International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 13, No. 3, November 2024, pp. 724~734
ISSN: 2089-4864, DOI: 10.11591/ijres.v13.i3.pp724-734

Precision medicine in hepatology: harnessing IoT and machine learning for personalized liver disease stage prediction

Satyaprakash Swain[1,2], Mihir Narayan Mohanty[3], Binod Kumar Pattanayak[1]
[1]Department of Computer Science and Engineering, Institute of Technical Education and Research, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, India
[2]Department of Computer Science and Engineering, Institute of Management and Information Technology, Biju Patnaik University of Technology, Odisha, India
[3]Department of Electronics and Communication Engineering, Institute of Technical Education and Research, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, India

Article Info

Article history:

Received Oct 25, 2023
Revised Mar 24, 2024
Accepted Apr 19, 2024

Keywords:

Internet of things sensors
K star
Multi-class classification
Naïve Bayes
Random forest
Sequential minimal optimisation

ABSTRACT

In this research, we used a dataset from Siksha 'O' Anusandhan (S'O'A) University Medical Laboratory containing 6,780 samples collected manually and through internet of things (IoT) sensor sources from 6,780 patients to perform a thorough investigation into liver disease stage prediction. The dataset was carefully cleaned before being sent to the machine learning pipeline. We utilised a range of machine learning models, such as Naïve Bayes (NB), sequential minimal optimisation (SMO), K-STAR, random forest (RF), and multi-class classification (MCC), using Python to predict the stages of liver disease. The results of our simulations demonstrated how well the SMO model performed in comparison to other models. We then expanded our analysis using different machine learning boosting models with SMO as the base model: adaptive boosting (AdaBoost), gradient boost, extreme gradient boosting (XGBoost), CatBoost, and light gradient boosting model (LightGBM). Surprisingly, gradient boost proved to be the most successful, producing an astounding 96% accuracy. A closer look at the data showed that when AdaBoost was combined with the SMO base model, the accuracy results were 94.10%, XGBoost 90%, CatBoost 92%, and LightGBM 94%. These results highlight the effectiveness of proposed model i.e. gradient boosting in improving the prediction of liver disease stage and provide insightful information for improving clinical decision support systems in the field of medical diagnostics.

Corresponding Author:

Satyaprakash Swain
Department of Computer Science and Engineering, Institute of Technical education and Research
Siksha 'O' Anusandhan (Deemed to be University)
Bhubaneswar, Odisha, India
Email: satyaimit@gmail.com

1. INTRODUCTION

Millions of people worldwide are affected by liver disease, which also places a heavy cost on healthcare systems [1] around the world. Accurate staging is required for effective care of liver diseases, in addition to prompt diagnosis, in order to direct the right clinical interventions. While reliable, conventional techniques of liver disease diagnosis have some drawbacks, particularly when it comes to assessing the severity and course of the disease. In this context, combining machine learning and internet of things (IoT) technology presents a viable path for enhancing patient care and diagnostic accuracy.

The goal of this study is to use IoT secure framework [2] and machine learning to predict the stages of liver disease [3], taking into account the important differences between stages 1 to 4. Also, this study investigates the viability of using a collection of machine learning models to accomplish this crucial diagnostic goal by leveraging a real time dataset of 6,780 samples which are obtained through a combination of manual sample tests and data collected through IoT sensors from the patients of Siksha ‘O’ Anusandhan (S’O’A) University Medical Laboratory, Bhubaneswar. The potential to improve liver disease diagnosis' precision and effectiveness, leading to better patient care and therapeutic results [4], is what spurred this Endeavour.

The real time dataset under investigation includes a wide range of 15 different parameters collected through IoT sensors that have been carefully chosen to capture a comprehensive picture of liver health, including the number of days the patient has been receiving treatment, ascites, age, sex, hepatomegaly, spiders, edoema, bilirubin, cholesterol, albumin, copper, alk_phospho, serum glutamic oxalacetic transaminase (SGOT), tryglicerides, platelets, and prothrombin. When taken as a whole, these parameters provide a comprehensive understanding of the clinical, biochemical and demographic profile of the patient. A liver disease's precise identification and assessment [5], [6] is made possible by the presence of specific clinical symptoms in conjunction with elevated levels of certain markers, such as SGOT and bilirubin, which can help with diagnosis and treatment. These metrics range from more modern data collected from IoT devices like in Figure 1, such as continuous monitoring of vital signs and other pertinent health indicators [7], [8], to more classic clinical measures, such as liver enzyme levels and bilirubin concentration. This study aims to utilise the complete range of information available for forecasting the stages of liver disease by combining the traditional and contemporary elements of healthcare data.

