An effective classification technique for XML documents using hyper parameter optimized classifiers

S.Sahunthala [a], Angelina Geetha [b], Latha Parthiban [c]

[a] Research Scholar, Department of Computer Science Engineering, Hindustan Institute of Technology and Science, India
[b]Professor, Department of Computer Science Engineering, Hindustan Institute of Technology and Science, India
[c]Assistant Professor, Department of Computer Science Engineering, Pondicherry, University, Puducherry, India

Article History: *Do not touch during review process(xxxx)*

Abstract: In real world XML data plays a significant role in the application of World Wide Web. Now a days, in research the data classification in XML document for heterogeneous structure proves to be a challenging task. A number of algorithms are available in XML data classification process. In the existing technique the performance is degraded in the classification process of XML document. In this paper the machine learning technique TSRSA (Tuning Swarm Rapid Swarm Algorithm) is proposed to classify the XML data. First, the elements are extracted by using kernel vector space model. Second, we classify the XML data using the algorithm of TSRSA optimization technique. TSRSO is using hyper parameters to obtain the better classifier. The experiments are demonstrated in the existing technique ELM (Extreme Machine Learning), Standard algorithms (SVM Support Vector Machine, DT-Decision Tree, NB-Navie Bayes, and KNN-K Nearest Neighbor), KPCA-Kernel Principal Component Analysis and KELM Kernel Extreme Machine. In this research the proposed TSRSA algorithms are compared with the existing technique. The various performance parameters are compared with reference to the existing and the proposed model.

Keywords: XML data, classification data, Vector space model, Tuning Swarm Rapid swarm Optimization Algorithm TSRSA

1.Introduction

In real world, the growth of numerous web applications plays a very vital role in business. Most applications are based on the de-facto standard format of XML (Extensible Markup Language).The heterogeneous structure of data has become a tree model to perform the query .The classification process of XML document plays a very complex role in real time applications. The integration of data is required while processing the query between more than one XML documents. Research is very much required in the classification process of XML hierarchical. The semantics structure is used to classify the XML dataset **(Thasleena, N. T., & Varghese. S. C.2015)**.The XML document classification is similar to text document classification. The text document is based on the semantics of the dataset. The XML document has framed the tree structure to view the information of the document. More than one level is maintained in the file structure. The structure is started from the root node. Then children nodes are created based on the information of the XML document. The structure and the semantics are considered in XML document classification. The XML semantics matching process is introduced in the template of XML matcher template. The component of XML matcher template and the various challenges of matcher template are described in **(Agreste, Santa, Pasquale De Meo, Emilio Ferrara, &Domenico Ursino 2014)**. Most of the comparisons are done using tree edit distance in the tree structure **(Tekli, Joe, & Richard Chbeir. 2012)(Tekli, Joe, Richard Chbeir, Agma JM Traina, Caetano Traina Jr, & Renato Fileto. 2015)**. In general XML documents are classified as deterministic. Fuzzy information is vague rather than being definite information. In real time applications vague values are subjective information to process the data. Most of the real world applications have used fuzzy set basics concept. Various areas of the fuzzy applications are given by **(Zadeh, Lotfi A. 1999)**. The fuzzy XML document is processed based on the XML data in **(Oliboni, Barbara, and Gabriele Pozzani. 2008)**.The approaches of fuzzy XML data have been found in**(Nierman A, Jagadish H. V. 2002)**.The web based fuzzy XML data growth is there in real world internet communication. This creates necessity to classify more than one fuzzy XML document into one format to process the data , so that various documents are integrated in the data processing. The current research challenges have not generated the report of the classification on the fuzzy XML data unfortunately. The classification process has been analyzed in uncertain XML document data (**Zhao X, Bi X, Wang G, Zhang Z, Yang H. 2016**).The Extreme Learning Machine approach is discussed with various datasetas can be found in (**Huang G. B. 2014**) (**Huang GB, Chen L. 2007**).The ELM with semi supervised and supervised is explained in (**Huang G, Song S, Gupta JN, Wu C. 2014**).The various classifications are analyzed in**(Sahunthala S, Geetha A, Parthiban L. 2020)(Sahunthala S, Geetha A, Parthiban L. 2020)(Vidhyalakshmi. M & Sudha. S. 2019)**.The error in the decoding of XML data is discussed in **(Escalera S, Pujol O, Radeva P. 2019)**.

The remaining part of the paper is divided into four sections.Section.2 discusses the works relating to the proposed technique. The proposed methodology is explained in section.3. The experimental evaluation is given in Section.4. Section.5 gives the conclusion of this research.

2. Related Work

The existing approach classification of the XML data in **(Zhao, Zhen, Zongmin Ma, and Li Yan. 2019)**. The techniques ELM, Standard Classification algorithms and KPCA-KELM are used to classify the XML data in existing work. This review focuses on only the classification part of the XML data.

2.1 KPCA (Kernel Principal Component Analysis)

It is derived from PCA. This is widely used for non-linear data feature extraction. A kernel function is used in the feature space association. KPCA is the mapping process for all samples from the linear data space to non-linear feature space. In the feature space the PCA features are extracted. The mapping function θ is defined implicitly by using kernel function. The M samples are considered for training data. In this technique the mapped samples θ(y1), θ(y2),…., θ(ym) are considered for classification.

$$\sum_{j=1}^{m}(y_j) = 0 \tag{1}$$

In feature space the correlation of the mapped sample is given by

$$Cr = \frac{1}{m}\sum_{j=1}^{m} \theta(y_j)\theta(\mathrm{y_j})^{\mathrm{T}} \tag{2}$$

The set of Eigen vectors Cr must be extracted features in feature space. This technique looks for the Eigen values μδ. Then the associated Eigen vectors Vec satisfying

$$CrVec = \delta\mathrm{Vec} \tag{3}$$

The mapping data process depends on the Eigen vectors which are defined in Eqn.3.The performance parameters- accuracy and training time are measured in the experimental demonstration. The performances of these parameters are decreased.

