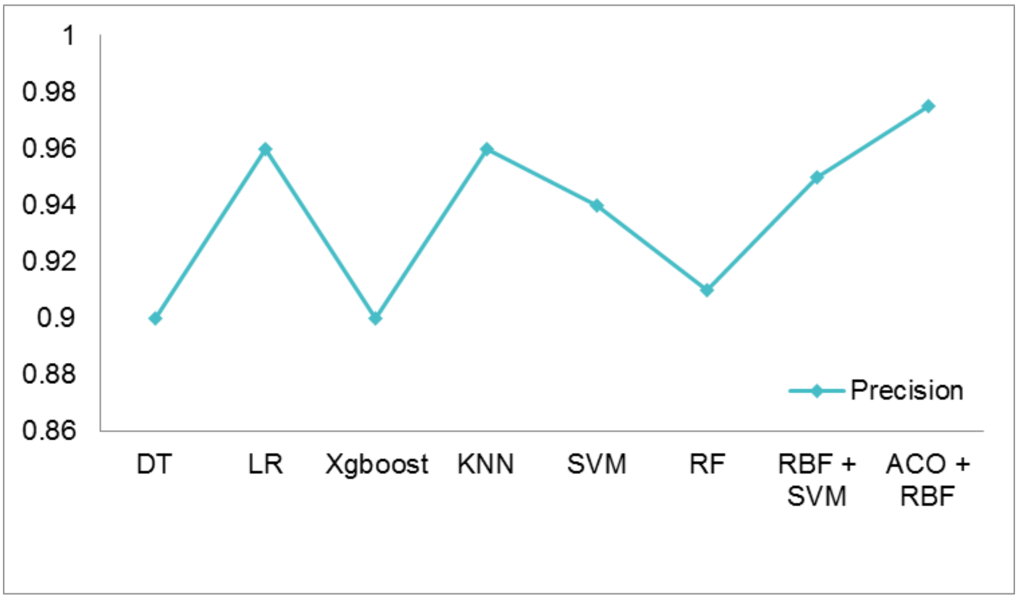


Figure 4: The proposed novelty's precision in comparison to other algorithms

$$Precision = \frac{Tp}{Tp+Fp} \qquad (9)$$

Table 2 displays the results of our analysis, which demonstrates that our proposed method has a higher precision rate than other ML models. In this case, ACO+RBF yields a 97.5% improvement.

Our new approach improves precision by 4.5% on average compared to traditionally used ML methods, as Fig. 4 shows the precision of our novelty and others. Remember that we divide the number of positive test values by the number of false negative test values to get the proportion of true positives. A different name for it is True Positive Rate (TPR). It is a metric for evaluating the all-encompassing nature of a classifier. A low recall can be inferred from numerous false negatives.

$$Recall = \frac{Tp}{Tp+Fn} \qquad (10)$$

ACO + RBF had a 97.1% recall rate in this study as Figure. 5 shows the recall of our novelty and others. This is the highest success rate amongst ML algorithms. Figure 6 depicts the recall of various ML approaches. The recall is improved by 6.2% on average using our novel approach compared to previously used ML techniques.

