A Novel Binary Search Tree Method to Find an Item Using Scaling

Praveen Pappula
School of Computer Science and Artificial Intelligence, SR University, India
prawin1731@gmail.com

Abstract: *This Approach comprises of methods to produce novel and efficient methods to implement search of data objects in various applications. It is based on the best match search to implement proximity or best match search over complex or more than one data source. In particular with the availability of very large numeric data set in the present day scenario. The proposed approach which is based on the Arithmetic measures or distance measures called as the predominant Mean based algorithm. It is implemented on the longest common prefix of data object that shows how it can be used to generate various clusters through combining or grouping of data, as it takes O(log n) computational time. And further the approach is based on the process of measuring the distance which is suitable for a hierarchy tree property for proving the classification is needed one for storing or accessing or retrieving the information as required. The results obtained illustrates overall error detection rates in generating the clusters and searching the key value for Denial of Service (DOS) attack 5.15%, Probe attack 3.87%, U2R attack 8.11% and R2L attack 11.14%. as these error detection rates denotes that our proposed algorithm generates less error rates than existing linkage methods.*

Keywords: *Clustering, classification, KNN, vector quantization, mean based search, scaling.*

Received May 1, 2020; accepted July 4, 2021
https://doi.org/10.34028/iajit/19/5/2

1. Introduction

In the present day era of data mining almost 99% of the users will search the data or a particular requirement over internet or any other sources. The searching of the data over a small database or file system will be an easy approach. But when search is implemented over a huge data set that comprises of trillions of datasets where every single data set may consists of many attributes (fields) or properties which is a typical or difficult task when implemented or searched. Almost the servers will drain or die while searching the data.

When we deal with a massive datasets that comprises of different things or data pertaining to different people over a dynamic or ever changing data, as the technology that tends to store, retrieve, or analyze the data. And while illustrating in a more generalized manner it represents that data is merged or combined and leads to the size or complexity and cost for utilizing the current technology which leads to a huge dataset based on previous or ongoing research the search operation is used up to a maximum extent in the areas such as: practical physics and its attained results, astronomy which includes data related to planets and stars identification and measures, geology which deals with movement in totemic plats and soil patterns. In the climatology which deals with seasons and climatic variations then medicine related data that

Includes up gradations and Approaches. Chemistry that includes redefining of new elements and replicating them back, genomics deals with the Approaches and intrusions related data attained, banking data is related to various transactions. As the telecommunications, data is related to cutting edge technologies and frequency bands, and then data related to public and private enterprise, which comprises of many aspects of data.

The process of calculating the data utilization in the present day digital universe. Where the technology barriers are represented to be reached to 550 hexa bytes (681 billion gigabytes) by 2020 and by 2025. It will be expected to reach 10 times more than of 2018.It is considered to be a benchmark because for the first time the amount of data that is created, captured or replicated as at the maximum ever with respect to storage.

As the trend is expected to be increased by almost 50% by 2025 and the estimation represents that 95% of digital era. The universe pumps unstructured data because the extra effort is kept on the process implementation to identify or locating the information because most of the challenges represent massive growth in data. The information is implemented to be economical as the environmental cost incurred or attained in all the techniques needs to access, store or visualize and analyze the data are lesser when compared to searching a data object on the same data.

For performing data analysis over exploratory data that shows more impact or analysis over the efforts pertaining throughout this work. Due to massive increase in attainment or availability of data complexity or scalability which is the most typical problem as the conventional data analysis.

The structure of paper comprises of section 2 with Related work followed by section 3 with problem statement which is further followed by conventional models to search an object then in section 4 proposed model to searching an object in a cluster (mean-based search) is illustrated then in section 5.1 proposed model to searching an object in a cluster (mean-based search) is proposed then in section 6 results obtained are explored and this section is followed with conclusion of the paper.

2. Related Work

A binary search is a search algorithm that performs identification of the position of a specified value being searched within a sorted array as the algorithm accomplishes comparing of the input key value with the middle element value in a given list. And when the keys are matched it denotes that the search element is found at that specific index which is returned. Else when the searched key is smaller than that of compared middle element then the algorithm repeats its action on the left sub-array and when the searched key is greater than the that of compared middle element then the algorithm repeats its action on the right sub-array and this process is repeated in an iterative manner till either search value is found or the sub-array is reduced to 0 elements which represents that “Search element is Not found” [18]. Every iteration will reduce half of the searching values that leads binary searches to be very efficient over huge collections and the only concern in binary search is it requires a sorted collection of elements [20].