2.2 Extreme Learning Machine

ELM is the free forwarded network for classification **(Zhao X, Bi X, Wang G, Zhang Z, Yang H. 2016)**. This technique avoids multiple iteration and local minimization in the classification process. In general, the learning speed is faster in the process. The random assignment is generated of network's hidden layer and the output weights can be manipulated by matrix operations. The ELM model is given by

$$f(x) = \sum_{k=1}^{M} b_k\, g_k(x_j) = h(y)bj = 1,2,\dots,N \tag{4}$$

where M is the number of hidden layer nodes, g is the activation function, and bk is the weight vector between the kth hidden node and the output node. The ELM is used to minimize the training errors ϵ.The mathematical manipulation to obtain the optimal classification in the existing approach is discussed in **(Dhiman, Gaurav, Meenakshi Garg, Atulya Nagar, Vijay Kumar, and Mohammad Dehghani. 2015)(Kaur, Satnam, Lalit K. Awasthi, A. L. Sangal, and Gaurav Dhiman. 2020)**.

2.3 Standard Classification Algorithms

2.3.1 Support Vector Machine

This is the supervised learning approach to perform the classification. This technique classifies linear and non-linear data set. The feature value is adjusted using the kernel value. This approach adjusts the input in each layer to receive the desired output. This has more than one linear model. The kernel linear or non- liner or Gaussian is used to adjust the feature value. The existing technique classifies XML data without using optimized hyper parameters for classification. So the performance is reduced in the classification of XML data.

2.3.2 Decision Tree

This technique is the supervised learning in which the data is continuously classified in the given data set. In general the outcome is 0 or 1 in this approach. Another outcome is the continuous data in the dataset. This technique has the parameters entropy and information gain to predicate the outcome. The probability distribution is manipulated based on the training data set. The system is changed by the stochastic process dynamically. The input feature is correlated with another feature in the data set.

2.3.3 Navie Bayes

This approach uses a huge volume of data. This supports fast process. It also handles an uncomplicated classification technique. This technique uses the conditional probability concept to classify the data. This uses the prior probability value. This can be used in binary and multiclass classification. Navie Bayes theorem is the back bone of this algorithm.

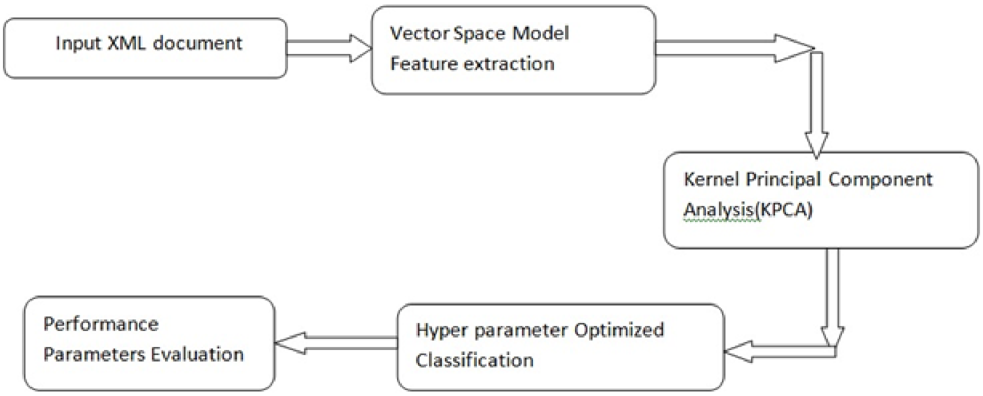
2.3.4 KNN-K Nearest Neighbor

This is supervised learning to perform the classification in the data. KNN has the properties of lazy learning and non-parametric process. This uses feature similarity to predict the new position of the data based on the nearest point in the training dataset. To find the new point using the Euclidean distance computation mechanism. The performances of the existing techniques are degraded in the classification of XML data. This can be significantly increased by the proposed model TSRSO algorithm.

3. Proposed Work

TSRSO classification technique is proposed in this research. The proposed work contains two modules. First one is the vector space model for feature extraction and the other is the technique TSRSO for the classification of XML data with hyper parameters. This technique is highly considerable for high dimensional non-linear data. The preprocessing is the technique to produce the raw data to implement the machine learning technique. The overall system architecture is given in Figure.1.

Figure.1. Overall System Architecture of the proposed technique.


Input XML document
Vector Space Model Feature extraction
Kernel Principal Component Analysis(KPCA)
Hyper parameter Optimized Classification
Performance Parameters Evaluation

3.1 Vector space model

Vector space model or term vector model is an algebraic model for representing text documents (and any objects, in general) as vectors of identifiers (such as index terms). Documents and queries are represented as vectors. The non- linear data features are extracted from the given XML raw dataset. The vector space model is used to extract XML documents being represented by the tree model which is having the ordered number of labels **(Tekli, Joe, Richard Chbeir, Agma JM Traina, Caetano Traina Jr, & Renato Fileto. 2015)**. In this tree the node represents the element’s name and the attribute value with the respective label in the XML document structure. The XML document is represented as rooted ordered labelled tree structure. The tree structure represents the type of the element based on the information from the XML document. The unnecessary information is removed from the tree structure. This increases the performance task of the process. The root node has more number of parent nodes in the XML document. The parent node has more number of children nodes and their values.

