Engineering Applications of Artificial Intelligence 164 (2026) 113240

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Research paper

A hybrid quantum-classical approach for liver disease detection using quantum machine learning

Laura María Donaire [a,*], Gloria Ortega [a], Francisco Orts [a], Ester Martín Garzón [a], Ernestas Filatovas [b]

[a] Department of Informatics, UAL-Health Research Center (CEINSA), University of Almería, Ctra. Sacramento s/n, La Cañada de San Urbano, 04120 Almería, Spain

[b] Institute of Data Science and Digital Technologies, Faculty of Mathematics and Informatics, Vilnius University, Lithuania

ARTICLE INFO

Dataset link: https://github.com/LauraMDonaire/QML-Liver

Keywords:
Liver diseases
Quantum machine learning
Quantum computing
Machine learning
Quantum layers

ABSTRACT

Quantum Machine Learning (QML) combines principles of quantum computing with traditional Machine Learning (ML) to explore computational advantages in data processing and model efficiency. With the rise of Noisy Intermediate-Scale Quantum (NISQ) devices, hybrid quantum–classical approaches are gaining momentum, especially in domains requiring high precision such as healthcare. In this work, we investigate whether hybrid quantum computing can enhance certain aspects of classical ML, specifically in dataset balancing and the complexity of the neural network involved in training. To this end, we use the Indian Liver Patient Dataset as a case study to determine the presence of liver disease. We present the methodology for developing 'QML-Liver', a hybrid approach that seamlessly integrates classical and QML techniques. This includes data preprocessing, model design, and optimal configuration. Our results demonstrate that 'QML-Liver' improves key performance metrics, such as accuracy and F1-Score. Additionally, we successfully reduce the number of required qubits to just two, making practical deployment more feasible. These findings underscore the potential of QML for medical diagnostics, particularly in the NISQ era.

1. Introduction

Quantum Machine Learning (QML) is an emerging field that combines the power of quantum computing with traditional Machine Learning (ML) techniques, offering new opportunities to tackle challenging problems in various domains, including medicine, finance, and logistics (Biamonte et al., 2017; Schuld et al., 2014). ML has transformed numerous fields by enabling automated pattern recognition and decision-making in large datasets. In healthcare, ML models have proven to be powerful tools for disease classification, early diagnosis, and predictive analytics, with traditional techniques such as neural networks, support vector machines, and decision trees being widely used for medical diagnostics, including liver disease detection (Azam et al., 2020; Mutlu et al., 2022; Nahar and Ara, 2018). However, these classical models often face challenges related to data quality, computational limitations, and the complexity of medical patterns, which can hinder their performance. Unlike classical methods that typically rely on large, balanced datasets to achieve optimal performance, quantum models can exploit phenomena such as superposition and entanglement to learn complex patterns with fewer data and without the need for class balancing (Sinno et al., 2025). These properties make QML particularly promising for applications where data collection is costly or where classes are inherently imbalanced, such as rare disease detection or cybersecurity problems.

Numerous studies have explored the application of QML in healthcare, addressing tasks such as medical image analysis (Houssein et al., 2022; Wei et al., 2023), disease detection (Dutt et al., 2020), and large-scale clinical data classification (Dasari et al., 2023). In this domain, supervised learning models, particularly those focused on classification, have shown promising results in medical diagnostics (Maheshwari et al., 2023). Techniques such as quantum preprocessing, error reduction, and hybrid quantum–classical neural networks have contributed to the development of more efficient models for analyzing clinical data (Aishwarya et al., 2020; Cong et al., 2019; Moradi et al., 2022; Sierra-Sosa et al., 2021). Despite its potential, quantum computing faces several limitations, including the restricted availability of qubits in current hardware and the susceptibility of quantum systems to noise and errors due to decoherence, while converting classical data into quantum representations remains a significant technical

* Corresponding author.
E-mail addresses: laura.donaire@ual.es (L.M. Donaire), gloriaortega@ual.es (G. Ortega), francisco.orts@ual.es (F. Orts), gmartin@ual.es (E.M. Garzón), ernestas.filatovas@mif.vu.lt (E. Filatovas).

https://doi.org/10.1016/j.engappai.2025.113240
Received 12 May 2025; Received in revised form 22 September 2025; Accepted 16 November 2025
Available online 22 November 2025

challenge (Preskill, 2018; Ranga et al., 2024). These constraints have motivated the development of hybrid quantum–classical approaches, providing practical solutions within the current Noisy Intermediate-Scale Quantum (NISQ) era. Although scalability and hardware limitations remain challenges for large-scale clinical deployment, recent advances, including hybrid architectures, error mitigation, and efficient quantum preprocessing, have already enabled meaningful applications of QML to clinical datasets. Overcoming remaining hardware and algorithmic bottlenecks will further enhance the deployment of QML systems at clinical scale (Devadas and Sowmya, 2025).