3. Problem Statement

In Present literature many BST algorithms have been proposed to maintain with BST in optimal shape as illustrated in Table 1 [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. Each technique has its own pros and cons. In this paper, we proposed an algorithm, which yields best accuracy than existing models as all the existing models use Euclidian distance with BST algorithms for generating clusters and searching key value whereas in the proposed algorithm we use Mean based divisive algorithm. The advantage of using mean based divisive clustering algorithm over Euclidian is lesser comparisons and simpler cluster creation process.

Table 1. Characteristics of existing BST techniques [18].

Technique	Characteristics
AVL [1]	The difference of height of n’s left subtree as well as right subtree is not more than 1
Red Black [5]	Requires minimal operations when compared with AVL tree while performing deletion of a node does not require recursive implementation
Balancing with Sorting Algorithm [14]	Is the initial suggested technique of static algorithm for generating perfectly balance tree by subdividing the tree into two partitions
ThreadedBST [28]	Recursion is not required
Globally Balancing Algorithm [10]	Performs global balancing tree by rearranging the pointers within the linear time for traversing recursively in in-order process over the sorted data and further rebalances it
Splay Tree [7]	Implemented based on aspect of temporal locality
DSW Algorithm [22]	Does not require additional memory for translating a tree into its intermediate form
AA Tree [3]	Computation is simplified by omitting half of the tree which simplifies computation process
Treap [26]	Priority based BST is maintained
Random BST [29]	Rebalances portions which have lost their balance

4. Conventional Models to Search an Object

An active search methodology toward finding objects with optimum or near-optimal values. The premise of variables obtained by indirect and more cost effective ways where the several analytical applications all over the substance consists of finding an object with optimal or near-optimal value for a x and y-property of significance within a given group of objects [6, 26].

4.1. K Nearest Neighbor Search (KNN)

The problem to identify a point in the “Nearest Neighbor Search (NN)” which is “nearest” to any input query for a given possible set of data points [12]. The “nearness” is calculated between any two possible points are assessed by attaining or providing a probable distance metric as shown in Figure 1 [7].

In the present day scenario all the vital problems in NN is considered to be a significant problem in most of the research areas that includes most of the computational geometry area which is limited [15], as per the attained computer vision implementation of data mining, core of machine learning and pattern recognition [14].

NN is considered to be the root of many applications that retrieve the data in distinct fields related to image ranking and information retrieval [31] then performing multimedia information retrieval [27] and data attained through audio information retrieval [8] system. When implemented in machine learning where the instance oriented learning is attained through the nearest neighbor classifier that purely relies on the implementation of NN [6].

The implementation of NN search is implemented by considering the sample set that consists of possible set of random samples say X such that $X = \{X_1, X_2, \ldots, X_N\}$ for all $X \in R^D$, where the item set comprises of query $q \in R^D$ that is implemented using or identifying the item sk is used to predict the nearest data node that comprises of distinct set possible values represented by q for distinct possible values represent metrics, M: D X D $\rightarrow$R q is denoted by $\text{argmin}_{x\in X}$ M(q,x) [33].

This problem comprises of naive solution that calculates the results by implementing the equation $M(q,x)$ for all datasets $\boldsymbol{x}\in$X, which generates a linear query time is represented by N and D constants ($O(D(N))$). One of the best approaches to reduce the complexity is briefly proposed by Silpa-Anan and Hartley [28] that provides the complexity attained in query time $O(2D\log N)$ and attained data preprocessing cost of $O(2D+1)$ these two follows the proposals [17, 19] as the complexity attained is ended up with the exponential value (or polynomial) value with possible D [14, 36].

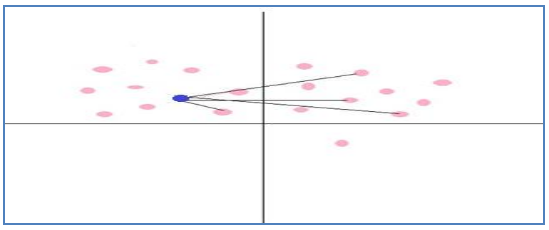


Figure 1. Implementation of two-dimensional search using KNN.

This condition can be illustrated as "the curse of dimensionality" which represents to be strict Nearest Neighbor [17] solutions that are unrealistic for D greater than 2 which is greater when considered [20].

The main drawback of KNN is the aspect of distance which is based on learning process as it is not clear with respect to computational cost is moderately higher because the computational process of measuring the distance for each of the query with various samples [21, 34].

4.1.1. Approximate of Nearest Neighbor Search Using Vector Quantization

"Vector Quantization (VQ) for Approximate Nearest Neighbor Search (ANN)" is search an element by using nearest of two points. The ANN algorithms propose to implement VQ for ANN based techniques consists of various datasets that store binary values for hashing methods [17].