3.1.1 Definition 1: XML document tree

The XML document has a rooted ordered labelled tree XDTS=(n,e,l,d,p,v) where

- N is the set of nodes in the tree XDTS

- The set of edges is represented by e, which presents the hierarchical structure of the tree XDTS
- The labels of the element is represented by l
- D is the symbol which represents the data type of the element
- The position the element is given by p in the XDTS
- The value of the element is represented by v.

We should maintain the related information of the element and their information to perform the conversion from the XML document into XML tree structure.

3.1.2 Definition 2: XML document tree node (XDTN)=(NL,ND,NP,NV)

- NL is the label of the node
- ND is the depth of the node. In general the depth of the root node is 1.This value is incremented based on the child nodes of the root node.
- NP is the position of the node
- NV is the value of the node

Table.1. Hyper parameters in classification technique.

Machine learning classifier	Parameter name	Parameter value range
KNN	Number of neighbours	(1,50)
	Distance	(minkowski,cosine,euclidean,mahalanobis,seuclidean,jaccard,chebychev)
NB	Distribution names	(Kernel, Multinomial distribution, Multivariate multinomial distribution, Gaussian distribution)
DT	Minimum leaf size	(1,50)
	Minimum parent size	(1,50)
SVM	Coding	(binary complete, dense random, one vs all, ordinal,sparse random,ternary complete)

3.2 Classification with Tuning Swarm Optimized algorithm

The feature is extracted using vector space model. The training stage and testing stage is considered in the overall architecture of the proposed system.

3.2.1 Preprocessing

Preprocessing is the process to clear the data. This should be applied to the data before implementing the data mining techniques. To receive the raw data the properties of the XML document, The XML document is converted into XDT structure. The feature value is calculated for each element of the hierarchical document. In training or testing data the necessary preprocessing technique is considered to obtain the raw data.

3.2.2 Classification Algorithm

The algorithm TSRSO contains two procedures. First one is the TSRSA. This has set the initial parameters required and classification for our algorithm. The second procedure init has the hyper parameters are used in the classification process. To obtain the optimal solution the iteration is considered in the algorithm. The hyper parameters maximum split is referred from **(Khan, Faiza, Summrina Kanwal, Sultan Alamri, & Bushra Mumtaz. 2020)**. The various hyper parameters are given in Table 1.This procedure updates the optimal neighbor points in the XML document based on the class which is specified. The data flow of the proposed TSOA algorithm is given in Figure.2.

In the algorithm TSRSA the new point P of the search agent is given by the Equation.5.

$$P = B.P(x) + D.\left(P_i(x) - P_r(x)\right) \tag{5}$$

Where Pi (x) indicates the position of the rate, Pr(x) represents the best optimal position to find the nearest neighbor. For better exploration and exploitation the subject of the iteration is explored by the parameters B and D. B and D are calculated from Equations 6 and 7 respectively.

$$B = S - x * \left(\frac{S}{maxi_{Iteration}}\right) \tag{6}$$

$$D = 2.random() \tag{7}$$

S is the random number between [1,5] and D is also the random number between [0,2]. The next neighbour point with respect to the current iteration (x) Pi(x+1) is computed in the Equation.8.

$$P_i(x+1) = |P_r(x) - P| \tag{8}$$

Pi defines the updated next position of the neighbour.

$$Po = |P - r_{random}\ P_m(x)| \tag{9}$$

Where Po is the distance between source and the search agent, x indicates the current iteration, P represents the position of the source, and Pm represents the new tuning position. The updated position with respect the point P is given in Equation 10.