Advancements in medical science are fueling interest in emerging technologies like quantum computing to enhance disease diagnosis and treatment, particularly for high-risk organ systems. The liver, located in the upper right part of the abdominal cavity, is the largest organ in the body and the largest gland, second only to the skin. It represents approximately 4% of body weight (Mutlu et al., 2022). Wedge-shaped, it is essential for digestion and the elimination of toxins, performing over 500 vital functions necessary for human survival (Gupta et al., 2022). Liver conditions, ranging from hepatitis (often caused by viral infections like hepatitis A, B, C, D and E) and cirrhosis to non-alcoholic fatty liver disease, liver tumors, and liver cancer, represent a significant global health issue, claiming approximately 2 million lives worldwide (Dritsas and Trigka, 2023). According to the Global Burden of Disease project, in 2010 alone, one million people died from cirrhosis, while many more were diagnosed with liver cancer (Nigatu et al., 2023). Factors such as excessive alcohol consumption, viral infections, obesity, diabetes, and autoimmune conditions can severely impair liver function (Kumar and Rani, 2024).

Early detection of liver diseases is therefore crucial to improve patient prognosis and reduce mortality rates. Traditional diagnostic methods, such as blood tests, biopsies, and medical imaging, can be invasive, costly, and prone to human error, which has driven the growing adoption of artificial intelligence (AI) in healthcare. AI enables automation in diagnostics and optimization of medical decision-making.

This work presents the design and optimization of a hybrid quantum deep learning model for liver disease classification. Our model integrates both classical and quantum layers, leveraging quantum computing advantages while maintaining an efficient architecture.

The main contributions of this study are:

- Design of a resource-efficient hybrid model featuring a two qubits quantum layer, optimized for deployment on NISQ-era quantum hardware.
- Application of the model to the Indian Liver Patient Dataset (ILPD), a widely-used benchmark dataset for liver disease classification.
- Extensive comparison with both classical and quantum approaches, showing competitive results while minimizing quantum resource usage.
- Full release of code and model parameters to promote transparency and enable reproducibility for future research.

The remainder of the paper is structured as follows. Section 2 discuss QML and its challenges and opportunities. Section 3, reviews classical and quantum classification methods for liver diseases, with a focus on state-of-the-art approaches that use the ILPD. Section 4 details the development of our hybrid quantum model, 'QML-Liver', including the preprocessing steps, model structure, configuration, and training techniques employed. Section 5 presents experimental results and compares our model against leading classical and quantum approaches. Finally, Section 6, discusses the general conclusions of this research and outlines future directions.

2. Quantum machine learning

Quantum computing is distinguished by its ability to process information using the rules of quantum mechanics, enabling the tackling of problems in new and potentially revolutionary ways (Mermin, 2007; Nielsen and Chuang, 2010; Ying, 2010). However, we are currently in the NISQ era, characterized by the use of quantum devices with a limited number of qubits and considerable noise. Despite these constraints, these systems provide valuable opportunities to explore quantum algorithms that, for specific tasks, may outperform classical methods.

QML sits at the intersection of classical ML and quantum computing, aiming to exploit quantum properties to develop models with improved learning capabilities (Biamonte et al., 2017; Tychola et al., 2023; Ganguly, 2021). There are four approaches that combine quantum computing and ML (Ranga et al., 2024):

- Classical-Classical (CC): Classical algorithms with classical datasets.
- Quantum-Classical (QC): Classical algorithms processing quantum datasets.
- Classical-Quantum (CQ): Classical datasets processed on quantum hardware.
- Quantum-Quantum (QQ): Quantum algorithms with quantum datasets.

