https://doi.org/ 10.33472/AFJBS.6.9.2024.2182-2195

AFJBS

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

An Algorithm for Mining and Prediction of Medical Dataset Based on Dynamic Threshold Value Rule

Abu Sarwar Zamani[1*], Aisha Hassan Abdalla Hashim[2], Sumbul Alam[3], Ahmad Talha Siddiqui[4]

[1,2]Department of Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia.

[3]Department of Computer Science Engineering, Presidency University, Bengaluru-560064, Karnataka, India.

[4]Department of CS & IT, Maulana Azad National Urdu University, India.

Corresponding author: Sarwar_zamani@yahoo.com

Article History

Volume 6,Issue 9, 2024

Received:21 Mar 2024

Accepted : 30 Apr 2024

doi: 10.33472/AFJBS.6.9.2024.2182-2195

Abstract— The rule-based classification algorithm now includes a dynamic threshold value. The significance of threshold values, classification algorithms that use threshold values, issues with setting static threshold values and dynamic threshold values, and rule generation. However, for some datasets, it is necessary to have a large number of rules, and for those datasets, a greater number of rules should be extracted, which affects reduction of generalisation and makes the system less transparent. Logic rules framed for small datasets have minimum number of rules that interns are very easy to understand and compare. One of the best, most flexible solutions is to frame fuzzy logic rules; however, fuzzy rule support is less for datasets with symbolic and nominal attributes. These alternative rules extraction systems are driven by similarity-based learning and are based on prototype rules, presented threshold rules algorithm, which uses a small number of highly precise ordered rules to extract threshold rules from data.

Keywords— Rule based algorithm, Fuzzy logic, Classification algorithm, Threshold, CMAR, CBA, KNN

1. Introduction

To best predict the class, classification algorithms heavily rely on the threshold value K (the number of cluster). The threshold value for K is used by the KNN and K-means clustering algorithms to predict the class [1]. The other classification algorithms employ fixed sensitivity and specificity threshold values that are calculated by an algorithm and verified to predict the final class [2]. To accurately predict the class and lower the rate of misclassification, the choice of the ideal threshold value must be made. Decision threshold values have been the subject of numerous studies and analyses. The various threshold concepts and various rules for the suggested classification algorithm are covered in this paper.

The creation of rules based on objects that appear frequently in a transaction is the fundamental idea behind an association rule. This involves two primary steps: identifying the frequently occurring item set and creating rules. A group of items with a higher frequency of recurrence than the threshold value specified in the transaction is called a frequent item set. The minimal support is another name for this number. The availability of numerous, diverse, and heterogeneous data sources poses a number of obstacles for the real-time applications of the association rule, including the intricacy of determining the value of minimum support [13]. Additionally, establishing the minimal support value is the initial stage in the association rule [14, 15]. The user currently chooses this value. Right now, the user chooses this value. Given that the user is thought to be the dataset's most knowledgeable source, this is done. The amount and extent of the intended output can also be limited by the user. The value of minimum support is actually something that many users find challenging to determine.

The value of various rules that are generated is significantly influenced by the minimum support value [16]. The intuitiveness of the user is the basis for the method used to calculate the value of minimum support. If the rule acquired does not match, the process of creating an association rule is repeated by changing the minimum support value. This does not, however, ensure that a rule with the input minimum support value will be produced.

The evaluations in this study, which suggests a way to calculate the minimal support value depending on the dataset's properties and additional factors, can also be applied to the process of creating the rule. The minimal support value in the suggested technique is automatically determined based on the dataset's features, negating the need for the user to decide it at the outset.

Furthermore, other characteristics that are taken into consideration throughout the rule-formation process are also taken into account when determining the minimum threshold value. This is in addition to the frequency of occurrence of items. Using this approach, the process of creating rules becomes more user-responsive. The suggested approach is limited to the step of figuring out the lowest threshold required to choose often occurring item set. Several existing algorithms can be used in the rule formation process once the frequent item set has been produced. By using the suggested approach, the user is spared the trouble of figuring out the minimum support value and is forced to repeatedly go through the rule-formation process due to improper minimum support value selection.

2. Related Work

The user must first define a minimum support value according to the association rule's core notion. Although in practice various things may have varied criteria for assessment, this value often applies consistently to all of them. Therefore, research on multiple minimum support has emerged, stating that distinct items should have various support values [16, 17, 18, 19]. Finding the minimal support for each item is a new task that the user must perform as a result of the system's implementation.