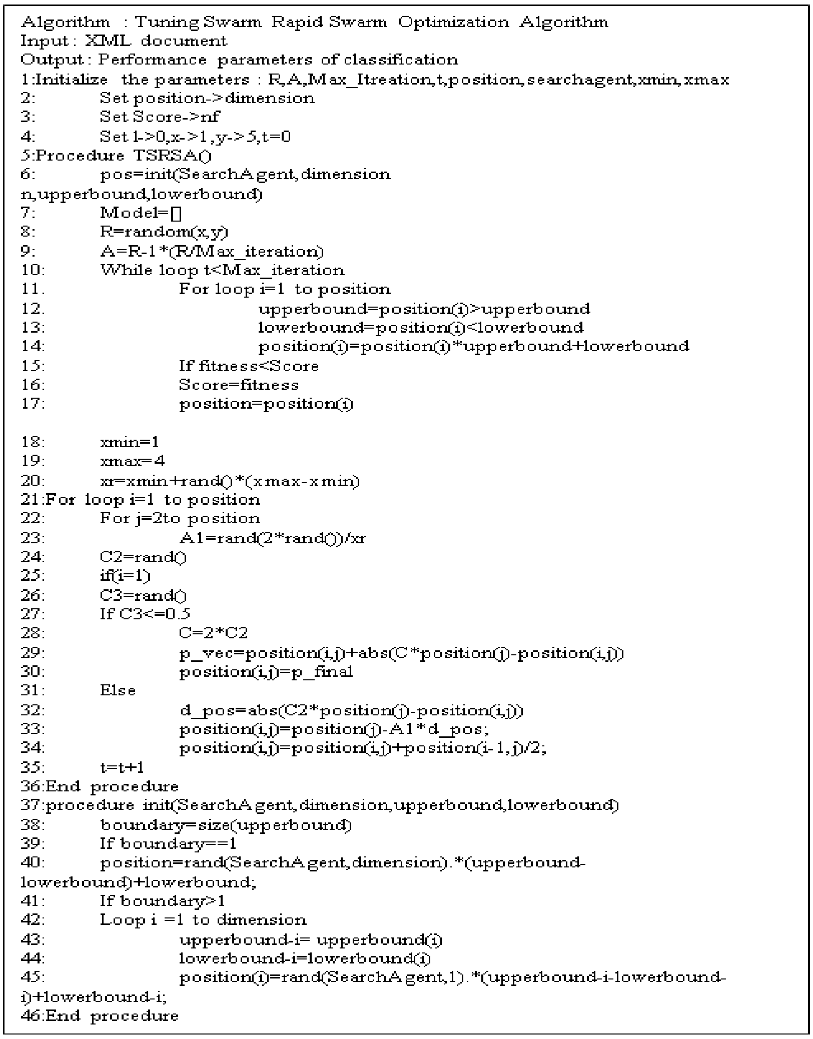
$$P_m(x) = \begin{cases} P + A.Poif r_{random} & \geq 0.5 \\ P - A.Poif r_{random} & < 0.5 \end{cases} \tag{10}$$

r_{random} is the number between 0 and 1 [0,1]. The best optimal solution of next point of Pm(x+1) is given by the Equation. (11).

$$P_m(x+1) = \frac{P_m(x) + P_m(x+1)}{2 + c} \tag{11}$$

where c is the random number between 0 and 1[0,1].

Figure.2. TSRSO algorithm.



```
Algorithm  : Tuning Swarm  Rapid Swarm  Optimization  Algorithm
Input : XML  document
Output : Performance  parameters  of classification
Initialize  the parameters  : R,A,Max_Itreation,t,position,searchagent,xmin,xmax
    Set position->dimension
    Set Score->nf
    Set l->0,x->1,y->5,t=0
Procedure  TSRSA()
    pos=init(SearchAgent,dimension n,upperbound,lowerbound)
    Model=[]
    R=random(x,y)
    A=R-1*(R/Max_iteration)
    While loop t<Max_iteration
        For loop i=1  to position
            upperbound=position(i)>upperbound
            lowerbound=position(i)<lowerbound
            position(i)=position(i)*upperbound+lowerbound
        If fitness<Score
        Score=fitness
        position=position(i)

    xmin=1
    xmax=4
    xr=xmin+rand()*(xmax-xmin)
For  loop i=1  to position
    For j=2to position
        A1=rand(2*rand())/xr
    C2=rand()
    if(i=1)
    C3=rand()
    If C3<=0.5
        C=2*C2
        p_vec=position(i,j)+abs(C*position(j)-position(i,j))
        position(i,j)=p_final
    Else
        d_pos=abs(C2*position(j)-position(i,j))
        position(i,j)=position(j)-A1*d_pos;
        position(i,j)=position(i,j)+position(i-1,j)/2;
    t=t+1
End  procedure
procedure  init(SearchAgent,dimension,upperbound,lowerbound)
    boundary=size(upperbound)
    If boundary==1
    position=rand(SearchAgent,dimension).*(upperbound-lowerbound)+lowerbound;
    If boundary>1
    Loop i =1  to dimension
        upperbound-i= upperbound(i)
        lowerbound-i=lowerbound(i)
        position(i)=rand(SearchAgent,1).*(upperbound-i-lowerbound-i)+lowerbound-i;
End  procedure
```

Figure.3. Data Flow diagram of the proposed algorithm.

