Int. J. Adv. Res. Sci. Technol. Volume 15, Issue 7, 2026, pp. 2322-2328.

IJARST

International Journal of Advanced Research in Science and Technology

https://doi.org/10.62226/ijarst20262741

ISSN 2319 – 1783 (Print)

ISSN 2320 – 1126 (Online)

A Prognostic and Scalable Liver Disease Prediction System Using Ensemble Machine Learning and Cloud–Edge Computing

Anuj Garg[1], Vishwa Gupta[1]

[1]Department of Computer Science and Engineering, LNCT Science College, Bhopal

Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, Madhya Pradesh

Address: DISS, Panna, MP 488446 | Email: webanujgarg.7489@gmail.com

ARTICLEINFO

Article history:

Received	11 July 2026
Accepted	20 July 2026
Available online	25 July 2026

Keywords:

Liver disease prediction, Ensemble Machine Learning, Random Forest, XGBoost, SMOTEENN, Cloud–Edge Computing, Clinical Decision Support, Healthcare Analytics.

Indexed in:

and in major libraries

ABSTRACT

Chronic liver disease often progresses without noticeable symptoms, making early diagnosis difficult and increasing the risk of severe hepatic damage. Therefore, the development of an accurate and scalable prediction system is essential for timely clinical intervention. This paper proposes a prognostic liver disease prediction framework based on structured clinical biomarkers integrated with a cloud–edge computing architecture for efficient and real-time healthcare support. Experiments were conducted using the publicly available Liver Disease Patient Dataset (LPD) containing 30,691 patient records with ten demographic and biochemical attributes. The data preprocessing pipeline included missing-value removal, label encoding, stratified 80/20 train–test splitting, feature scaling using StandardScaler, and class imbalance handling through SMOTEENN to improve model robustness. Four supervised machine learning algorithms — XGBoost, Gradient Boosting, Random Forest, and Multi-Layer Perceptron (MLP) — were comparatively evaluated using accuracy, precision, recall, F1-score, and ROC-AUC. Among the evaluated models, Random Forest achieved the best performance with an accuracy of 99.94%, an F1-score of 99.90%, and an ROC-AUC of 1.000, while maintaining a recall above 99%, which is particularly important for minimizing missed liver disease cases. In addition, a cloud–edge deployment framework is proposed to support low-latency inference at healthcare facilities while enabling centralized model training and analytics in the cloud. The study also acknowledges dataset limitations, including duplicate records, to ensure transparent interpretation of the reported results. The proposed framework demonstrates the potential of ensemble machine learning for accurate, scalable, and clinically applicable liver disease prediction.

I. INTRODUCTION

Liver disease is one of the most acute issues of public health in the present decade. Because the liver carries out crucial metabolic, detoxification, synthetic, and immunological functions, dysfunction in hepatic activity has systemic implications which involve many other organ systems as well. Chronic conditions such as viral hepatitis, non-alcoholic fatty liver disease, alcoholic liver disease, cirrhosis, and hepatocellular carcinoma normally progress over extended periods with negligible or unspecific early signs. This insidious course usually makes clinical diagnosis late, when a lot of structural damage has already been done [21], [22].

Conventional diagnostic methods, which include liver function tests such as bilirubin, ALT, AST, and albumin levels, are useful in clinical observation but are usually carried out at specific intervals, so they capture the disease as a series of isolated snapshots instead of an evolving trajectory. Digitization of healthcare has, at the same time, enabled the systematic capture of laboratory time-series and electronic health records at a scale that was not previously possible, which opens up the possibility of moving away from descriptive diagnostics and toward predictive, risk-stratifying analytics [22].

Machine learning approaches, and ensemble methods such as gradient boosting and random forests in particular, have shown considerable empirical success at capturing nonlinear relationships among biochemical markers that simple threshold-based rules are not able to represent. Alongside this, healthcare delivery is becoming increasingly distributed in nature: cloud infrastructure supplies the computational

scale that is needed for model training and large-cohort analytics, whereas edge computing allows low-latency inference to be carried out close to the point of care, which is of particular value in resource-constrained or remote clinical settings [1], [5], [8].