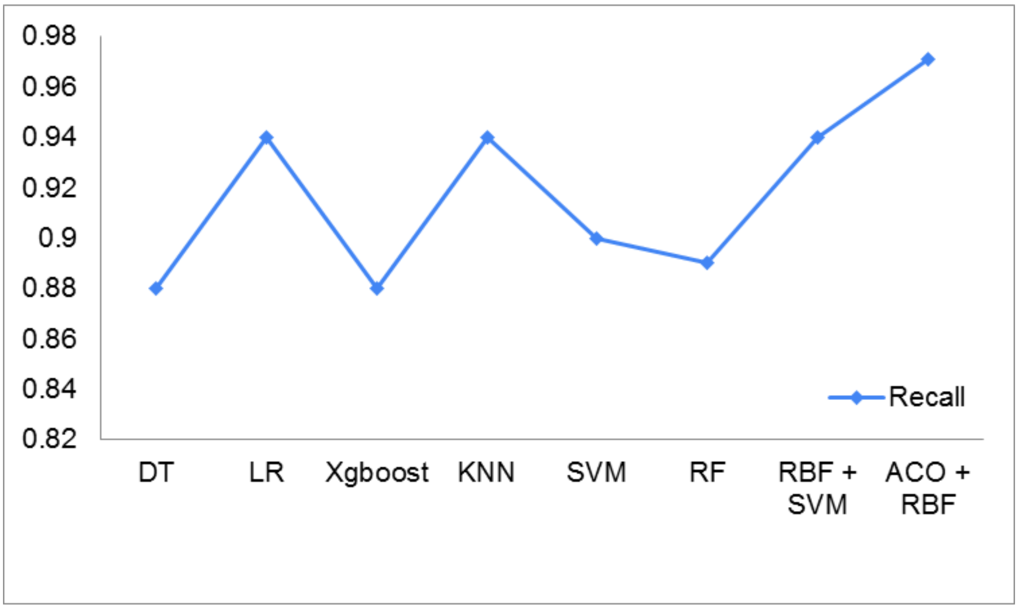


Figure 5: Recall the measurement of our novelty and others.

F-Measure: the sum of a measurement's recall and precision weights[25].

$$F - Measure = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall} (11)$$

Table 2 shows that when combined with other methods, our proposed method, ACO+RBF, has a 96.7% success rate. Compared to other machine learning algorithms, it achieves the highest percentage of F-scores as Figure. 6 shows the F-Measure of our novelty and others. The ACO algorithm can be used to train the RBF network to recognize ambiguous and noisy information.

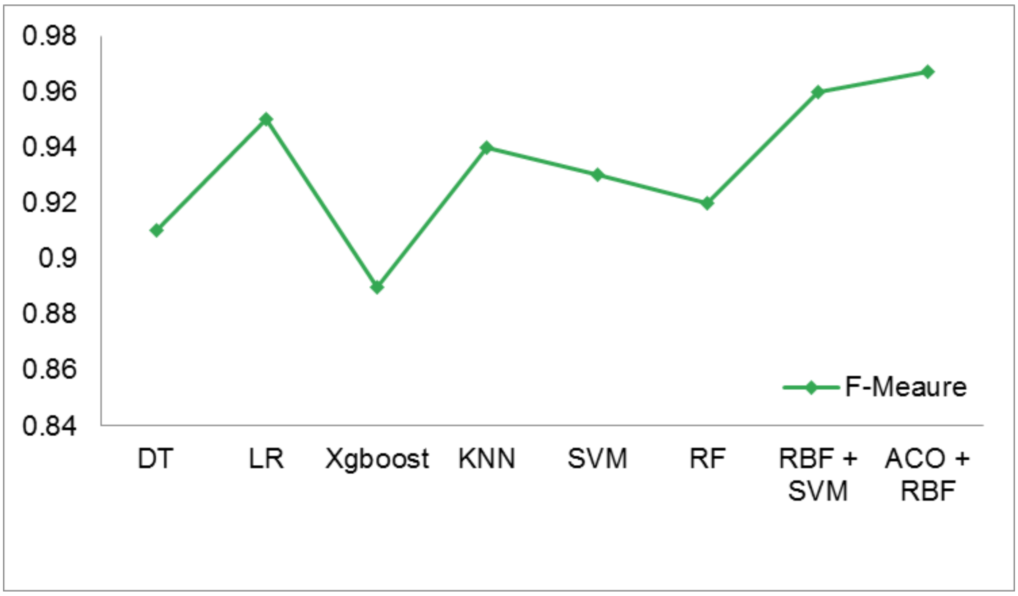


Figure 6: F-Measure of our novelty and others.

We implemented our proposed method and various ML approaches, as provisionally mentioned, to obtain results and compared which ones were most effective. Experiments comparing our approach to standard ML methods produce the results shown in Table 2.

Table 2: Comparative Metrics to Classify Uncertain Data.

Metrics	DT	LR	Xgboost	KNN	SVM	RF	RBF + SVM	ACO + RBF
Accuracy	0.92	0.96	0.9	0.95	0.94	0.93	0.97	**0.985**
F-Measure	0.91	0.95	0.89	0.94	0.93	0.92	0.96	**0.967**
Precision	0.9	0.96	0.9	0.96	0.94	0.91	0.95	**0.975**
Recall	0.88	0.94	0.88	0.94	0.9	0.89	0.94	**0.971**

ACO was then used as the optimization algorithm for the RBF network. Metrics show that our approach is superior to the most popular and established machine learning algorithms. The results also show that our proposed method is superior to others in locating uncertain data. Our novel approach improves accuracy by an average of 4.9% compared to historically used ML techniques, using industry-standard metrics of performance. With reference to Fig. 7.