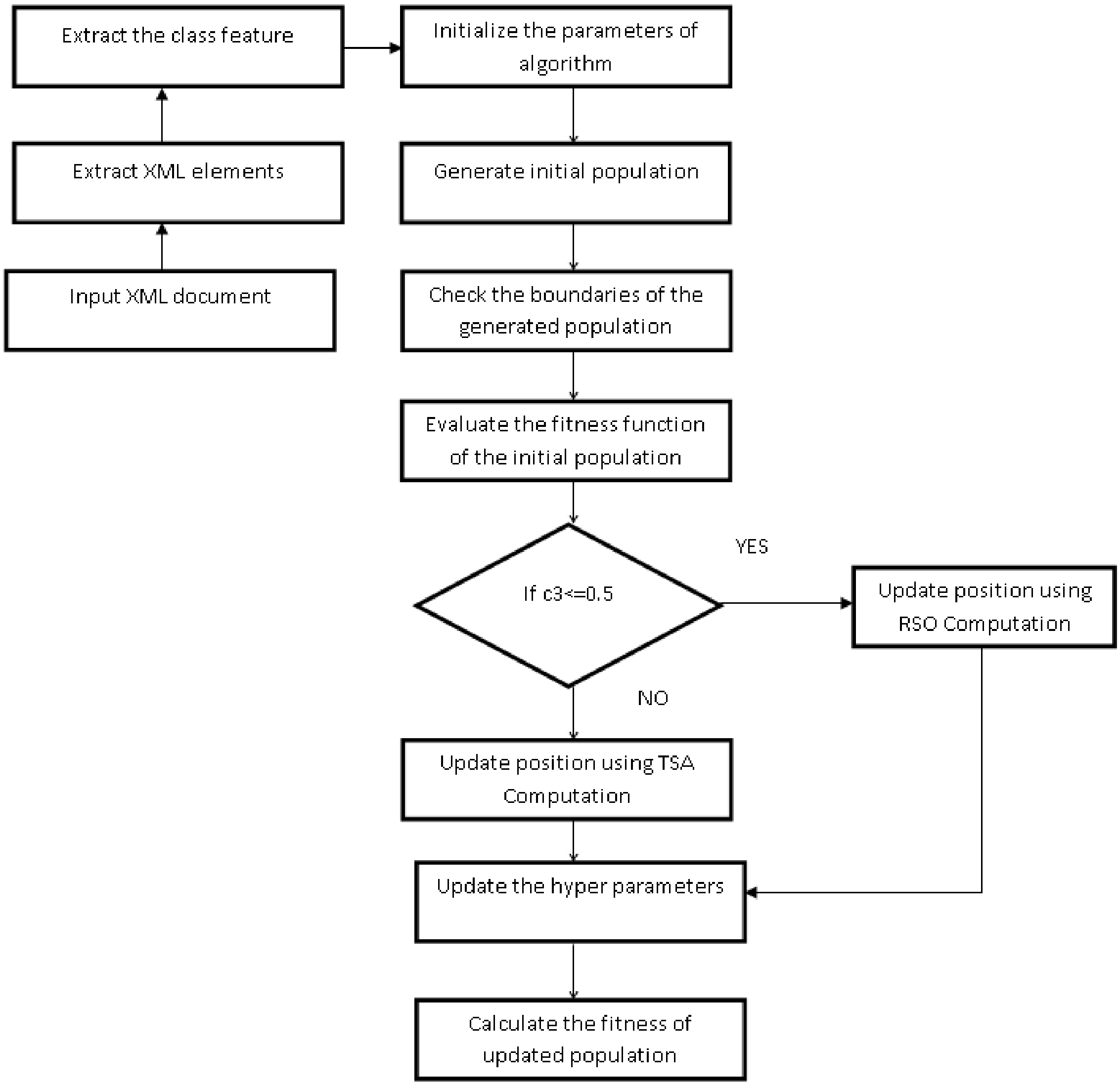
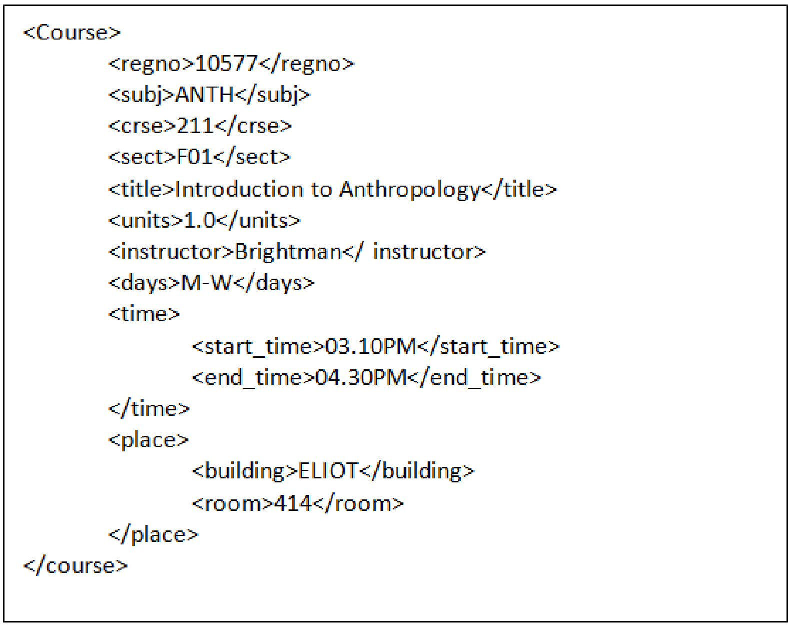

Extract the class feature
Initialize the parameters of algorithm
Extract XML elements
Generate initial population
Input XML document
Check the boundaries of the generated population
Evaluate the fitness function of the initial population
If c3<=0.5
YES
Update position using RSO Computation
NO
Update position using TSA Computation
Update the hyper parameters
Calculate the fitness of updated population

Figure.4. The sample XML data in "reed.XML".



```
<Course>
        <regno>10577</regno>
        <subj>ANTH</subj>
        <crse>211</crse>
        <sect>F01</sect>
        <title>Introduction to Anthropology</title>
        <units>1.0</units>
        <instructor>Brightman</ instructor>
        <days>M-W</days>
        <time>
                <start_time>03.10PM</start_time>
                <end_time>04.30PM</end_time>
        </time>
        <place>
                <building>ELIOT</building>
                <room>414</room>
        </place>
</course>
```

The part of the reed.xml document is given in Figure.4. In this research the feature “Units “is considered for the classification. The dataflow diagram of the proposed algorithm is given in Figure.3. The input is the XML document which is being classified. The elements of the input document are extracted by using vector space model. The sample fragment of reed.xml is given in Figure.4. The class feature is extracted for classification. The initial population of the TSRSA algorithm is set. The new population based on the initial population for the given input xml document is generated. The fitness or objective function to obtain the better accuracy of the classification is evaluated. In this research the class label is Units. So the class label value 0.5 is set for the classification process. The class label is verified by two manipulations. First one is, if class label (C3) <=0.5, generate the new position and the population by using Equations 5, 6, 7, and 8. Otherwise the new point and the population is obtained using the Equations 9, 10, and 11. In this research four objective functions are used to improve classification performance .They are Decision Tree (DT) fitness, Navie Bayes (NB) fitness, K-Nearest Neighbour (KNN) fitness, and Support Vector Machine (SVM) fitness. By using these objective functions the error is reduced to obtain a better accuracy.