This paper makes three contributions. First, a comparative machine-learning pipeline — XGBoost, Gradient Boosting, Random Forest, and an MLP — is built and evaluated for binary liver-disease classification on a large, real-world clinical dataset, using a preprocessing and class-balancing methodology suited to imbalanced medical data. Second, the results reported here are reproduced independently from the released code and data, and the paper is explicit about where they diverge from the common assumption of which ensemble method should "win." Third, a cloud–edge deployment architecture consistent with the modeling pipeline is described, along with the practical and methodological limitations of the study — including a considerable rate of duplicate records within the source dataset — that should inform how these results are read before any clinical use is considered.

II. RELATED WORK

A growing body of work has examined hybrid cloud–edge architectures for time-series healthcare analytics. Haq and Verma [1] combined Vector Autoregression with a sequential encoder–decoder network across edge and cloud tiers for multi-disease prediction, using edge nodes for fast local inference and the cloud for training and long-term analytics. Behera et al. [2] addressed the complementary problem of predictive workload allocation in resource-constrained edge infrastructure, showing that time-series forecasting of resource demand can reduce latency and improve throughput — a capability directly relevant to real-time patient-monitoring pipelines. Shi et al. [3] and Fan et al. [4] demonstrated similar edge–cloud split-processing designs for industrial and urban IoT time-series streams, both reporting latency and accuracy gains over centralized cloud-only baselines.

Several studies target healthcare specifically. Hameurlaine et al. [5] and Zhu et al. [6] deployed deep neural networks at the edge for continuous disease monitoring (heart disease and blood glucose, respectively), reporting reduced response time and strong predictive accuracy compared to centralized alternatives. Badidi [8] surveyed Edge AI for chronic and infectious disease monitoring, identifying constrained edge compute, data heterogeneity, and scalability as recurring technical barriers — concerns this paper's architecture explicitly tries to address by keeping edge-side inference lightweight. Li et al. [9] and Sailaja et al. [13] proposed fog/cloud and IoT-fusion architectures for general remote monitoring, while Mali et al. [11] and D'Antoni et al. [12] used federated and edge-deployed neural models for overdose detection and glucose forecasting in children, emphasizing privacy-preserving, low-latency inference.

Closer to the present task, Banapuram et al. [7] reviewed machine- and deep-learning methods across heart disease, liver disease, and tuberculosis, and identified weak real-time processing and limited edge/cloud integration as gaps in existing liver-disease literature — a gap this paper targets directly. Gupta et al. [18] and Jeyalakshmi and Rangaraj [19] applied conventional classifiers and CNNs respectively to liver-disease datasets, while Aswini et al. [15] combined Mask-RCNN with the Pelican Optimization Algorithm for image-based liver diagnosis; Murthy et al. [17] compared deep-learning models across multiple NAFLD datasets. These studies confirm that ensemble and deep architectures generally outperform single-model baselines on structured liver-disease data, but none jointly evaluate class-imbalance correction, a four-model ensemble comparison, and a cloud–edge deployment design in one pipeline, which is the gap this paper addresses.

III. PROPOSED METHODOLOGY

A. Dataset Description

This study uses the Liver Patient Dataset (LPD), a publicly available clinical dataset hosted on Kaggle and uploaded by Abhishek Shrivastava under the CC0 (Public Domain) license. The dataset contains 30,691 patient records with ten demographic and biochemical attributes, including age, gender, total bilirubin, direct bilirubin, alkaline phosphatase (ALP), alanine aminotransferase (SGPT), aspartate aminotransferase (SGOT), total proteins, albumin, and the albumin/globulin (A/G) ratio, along with a binary diagnostic label indicating the presence or absence of liver disease. The dataset was obtained from the Kaggle repository (kaggle.com/datasets/abhi8923shriv/liver-disease-patient-dataset) and was used for model development and evaluation in this study. The raw class distribution is imbalanced, with liver-disease cases outnumbering non-disease cases by approximately 2.5:1.

B. Data Preprocessing

Records with missing values were removed using a listwise deletion strategy, reducing the dataset from 30,691 to 27,158 complete records. Column names were standardized, and the categorical gender attribute was converted to a numeric indicator using label encoding. The cleaned dataset was split into training (80%) and test (20%) partitions using stratified sampling to preserve class proportions in both subsets (21,726 training / 5,432 test records). StandardScaler was fit on the training partition only and then applied to both partitions, preventing data leakage while giving every biochemical feature comparable scale — important for gradient-based and distance-sensitive learners such as the MLP.