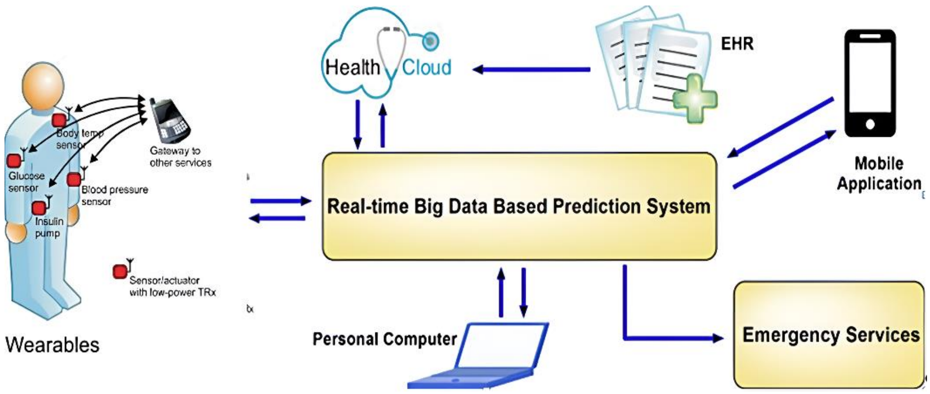

Figure 1. Cloud service in health care

Traditional machine learning models [9] like Naïve Bayes (NB), sequential minimal optimisation (SMO), K star (K*) and random forest (RF) are used to accomplish this predicting task. The predictive power of this study is derived from these models, which were chosen for their diversity and proven efficacy in classification tests. We dive deeper into the world of boosting strategies, however, as we are aware that predicted performance may always be improved. We intend to improve the accuracy and robustness of the chosen traditional models in predicting the stages of liver disease by employing boosting techniques like adaptive boosting (AdaBoost) and gradient boosting on the result traditional model.

This research makes a substantial contribution to the diagnosis and stage prediction of liver disease. The findings have the potential to fundamentally alter how liver disease is managed by giving clinicians a quicker and more precise tool for diagnosing and monitoring disease progression. Additionally, the incorporation of IoT sensors [10] into the diagnostic procedure highlights the possibility for ongoing patient monitoring, ushering in a new era of individualised healthcare (Figure 2). In bellow methodology, experimental plan and findings sections of this study, providing light on the effectiveness of conventional machine learning models and the revolutionary potential of boosting techniques which is used to forecast liver disease [11]. This research is essential for boosting patient outcomes and raising the standard of care for liver illnesses because of its potential effects on healthcare and the management of liver disease in the future.

It is a popular machine learning method called NB is extensively used for classification jobs [12]. This algorithm is based on the Bayes theorem and simplifies by assuming feature independence, even though this assumption might not hold true in real-world situations. Support vector machines (SVMs), which are frequently used in machine learning for classification and regression problems [13], are trained using the specialised technique SMO. The quadratic optimisation problem that arises during SVM training must be efficiently resolved and SMO is essential to this process. K* is a k-nearest neighbour (KNN)-based instance-based classifier. It makes an effort to cluster k data points into n data points. K* uses an entropic distance measurement depending on the probability of transforming one occurrence into another. RF is another ensemble learning method that is used for a variety of tasks, including classification and regression. During the training phase of this approach, several decision trees are created. The class that appears the most frequently throughout the trees is chosen to produce the RF output in classification scenarios. A useful indicator for evaluating the effectiveness of classification models, particularly in binary classification scenarios, is the matthews correlation coefficient (MCC) [14]. It evaluates the model's ability to predict outcomes by taking into account crucial elements like true positives (cases that were correctly predicted as positive), true negatives (cases that were correctly predicted as negative), false positives (cases that were incorrectly labelled as positive when they were negative) and false negatives (cases that were incorrectly labelled as negative when they were positive).