Applying association rules might be challenging at times, especially when deciding on a minimum support value. This is due to the fact that most approaches make the assumption that every database item is similar and occurs frequently. This assumption, however, is false since certain entries appear in the database more frequently than others [20].

The user estimates these parameters intuitively because current methods, like apriority and fpgrowth, are unable to identify the minimal support and threshold values. Due to its ability to produce a huge number of rules, the association rule mining algorithm might suffer from lengthy execution times and high memory consumption, and vice versa. But this depends on the selection of the threshold [16].

Apriori-based mining methods, a frequent and appealing item set, were developed as a result of users' difficulty setting minimal support. This is a difficult issue since the algorithm's performance heavily depends on a user-defined threshold. For instance, if the minimum support value is set too high, the database will be empty. Conversely, low minimum support causes inadequate mining performance and an abundance of undesirable association rules. This means that consumers are being unreasonable in their requests for information about the characteristics of the database to be mined and the appropriate threshold. Results were not in line with customers' needs, even though the minimum assistance was investigated under the guidance of seasoned miners [21].

Zhang [21] conducted a study whose primary contribution was to offer a method for transforming ambiguous (user-defined) thresholds into real minimal support. Therefore, building a conversion function requires a technique that can identify certain elements of the database to be mined. When utilizing current Apriori algorithms, users still need to specify the real minimum support corresponding to the database to be mined. It is impossible to determine the minimal support that matches the database, though, without the right information. Zhang suggested a computational approach to solve the minimum support settings issue. This approach is different from the current Apriori algorithm in that it lets users specify their mining requirements in a mode that is often used and automatically translates the given threshold into the real minimum support.

In order to examine the Semi-Apriori method, Trivedi [20] integrated the average support threshold. Subsequently, a frequently occurring item set was created by employing an automatically constructed support threshold to assess the data. This lowers the complexity of both space and time.

A technique for choosing a suitable minimum threshold value for efficient support was created by Dahbi [16]. The first contribution was that this study automatically calculated the minimum support (minsup) for each data set, rather than relying on user-defined constant values. As this was going on, the second updated this minsup dynamically by giving each level a single, uniform minimum support threshold. That being said, not every item in an item set functions in the same way; some were used regularly, while others were not. The minsup threshold must therefore change based on the item level.

3. Methodology

3.1 Threshold Value On Classification Algorithm

The majority of classification methodologies use the threshold value in a variety of ways to determine the final class. Many classification algorithms predict the class using the threshold value and support and confidence values [3, 22]. The threshold value must be compared with by the algorithm that predicts the final class using the support and confidence values.

The threshold value serves as a class boundary value. The accuracy of the classifier is improved by the classification algorithm KNN, which uses various modified threshold value as a key value. It is one of the most straightforward and widely used classification models, using threshold value to foretell the next-closest value. In contrast to most others, KNN does not require any training phases, making it a quick classification method. Using a distance metric like the Euclidean distance, k training samples are obtained for each test sample in this method [4, 24]. Within these k samples, the test sample's class is determined by majority voting.

With support and confidence threshold values, the proposed algorithm is compared to rule-based algorithms like CMAR, CBA, and C4.5 [5]. The implementations of these three classification algorithms—CMAR, CBA—on various medical datasets with fixed threshold values are covered in this section. The Support and confidence to predict the class are calculated by the Association rule-based algorithm. [6, 23] How frequently the items appear in the database is indicated by their support. The number of times the if/then statements have been verified as true is indicated by confidence.

Equation 1 is used to calculate the support in the Association rule mining.

$$Support\ =\ occurrence\ of\ the\ instances\ /\ Total\ support \qquad (1)$$

and the confidence is calculated for given x=> y using the equation 2

$$Confidence =\ occurrence\{y\}\ /\ occurrence\{x\} \qquad (2)$$

The authors in [7] developed the new associative classification method known as CMAR, or Classification based on Multiple Association Rules, in 2001. The author of this algorithm used the confidence values and support threshold value to predict the class as CBA. The confidence difference threshold is set to 20%, and the database coverage threshold is set to 80%. The author disabled the cap on the number of rules for CBA and set the support threshold to 1% and confidence threshold to 50%. Other settings are left at default. It only reports the accuracy for the rule method because it is more accurate.