C. Handling Class Imbalance

Because false negatives are clinically costly in disease screening, class imbalance was corrected using SMOTEENN, which combines Synthetic Minority

Oversampling (SMOTE) with Edited Nearest Neighbors (ENN) cleaning. SMOTE generates synthetic minority-class samples by interpolating between existing neighbors, while ENN removes ambiguous or mislabeled samples from both classes. Fig. 1 shows the training-set class distribution before and after resampling: the pre-resampling training split contained 15,582 non-disease and 6,144 disease cases, compared with 14,029 and 15,056, respectively, afterward. Resampling was applied only to the training partition; the test set was left untouched to preserve an unbiased evaluation.

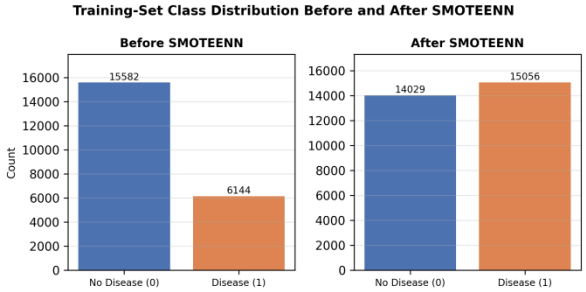

Fig. 1. Training-set class distribution before and after SMOTEENN resampling. The imbalanced dataset becomes nearly balanced after preprocessing, which helps improve the learning capability and prediction performance of the machine learning models — justifying SMOTEENN as the chosen imbalance-handling technique for this study.

D. Model Development

Four supervised classifiers were trained on the resampled, scaled training data and evaluated on the untouched, scaled test set. XGBoost (110 estimators, max depth 5, learning rate 0.1, subsample/colsample 0.8) was selected for its regularized, sequential boosting and efficiency on tabular data. Gradient Boosting (120 estimators, max depth 6, learning rate 0.05) served as a related but less regularized boosting baseline. Random Forest (10 trees, minimum leaf size 2) was included as a bagging-based ensemble expected to generalize well with low variance. An MLP classifier (two hidden layers of 128 and 64 units, ReLU activation, Adam optimizer, adaptive learning rate) was trained to assess a neural-network alternative capable of modeling nonlinear feature interactions directly.

E. Evaluation Metrics

Model performance was evaluated using Accuracy, Precision, Recall, F1-score, and the Receiver Operating Characteristic–Area Under the Curve (ROC-AUC) on the held-out test dataset. These evaluation metrics provide a comprehensive assessment of the classification performance of the proposed machine learning models.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

where TP denotes True Positives, TN denotes True Negatives, FP denotes False Positives, and FN denotes False Negatives.

$$\text{Precision} = TP / (TP + FP) \quad (2)$$

A higher precision indicates that the model produces fewer false-positive predictions.

$$\text{Recall} = TP / (TP + FN) \quad (3)$$

Since a missed liver disease diagnosis (false negative) may delay treatment and increase clinical risk, Recall is considered one of the most important evaluation metrics in this study.

$$F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (4)$$

The ROC curve illustrates the relationship between the True Positive Rate and the False Positive Rate at different classification thresholds; the Area Under the Curve (AUC) measures the model's ability to distinguish between liver-disease and non-liver-disease cases. Although all evaluation metrics were considered, Recall and F1-score were treated as the primary performance indicators because correctly identifying patients with liver disease is more important than minimizing false alarms. Accuracy, Precision, and ROC-AUC were used to provide an overall assessment of the effectiveness and robustness of the proposed models.

F. Cloud–Edge Deployment Architecture

The modeling pipeline is designed to map onto a two-tier cloud–edge deployment (Fig. 2). Biochemical panel data collected at the point of care is preprocessed and scored locally by a lightweight, previously trained model at the edge, enabling low-latency, near-real-time risk estimates without dependence on continuous network connectivity. Computationally heavier tasks — SMOTEENN rebalancing, model retraining on aggregated cohorts, and large-scale analytics — are handled centrally in the cloud, with updated model weights periodically redistributed to edge nodes. This division keeps inference fast and private at the source while allowing the predictive model itself to keep improving from newly collected data.