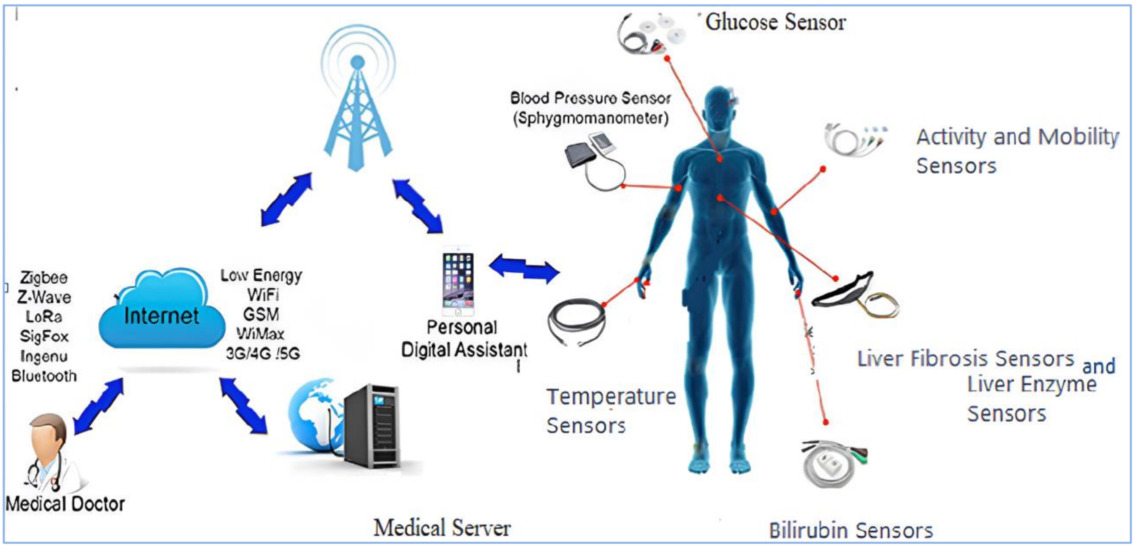

Figure 2. Internet of medical things (IoMT) architecture

Attributes for performance comparison: i) mean absolute error (MAE). The term "MAE" refers to the average of errors between paired observations that capture the same phenomena. It is one important indicator for assessing how well machine learning models work is MAE. In a dataset, it measures the mean absolute difference between the expected and actual values. In contrast to other error metrics, MAE gives a clear picture of the correctness of the model without taking the errors' direction into account. Since a lower MAE means that the model's predictions are often closer to the true values, it can be used to determine whether the model is producing more accurate predictions. MAE is very helpful in datasets with outliers since it assigns equal weight to each error. As such, it is a trustworthy metric for evaluating a machine learning model's overall effectiveness across a range of applications and domains; ii) root mean square error (RMSE). It is one of the most important metrics for evaluating the effectiveness of machine learning models is the RMSE. It computes the square root of the mean squared deviation between a dataset's actual and expected values. By penalizing greater errors more severely than smaller ones, RMSE offers a thorough picture of the accuracy of the model. Because of this, it is more susceptible to outliers and the effects of significant errors on the model's overall rating are magnified. A smaller RMSE denotes more accuracy since it shows that the model's predictions are more accurate when compared to the real data. RMSE is a commonly used machine learning metric that offers a reliable way to assess model performance and make efficient comparisons between different techniques; iii) relative absolute error (RAE). It is one of the important metric for assessing how well machine learning models are doing is the RAE. The ratio of the mean absolute error to the mean absolute value of the dataset's actual values is what it represents. RAE provides a normalized evaluation of model accuracy, allowing comparisons between various datasets and domains. A lower RAE denotes greater model performance, and values nearer zero imply that the model's predictions and actual values are closely aligned. Since RAE takes into consideration relative error rather than just absolute error, it is very helpful when working with datasets of different sizes and scales. Consequently, RAE is an effective instrument for evaluating and optimizing machine learning models in a variety of contexts and applications; and iv) root relative squared error (RRSE). It is one of the important indicator for assessing how well machine learning models perform is the RRSE. It computes the square root of the ratio between the variance of the dataset's actual values and the mean squared error of the model. By taking into consideration the variability in the dataset, RRSE offers a normalized assessment of the accuracy of the model. In relation to the variability in the dataset, a lower RRSE denotes better model performance, and values closer to zero suggest that the model's predictions agree with the actual values. RRSE offers important insights into the robustness and dependability of machine learning models by taking into account both the mean squared error and the variability of the data.

2. METHOD

2.1. Data collection

We collect data from 6,780 liver disease patients who were coming to S’O’A University Medical Laboratory, Bhubaneswar for treatment by manually and using several implant IoT devices. These data are collected either using IoT sensors and manually. Some common IoT sensors of health care applications used for liver diseases symptom identification are:

– Liver function tests (LFT) sensors: these could measure levels of enzymes, bilirubin and other substances indicative of liver function.
– Wearable health monitors: devices that track vital signs such as heart rate, blood pressure, and activity levels can provide valuable health data.
– Blood glucose monitors: monitoring glucose levels can be relevant, as liver health is interconnected with metabolic processes.

We have taken the help of IoT framework from collection of data to storage of data at cloud, which can be only possible by use of different layers of IoT architecture [15], [16]. IoT framework takes a vital role to transimit the data from sensors to cloud in a secure path by help of its inbuilt IoT communication protocols. This helps to reduce the loss of important data in the middle of communication.

2.2. Data cleaning and preparation

Data cleansing comes first in the data preparation phase after a dataset has been gathered. In this essential stage, problems with the dataset that can prevent correct analysis and modeling are found and fixed. The central tendency measurements mean, mode and median are important in this data cleaning process.

2.2.1. Handling missing values

Prior to training a machine learning model, handling missing values in a dataset is important for a number of reasons. First off, when missing values are available, a lot of machine learning algorithms are unable to deal with them directly and may malfunction or yield incorrect results. Second, biased conclusions and predictions may result from the introduction of missing values into the model training process. Furthermore, the model's generalizability and forecast accuracy may suffer if missing variables are ignored. Furthermore, missing numbers have the potential to skew statistical calculations and analyses, which lowers the model's overall quality. As a result, filling in the missing values by methods like imputation or removal guarantees that the model is trained on correct and comprehensive data, which produces predictions that are more resilient. Following are different methods which we have used to handle missing values.

– Mean imputation: if the dataset has missing numerical values, one common approach is to replace these missing values with the mean of the available data in the same column. This helps to preserve the overall distribution of the data.
– Mode imputation: for categorical data, missing values can be replaced with the mode, which is the most frequent value in the column.
– Median for robustness: when dealing with outliers (extreme values that can skew the analysis), the median is often preferred over the mean. The median is less sensitive to outliers, making it a more robust measure of central tendency.
– Data validation: using these central tendency measures can also help identify potential errors or inconsistencies in the dataset. Extreme values that are far from the mean or median may warrant further investigation as potential data entry errors.
– Impact on analysis: it's important to note that data cleaning decisions, such as imputing missing values or handling outliers, can impact the results of subsequent analyses or machine learning models. Therefore, these steps should be performed with careful consideration of the specific goals of the analysis.

3. RESEARCH DESIGN

The data set used for this study was collected from 6,780 liver disease patients of S'O'A University Medical Laboratory, Bhubaneswar. In this study, various machine learning models were employed to predict the accuracy of the various stages of liver disease, such as stage 1 to 4 based on its complications. The process of our work is shown in Figure 3.