Although QML is still in its early stages and faces considerable challenges, it holds great promise for solving complex problems more efficiently (Dunjko and Briegel, 2018; Perelshtein et al., 2022). If we focus on the exclusively quantum case (QQ), we can say that, currently, it has several limitations:

- Limitations in the number of qubits: Current devices have few qubits (on the order of hundreds), making it difficult to implement complex quantum neural networks.
- Noise and errors in quantum systems: Decoherence and other noise sources affect the stability and reliability of calculations.
- Encoding classical data into quantum states: Efficiently converting classical information into data that quantum systems can process is a challenge in itself.
- Difficulties in training quantum models: Optimizing parameters in quantum neural networks requires the development of more robust and efficient algorithms.

These constraints originate in the intrinsic physical and architectural features of NISQ hardware. In particular, the lack of fault-tolerant error correction mechanisms and the exponential overhead of characterization methods such as full quantum state tomography severely limit both the reliability and scalability of QQ implementations (Keçeci, 2025). Gate errors, qubit decoherence, and crosstalk introduce complex noise dynamics that accumulate in deep circuits, degrading model fidelity. Furthermore, the probabilistic nature of quantum measurements makes it harder to obtain precise feedback during training, forcing repeated evaluations and indirect estimation methods to guide the optimization of quantum neural networks (Beer et al., 2020).

Beyond the architectural limitations of NISQ hardware, scalability is particularly critical for QML models applied to healthcare, where datasets are often high-dimensional, heterogeneous, and privacy-sensitive (Rasool et al., 2023). Real-world clinical tasks, including molecular simulation, medical precision, radiotherapy, and drug development, require models capable of generalizing across diverse patient populations (Ullah and Garcia-Zapirain, 2024). Techniques such as quantum transfer learning, federated quantum models, and distributed quantum computing architectures have shown promise in extending QML capabilities to larger datasets without compromising privacy or interpretability (Acar and Yılmaz, 2021; Chen and Yoo, 2021; Kawase, 2024). These strategies are especially relevant in precision medicine

and diagnostic imaging, where scalability is not only a technical requirement but also a clinical imperative. As quantum hardware matures, integrating scalable QML pipelines into healthcare workflows will be essential to unlock their full potential.

Complementing these scalability-oriented strategies, several algorithmic approaches have been proposed to enhance QML performance on healthcare data. Among these, quantum kernel methods (QKM), variational quantum circuits (VQCs), and hybrid quantum–classical models have emerged as particularly promising. Each approach offers distinct advantages and is better suited to specific problem settings, depending on factors such as dataset size, model complexity, and hardware limitations.

QKMs exploit quantum computers to map classical data into high-dimensional Hilbert spaces, enabling efficient separation through linear classifiers (Havlíček et al., 2019; Schuld and Killoran, 2019). By computing the kernel matrix via quantum circuits, QKMs can reveal subtle patterns in data that often remain hidden to classical kernel methods (Raubitzek and Mallinger, 2023). However, despite their effectiveness on small datasets, their scalability is hindered by the computational overhead of quantum kernel evaluation on current hardware (Tscharke et al., 2024; Wang et al., 2021).

VQCs, in contrast, are hybrid algorithms that function as quantum counterparts to classical neural networks, particularly multilayer perceptrons (Cerezo et al., 2021; Griol-Barres et al., 2021). They consist of parameterized quantum circuits whose parameters are optimized using classical techniques. Although VQCs can effectively perform classification tasks (Raubitzek and Mallinger, 2023), they often face challenges with high-dimensional data due to the expressiveness limits of shallow quantum circuits (Qi et al., 2024).

Given the trade-offs of these approaches, we adopt a hybrid quantum–classical neural network (QNN) architecture for the task of liver disease classification. Hybrid QNNs align well with current NISQ devices, and offer a practical compromise between expressivity and hardware feasibility. They have shown competitive performance in supervised learning tasks (Combarro and Gonzalez-Castillo, 2023; Skolik et al., 2022), and their adaptability makes them promising tools for biomedical data analysis.

Our hybrid model combines classical neural layers with a quantum layer, leveraging the strengths of both paradigms. The hybrid setup helps address these limitations by offloading part of the learning to classical layers, reducing quantum circuit depth while still harnessing quantum processing. This is especially advantageous given the characteristics of our dataset, which contains 10 features and 583 samples (see Section 4.1), small enough to benefit from quantum components, but also requiring a robust classical backbone for generalization.

3. Review of classical and quantum classification of liver diseases

In this section, we will examine the state-of-the-art using the ILPD database and both approaches: classical ML and QML (from 2017 to 2024). The literature review was conducted using the snowballing methodology, starting from key references and expanding the search iteratively based on the citations and references of relevant works.