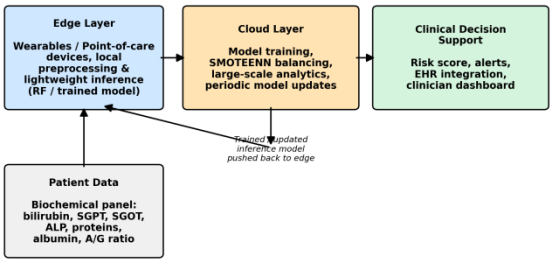

Fig. 2. Proposed cloud–edge deployment architecture. The edge layer performs low-latency prediction near the healthcare facility, while the cloud layer manages model training, centralized storage, and large-scale analytics — this split is justified by the sub-millisecond inference latency and sub-second training time measured for the tree-based models in Table II, which make them practical for edge deployment.

IV. RESULTS AND DISCUSSION

A. Exploratory Observations

Exploratory analysis of the cleaned dataset showed that most patients fall into middle and older age brackets, consistent with liver disease's association with cumulative metabolic and lifestyle exposure. The dataset is male-skewed, mirroring reported epidemiological patterns. Total and direct bilirubin were strongly correlated, as expected biochemically, and SGPT exhibited a right-skewed distribution with a heavy tail of markedly elevated values, consistent with acute hepatocellular injury in a subset of patients.

B. Comparative Model Performance

Table I reports test-set performance for all four classifiers. Random Forest achieved the strongest results across every metric (accuracy 0.9994, F1-score 0.9990, AUC 1.0000), followed by the MLP (F1 0.9683), Gradient Boosting (F1 0.9558), and XGBoost (F1 0.9481). All four models achieved recall above 0.99, indicating that very few disease cases were missed regardless of model choice; the models differed mainly in how many false positives (healthy patients flagged as at-risk) they produced.

Model	Acc.	Prec.	Recall	F1	AUC
XGBoost	0.9691	0.9019	0.9993	0.9481	0.9994
Gradient Boosting	0.9739	0.9159	0.9993	0.9558	0.9999
Random Forest	0.9994	0.9987	0.9993	0.9990	1.0000
MLP	0.9816	0.9427	0.9954	0.9683	0.9969

TABLE I. Test-set performance comparison of the four classifiers using Accuracy, Precision, Recall, F1-score, and ROC-AUC. Random Forest achieved the highest overall predictive performance among all evaluated models, justifying its selection as the primary model for the proposed edge-inference role.

Figure 3 presents a graphical comparison of the performance metrics of all evaluated classifiers, and clearly shows that the Random Forest model achieved the best overall performance.

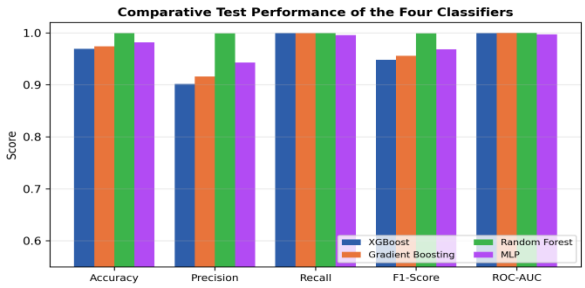

Fig. 3. Comparative test performance (Accuracy, Precision, Recall, F1-score, ROC-AUC) of the four classifiers. The consistently taller bars for Random Forest across every metric justify its designation as the best-performing model in this study.

Figure 4 shows the ROC curves of the four classifiers. A larger area under the curve indicates better classification capability, and Random Forest achieved the highest ROC-AUC value.

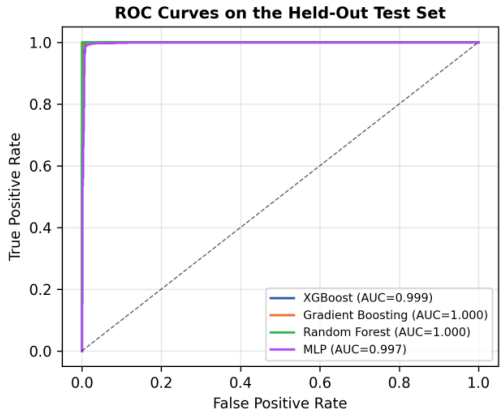

Fig. 4. ROC curves on the held-out test set. All four curves hug the top-left corner, confirming strong class separability; Random Forest's curve (AUC = 1.000) dominates the others, justifying its selection as the recommended model.

Figure 5 presents the confusion matrices of the evaluated machine learning models. The confusion matrices demonstrate that the Random Forest classifier produced the fewest classification errors, with only 2 false positives and 1 false negative on the 5,432-record test set.