4. Experimental Evaluation

This section presents the effectiveness and efficiency of the proposed TSOA machine learning approach. The proposed work is compared with the existing techniques ELM, SVM, and KPCA-KELM by various performance parameters.

4.1 Experimental Settings

Various classification techniques are presented in 2019, 2019, 2018.The familiar classification techniques are selected to compare the performance behavior of the classification process. This research has considered the data set “reed.xml” for demonstration.The comparison between the existing techniques ELM, SVM, KPCA-KELM has been presented and the proposed technique TSOA has been combined with NB, DT, SVM, and KNN. All the experiments are developed [using?] Corei3 processor, 4GB RAM, and MATLAB 2019a at Windows platform. The following performance parameters are evaluated in the experiment. The performance parameters are defined as the following terms :

True Positive (TPD): The data that is correctly classified in the algorithm
False Positive (FPD): The data that is incorrectly classified in the machine learning technique
True Negative (TNP): The data that is correctly classified as normal
False Negative (FN): The data that is incorrectly classified as normal

Accuracy: This metric evaluates the classification models. The prediction model is correct in the data.

$$A = \frac{TPD + TND}{TPD + FPD + FND + TND} \tag{12}$$

Precision: Number of correct data returned by the machine learning technique which is implemented

$$Precision = \frac{TPD}{TPD + FPD} \tag{13}$$

Recall or sensitivity: Number of correct data being returned by the ML algorithm in the process

$$Recall = \frac{TPD}{TPD + FN} \tag{14}$$

Specificity: Number of incorrect data being returned by the ML algorithm in the process

Training time: How long the training of the model will take to classify the data

4.2 Performance Comparison

The performance comparison of classification metric between the existing and the proposed technique is given in Table.2.

Table.2. Performance Comparison between existing and proposed approach.

	Existing	**Proposed**	Existing	**Proposed**	Existing	**Proposed**	Existing	**Proposed**
Parameter	SVM	**TSRSO-SVM**	KNN	**TSRSO-KNN**	DT	**TSRSO-DT**	**NB**	**TSRSO-NB**
Accuracy (%)	61	69	74	88	69	85	69	82
Training	6.92	1.14	4.1	3.9	2.62	2.44	4.3	1.19

Time(Sec)								
Sensitivity (%)	33	46	51	66	48	68	42	54
Specificity (%)	71	81	82	94	81	94	74	90
Precision (%)	20	44	48	92	47	68	48	53
FPR (%)	32	20	17	07	18	08	25	13
RMSE (%)	**1.15**	1.04	**9.45**	**0.5**	**9.8**	**0.57**	**1.04**	**0.79**

Figure. 5a. Comparison of Accuracy between standard and proposed algorithms.

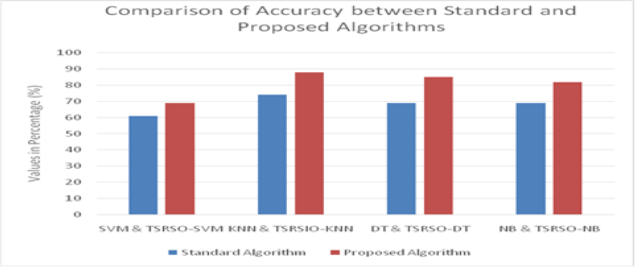


Figure. 5b. Comparison of Training between standard and proposed algorithms.

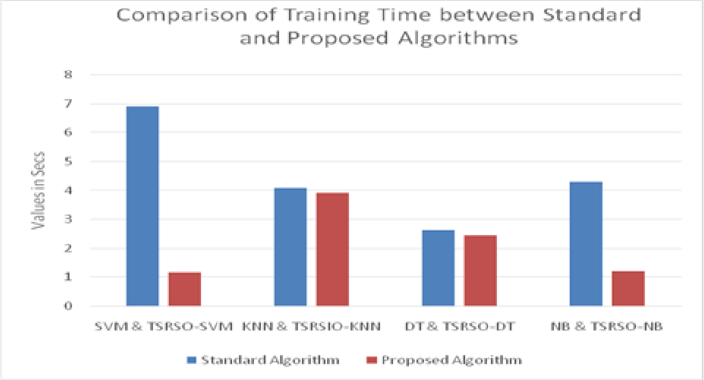


Figure. 5c. Comparison by Sensitivity metric between proposed and existing algorithms.