3.1. Classical machine learning approaches

Several studies have employed the ILPD dataset with various classical ML algorithms to predict liver disease outcomes. Table 1 summarizes the review of classical methods conducted from 2017 to 2024, highlighting the best-performing models based on accuracy and recall. The table also includes other reported metrics such as precision and F1-Score.

One of the early works, by Sontakke et al. (2017), employed techniques like over-sampling and under-sampling to address class imbalance in the dataset. The best model, in terms of accuracy and recall, was the Back Propagation Neural Network (BP). In comparison, the Support Vector Machine (SVM) model also achieved competitive results. The author does not provide the F1-Score and recall metrics directly, but instead reports sensitivity, which is equivalent to recall. The F1-Score has been calculated as the harmonic mean of precision and sensitivity (recall), as can be seen in Table 1.

In a subsequent study, Nahar and Ara (2018) explored a range of decision tree-based algorithms, including Random Forest (RF), Random Tree, Decision Stump, Hoeffding Tree, and others. They performed a 10-fold cross-validation on the dataset. The best model, in terms of accuracy and recall, was the Decision Stump (with a single level). The second-best model in terms of accuracy and recall was the Hoeffding Tree (with a tree size of one), as can be observed in Table 1.

In 2020, Sokoliuk et al. (2020) applied several classification algorithms, including Decision Tree (DT), RF, SVM, Multi-Layer Perceptron (MLP), Naïve Bayes (NB), and others, to analyze the ILPD dataset. This author does not mention the precision and F1-Score metrics, so they will not be included in this review. The author provides results for both balanced and unbalanced datasets. They used GridSearchCV to find the optimal parameters for each Scikit-learn algorithm and preprocessed the data in the most suitable way for each explored algorithm. The best model for the balanced dataset, in terms of accuracy and recall, was K-Nearest Neighbors (KNN). The second-best model was MLP. Using an unbalanced dataset, their best models were, in terms of accuracy, the GradientBoosting model, and in terms of recall, the GaussianNB model, as can be seen in Table 1.

That same year, Azam et al. (2020) explored the use of RF, Perceptron, DT, KNN, and SVM for classification. They applied feature selection techniques and reported results both with and without feature selection. With feature selection (WFS), the highest performance was achieved using KNN. The second-best model in terms of accuracy and recall was the RF model. Without feature selection (WOFS), the best-performing model was SVM, while KNN also yielded competitive results (see Table 1). Gajendran and Varadharajan (2020) experimented with a Mathematical Approach on Multilayer Feedforward Neural Networks with Backpropagation (MAMFFN). They reported accuracy results both before and after applying feature selection, achieving the highest accuracy with feature selection, as can be observed in Table 1.

In 2021, Kumar and Thakur (2021) proposed a method called Variable-Neighbor Weighted Fuzzy K Nearest Neighbor Approach (V-NWFKNN), based on existing NWKNN and Fuzzy-NWKNN methods. They used normalization and standardization. For the majority of instances, they used undersampling by eliminating of Tomek link pairs and redundant pairs (TL_RUS). Without using TL_RUS, their method achieved moderate success. When TL_RUS was applied, the performance improved significantly (see Table 1). Geetha and Arunachalam (2021), also in 2021, evaluated two ML techniques: Logistic Regression (LR) and SVM. The SVM achieved the highest performance, while LR also performed well. The F1-Score has been calculated as the harmonic mean of precision and sensitivity, as can be seen in Table 1.

In 2022, Gupta et al. (2022) also conducted research on liver disease prediction by employing various ML algorithms, including LR, RF, Gradient Boosting, Light GB, and others. Among the studied algorithms, RF achieved the best results, followed closely by Light GB (see Table 1). Mutlu et al. (2022) investigated various ML techniques, such as KNN, LR, SVM, and NB, and proposed a CNN model with four layers consisting of 68, 70, 70, and 2 neurons, respectively. They also incorporated Principal Component Analysis (PCA) and SMOTE techniques to optimize model performance on the ILPD dataset. With these enhancements, Mutlu et al. (2022)'s CNN model achieved the highest performance among the studied models, as summarized in Table 1.