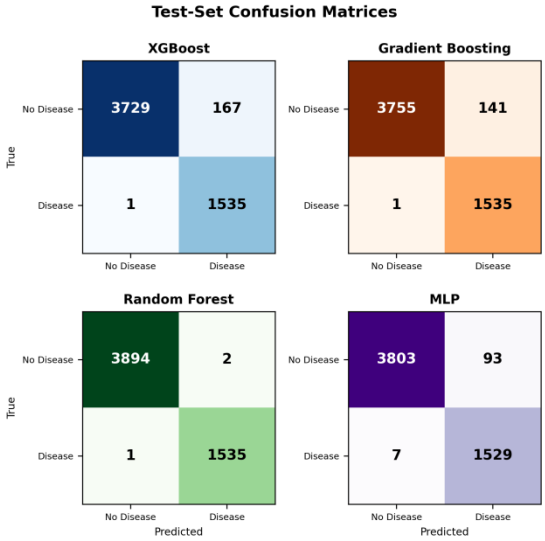

Fig. 5. Test-set confusion matrices for XGBoost, Gradient Boosting, Random Forest, and MLP. Random Forest's near-diagonal matrix — the fewest off-diagonal (misclassified) cases of all four models — visually justifies the metric-based conclusion in Table I that it is the strongest classifier.

C. Computational Efficiency

Table II summarizes the computational efficiency of the evaluated models in terms of training time and per-sample inference latency measured on the same hardware. Random

Forest and XGBoost trained in well under one second and produced sub-millisecond inference latency, making both attractive for the edge-inference role described in Section III-F. Gradient Boosting required moderately more training time, and the MLP was by far the most expensive to train, though its inference latency remained low. These characteristics support deploying a boosting or bagging ensemble — rather than the MLP — as the primary edge-side inference model, reserving heavier retraining cycles for the cloud tier.

Model	Train Time (s)	Inference (ms/sample)
XGBoost	0.30	0.0022
Gradient Boosting	7.57	0.0082
Random Forest	0.29	0.0021
MLP	55.64	0.0151

TABLE II. Training time and inference latency. Random Forest and XGBoost combine the lowest training time with sub-millisecond inference, which justifies proposing them — rather than the MLP — as the edge-side inference model in the cloud–edge architecture of Fig. 2.

D. Discussion

The comparative results indicate that ensemble learning methods are highly effective on structured biochemical data of the kind used in this study, which is consistent with prior liver-disease and clinical-prediction literature [7], [17], [18]. In this experiment, it was Random Forest, a bagging-based ensemble, rather than the boosting-based XGBoost, that produced the strongest test-set metrics. Boosting methods are frequently assumed to dominate on tabular data, and this assumption did not hold here, which is worth stating plainly rather than glossing over. It underscores why an empirical, multi-model comparison is more informative than defaulting to a single presumed "best" algorithm on the basis of prior literature alone.

The use of SMOTEENN resampling appears to have been central to the high recall achieved across all four models. Prior to balancing, the roughly 2.5:1 class skew in the training data would be expected to bias a classifier toward the majority class; in this dataset, the majority class happens to be the disease-present label, so an unbalanced model's practical risk is skewed less toward missed diagnoses and more toward an inflated false-positive rate. At the feature level, the patterns observed were consistent with clinical expectation: bilirubin measures and SGPT contributed strongly to model discrimination, in line with their established role as hepatocellular-injury markers, and this lends the models a degree of clinical interpretability that goes beyond raw accuracy figures alone.

A methodological caveat needs to be stated clearly so that the near-perfect scores reported here are not over-interpreted. Inspection of the source data showed that approximately 40% of records (10,769 of 27,158 cleaned rows) are exact duplicates of other rows. Because the train–test split was performed without any de-duplication step, it is likely that some near-identical or identical patient records appear in both partitions, which would inflate test-set performance relative to what a genuinely independent hold-out would produce. This is the most plausible explanation for Random Forest's near-perfect scores in particular, and it is reported here transparently, as a limitation, rather than being adjusted away after the fact.