However contrary to hashing [23] approximation of obtaining distances [11, 19] between various database samples over a query is performed by look-up tables instead of binary distances [9]. This gives a significant improvement in performance levels and the proposed gap approximation performance. The values are represented using VQ for ANN is being considered adequately well to retrieve and classify the tasks representing to replace the precise distance calculation different ways [10, 24, 28].

4.1.2. Attainment of Dimensional Using Approximate Nearest Neighbor Search

Because of the rapid growth in datasets that consists of descriptor cardinalities that are attained because of massive requirement of algorithms that can effectively execute smaller queries [8, 9]. By using the NN approach the value with possible solutions for implementing the nearest neighbor search [5] with best possible results uses ANN algorithm [9, 20]:

For the specific data items that are represented using M={x_1, x_2, … , x_n} where every possible sample that belongs to R, which is a single dimensional vector that finds or searches a sample set value that belongs to the neighborhood for a possible query q in a given metric [2]:

$$\text{M: } D\times D \rightarrow R \ \forall \ x\in X \text{ and } p\in P, \tag{1}$$

$$M(q, p) \leftarrow (1+\varepsilon)\, M(q,x) \tag{2}$$

For instigating NN methods over ANN which tends to decrease constraints based on the sample p which is the exact nearest neighbor over a given query q as most of the applications are employed based on core ANN for identifying the exact NN which tends to replaced by ANN [16]. And based on the Equation (2) illustrates the generic mathematical definition that is error bounded ANN [28] binds the time spent during the implementation of search is limited to a specific time bound. For imposing the additional constraints can be modified based on the use cases.

4.2. Search based on Partitioned Tree Structures

The "Partition trees" [9, 24] are the data structures that are most often used for searching an object by implementing the division of datasets hierarchically for organizing the dataset sample as each of the node directly corresponds with three sub partitions with mere possible dataset in a hierarchical organization structure. The structure implementation starts from root node that comprises of the whole dataset where each traverse is performed downwards on disjoint subsets. There is a chance that all leaf nodes may represent with either a single sample or a set of possible samples [8, 30, 32].

This method is briefly illustrated into categories that are mainly derived from distinct K-d trees which based on most are well known partition tree structure whose aim is to extend single dimensional binary search over multi dimensional data [4]. All the dimension are denoted as D vectors being stored in a K-d tree structure where each of the branch is divided into two distinct datasets by generating the median value based on threshold value provided. All sample vectors are denoted by a leaf node in a generated tree as illustrated in Figure 2.

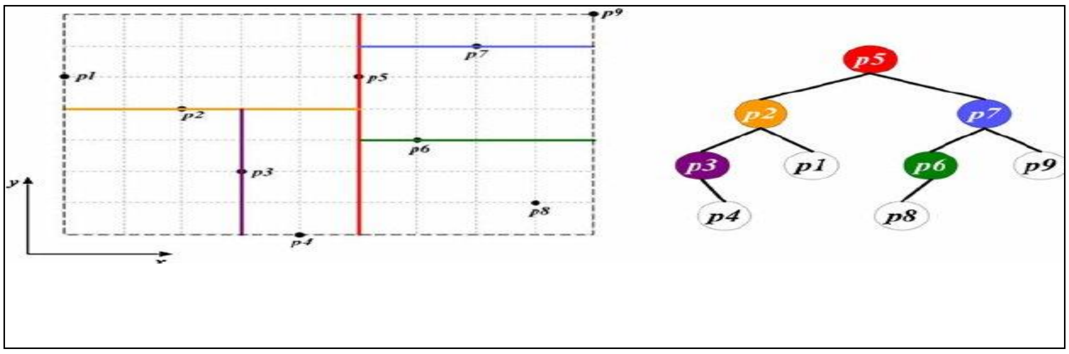


Figure 2. Two-dimensional K-d tree structure illustration.

The novel query that represents the values in probable selected dimensions that are compared for generating the corresponding threshold value is traversed in the tree according to the selected sub branch for all generated nodes [19, 35]. The binary split operation is implemented at every hierarchy level where the maximum depth of tree is bounded by $\log_2 N$.

The split operation is performed by using split or partition operation over selected dimension is efficiently implemented using K-d trees on datasets where $D > \log_2 N$. Most of the dimensions are discarded based on the rule of thumb K-d trees might be used efficiently only if N> 2^{2D} is efficiently implemented as specified [22].

4.3. Research Gap

The traditional object search methods are developed based on handcrafted features and insubstantial trainable architectures that are classified into three stages:

- Informative area section: as distinct objects are available in a cluster at distinct locations holding distinct values and it is a natural process to scan whole cluster using multi scan sliding window which revels all the objects along with their data and other disadvantage is due to bigger number of candidate windows it will become computationally expensive and creates many unnecessary windows.
- Feature Extraction: to identify distinct objects we need to extract all the features, which represents association between agents, though due to diversity in values it is merely possible to design a vigorous feature descriptor that defines all types of objects.
- Classification: a classifier has to identify the target object from all other object classes.