In 2023, Dritsas and Trigka (2023) applied various ML algorithms to the ILPD dataset, including NB, AdaBoostM1, Voting, KNN, and others. They applied SMOTE to balance the dataset and evaluated with 10-fold cross-validation. The best-performing model was the Voting model, which combined RF and AdaBoostM1 classifiers. The second-best model

Table 1
Summary of studies using the ILPD dataset with classical ML algorithms for liver disease prediction. The table highlights the two best-performing models from each study based on accuracy and recall, including their reported precision and F1-Score when available. An asterisk (*) in the F1-Score column indicates that the metric was computed rather than explicitly reported by the authors. Likewise, 'No' in the 'Balancing' column signifies that it was not specified whether data balancing techniques were applied. A dash (–) denotes that the corresponding metric was not reported.

Author	Model	Balancing	Accuracy	Precision	Recall	F1-Score
Sontakke et al. (2017)	SVM	Yes	71	64	72	68*
	BP	Yes	73	66	73	69*
Nahar and Ara (2018)	Decision Stump	No*	71	50	71	59
	HoeffdingTree	No*	70	63	70	62
Sokoliuk et al. (2020)	GradientBoosting	No	72	–	40	–
	GaussianNB	No	55	–	95	–
	KNN	Yes	74	–	**97**	–
	MLP	Yes	63	–	92	–
Azam et al. (2020)	KNN_WFS	No*	74	72	74	72
	RF_WFS	No*	73	74	73	73
	SVM_WOFS	No*	71	80	71	60
	KNN_WOFS	No*	66	64	66	65
Gajendran and Varadharajan (2020)	MAMFFN_WOFS	Yes	72	–	–	–
	MAMFFN_WFS	Yes	75	–	–	–
Kumar and Thakur (2021)	V-NWFKNN (No TL_RUS)	No	78	90	82	86
	V-NWFKNN (TL_RUS)	Yes	88	**95**	90	**93**
Geetha and Arunachalam (2021)	LR	No*	73	79	88	83*
	SVM	No*	75	77	79	78*
Gupta et al. (2022)	RF	Yes	63	64	63	63
	Light GB	Yes	63	63	62	63
Mutlu et al. (2022)	CNN	Yes	72	74	75	75
Dritsas and Trigka (2023)	Voting	Yes	80	80	80	80
	AdaBoostM1	Yes	80	80	80	80
Nigatu et al. (2023)	ANN	No*	87	–	–	–
	SGD	No*	81	–	–	–
Elsayed et al. (2024)	SVM	No*	71	71	71	83
	ZeroR	No*	71	71	71	83
Raj et al. (2024)	RF	No*	72	–	–	–
	KNN	No*	81	–	–	–
Kumar and Rani (2024)	RF_WOOP	Yes	80	79	82	81
	SVM_WOOP	Yes	68	63	88	73
	RF_AFOP	Yes	81	78	86	82
	SVM_AFOP	Yes	81	79	89	82
Alyasin and Ata (2024)	Stacking	Yes	**90**	90	90	90
	ET	Yes	88	92	82	87

was AdaBoostM1 with an RF classifier, as can be observed in Table 1. Similarly, Nigatu et al. (2023) implemented RF, DT, Stochastic Gradient Descent (SGD), ANN, and others. They applied hyperparameter tuning with GridSearchCV. They only reported the accuracy metric, with ANN achieving the highest accuracy, followed by the SGD model (see Table 1).

In 2024, Elsayed et al. (2024) employed several ML algorithms, including NB, SVM, ZeroR, and Voting Feature Intervals. Their classification algorithms were implemented using the WEKA tool and 10-fold cross-validation. They reported that both the SVM and ZeroR algorithms achieved the same accuracy, precision, recall, and F1-Score. Raj et al. (2024), also in 2024, applied LR, KNN, DT, SVM, and RF. They only reported the accuracy metric, with RF achieving the highest accuracy, followed closely by the KNN model (see Table 1). Kumar and Rani (2024) explored the use of AdaBoost, XGBoost, SVM, and RF for classification. Their best-performing model using optimization (AFOP) was RF, followed by SVM. Without hyperparameter optimization (WOOP), their two best models were RF and SVM, as can be observed in Table 1. Lastly, Alyasin and Ata (2024) proposed five ML models: RF, XGB, DT, ExtraTrees, and Stacking. Their stacking model, which used RF, DT, XGB, and ExtraTrees as fundamental classifiers, achieved the highest performance. The second-best model was ExtraTrees (see Table 1).

Training classical systems with imbalanced datasets poses a significant challenge, as it can lead to biased models and reduced predictive performance. In the following, we examine how QML performs with this imbalanced dataset.