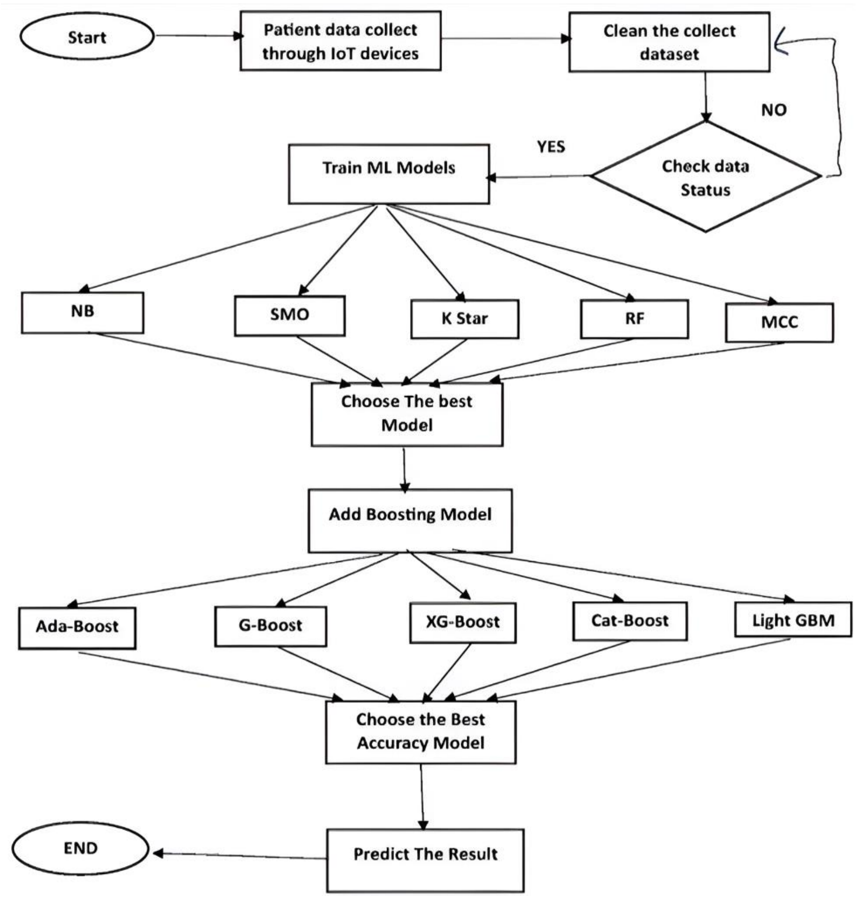

Figure 3. Flow of work

3.1. Proposed boosting models

The "boosting" ensemble modeling technique [17] seeks to build a strong classifier from a large number of weak classifiers. A model is built by stringing together weak models. A model is first built using the training data. Then, in an effort to correct the first model's flaws, the second model is built. This procedure is repeated until the maximum number of models has been added or until all of the training data set has been successfully predicted. In order to raise the accuracy of the final model, boosting might combine the accuracies of numerous weak models and average them for regression or vote over them for classification. In our experimental comparisons, we included 5 well-known strategies among the numerous types of boosting approaches. AdaBoost, GradientBoost, XGBoost, CatBoost and light gradient boosting model (LightGBM) are the techniques.

3.2. Adaptive boosting

AdaBoost is an ensemble learning technique that emphasises rectifying the errors of weak learners by giving extra weight to data points that have been incorrectly categorised. It creates a strong classifier by combining a number of weak classifiers, frequently decision trees. Each weak classifier is trained one at a time and samples that were incorrectly identified are given heavier weights in the following model. According to their performance, AdaBoost modifies the model weights, with more accurate models having a greater impact on the outcome of the prediction.

3.3. Gradient boosting

Gradient boosting is a broad boosting framework that creates a group of decision trees to increase the accuracy of predictions. Gradient boosting, in contrast to AdaBoost, which focuses on data point weights, optimises the ensemble by minimising a loss function in relation to the predictions of the individual models. Decision trees are incrementally added to the ensemble and each tree employs gradient descent optimisation to try to rectify the flaws of the previous one. Gradient boosting is frequently used and popular examples include XGBoost, LightGBM and CatBoost.

3.4. Extreme gradient boosting

Gradient boosting is implemented effectively and scalably in XGBoost, which has been popular in both machine learning contests and practical applications. It manages missing data, employs regularisation techniques to avoid overfitting and supports parallel and distributed computing. Advanced features like early halting to avoid overtraining and intelligent handling of sparse data are also available with XGBoost.

3.5. CatBoost

CatBoost is a library for gradient boosting that was created specifically to support category features. It is practical for many real-world datasets since it automatically accommodates categorical variables without the need for explicit encoding. CatBoost offers robust performance right out of the box and has built-in mechanisms to minimise over fitting.

3.6. Light gradient boosting model

Another gradient-boosting library with a solid reputation for speed and effectiveness is LightGBM. Data is divided using a histogram-based method, which utilises less memory and hastens training. For parallel processing and huge datasets, LightGBM is a good choice. It offers different types of boosting, such as conventional gradient boosting, and RF. These boosting algorithms have special qualities that make them ideal for many types of data and tasks, making them effective instruments for increasing the accuracy of predictive modeling.

4. RESULTS AND DISCUSSION

The contents of Table 1 are representing the accuracy percentage which is received from the python simulation of above five machine learning models by taking the dataset. Here we divided the 6,780 samples of dataset into six numbers of classes as specified in Table 1 and then applied different machine learning models to get correct accuracy percentage. Figure 4 represents its corresponding graphical representation.