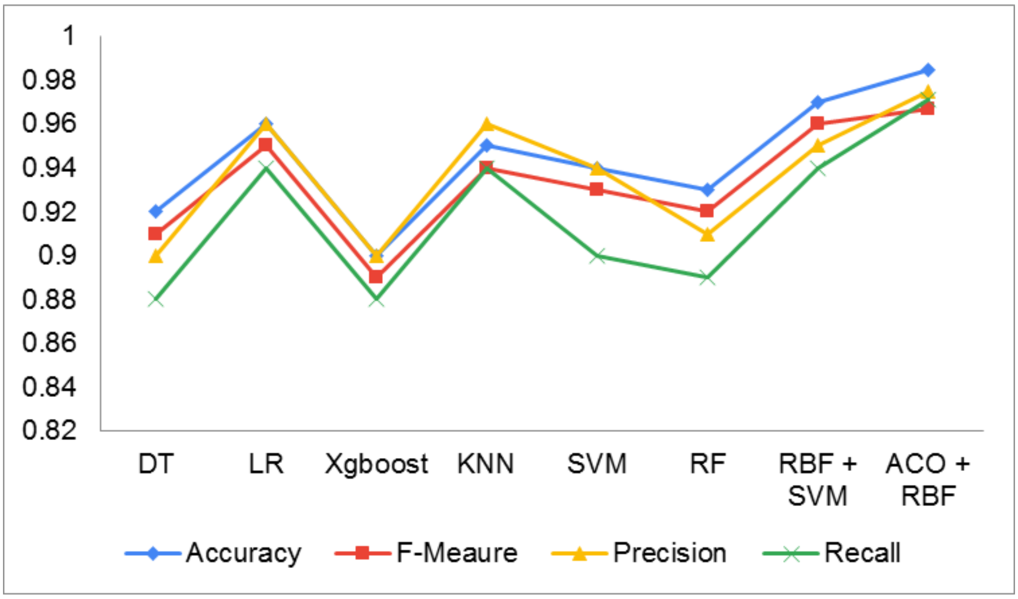


Figure 7: Evaluation of our novelty and other methods for classifying uncertain data

5. CONCLUSION

In this work, we propose a machine learning-based method for classifying data with some degree of uncertainty. The proposed method combines elements of the RBF network and the ACO algorithm to achieve optimal results. The RBF network is trained with ACO and then used for the classification of uncertain data. A number of experiments were conducted to evaluate the proposed method and to compare it to other popular methods such as the k-NN classifier, the SVM, the DT, the RF, the LR, the Xgboost, and the RBF SVM. Accuracy = 98.5%, precision = 97.5%, recall = 97.1%, and F-Measure = 96.7% were the specific results of the simulation. This is conclusive evidence that the proposed model outperforms baseline ML techniques when it comes to classifying data with a degree of uncertainty. On average, the proposed method is 4.9% more precise than the ML-based method. In the future will modifying the Ant colony which is a fully connected graph into a clustering graph for better performance and efficiency of the system.

REFERENCES

[1] T. Agrawal, Rakesh, 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] G. Suresh, S. Shaik, O. Reddy, & Munibhadrayya, B. "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 NetworkBased 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.

[23] S.R.. Mahmood, M. Hatami and Moradi, P., 2020, October. A Trust-based Recommender System by Integration of Graph Clustering and Ant Colony Optimization. In 2020 10th International Conference on Computer and Knowledge Engineering (ICCKE) (pp. 598-604). IEEE.

[24] C.W. Fisher, E.J. Lauria, and Matheus, C.C., 2009. An accuracy metric: Percentages, randomness, and probabilities. Journal of Data and Information Quality (JDIQ), 1(3), pp.1-21.

[25] G. Hripcsak, and A.S Rothschild, 2005. Agreement, the f-measure, and reliability in information retrieval. Journal of the American medical informatics association, 12(3), pp.296-298.