5. Proposed Model to Searching an Object in A Cluster (Mean-Based Search)

As per the literature in this Approach, the conventional algorithms or techniques that are considered are KNN and Partitioned tree search. Both these algorithms implement Euclidian distance measure for searching a data object in a data set will consume more amount of processing time because of iterations or CPU cycles required to implement search, as the whole data set is to be loaded before implementing the search operation. In other words, we need to create and dump the whole cluster before performing the search operation [26].

In KNN nearest neighbor is identified and searched for the data object whereas in partition tree algorithms median is calculated and all the data points near the median are considered and formed as a cluster. Then search is implemented on this cluster data.

The proposed model is implemented efficiently by finding or calculating mean of data by finding the mean value in every vector. Then the proposed system will implement the horizontal scaling and vertical scaling till lleaf, rleaf, tleaf, bleaf values are attained. And the search operation on the cluster data is implemented using “Mean based divisive clustering algorithm”, where the algorithm will search the data by performing same vertical and horizontal scaling [25, 30].

The advantage of proposed method over existing ones is superior because of vector quantization. Where the binary tree implementation of ANN will store and retrieve the data in a hierarchical method due to which the upper memory location is considered to be a wastage and whereas the proposed model will use the vectors concept which is a dynamic array and there won’t be any wastage of space in memory.

Algorithm 1: Mean-Based Search

```
Algorithm: Search- item ( Dct, $S_k$, N)
Input: Dtc= {I1, $I_2$,……………………………………..,$I_n$}
Output: $S_k$ id exist or not
Step 0: Begin
Step 1:   I←0
Step 2:   Flag ←0
Step 3:   Dct ←{ $I_1$, $I_2$, …………………..,$I_n$}
Step 4:   Nro_C ←n
Step 5:   CI ←n
Step 6:   Initial sort All the items of Cluster DCT { $I_1$, $I_2$,…..,$I_n$}
Step 7:   Find mean of { $I_1$, $I_2$,…………..,$I_n$} such that mean =$i_m$
Step 8:   Split cluster when $S_k$ compare with $i_m$
Step 9: If ($S_k$ < $i_m$) then find $S_k$ in { $I_1$, $I_2$,……….,$I_m$}, then goto
step 10, else goto step 12
Step 10: When found print $S_k$ exist in cluster and exit.
Step 11: Cj ←$I_{m-1}$
Step 12: If ($S_k$ > $i_m$) then find $S_k$ in { $I_{m+1}$, $I_{m+2}$, ……..,$I_n$}, then
goto step 13, else goto step 15
Step 13: When found print Sk exist in cluster and exit.
Step 14: Ci ←$I_{m+1}$
Step 15: If flag is even, then goto step 16, else goto step 18
Step 16: select_clust($C_{ij}$top, $C_{ij}$bottom)
Step 17: select_clust($C_{ij}$left, $C_{ij}$right)
Step 18: I ← I + 1
Step 19: if Nro_C <= 1, then got step 6, else goto step 20.
Step 20: End.
```

Referring the above algorithm there is a vital scope to execute some of the steps in a iterative or parallel way in a distributed environment where some of the possible steps that can be executed parallel are briefly defined below in following steps:

In step 2 we initialize the item value to zero, stating that the item set is empty. In next step that is step 3 we take the dataset objects that are numerous in number from database into Dct. In next step, we have defined Nro_c a parameter represent the number of objects that are loaded into the memory that takes into consideration for example objects such as {A, B, C, D, E, F, G, H}. We take the search key as input parameter to the function, we compare the search value with the mean value, and if it is same then we display a message stating the value is found in cluster number.

If the mean value is not equal to the search key S_k then we need to move to top or bottom if we select horizontal scaling, we need to search in left or right if we select vertical scaling based on the flag value. If no elements are left over that is I > n condition is met then we will stop executing the search operation and display search key not found.

As per the Figure 3, partitions are created by primarily calculating the mean value of the whole data set in a vertical manner and the root node is created. Then in both the partitions further mean values are calculated and partitions are formed, along with which a node is created in the tree in both the partitions. And this process is continues until the last partition is performed and node is created and leaf nodes are generated. Tree comprises of M1, M4, M8, and M9 as the leaf nodes and M5 is the root node. The whole data set is divided into one or more vectors based on the size of data set. In a vector full of data the mean value is being calculated.