Table 1. Comparison of NB, SMO, K-STAR, RF and MCC on the basis of correct accuracy

	Correct accuracy				
Class	NB	SMO	K STAR	RF	MCC
1-1130	85	91.06	73.86	88.75	90.62
1130-2260	86.76	93.94	75.42	92.7	93.49
2260-3390	84.99	89.95	75.96	87.91	89.33
3390-4520	86.05	92.08	75.87	91.1	91.55
4520-5650	86.23	93.58	77.82	92.7	93.23
5650-6780	81.52	98.78	75.98	86.45	98.18

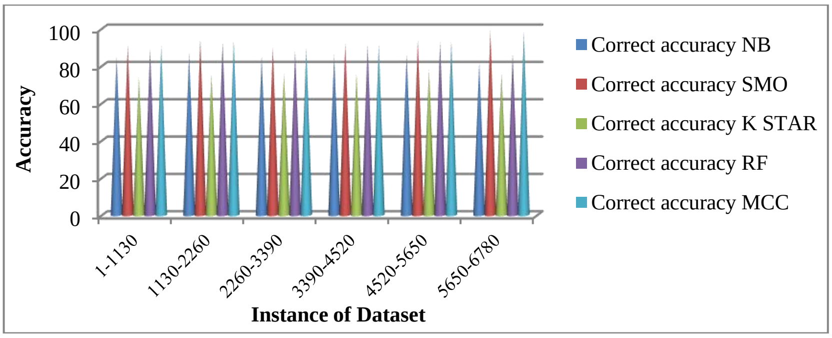

Figure 4. NB, SMO, K-STAR, RF and MCC on the basis of correct accuracy

The contents of Table 2 representing the MAEs received from the python simulation of above selected five machine learning models by taking the dataset and Figure 5 representing its graphical representation. The contents of Table 3 representing the RMSE received from the python simulation of above selected five machine learning models by taking the dataset and Figure 6 representing its graphical representation. The contents of Table 4 representing the RAE received from the python simulation of above selected five machine learning models by taking the dataset and Figure 7 representing its graphical representation. The contents of Table 5 representing the RRSE received from the python simulation of above selected five machine learning models by taking the dataset and Figure 8 representing its graphical representation.

Table 2. Comparison of NB, SMO, K-STAR, RF and MCC on the basis of MAE

	MAE				
Class	NB	SMO	K STAR	RF	MCC
1-1130	0.3327	0.3195	0.3303	0.3265	0.3232
1130-2260	0.3215	0.3149	0.3203	0.3159	0.3118
2260-3390	0.324	0.3169	0.3224	0.3271	0.3232
3390-4520	0.3268	0.3183	0.3211	0.3219	0.3212
4520-5650	0.3324	0.3161	0.3136	0.3174	0.314
5650-6780	0.3423	0.3226	0.3217	0.3325	0.331

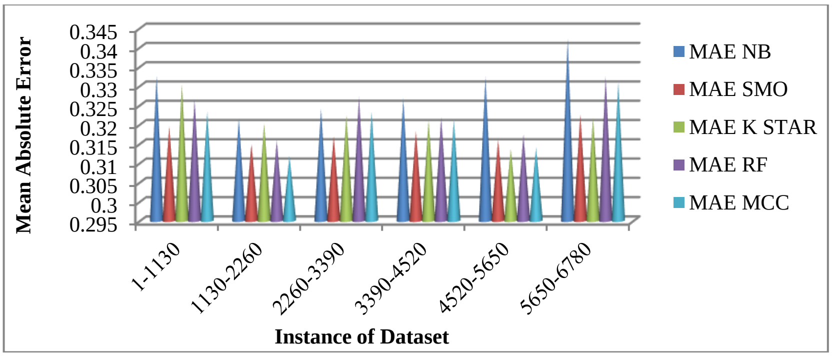

Figure 5. Graphical comparison of NB, SMO, K-STAR, RF and MCC on the basis MAE

Table 3. Comparison of NB, SMO, K-STAR, RF and MCC on the basis of RMSE

	RMSE				
Class	NB	SMO	K STAR	RF	MCC
1-1130	0.4243	0.4083	0.5396	0.4105	0.4052
1130-2260	0.4202	0.4026	0.534	0.4031	0.3978
2260-3390	0.4216	0.4052	0.5343	0.4109	0.4053
3390-4520	0.4284	0.4069	0.5347	0.4053	0.4044
4520-5650	0.4226	0.4041	0.5281	0.4029	0.3992
5650-6780	0.4334	0.4121	0.5328	0.414	0.41

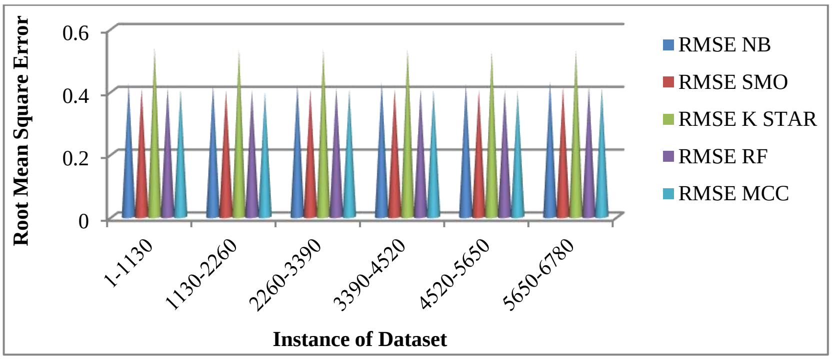

Figure 6. Graphical comparison of NB, SMO, K-STAR, RF and MCC on the basis of RMSE

Table 4. Comparison of NB, SMO, K-STAR, RF and MCC on the basis of RAE

	RAE				
Class	NB	SMO	K STAR	RF	MCC
1-1130	103.08	99.01	102.34	101.17	100.15
1130-2260	102.54	100.42	102.14	100.73	99.45
2260-3390	100.13	97.94	99.62	100.09	99.88
3390-4520	101.93	99.29	100.14	100.41	100.18
4520-5650	105.21	100.04	99.28	100.48	99.4
5650-6780	103.74	97.76	97.49	100.77	100.3