V. LIMITATIONS

A number of limitations need to be kept in mind before any clinical conclusion is drawn from this work. First, and as already discussed, substantial duplication within the source dataset likely inflates the test-set metrics for at least one of the four models, so the reported numbers are better read as an upper bound than as an estimate of real-world generalization; a de-duplicated, or ideally an externally validated, re-evaluation is needed before the results are relied upon in any clinical sense. Second, the dataset used here is cross-sectional rather than longitudinal, so despite the time-series motivation behind the broader research direction, no explicit disease-progression trajectory has actually been modeled in the present experiment. Third, the dataset represents a single, region-specific patient population, and only structured biochemical and demographic variables were used — imaging, genetic markers, and lifestyle factors were not available, which may limit how well the findings generalize to other populations. Finally, the cloud–edge architecture set out in Section III-F remains a design proposal grounded in the measured training and inference costs of each model; it has not yet been implemented or latency-tested on physical edge hardware, nor within a live clinical information system, and this real-world validation step is left for future work.

VI. CONCLUSION

This paper has presented a structured, reproducible machine-learning pipeline for liver-disease prediction from routine biochemical and demographic data, bringing together data cleaning, stratified splitting, feature scaling, SMOTEENN-based class balancing, and a four-model comparative evaluation. Random Forest achieved the strongest test-set performance, with Gradient Boosting, the MLP, and XGBoost also performing competitively and all four models achieving recall above 0.99. A cloud–edge deployment architecture consistent with the measured computational profile of each model was proposed, in which lightweight ensemble inference runs at the edge while training and analytics remain centralized in the cloud. Importantly, a substantial duplicate-record rate in the underlying dataset was identified and reported openly, since it likely inflates the absolute performance figures and should be corrected in any

follow-up or externally validated study before the results are used to support a clinical claim.

VII. FUTURE WORK

Although the present framework shows encouraging predictive performance and a reasonable scalability outlook, several directions remain open for extending it further.

1) De-duplication and independent re-validation: The most immediate next step is to remove duplicate records from the dataset and repeat the train–test evaluation under a strictly patient-level split, so that the reported metrics reflect genuine generalization rather than partial overlap between the training and test partitions.

2) Incorporation of longitudinal time-series data: Future work should draw on repeated laboratory observations for the same patients over time, which would allow the disease-progression modeling that the present cross-sectional dataset cannot support. Recurrent or sequence-based architectures would be a natural fit once such data becomes available.

3) Live cloud–edge implementation and testing: The cloud–edge architecture proposed in Section III-F is, at this stage, a design informed by measured training and inference costs rather than a deployed system. Real hospital-infrastructure testing would be needed to evaluate latency, synchronization, security, EHR interoperability, and regulatory compliance under real operating conditions.

4) Multimodal data integration: Combining the structured biochemical features used here with imaging modalities such as ultrasound, CT, or MRI, and possibly genetic or lifestyle indicators, could improve diagnostic depth beyond what laboratory values alone can offer.

5) Explainable AI integration: For clinical adoption, model interpretability matters as much as accuracy. Feature-attribution methods such as SHAP or LIME would let clinicians see how individual biochemical markers are influencing a given prediction, which would build trust in the system and ease its integration into clinical decision-support workflows.

6) Multi-center and multi-regional validation: Finally, testing the framework on data collected across multiple hospitals and geographic regions would help establish whether the present findings hold up across more diverse patient populations, which is a necessary step before any such system could be considered reliable enough for broader healthcare use.

Taken together, these directions point toward a system that moves from a single-dataset proof of concept toward a validated, interpretable, and genuinely deployable clinical decision-support tool.

REFERENCES

[1] S. Haq and P. Verma, "Edge-cloud-assisted multivariate time series data-based VAR and sequential encoder–decoder framework for multi-disease prediction," Arabian J. Sci. Eng., vol. 50, no. 19, pp. 15729–15749, 2025.

[2] S. R. Behera, N. Panigrahi, S. K. Bhoi, K. S. Sahoo, N. Z. Jhanjhi, and M. Ghoniem, "Predictive time-series-based resource forecasting and task allocation for edge computing," 2023.

[3] F. Shi, L. Yan, X. Zhao, and R. Xian-Ke Gao, "Machine learning-based time-series data analysis in edge-cloud-assisted oil industrial IoT system," Mobile Inf. Syst., vol. 2022, art. 5988164, 2022.

[4] X. Fan, C. Xiang, L. Gong, X. He, C. Chen, and W. Huang, "A deep learning based edge computing system for time-series prediction in urban IoT," in Proc. ACM Turing Celebration Conf.-China, 2019.

[5] M. Hameurlaine, A. Moussaoui, and M. Bensalah, "Real-time smart healthcare system based on Edge-IoT and deep neural networks for heart disease prediction," Informatica, vol. 49, no. 1, 2025.