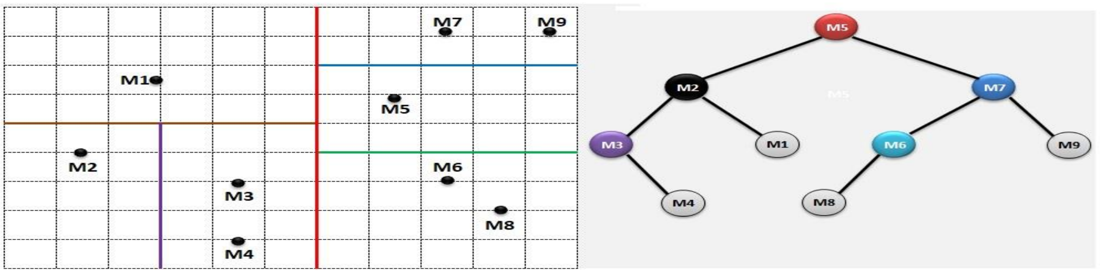


Figure 3. Illustration of mean-based search in particular cluster.

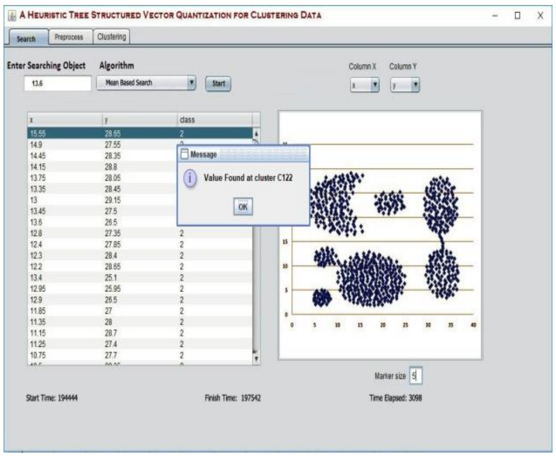
Then if required a vector can be sub divided into sub vectors based on the conditions implied on the clusters. Most of the times, this approach is also used for efficiently eliminating the outliers in a cluster or group of clusters.

Search operations are implemented on unique attributes in the cluster that are stored in the form of vectors. Initially prediction of primary attribute is to be performed then based on that attribute; the above procedure is implemented on that attribute identified uniquely.

The search operation is performed on the mean value attained from the overall cluster then primarily horizontal scaling is implemented then vertical either scaling and so on till the value is matched or elements are finished.

6. Results

The below results represents the results attained while searching the data in a data set that comprises of millions of transactions or data objects using Mean Based Search where a dialog box will be displayed illustrating whether the value is found or not found in the dataset through Figure 4.


A Heuristic Tree Structured Vector Quantization for Clustering Data
Search
Preprocess
Clustering
Start
OK

Enter Searching Object: 13.6
Algorithm: Mean Based Search
Column X: x
Column Y: y

x	y	class
15.55	28.65	2
14.9	27.55	2
14.45	28.35	
14.15	28.8	
13.75	28.05	
13.35	28.45	
13	29.15	
13.45	27.5	
13.6	26.5	
12.8	27.35	2
12.4	27.85	2
12.3	28.4	2
12.2	28.65	2
13.4	25.1	2
12.95	25.95	2
12.9	26.5	2
11.85	27	2
11.35	28	2
11.15	28.7	2
11.25	27.4	2
10.75	27.7	2

Message: Value Found at cluster C122

Marker size: 5

Start Time: 194444 Finish Time: 197542 Time Elapsed: 3098

Figure 4. Implementation of the proposed Algorithm for searching the data object.

The above figure represents the result attained while searching the data 13.6 and it is found in the cluster C#122. The process of searching can be illustrated using the below example by taking the previous chapter 5 example data set that comprises of values such as {66, 59, 1, 22, 8, 7, 6, 17, 16, 11, 9, 76, 5, 3, 19, 38, 54, 15, 88, 4, 75} where we will try to search the value 7 can be searched using the process: Value 7 is compared with the media value which is 66, it is compared with the median as the value is lesser than median then the search moves to the top partition then it is compared with 4 as it is lesser, it is compared with the right partition value which is 1, as the value is greater than 7 is compared with bottom partition whose value is 22, then 7 is compared with right partition as the value 8 is greater than 7, it is compared with the top partition value which is 7, hence the value is same or equal the algorithm displays a message stating value found in its cluster number.

The Table 2 represents the time consumed for searching a value in spatial dataset where the start time and end time are represented and time elapsed is calculated which value is leaser than the previous algorithm as the elapsed time is lesser for searching a value than that of creation step. And the attribute "is_found" represents 1 for value found and 0 for value not found.

Table 2. Processing time attained for search being implemented.

Spatial Dataset (Millions)	Start Time	End Time	Time Elapsed	is_found
1.2	12862	15000	2138	1
2.1	16012	19563	3551	1
3.2	20845	26325	5480	0
4.1	27768	34422	6654	1
5.2	36221	44563	8342	0

The Table 2 is graphically represented in the below figure where "is_found" attribute or property is not considered as the value consists of 0,1 which is almost ignored by the chart that is generated and represented same in Figure 5.