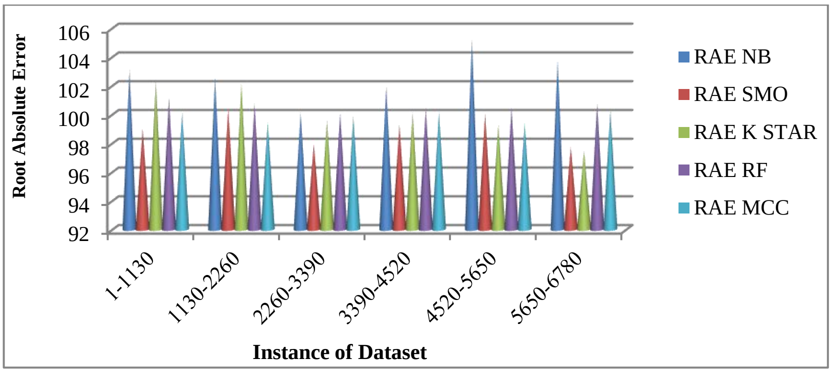

Figure 7. Graphical Comparison of NB, SMO, K-STAR, RF and MCC on the basis of RAE

Table 5. Comparison of NB, SMO, K-STAR, RF and MCC on the RRSE

	RRSE				
Class	NB	SMO	K STAR	RF	MCC
1-1130	105.67	101.68	134.37	102.21	100.92
1130-2260	106.15	101.72	134.91	101.85	100.5
2260-3390	104.85	100.75	132.87	102.18	100.8
3390-4520	107.03	101.66	133.58	101.26	101.03
4520-5650	106.35	101.7	132.93	101.41	100.47
5650-6780	106.73	101.47	131.19	101.94	101

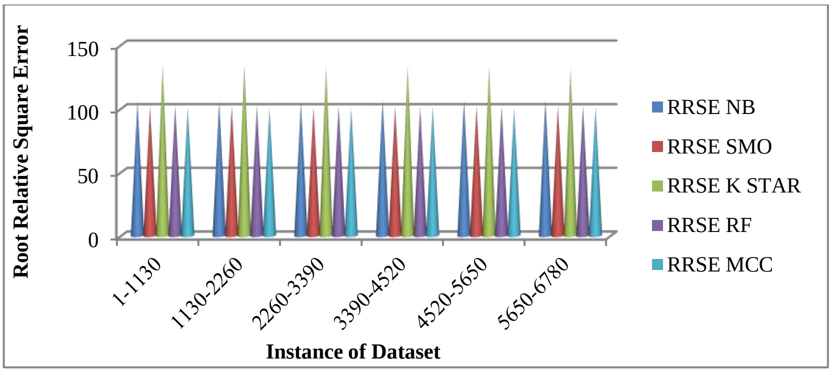

Figure 8. Graphical comparison of NB, SMO, K-STAR, RF and MCC on the RRSE

Tables 1 to 5 and its corresponding figures are used for the performance comparison [18]–[21] of all the above models. Correct accuracy, MAE, RAE, RMSE, and RRSE are five parameters of above machine learning models for measuring its performance. In machine learning models, the model which has highest accuracy and lowest error values is considered as best models. After examining all data of Tables 1 to 5 and its corresponding figures, we have found that SMO model is giving better results on this particular dataset as compared to other models. So, here we consider that SMO model is best model for prediction of liver daises stages. Then we have applied the machine learning boosting models with the selected SMO results. Table 6 representing the accuracy percentage of different boosting models with collaboration to SMO model and Figure 9 showing its graphical representation.

Table 6. Accuracy level after adding boosting models with SMO

Boosting with SMO	
AdaBoost	94.10%
Gradient boost	96%
XGBoost	90%
CatBoost	92%
LightGBM	94%

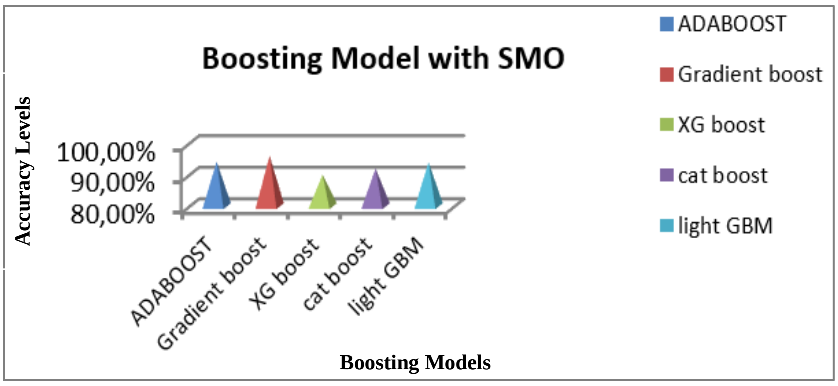

Figure 9. Accuracy after adding boosting models

In this research work we have reviewed some recent existing works on prediction of liver disease, in which we found that, there are various research gaps regarding implementation of hybrid machine learning models. By implementation different machine learning boosting models on the base models we can enhance the disease prediction accuracy. Though in this research we have implemented the machine learning boosting models and found the results are remarkable as compared to other machine learning models [22], [23] that are proving the novelty of our work on prediction of liver disease. Table 7 showing the performance comparison between the existing works and our present work.