[6] T. Zhu, L. Kuang, J. Daniels, P. Herrero, K. Li, and P. Georgiou, "IoMT-enabled real-time blood glucose prediction with deep learning and edge computing," IEEE Internet Things J., vol. 10, no. 5, pp. 3706–3719, 2025.

[7] C. Banapuram, A. C. Naik, M. K. Vanteru, V. S. Kumar, and K. Vaigandla, "Machine learning and deep learning approaches for heart, liver, and tuberculosis disease detection: a review," 2024.

[8] E. Badidi, "Edge AI for early detection and continuous monitoring of chronic and infectious diseases," 2023.

[9] Z. Li, L. Wen, J. Liu, Q. Jia, C. Che, C. Shi, and H. Cai, "Fog and cloud computing assisted IoT model based personal emergency monitoring and diseases prediction services," Computing and Informatics, vol. 39, no. 1-2, pp. 5–27, 2020.

[10] S. Kanagaraj, "Leveraging cloud-based deep learning for cardiovascular diseases prediction: challenges and solutions," in Proc. IEEE ICAECA, 2025, pp. 1–6.

[11] S. Mali, N. Mali, F. Zeng, and L. Zhang, "Edge computing with federated learning for early detection of citric acid overdose and adjustment of regional citrate anticoagulation," BMC Med. Inform. Decis. Mak., vol. 25, no. 1, art. 320, 2025.

[12] F. D'Antoni et al., "Edge computing-based neural network glucose concentration prediction in children with type 1 diabetes," Bioengineering, 2020.

[13] P. Sailaja, A. Dinesh, and L. Blossom, "HealthSenseFusion for healthcare industry: IoT based integrated model for remote monitoring and disease prediction," in Proc. GITCON, 2025, pp. 1–7.

[14] Y. Liao, Z. Tang, K. Gao, and M. Trik, "Optimization of resources in intelligent electronic health systems based on Internet of Things to predict heart diseases via artificial neural network," Heliyon, vol. 10, no. 11, 2024.

[15] J. Aswini, B. Yamini, K. Venkata Ramana, and J. Jegan Amarnath, "Mask-RCNN and Pelican Optimization Algorithm for liver disease prediction," 2024.

[16] B. Shayesteh, C. Fu, A. Ebrahimzadeh, and R. Glitho, "Auto-adaptive fault prediction system for edge cloud environments in the presence of concept drift," in Proc. IEEE IC2E, 2021, pp. 217–223.

[17] K. V. S. S. Murthy, R. S. Shankar, S. N. Pradhan, B. Mohanty, and V. V. R. M. Rao, "Comparative analysis of deep learning models for various nonalcoholic fatty liver disease datasets," Int. J. Public Health, vol. 13, no. 4, pp. 2033–2043, 2024.

[18] K. Gupta, N. Jiwani, and N. Afreen, "Liver disease prediction using machine learning classification techniques," in Proc. IEEE CSNT, 2022, pp. 221–226.

[19] K. Jeyalakshmi and R. Rangaraj, "Accurate liver disease prediction system using convolutional neural network," Indian J. Sci. Technol., vol. 14, no. 17, pp. 1406–1421, 2021.

[20] B. Dhiyanesh, M. Rameshkumar, K. Karthick, and R. Radha, "Cloud computing and machine learning for analysis of health care data based on neuro fuzzy logistic regression," J. Intell. Fuzzy Syst., vol. 44, no. 6, pp. 9955–9964, 2023.

[21] H. K. Rudsari et al., "Digital twins in healthcare: a comprehensive review and future directions," Front. Digit. Health, vol. 7, art. 1633539, 2025.

[22] J. Meng, M. Wu, F. Shi, Y. Xie, H. Wang, and Y. Guo, "Medical laboratory data-based models: opportunities, obstacles, and solutions," J. Transl. Med., vol. 23, no. 1, art. 823, 2025.

[23] M. A. Talukder, A. S. Talaat, M. Kazi, and A. Khraisat, "XAI-HD: an explainable artificial intelligence framework for heart disease detection," Artif. Intell. Rev., vol. 58, no. 12, 2025.

[24] M. K. Rahman, M. A. R. Khan, I. Ahammad, and J. R. Das, "Comparative evaluation of machine learning models for stroke prediction in clinical settings," Cloud Comput. Data Sci., pp. 196–216, 2025.