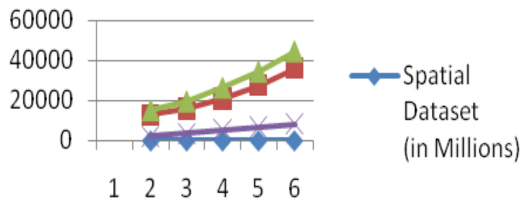


Figure 5. Results generated based on the table attined.

To illustrate further performance of proposed algorithm we have implemented it on KDD 99 dataset by adding 5656 additional records, as KDD comprises of four types of identified attacks: the Denial of Service (DOS), the Remote to Local (R2L), the User to Root (U2R) and Probing (Probe) attacks and further these are depicted in Table 3.

Table 3. Reduced training and testing dataset used in this work.

Set	DOS	R2L	U2R	Probe	Total
Training	55483	982	45	2056	58566
Testing	22123	2015	204	2569	26911

And the following results have been obtained by comparing proposed algorithm with other existing classifiers.

Table 4. Overall detection error rates.

Class	Proposed	Single Link	Complete Linkage	Average Linkage	Average Weighted
DOS	5.15%	6.89%	10.59%	8.32%	6.88%
Probe	3.87%	7.25%	11.98%	7.95%	6.57%
U2R	8.11%	18.44%	36.75%	8.45%	9.89%
R2L	11.14%	19.12%	25.14%	9.75%	46.89%

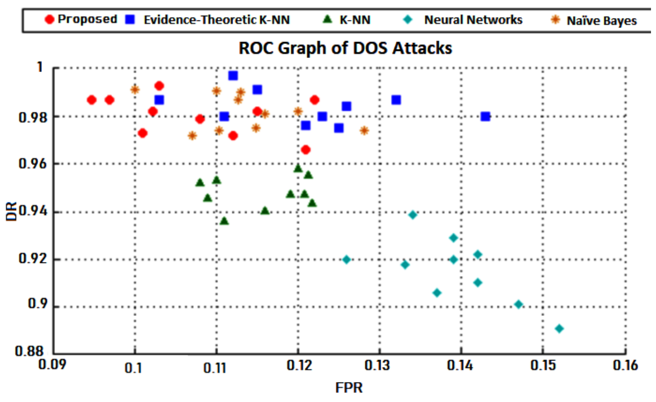


Figure 6. ROC plot for detecting DOS attacks to search object with proposed method.

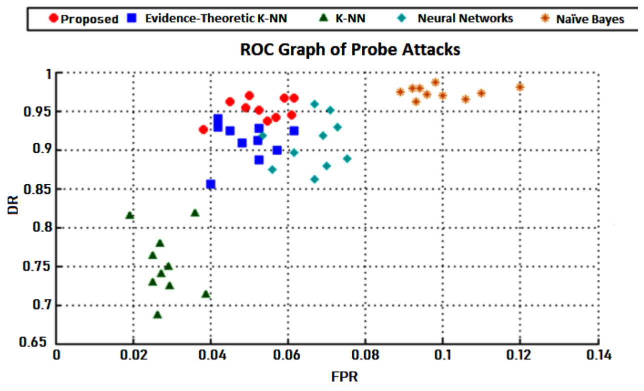


Figure 7. ROC plot for detecting Probe attacks to search object with method.

Table 4 illustrates overall detection error rates that effects both DR and FPR and it represents that the proposed algorithm is better in detecting attacks which are considered as agents.

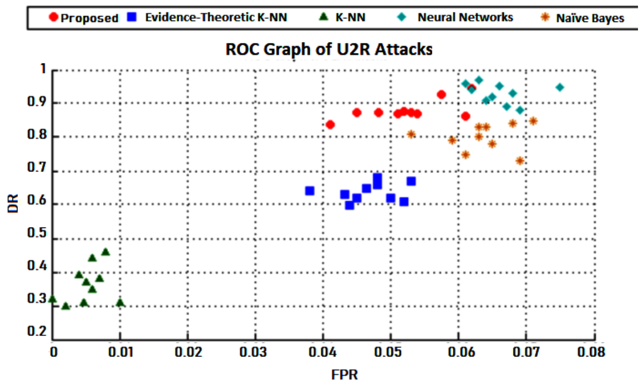


Figure 8. ROC plot for detecting U2R attacks to search object with proposed method.