Table 7. Result comparison with some existing work

Author	Year	Machine learning models	Accuracy
Dritsas *et al.* [3]	2023	NB and LR	80.1%
Singh *et al.* [24]	2020	LR, SMO, RF, NB, J48, and K-Nearest	72.50%
Amin *et al.* [25]	2023	LR, RF, K-Nearest, SVM, MLP and voting models	91.40
Our work	2023	NB, SMO, K-STAR, RF, MCC and machine learning boosting models	96%

5. CONCLUSION

This research uses a real time dataset made up of 6,780 records that were collected manually and using IoT sensor from the patients of S'O'A University Medical Laboratory to successfully forecast the stages of liver disease. To predict the phases of liver damage in patients, this study used a number of machine learning models, including NB, SMO, K-STAR, RF, and MCC. After careful simulation and analysis, it became clear that the SMO model performed better than the others and offered promising outcomes. But the search for better precision did not stop there. The SMO model's data was then subjected to analysis using a variety of machine learning boosting models, including AdaBoost, gradient boost, XGBoost, CatBoost and LightGBM. Then we get the result that, gradient boosting emerged as the most accurate and practical option in the end of the research, with an astounding accuracy rate of 96%. The ability of machine learning models, particularly gradient boosting, to support early detection and intervention for patients with liver disorders is demonstrated by this study, which represents a significant improvement in the field of liver disease stage prediction. This study paves the path for quicker and more precise medical interventions, which will eventually enhance patient outcomes and healthcare productivity. Future research will focus on bridging the gap between theoretical machine learning applications and real-world healthcare applications, which will ultimately enhance patient outcomes and increase the effectiveness of healthcare delivery.

REFERENCES

[1] S. H. Afrizal, P. W. Handayani, T. Eryando, and A. Sartono, "Primary care functional requirements of a health information system in Indonesia," in *Proceedings of the 3rd International Conference on Informatics and Computing, ICIC 2018*, Oct. 2018, pp. 1–7. doi: 10.1109/IAC.2018.8780501.

[2] P. K. Panda and S. Chattopadhyay, "An enhanced mutual authentication and security protocol for IoT and cloud server," *Information Security Journal*, vol. 31, no. 2, pp. 144–156, Mar. 2022, doi: 10.1080/19393555.2020.1871534.

[3] E. Dritsas and M. Trigka, "Supervised machine learning models for liver disease risk prediction," *Computers*, vol. 12, no. 1, pp. 1–15, Jan. 2023, doi: 10.3390/computers12010019.

[4] A. Sivasangari, B. J. Krishna Reddy, A. Kiran, and P. Ajitha, "Diagnosis of liver disease using machine learning models," in *Proceedings of the 4th International Conference on IoT in Social, Mobile, Analytics and Cloud, ISMAC 2020*, Oct. 2020, pp. 627–630. doi: 10.1109/I-SMAC49090.2020.9243375.

[5] R. V. Manjunath, A. Ghanshala, and K. Kwadiki, "Deep learning algorithm performance evaluation in detection and classification of liver disease using CT images," *Multimedia Tools and Applications*, vol. 83, no. 1, pp. 2773–2790, Jan. 2024, doi: 10.1007/s11042-023-15627-z.

[6] G. Singh and C. Agarwal, "Prediction and analysis of liver disease using extreme learning machine," *Sentiment Analysis and Deep Learning, Advances in Intelligent Systems and Computing*, vol 1432, 2023, pp. 679–690. doi: 10.1007/978-981-19-5443-6_52.

[7] T. Malapane, W. Doorsamy, and B. S. Paul, "An intelligent IoT-based health monitoring system," in *2020 International Conference on Intelligent Data Science Technologies and Applications, IDSTA 2020*, Oct. 2020, pp. 100–105. doi: 10.1109/IDSTA50958.2020.9264102.

[8] M. Senbagavalli and S. K. Singh, "Improving patient health in smart healthcare monitoring systems using IoT," in *2022 International Conference on Futuristic Technologies,* Nov. 2022, pp. 1–7. doi: 10.1109/INCOFT55651.2022.10094409.

[9] S. S. Rasheed and I. H. Glob, "Classifying and prediction for patient disease using machine learning algorithms," in *3rd Information Technology to Enhance e-Learning and Other Application, IT-ELA 2022*, Dec. 2022, pp. 196–200. doi: 10.1109/IT-ELA57378.2022.10107935.

[10] F. Wu, T. Wu, and M. R. Yuce, "Design and implementation of a wearable sensor network system for iot-connected safety and health applications," in *IEEE 5th World Forum on Internet of Things, WF-IoT 2019 - Conference Proceedings*, Apr. 2019, pp. 87–90. doi: 10.1109/WF-IoT.2019.8767280.

[11] S. S. G. K. Das, K. D. Mahato, C. Azad, and U. Kumar, "Heart disease prediction using different boosting models," in *Proceedings - 2023 International Conference on Advanced and Global Engineering Challenges, AGEC 2023*, Jun. 2023, pp. 131–136. doi: 10.1109/AGEC57922.2023.00036.

[12] F. Harahap, A. Y. N. Harahap, E. Ekadiansyah, R. N. Sari, R. Adawiyah, and C. B. Harahap, "Implementation of Naïve Bayes classification method for predicting purchase," in *2018 6th International Conference on Cyber and IT Service Management, CITSM 2018*, Aug. 2019, pp. 1–5. doi: 10.1109/CITSM.2018.8674324.

[13] M. W. K. Mbukani and N. Gule, "Implementation of an SMO-based MRAS estimator for sensor-less control of rdfig systems," in *Proceedings - 2020 International Conference on Electrical Machines, ICEM 2020*, Aug. 2020, pp. 1143–1149. doi: 10.1109/ICEM49940.2020.9270923.

[14] K. Abhishek and G. Hamarneh, "Matthews correlation coefficient loss for deep convolutional networks: application to skin lesion segmentation," in *Proceedings - International Symposium on Biomedical Imaging*, Apr. 2021, vol. 2021-April, pp. 225–229. doi: 10.1109/ISBI48211.2021.9433782.