3.2 Threshold Value Fixing Problem

When the closest value to the threshold is obtained, the threshold value-based classification algorithm predicts the class. It is difficult to predict the class in a multiclass classification when two or more classes have the same nearest distance to the threshold value. The nearest neighbor concept is used by the k-NN. In our study, the rule-based classifier's threshold value is determined by the nearest neighbor. The optimum value is used to predict the class rather than the approximate value.

The threshold value is necessary for the classification algorithms that use support and confidence values to compare and accurately predict the class. In those algorithms, the support and confidence threshold values are predetermined by the user and are not altered as the algorithm is being run. The user-set threshold is regarded as static because it is constant. The classification accuracy will rise with the change in threshold value's percentage. The greatest value out of the n values is selected as the threshold value for the modified threshold value, which is dependent on the determination of the support factor for each class. The association rule's support and confidence equation is used to calculate the support factor, which results in the formation of the following equation 3.

$$Support\ Factor\ =\ number\ of\ factors\ satisfies\ the\ rule\ /\ total\ factors \qquad (3)$$

And the support factor percentage is calculated using the equation 4.

$$Support\ percentage\ =\ Support\ factor/100 \qquad (4)$$

The accurate threshold percentage should be set so that the classifier is as accurate as possible. Choosing and establishing an accurate threshold value is another difficult task in obtaining the best classification percentage.

The dataset, which has "m" attributes and "n" classes that need to be predicted, does not require that all "m" attributes be checked in order to predict all "n" classes. The classification time will be shortened by identifying the attributes associated with the particular class. Certain attributes do not need to be checked, and doing so will prolong the classification process. Only a small subset of the "m" attributes that support class C1 must be taken into account in order to predict class C1, and only a small subset of other attributes must be taken into account in order to predict class C2. Certain characteristics that support class C1 also support class C2, so those characteristics must also be considered when predicting class C1.

The number of attributes for each class will vary, and different numbers of attributes must be checked in order to predict each class. Additionally, the threshold value percentage used to determine the class in each instance will vary. Therefore, it is debatable whether the threshold value should be set as a constant. The support value with the closest value is predicted as the resultant class when the threshold value is fixed as a constant. Another issue arises when two or more classes have percentages that are close to the fixed threshold value, leading to a tedious class as a result. Example: Consider a multi-class problem where the threshold value is fixed at 80% and there are two classes, each with a percentage of 70% and 90%, and both classes are within the same distance of the threshold value. Therefore, predicting the resultant class becomes difficult. The concept of nearest value does not result in the best class prediction; instead, the best threshold value is required to produce an accurate class prediction.

Fixing a static threshold value will have an impact on the accuracy and performance of sensitive datasets [9,10]. The characteristics specific to those cases alone will be taken into account more than the other characteristics in order to determine the cause of a given case. Using any of the statistical techniques, the threshold values were established, and the fixed value is static. The algorithm checks each class' attributes and calculates its support counts when there are a total of two classes in the sample, C1, C2, and when the threshold value is fixed as some constant value "N%". If it reaches the threshold value "N%", the algorithm fixes the cause for the specific case as C1, otherwise it fixes the cause as C2. Fixing a static threshold value raises the wrong prediction rate when there are multiple classes (C1, C2, C3..., Cn). When the attributes for each class vary in size according to its attributes, the mean value for each case must be calculated separately.

Setting the proper threshold value will help you predict the class. A difficult task is selecting an accurate threshold value for an algorithm to predict the class [12]. The threshold value has been set using a variety of methodologies. In some circumstances, the minimum requirements must be met in order to predict the class. Depending on the user constraint, that minimum value may need to be fixed. However, the minimum constraint does not always work. Particularly for medical datasets, the maximum support should be counted when predicting a record's class. When the minimum threshold is set, there is a chance that the class will be predicted incorrectly, which can result in misclassification and an increase in the misclassification rate. In order for multiclass classification to correctly predict the class, a dynamic threshold value is required.

Considering these limitations in our research, the threshold value is established by treating the threshold value as the maximum acceptable value rather than the minimum. Additionally, the algorithm will need to calculate that maximum satisfactory value repeatedly because it will vary for each record. The rule set is applied to the dataset, and for multiclass classification, the maximum acceptable value is calculated for each class and fixed as a threshold value to predict the class. The correctly predicted class for the test dataset is compared using the training dataset as a prototype. Here, we're using the p-rule as our classification basis. However, the algorithm uses a dynamic threshold value and includes both P-rules and f-rules.