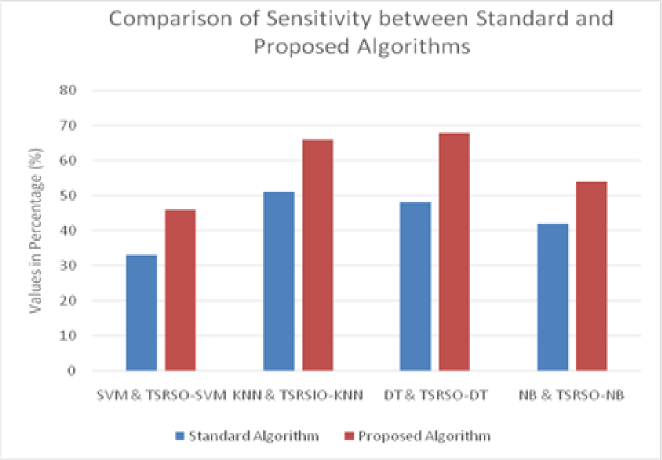


Figure. 5d. Performance Comparison by Specificity metric between existing and proposed algorithms.

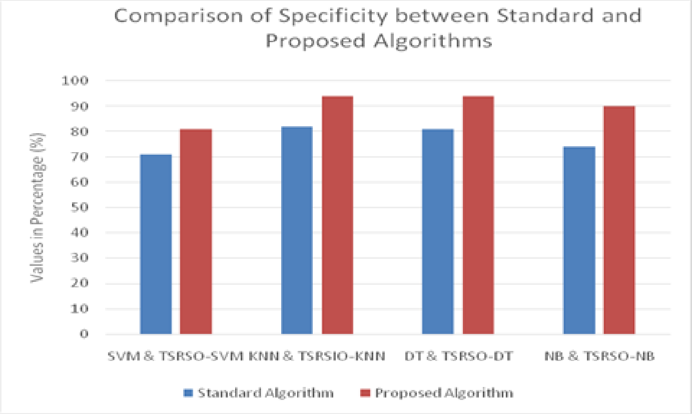


Figure. 5e. Performance Comparison by Precision metric.

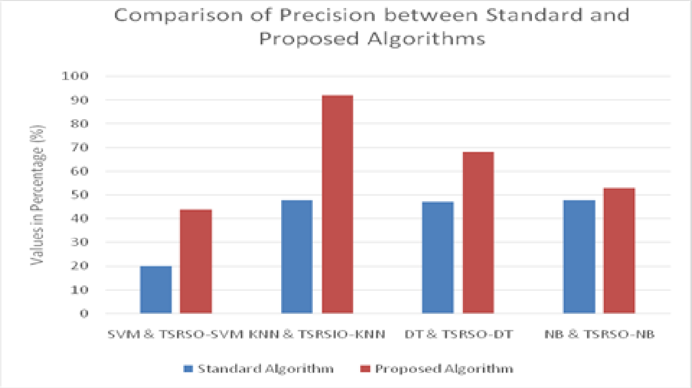


Figure.5f. Comparison by FPR performance metric.

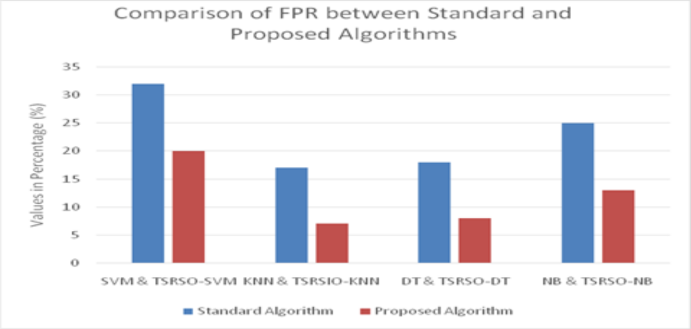
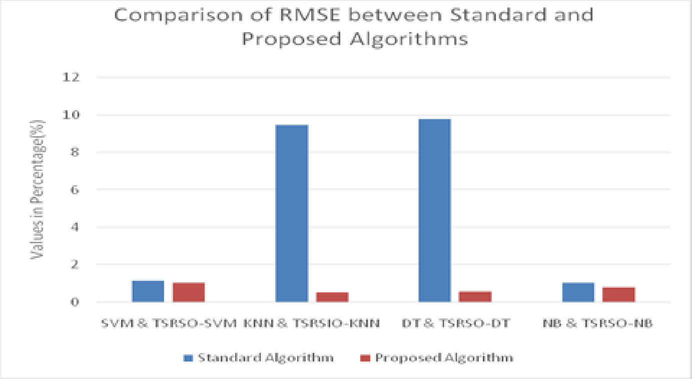


Figure. 5g. Comparison by RMSE metric between proposed and existing algorithms.



The figures 5a, 5b, 5c, 5d, 5e, 5f, and 5g show the result comparison between the existing standard classification algorithms (SVM, NB, DT, and KNN) and the proposed technique TSRSO by the performance metrics accuracy, training time, specificity, sensitivity, precision, FPR, and RMSE.

Figure.5a shows the performance of the accuracy between the existing and the novel algorithm TSRSA. The novel technique produces better result than the existing technique. The training times of the existing and the proposed algorithms is compared in Figure.5b.The proposed technique takes less time than the existing technique.

The sensitivity and the specificity performance comparison is shown in Figures.5c and 5d respectively. These parameters results should be high while implementing the novel algorithm. Figure.5c shows the proposed algorithm generating higher sensitivity than the existing standard algorithms. Figure.5d also gives higher specificity value in the proposed algorithm than the existing algorithm.

The parameter precision is compared between the existing and the proposed classification is given Figure.5e. The proposed technique gives higher precision than the existing algorithm. The metrics RMSE and FPR should be less in the new algorithm of the research. In this research the new algorithm TSRSO obtains the FPR more than the existing algorithms. This is shown in Figure.5f.The metric RMSE result comparison is given in Figure.5g. In the proposed approach RMSE is obtained less than the existing approach.