[15] A. E. Bouaouad, A. Cherradi, S. Assoul, and N. Souissi, "The key layers of IoT architecture," in *Proceedings of 2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications, CloudTech 2020*, Nov. 2020, pp. 1–4. doi: 10.1109/CloudTech49835.2020.9365919.

[16] S. Swain, M. N. Mohanty, B. K. Pattanayak, P. Mallik, K. J. Patra, and C. Panda, "A secure IoT-enabled machine learning framework for brain tumor classification and prediction using MR image data," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 9, pp. 1873–1882, 2023, doi: 10.17762/ijritcc.v11i9.9181.

[17] V. P. Rekkas, S. P. Sotiroudis, G. Athanasiadou, P. Sarigiannidis, G. V. Tsoulos, and S. K. Goudos, "Path loss prediction modelling for next-generation internet-of-things applications using different boosting machine learning methods," in *2022 Panhellenic Conference on Electronics and Telecommunications,* Dec. 2022, pp. 1–4. doi: 10.1109/PACET56979.2022.9976383.

[18] S. Hashem *et al.*, "Comparison of machine learning approaches for prediction of advanced liver fibrosis in chronic hepatitis c patients," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 3, pp. 861–868, 2018, doi: 10.1109/TCBB.2017.2690848.

[19] V. Anthonysamy and S. K. K. Babu, "Multi perceptron neural network and voting classifier for liver disease dataset," *IEEE Access*, vol. 11, pp. 102149–102156, 2023, doi: 10.1109/ACCESS.2023.3316515.

[20] C. G. Raji and S. S. Vinod Chandra, "Long-term forecasting the survival in liver transplantation using multilayer perceptron networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2318–2329, 2017, doi: 10.1109/TSMC.2017.2661996.

[21] S. El-Sappagh, F. Ali, A. Ali, A. Hendawi, F. A. Badria, and D. Y. Suh, "Clinical decision support system for liver fibrosis prediction in hepatitis patients: a case comparison of two soft computing techniques," *IEEE Access*, vol. 6, pp. 52911–52929, 2018, doi: 10.1109/ACCESS.2018.2868802.

[22] S. S. Nigatu, P. C. R. Alla, R. N. Ravikumar, K. Mishra, G. Komala, and G. R. Chami, "A comparitive study on liver disease prediction using supervised learning algorithms with hyperparameter tuning," *2023 International Conference on Advancement in Computation and Computer Technologies, InCACCT 2023*, pp. 353–357, 2023, doi: 10.1109/InCACCT57535.2023.10141830.

[23] C. Anuradha, D. Swapna, B. Thati, V. N. Sree, and S. P. Praveen, "Diagnosing for liver disease prediction in patients using combined machine learning models," *Proceedings - 4th International Conference on Smart Systems and Inventive Technology, ICSSIT 2022,* pp. 889–896, 2022, doi: 10.1109/ICSSIT53264.2022.9716312.

[24] J. Singh, S. Bagga, and R. Kaur, "Software-based prediction of liver disease with feature selection and classification techniques," *Procedia Computer Science*, vol. 167, pp. 1970–1980, 2020, doi: 10.1016/j.procs.2020.03.226.

[25] R. Amin, R. Yasmin, S. Ruhi, M. H. Rahman, and M. S. Reza, "Prediction of chronic liver disease patients using integrated projection based statistical feature extraction with machine learning algorithms," *Informatics in Medicine Unlocked*, vol. 36, pp. 1–11, 2023, doi: 10.1016/j.imu.2022.101155.

BIOGRAPHIES OF AUTHORS

Satyaprakash Swain completed his masters of technology (M.Tech.) in computer science from Utkal University, Odisha, India in 2015 and has since been actively involved in research, teaching and mentoring the students. He is currently working as a faculty in Institute of Management and Information Technology, Cuttack, India and as well as a part time Ph.D. scholar in Siksha 'O' Anusandhan University, Bhubaneswar, India. His research interest includes IoT, machine learning, network security, cloud computing, computational intelligence, and soft computing. He has 11 years of teaching and research experience. He is a life member of ISTE. He can be contacted at email: satyaimit@gmail.com.

Prof. Mihir Narayan Mohanty is currently working as a professor in the Department of Electronics and Communication Engineering, Institute of Technical Education and Research (ITER), Siksha ‘O’ Anusandhan (Deemed to be University), Bhubaneswar, Odisha. He has received his M.Tech. degree in communication system engineering from the Sambalpur University, Sambalpur, Odisha and obtained his Ph.D. degree in applied signal processing. He is the fellow of IE (I), and IETE. Also, he is the member of many professional societies including IEEE, IET etc. He has 25 years of teaching and research experience. He has published more than 400 papers in different journals, conferences including book chapters. He has authored two books. He is the successive reviewer of manuscripts from IEEE, Elsevier, Springer, and IGI Global. His areas of research interests include applied signal and image processing, speech processing, antenna and intelligent signal processing. He can be contacted at email: mihirmohanty@soa.ac.in.

Prof. Binod Kumar Pattanayak is currently working as a professor in the Department of Computer Science and Engineering, Institute of Technical Education and Research (ITER), Siksha 'O' Anusandhan University, Bhubaneswar, India since 1999. He got his Ph.D. from Siksha 'O' Anusandhan University, M.S. in computer engineering from National Technical University, Kharkov, Ukraine and B.Sc. from Revenshaw University, Cuttack, India. He has 25 years of teaching and research experience. He has published more than 400 papers in different journals, conferences including book chapters. His research interests include Ad-hoc networks, compiler design, software engineering, computer architecture, intelligent-systems. He can be contacted at email: binodpattanayak@soa.ac.in.