3.3 Dynamic Threshold Value

The nature of medical data sets is more delicate. It's critical to predict and categorize the root causes of each disease. The classes of medical datasets are predicted using the rule-based algorithm. For a rule-based algorithm to perform an accurate classification, a threshold value is required. When a threshold value for a classification is fixed statically, the algorithm declares the classification complete when it reaches the fixed value. For categorization into binary classes, it works well. The predicted outcome is impacted by fixing a static threshold value for the dataset with "N" classes. For medical data sets with multiple classes, a dynamic method of setting threshold values is especially important.

By setting the dynamic threshold value, the performance of classification is improved and the number of wrong classifications is cut down. The threshold values for sensitive data are modified periodically for each record. To accurately predict the class, a rule-based algorithm with a dynamic threshold value is required. The modified RBA algorithm is applied in two stages; in the first stage, the dataset is preprocessed using the discretization concept to reduce dataset complexity.

In the second phase, the modified RBA with predefined rules is put into use for classification. On various medical datasets, we implement the Rule Based Classification Algorithm with Dynamic Threshold Value in this paper, and its performance is assessed. Our implementation demonstrates that, on various medical datasets, the enhanced RBA with a dynamic threshold value outperforms other approaches.

When using a rule-based algorithm, the number of instances that satisfy the given rule is counted after the rule is applied to the data set. Here, when the threshold value is fixed as a constant, it ends the class when it reaches that value, which causes misclassification. The threshold value cannot be fixed as a static one when sensitive data uses multiple rules to predict multiple classes when using rule-based classification. whenever the threshold value is switched. As a result, the proposed rule-based algorithm applies multiple rules to the test data to determine the percentage that each class perceives. For each situation, a different percentage is received. Each record results in changes to the threshold value, and the class is predicted using Euclidean distance metrics.

When a rule-based algorithm is used for multiclass classification, it predicts that class C1 will come out on top with a threshold value of 60%. This threshold value serves as the first instance's cutoff value, and all other classes receive percentages that are lower than C1's threshold value. The algorithm predicts that the resultant class of the second instance will be c1 with a support factor value of 70% or 50%, which is also the instance's threshold value. Less than the predicted value of the support factor will be received by the other n-1 classes. The threshold value was either higher or lower than the previous threshold value. The maximum support factor is fixed as a threshold value and varies for each instance. Here, the threshold value is established in relation to the percentage of n-1 classes with satisfactory factors.

4. Rule Generation

The proposed algorithm employs a dynamic threshold value and is designed using the p-rule and f-rule. There are numerous rule-based algorithms, such as CBA and CMAR, which use various kinds of rules and are linked to association and decision rules, respectively. The suggested algorithm generates rules using a fuzzy rule (F-rule) to handle complex queries and a proto type to check for similarity. To accurately predict the class using our algorithm, we also use the distance function.

The dynamic threshold-based classification algorithm that is proposed is made with the help of a set of rules. The rules are developed by researching already-existing rules. Different kinds of rules are employed to express various knowledge types. The Threshold Rules Decision List Algorithm, which supports binary classes, is the foundation of our research. The rule iteratively tests the dataset and fixes the threshold value to be constant. In our proposed algorithm, the standard rule sets are applied to the chosen dataset for multiclass classification, and the threshold value is instantly calculated for each class.

Algorithm for rule generation:

Input: D, a data set class-labeled tuples; Attvals, the set of all attributes and their possible values.

Output: A set of IF-THEN rules.

Method:

```
Step 1: Rule set = {}; // initial set of rules learned is empty
Step2: For each class c do
    Step2.1: Repeat Rule = Learn One Rule (D, Attvals, c);
        Step 2.1.1: remove tuples covered by Rule from D;
    until terminating condition;
    Step2.2: Rule set = Rule set + Rule; // add new rule to rule set
 End for
Return Rule Set;
```

Many data sets from the UCI Machine Learning Repository [8] were tested by the author. Database coverage threshold and confidence difference threshold are two crucial CMAR parameters. The total number of rules chosen for classification is limited by these two thresholds. These two thresholds produced nominal accuracy when applied to the dataset, indicating that it was impossible to predict the ideal threshold values beforehand.