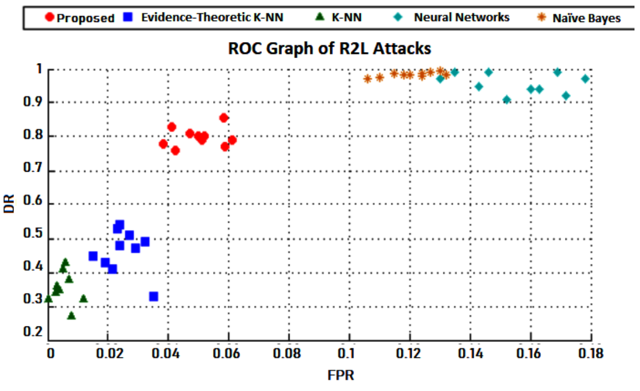


Figure 9. ROC plot for detecting R2L attacks to search object with proposed method.

And the time complexity of the proposed algorithm is O(logn) because in binary search tree cost of best case analysis is O($\log_n$) and all the previous algorithms are based on binary search tree with average case analysis O($n\log_n$) as our algorithm acquires O($\log_n$).

The detection process of U2R and R2L are more challenging and are unsatisfactory with existing classifiers than that of DOS and Probe attacks. As shown in Figures 6, 7, 8, and 9 the detection rate of all the four types considered are significantly improved.

7. Conclusions

In this Approach consists of the implementation of an algorithm search based on the scaling that includes lleaf (left leaf), rleaf (right leaf), tleaf (top leaf), bleaf (bottom leaf) where the sequence or an algorithmic step to be followed will be provided in detail as the algorithm works, what are the parameters that are based on Mean based divisive clustering which is efficiently includes in a particular clusters and finally how to enhance the computational complexity for handling huge data sets.

The proposed methodology yields best accuracy than existing models as all the existing models use Euclidian distance with BST algorithms for producing clusters.

The results illustrates overall error detection rates in generating the clusters for DOS attack 5.15%, Probe attack 3.87%, U2R attack 8.11% and R2L attack 11.14%. As these error detection rates denotes that our proposed algorithm generates less error rates than existing linkage methods.

Proposed algorithm generates clusters by using Mean based divisive algorithm whose advantage of over Euclidian is lesser comparisons and simpler cluster creation process.

References

[1] Adel'son-Vel'skii M. and Landis E., "An Algorithm for the Organization of information," Accession number: AD0406009, 1962.

[2] Akeem O., Ogunyinka T., and Abimbola B., "A Framework for Multime- Dia Data Mining in Information Technology Environment," *International Journal of Computer Science and Information Security*, vol. 10, no. 5, pp. 69-77, 2012.

[3] Andersson A., "Balanced Search Trees Made Simple," *in Proceeding of the Workshop on Algorithms and Data Structures*, Montreal, pp. 60-71, 993.

[4] Baeza-Yates R. and Ribeiro-Neto B., *Modern Information Retrieval*, Wesley Longman Publishing Co, 1999.

[5] Bayer R., "Symmetric Binary B-trees: Data Structure and Maintenance Algorithms," *Acta Informatica*, vol. 1, no. 4, pp. 290-306, 1972.

[6] Bentley J., "Multidimensional Binary Search Trees Used for Associative Searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509-517, 1975.

[7] Chang H. and Iyengar S., "Efficient Algorithms to Globally Balance a Binary Search Tree," *Communication of ACM*, vol. 27, no. 7, pp. 695-702, 1984.

[8] Day A., "Balancing a Binary Tree," *Computer Journal*, vol. 19, no. 4, pp. 360-361, 1976.

[9] Duro D., Franklin S., and Dubé M., "A Comparison of Pixel-based and Object-based Image Analysis with Selected Machine Learning Algorithms for the Classification of Agricultural Landscapes Using Spot-5 {HRG} Imagery," *Remote Sensing of Environment*, vol. 118, no. 3, pp. 259-272, 2012.

[10] Eamani R., VinodhKumar N., and Jakkamsetti G., "K-Means Clustering Algorithm and Architecture: A Brief Survey," *International Journal of Advanced Science and Technology*, vol. 29, no. 06, pp. 2955-2967, 2020.

[11] George A., "Efficient High Dimension Data Clustering using Constraint-Partitioning K-Means Algorithm," *The International Arab Journal of Information Technology*, vol. 10, no. 5, pp. 467-476, 2013.

[12] Jing X., Bi Y., and Deng H., "An Innovative Two-Stage Fuzzy KNN-DST Classifier for Unknown Intrusion Detection," *The International Arab Journal of Information Technology*, vol. 13, no. 4, pp. 359- 366, 2016.

[13] Khan S. and Khayal M., "A Survey on Maintaining Binary Search Tree in Optimal Shape," *International Conference on Information Management and Engineering*, Kuala Lumpur, pp. 365-369, 2009.