3.2. Quantum machine learning approaches

In recent years, several studies have explored the use of QML techniques with the ILPD dataset to predict liver disease outcomes. Table 2 provides a summary of quantum methods analyzed between 2023 and 2024, emphasizing the models with the highest accuracy and recall. Additionally, the table presents other reported metrics, including precision, F1-Score, number of qubits and quantum delay.

In 2023, Raubitzek and Mallinger (2023) explored the applicability of QML for classification tasks, utilizing two quantum classifiers: VQC and the Quantum Kernel Estimator (QKE). Their results demonstrate that both the VQC and QKE outperform basic ML algorithms, such as advanced linear regression models (Ridge and Lasso), achieving an accuracy of 74.4% (see Table 2). However, they conclude that while QML algorithms show potential in achieving competitive performance on certain datasets, they do not consistently outperform classical ML algorithms.

In 2024, Raubitzek et al. (2024) further advanced the field by employing QKE once again. By leveraging the diverse structures of Lie groups, the authors developed novel quantum-inspired feature maps

Table 2
Comparison of the two best models per study in terms of recall and accuracy for QML. Metrics marked with * indicate that 'EfficientSU2' is used, but the specific configuration is not provided. We assume the minimum delay setting. The symbol '–' denotes that the corresponding metric was not reported, while 'N/A' indicates that the concept is not applicable in the given context.

Author	Model	Balancing	Nº Qubits	Delay	Accuracy	Precision	Recall	F1-Score
Raubitzek and Mallinger (2023)	VQC	No	10	11*	74	–	–	–
	QKE	No	10	11*	74	–	–	–
Raubitzek et al. (2024)	Catboost	No	N/A	N/A	73	65	63	64
	Qz	No	N/A	N/A	72	65	65	65
Safriandono et al. (2024)	XGB_QFE Tomek	Yes	10	40	**81**	**77**	90	83
	LR_QFE	No	10	40	74	75	**99**	**85**
Bhaskaran and Prasanna (2024)	QKNN(K = 5)	No	10	23	69	–	–	–
	QKNN(K = 3)	No	10	23	67	–	–	–

that offer a more flexible and potentially powerful method for encoding and compressing classical data into quantum states. In particular, they employ the EfficientSU2 ansatz, it consists of alternating layers of single-qubit SU(2) gates and CNOT entanglement gates. The SU(2) group includes 2×2 unitary matrices with a determinant of one, such as Pauli rotation gates (Alami et al., 2025). They provide a comprehensive theoretical foundation for this approach, followed by a methodology that integrates these feature maps into quantum-inspired kernel classifiers. Their experiments consider kernel matrices based on Lie groups within the framework of a Support Vector Machine classifier. For the ILPD dataset, their best-performing model, CatBoost, achieved an accuracy of 73.14%, a precision of 64.78%, a recall of 62.79%, and an F1-Score of 63.49%. The second-best result was obtained using the Qz model, which achieved an accuracy of 72%, a precision of 64.51%, a recall of 64.70%, and an F1-Score of 64.60% (see Table 2). It is important to note that this study is based on a mathematical simulation and does not use actual qubits. The approach emulates quantum behavior through classical computations, simulating quantum-inspired methods rather than implementing them on quantum hardware.

In the same year, Safriandono et al. (2024) aimed to improve the accuracy of liver disease classification using Quantum Feature Engineering (QFE) combined with the Synthetic Minority Over-Sampling Technique and Tomek Links (SMOTE-Tomek) for data balancing. They employed LR, SVM, RF, and XGB, finding that the combination of QFE with SMOTE-Tomek, using the XGB model, achieved an accuracy of 81%, a precision of 77%, a recall of 90%, and an F1-Score of 83%. The second-best result was obtained using an imbalanced dataset with QFE in an LR model, achieving an accuracy of 74%, a precision of 75%, a recall of 99%, and an F1-Score of 85%, a summary of these metrics can be found in Table 2. The last study we reviewed is by Bhaskaran and Prasanna (2024), whose objective was to conduct a comprehensive accuracy analysis of classical and quantum-enhanced KNN algorithms using the Canberra distance metric across various datasets. Their experiment compared classical k-nearest neighbors (KNN) with $K = 3$ and $K = 5$ against quantum-enhanced KNN (QKNN) with $K = 3$ and $K = 5$ across five different datasets. For the ILPD dataset, their proposed QKNN ($K = 5$) model, using the Canberra distance, achieved the highest accuracy of 72.64%. The second-best accuracy was obtained with QKNN ($K = 3$), reaching 66.66% (see Table 2). Table 2 summarizes the best-performing QML models from the reviewed studies, based on recall an accuracy. The table includes key performance metrics such as accuracy, precision, recall, and F1-Score. Note that some models do not report certain metrics, which are left blank.