From Figures5a, 5b, 5c, 5d, 5e, 5f, and 5g it is clear that the proposed classification algorithm TSRSA generates better performance metric result than the existing standard classification techniques in XML data classifier.

All performance parameters have produced significantly better result than the existing ELM and KPCA-KELM algorithms.

5. Conclusion

In this paper the XML documents were effectively classified by the novel approach of TSRSA technique. This technique used the vector space model to extract the content of the XML document. The core part of this research is TSRSO algorithm. This algorithm used the hyper parameter to improve the performance of the classification of the XML data. The performance of the TSRSA classifier was improved based on the feature extracted from the XML document. The demonstration of the experimental results showed that the proposed classifier technique performed the classification of the XML document more efficiently than the existing technique. The accuracy of the proposed technique produced more robust result than the existing techniques. The training time of the novel classification technique significantly reduced compared with the existing technique. The other performance parameters were evaluated RMSE, sensitivity, Specificity, Precision and False Positive Rate. These parameters performance also significantly improved compared with the existing classification technique. In future the classification process will be evaluated in the distributed environment of the XML document.

References

1. Agreste, Santa, Pasquale De Meo, Emilio Ferrara, and Domenico Ursino. (2014). XML matchers: approaches and challenges. Knowledge-Based Systems, 66, 190-209.
2. Dhiman, Gaurav, Meenakshi Garg, Atulya Nagar, Vijay Kumar, and Mohammad Dehghani. (2020). A novel algorithm for global optimization: Rat swarm optimizer. Journal of Ambient Intelligence and Humanized Computing, 1-26.
3. Ding S, Zhao H, Zhang Y, Xu X, Nie R. (2015). Extreme learning machine: algorithm, theory and applications. Artificial Intelligence Review, 44(1), 103-15.
4. Escalera S, Pujol O, Radeva P. (2008). On the decoding process in ternary error-correcting output codes. IEEE transactions on pattern analysis and machine intelligence,32(1), 120-34.
5. Huang G, Song S, Gupta JN, Wu C. (2014). Semi-supervised and unsupervised extreme learning machines. IEEE transactions on cybernetics. 44(12), 2405-17.
6. Huang G. B, Chen L. (2007). Convex incremental extreme learning machine. Neurocomputing, 70(16-18), 3056-62.
7. Huang G. B. (2014).An insight into extreme learning machines: random neurons, random features and kernels. Cognitive Computation, 6(3), 376-90.
8. Kaur, Satnam, Lalit K. Awasthi, A. L. Sangal, and Gaurav Dhiman. (2020). Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization. Engineering Applications of Artificial Intelligence, 90, 103541.
9. Khan, Faiza, Summrina Kanwal, Sultan Alamri, and Bushra Mumtaz. (2020). Hyper-Parameter Optimization of Classifiers, Using an Artificial Immune Network and Its Application to Software Bug Prediction. IEEE Access, 20954-20964.
10. Vidhyalakshmi M and SudhaS. (2019). Text detection in natural images with hybrid stroke feature transform and high performance deep Convnet computing. Wiley Online Library.
11. Nierman A, Jagadish H. V. ProTDB: Probabilistic data in XML. InVLDB'02. Proceedings of the 28th International Conference on Very Large Databases 2002 Jan 1, pp. 646-657.

12. Oliboni, Barbara, and Gabriele Pozzani. (2008). Representing fuzzy information by using XML schema. In 2008 19th International Workshop on Database and Expert Systems Applications, pp. 683-687.
13. Sahunthala S, Geetha A, Parthiban L. (2020). Analysing Computational Complexity For Prediction Function In Health Record Dataset. In2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA) 2020 Nov 5, pp. 1643-1649.
14. Sahunthala S, Geetha A, Parthiban L. Computational Fuzzy Inference Logic For Effectively Analyzing Customer Survey. In2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) 2020 Oct 7, pp. 491-497.
15. Shitharth, S. (2017). An enhanced optimization based algorithm for intrusion detection in SCADA network. Computers & Security, 7016-26.
16. Shone, Nathan, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi.(2018). A deep learning approach to network intrusion detection. IEEE transactions on emerging topics in computational intelligence, 2(1), 41-50.
17. Tekli, Joe, and Richard Chbeir. (2012). A novel XML document structure comparison framework based-on sub-tree commonalities and label semantics. Journal of Web Semantics ,11, 14-40.
18. Tekli, Joe, Richard Chbeir, Agma J. M Traina, Caetano Traina Jr, and Renato Fileto. (2015). Approximate XML structure validation based on document–grammar tree similarity. Information Sciences, 295, 258-302.
19. Thasleena, N. T., and Varghese S. C. (2015). Enhanced associative classification of XML documents supported by semantic concepts. Procedia Computer Science, 46, 194-201.
20. Zadeh, Lotfi A. (1999). Fuzzy sets as a basis for a theory of possibility. Fuzzy sets and systems, 1009-34.
21. Zhao, Zhen, Zongmin Ma, and Li Yan. (2019). An Efficient Classification of Fuzzy XML Documents Based on Kernel ELM. Information Systems Frontiers, 1-16.
22. Zhao X, Bi X, Wang G, Zhang Z, Yang H. (2016). Uncertain xml documents classification using extreme learning machine. Neurocomputing, 174, 375-82.