Data Set	Attr#	Cls#	Rec	C4.5	CBA	CMAR
Wisconsin Breast Cancer	10	2	699	95	96.3	**96.4**
Cleveland heart disease	13	2	303	78.2	**82.8**	82.2
Crx heart disease	15	2	690	84.9	84.7	**84.9**
Pima Indian Diabetes dataset	8	2	768	74.2	74.5	**75.8**
Heart disease dataset	13	2	270	80.8	81.9	**82.2**
Hepatic	19	2	155	80.6	**81.8**	80.5

Table 1. Adaptive support mechanism tested on six datasets

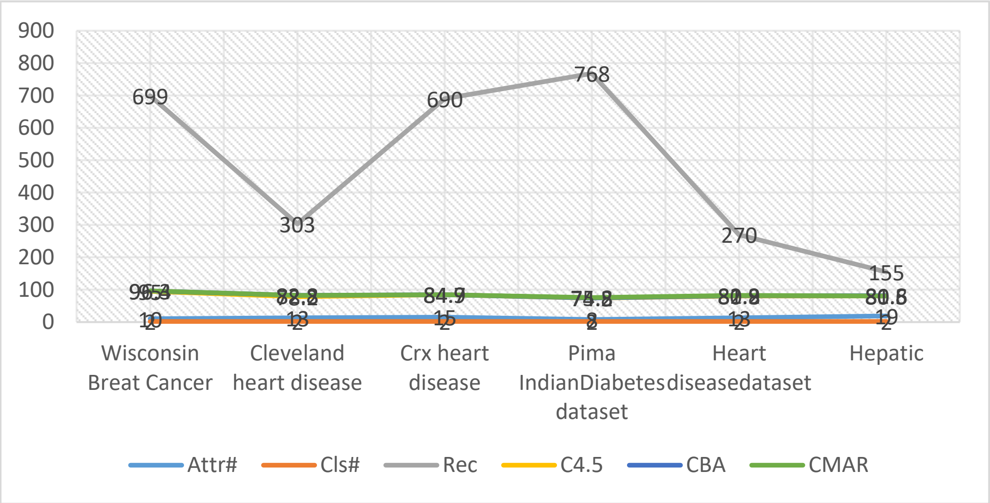

Figure 1. Accuracy on Medical Dataset

5. Conclusion

This paper explains the various rules that make up the rule-based algorithm and the significance of threshold values for classification. The difficulties with using the nearest neighbor concept and static threshold values, as well as the significance of dynamic threshold values. By setting the dynamic threshold value, the performance of classification is improved and the number of wrong classifications is cut down. The threshold values for sensitive data are modified periodically for each record. To accurately predict the class, a rule-based algorithm with a dynamic threshold value is required. The calculated threshold value is then used to draw conclusions about the resultant class. Table 1 and Figure 1 shows that the suggested adaptive support mechanism was tested on six datasets. Other criteria that were applied item-by-item in this study to establish the minimal threshold were the same. Moreover, the outcomes of all six datasets yielded rules with lift ratios greater than one and the suitable minimum support value.

References

1. Langfu, C. U. I., ZHANG, Q., Yan, S. H. I., Liman, Y. A. N. G., Yixuan, W. A. N. G., Junle, W. A. N. G., & Chenggang, B. A. I. (2023). A method for satellite time series anomaly detection based on fast-DTW and improved-KNN. *Chinese Journal of Aeronautics*, *36*(2), 149-159.
2. Peng, Y., Li, C., Wang, K., Gao, Z., & Yu, R. (2020). Examining imbalanced classification algorithms in predicting real-time traffic crash risk. *Accident Analysis & Prevention*, *144*, 105610.
3. Nusinovici, S., Tham, Y. C., Yan, M. Y. C., Ting, D. S. W., Li, J., Sabanayagam, C., ... & Cheng, C. Y. (2020). Logistic regression was as good as machine learning for predicting major chronic diseases. *Journal of clinical epidemiology*, *122*, 56-69.
4. Saadatfar, H., Khosravi, S., Joloudari, J. H., Mosavi, A., &Shamshirband, S. (2020). A new K-nearest neighbors classifier for big data based on efficient data pruning. *Mathematics*, *8*(2), 286.
5. Christopher, J. (2019, January). The science of rule-based classifiers. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 299-303). IEEE.
6. Dhanalakshmi, R., Anitha, K., Rukmani Devi, D., & Sethukarasi, T. (2021). Association rule generation and classification with fuzzy influence rule based on information mass value. *Journal of Ambient Intelligence and Humanized Computing*, *12*, 6613-6620.
7. Thanajiranthorn, C., &Songram, P. (2020). Efficient rule generation for associative classification. *Algorithms*, *13*(11), 299.
8. Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