As shown in Table 2, existing QML models applied to the ILPD dataset require a large number of qubits or have considerable circuit depth, which may limit their practicality on NISQ-era hardware. These constraints underscore the need for more efficient and balanced hybrid approaches like the proposed model 'QML-Liver', which can deliver competitive performance while minimizing quantum resource usage (see Section 5).

4. Developed quantum approach (QML-Liver)

In this section, we outline the dataset used, the methodology used to develop our QML model, 'QML-Liver'.

In Fig. 1, we detail the complete methodology for developing the hybrid neural network. Specifically, we outline the data preprocessing steps, the selected interfaces, and the construction of the hybrid model, which is based on the optimization of both classical and quantum hyperparameters of the quantum layer. After that, we describe the different model configurations designed to optimize performance. Finally, the model is evaluated using several metrics.

4.1. Indian liver patient dataset

In this work, we have used the Indian Liver Patient Dataset (ILPD), one of the most widely used databases for liver disease detection, available in the UCI Machine Learning Repository.[1] This dataset contains 583 records of patients from the northeast of Andhra Pradesh, India, presented by Ramana et al. (2012). Among them, 416 are diagnosed with liver disease and 167 are not. Each patient is described by 10 numerical variables, including age, gender, and biochemical markers such as Total Bilirubin (TB), Direct Bilirubin (DB), Total Proteins (TP), Albumin (ALB), Albumin and Globulin Ratio (A/G Ratio), Alamine Aminotransferase (SGPT), Aspartate Aminotransferase (SGOT), and Alkaline Phosphatase (Alkphos). These variables are represented using integer, floating-point values and categorical values.

Additionally, the dataset includes a categorical variable called 'Selector', which provides ground truth information by indicating whether the patient has liver disease or not, based on expert labeling.

4.2. Data preprocessing

Data preprocessing is a crucial step to ensure that ML models can handle the dataset effectively, especially in the case of medical data like the Indian Liver Patient Dataset, which contains categorical variables, missing values, and scale differences between its attributes. Fig. 2 shows a diagram of the main preprocessing steps carried out on the database.

Preprocessing steps are described in detail below.

1. **Dummy Encoding.** The dataset includes a categorical gender variable with values 'Female' and 'Male'. To facilitate analysis and make the data compatible with ML algorithms, these categories were encoded as binary values: '0' for Female and '1' for Male.

[1] https://archive.ics.uci.edu/dataset/225/ilpd+indian+liver+patient+dataset

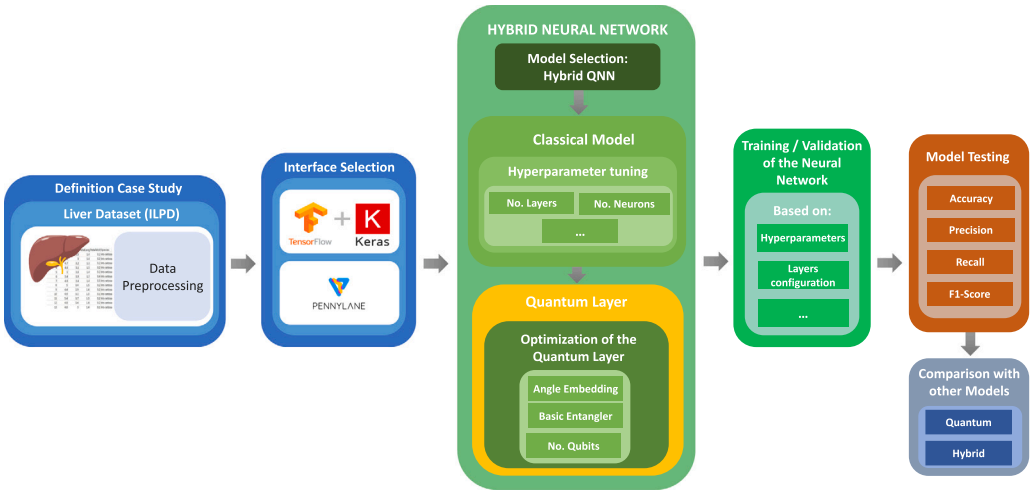

Fig. 1. Diagram of this work project, illustrating the stages from left to right for training a neural network for classification using the ILPD dataset.