[14] Kononenko I. and Kukar M., *Machine Learning and Data Mining*, Horwood Publishing, 2007.

[15] Kumar R., Reddy B., and Pappula P., "An Evaluation of Feature Selection Algorithms in Machine Learning," *International Journal of Scientific and Technology Research*, vol. 8, no. 12, pp. 2071-2074, 2019.

[16] Martin W. and Ness D., "Optimizing Binary Search Trees Grown with a Sorting Algorithm," *Communication of ACM*, vol. 15, no. 2, pp. 88-93, 1972.

[17] Praveen P. and Rama B., "An Efficient Smart Search Using R Tree on Spatial Data," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 4, 2017.

[18] Praveen P. and Rama B., "An Optimized Clustering Method To Create Clusters Efficiently," *Journal of Mechanics of Continua and Mathematical Sciences*, vol. 15, no. 1, pp. 339-348, 2020.

[19] Praveen P. and Rama B., "A Novel Approach to Improve the Performance of Divisive Clustering-BST," *Data Engineering and Intelligent Computing*, vol. 542, pp. 553-562, 2018.

[20] Praveen P. and Babu J., "Big Data Clustering: Applying Conventional Data Mining Techniques in Big Data Environment," *Innovations in Computer Science and Engineering*, pp. 509-516, 2019.

[21] Rama B., Praveen P., Sinha H., and Choudhury T., "A Study on Causal Rule Discovery with PC Algorithm," *in Proceedings of the International Conference on Infocom Technologies and Unmanned Systems*, Dubai, pp. 616-621, 2017.

[22] Rauber A., Merkl D., and Dittenbach M., "The Growing Hierarchical Self-organizing Map: Exploratory Analysis of High-dimensional Data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 13, no. 6, pp. 1331-1341, 2002.

[23] Shaik M., "A Survey on Text Classification methods through Machine Learning Methods," *International Journal of Control and Automation*, vol. 12, no. 6, pp. 390-396, 2019.

[24] Shaik M., Praveen P., and Prakash R., "Novel Classification Scheme for Multi Agents," *Asian Journal of Computer Science and Technology*, vol. 8, no. S3, pp. 54-58, 2019.

[25] Shaik M., "Time Series Forecasting Using Vector Quantization," *International Journal of Advanced Science and Technology*, vol. 29, no. 4, pp. 169-175, 2020.

[26] Sheshikala M., Rao D., and Prakash R., "Computation Analysis for Finding Co-Location Patterns using Map-Reduce Framework," *Indian Journal of Science and Technology*, vol. 10, no. 8, pp. 12-19, 2017.

[27] Seidel R. and Aragon C., "Randomized Search Trees," *Algorithmica*, vol. 16, no. 4, pp. 464-497, 1996.

[28] Silpa-Anan C. and Hartley R., "Optimised KD-Trees for Fast Image Descriptor Matching," *in Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, pp. 1-8, 2008.

[29] Sleator D. and Tarjan R., "Self Adjusting Binary Search Trees," *Journal of the Association for Computing Machinery*, vol. 32, no. 3, pp. 652-686, 1985.

[30] Stout Q. and Warren B., "Tree Rebalancing in Optimal time and Space," *Communications of the ACM*, vol. 29, no. 9, pp. 902-908, 1986.

[31] Tamilarasi A., Abarna A., Chitra K., Nagendhiran K., and Aarthi R., "Effective Data Clustering Using K Means Along with Lion Optimization Algorithm International," *Journal of Advanced Science and Technology*, vol. 29, no. 4, pp. 3835, 2020.

[32] Vinod P. and Maple C., "Maintaining a Binary Search Tree Dynamically," *in Proceedings of the 10th International Conference on Information Visualization*, London, pp. 483-488, 2006.

[33] Wang J., Wang N., Jia Y., Li J., Zeng G., Zha H., and Hua S., "Trinary-Projection Trees for Approximate Nearest Neighbor Search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 388-403, 2014.

[34] Wittek P., Clustering Structure and Quantum Computing. Quantum Machine Learning, Academic Press Elsevier, pp. 99-107. 2014.

[35] Wittek P., Unsupervised Learning. Quantum Machine Learning, Boston: Academic Press, pp. 57-62. 2014.

[36] Yadav S., "An Efficient Affinity Propagation Clustering Technique," *International Journal of Advanced Science and Technology*, vol. 29, no. 4, pp. 9555-9573, 2020.

Praveen Pappula received his Ph.D. degree in Computer Science from the Kakatiya University (India) in 2019. He is working as Associate Professor in Computer Science and Artificial Intelligence at SR University since 2007. He has published more than 40 referred research papers, 1 book, and has 5 patents filled and pre published. His interests includes Machine Learning, Algorithms Analysis, Data Mining and Programming Languages.