9. W. Pedrycz,"Fuzzy set technology in knowledge discovery", Fuzzy Sets and Systems 98,pp.279-290.
10. Kacker, S., Meredith, A., Kusters, J., Tomio, H., Felt, V., & Cahoy, K. (2022). On-orbit rule-based and deep learning image segmentation strategies. In *AIAA SCITECH 2022 Forum* (p. 0646).
11. Yanwei.X, Wang.J, Zhao.Z, Gao.Y,"Combination data mining models with new medicaldata to predict outcome of coronary heart disease", Proceedings International Conference on Convergence Information Technology.
12. Wenmin Li, j.Han, Jian, pei, "CMAR: Accurate and Efficient Classification Based onMultiple Class-Association Rules", International conference on Data mining.
13. Zhang S, Wu X. Fundamentals of association rules in data mining and knowledge discovery: fundamentals of association rules. Wiley Interdiscip Rev Data Min Knowl Discov. 2011;1:97–116.
14. Boley M, Grosskreutz H. Approximating the number of frequent sets in dense data. Knowl Inf Syst. 2009;21:65–89.
15. Wazir S, Beg MMS, Ahmad T. Comprehensive mining of frequent itemsets for a combination of certain and uncertain databases. Int J Inf Technol. 2020;12:1205–16.
16. Dahbi A, Balouki Y, Gadi T. Using multiple minimum support to auto-adjust the threshold of support in apriori algorithm. In: Abraham A, Haqiq A, Muda AK, Gandhi N, editors. Proceedings of the ninth international conference on soft computing and pattern recognition (SoCPaR 2017). Cham: Springer International Publishing; 2018. p. 111–9. https://doi.org/10.1007/978-3-319-76357-6_11.
17. Krishnamoorthy S. Efcient mining of high utility itemsets with multiple minimum utility thresholds. Eng Appl Artif Intell. 2018;69:112–26.
18. Gan W, Lin JC-W, Fournier-Viger P, Chao H-C, Zhan J. Mining of frequent patterns with multiple minimum supports. Eng Appl Artif Intell. 2017;60:83–96.
19. Wu CW, Shie B-E, Tseng VS, Yu PS. Mining top-K high utility itemsets. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining—KDD ’12. Beijing, China: ACM Press. pp. 78. 2012. http://dl.acm.org/citation.cfm?doid=2339530.2339546. Accessed 26 Sept 2019.
20. Trivedi J, Patel B. An automated support threshold based on apriori algorithm for frequent itemsets. Int J Adv Res Innovative Ideas Educ. 2017;3(6):446-52.
21. Zhang S, Wu X, Zhang C, Lu J. Computing the minimum-support for mining frequent patterns. Knowl Inf Syst. 2008;15:233–57.
22. Jasti, V. D. P., Zamani, A. S., Arumugam, K., Naved, M., Pallathadka, H., Sammy, F., ... & Kaliyaperumal, K. (2022). Computational Technique Based on Machine Learning and Image Processing for Medical Image Analysis of Breast Cancer Diagnosis. Computational Technique Based on Machine Learning and Image Processing for Medical Image Analysis of Breast Cancer Diagnosis.
23. Saxena, K., Zamani, A. S., Bhavani, R., Sagar, K. V., Bangare, P. M., Ashwini, S., & Rahin, S. A. (2022). Appropriate Supervised Machine Learning Techniques for Mesothelioma Detection and Cure. BioMed Research International, 2022.

24. Zamani, Abu Sarwar; Rajput, Seema H.; kaur, Harjeet; Meenakshi; Bangare, Sunil L.; Ray, Samrat,Towards Applicability of Information Communication Technologies in Automated Disease Detection, International Journal of Next-Generation Computing . Oct2022, Vol. 13 Issue 3, p396-403. 8p.