Fig. 2. Steps followed in preprocessing techniques for the ILPD dataset.

2. **Imputation of Missing Values.** Only 4 out of 583 values in the A/G Ratio column were missing, which were imputed using the mean (0.947), a value very close to the median (0.93) and consistent with the near-symmetric distribution of the feature. Given the minimal proportion of missingness and the concentration of most values between 0.70 and 1.10 (25th–75th percentile), mean imputation was considered appropriate, as it preserves the central tendency without introducing significant bias (Feature-engine, 2025; Joel et al., 2024). In the biomedical context, this approach also maintains the original scale and interpretability of the biochemical marker, ensuring that subsequent analyses and predictive model outputs remain clinically meaningful and robust (Williams, 2025).
3. **Re-coding the Target Variable.** The target variable, named 'Selector', was originally encoded with values '2' (representing 'healthy') and '1' (representing 'sick'). To improve the interpretation and consistency of the results, this column was re-coded to binary values, where '0' represents 'healthy' and '1' represents 'sick'. This change simplifies the analysis of prediction results in terms of liver disease probability, makes the interpretation of model performance metrics easier, and is also necessary for the correct visualization of the confusion matrix, which expects a binary variable.
4. **Split and Deduplicate.** We randomly selected an equal number of positive and negative cases for the test set. However, after conducting multiple evaluation runs, we observed that certain records were consistently misclassified across all 10 executions (see Appendix). This pattern suggested that these records could introduce ambiguity or inconsistencies in the evaluation process. To ensure a more reliable assessment of model performance, we decided to remove these cases from the test set and any duplicated rows (13 rows). The final datasets consist of 432 training cases (307 positive and 125 negative) and 146 test cases (80 positive and 66 negative).
5. **Data Standardization.** The numerical attributes of the dataset exhibit different scales, which could affect the performance of certain algorithms sensitive to scale. To prevent this, all attributes were standardized to have a mean of 0 and a standard deviation of 1 using the `StandardScaler` from the `scikit-learn` library. Importantly, the scaler was fitted only on the training set (X_{train}) to compute the mean and standard deviation, and then applied to transform both the training and test sets. During the cross-validation procedure, the scaler was fitted exclusively on the training folds and then applied to the corresponding validation folds. After model selection, it was finally fitted on the complete training set and applied to both the training and test sets. Standardization ensures that the variables have a balanced influence on the model, preventing features with broader ranges from dominating the analysis.
6. **SMOTE.** Given that the dataset is unbalanced, we applied the SMOTE technique from the `imbalanced-learn` library to generate synthetic instances for the minority class, which is the healthy class. SMOTE was applied only to the training set after standardization, ensuring that the test set remained untouched and unbiased. This resulted in a balanced dataset with 832 instances. The impact of applying SMOTE was analyzed, and the results of these tests can be observed in Table 7.

4.3. Proposed quantum machine learning model

The proposed model is developed within a computational framework that integrates several key tools for quantum and classical ML. PennyLane[2] provides essential tools for implementing hybrid quantum–classical models, while TensorFlow[3] and Keras[4] enable the construction and training of neural networks. Finally, Jupyter Notebook[5] serves as an interactive development environment, seamlessly integrating these resources.

After preprocessing, the dataset is divided into two sets: training and validation (75%) and test (25%) to evaluate new inference cases. To assess the performance of the ML models, we focused on Accuracy and Recall, as these are the most relevant metrics in the context of liver disease diagnosis. Accuracy provides an overall assessment of the classification task by measuring the proportion of correctly predicted cases. However, given the potential consequences of misdiagnosing

[2] https://pennylane.ai/
[3] https://www.tensorflow.org/
[4] https://keras.io/about/
[5] https://jupyter-notebook.readthedocs.io/en/latest/

Table 3
Hyperparameter configurations evaluated using GridSearchCV.

Hyperparameter	Value 1	Value 2	Value 3	Value 4	Value 5
Optimizer	Adam	RMSprop	–	–	–
Hidden Layers	1	2	3	4	–
Dropout rate	0	0.2	0.3	0.4	–
Hidden Neurons	32	64	128	256	512
Loss Function	Binary Crossentropy	Binary Focal Crossentropy	–	–	–

Table 4
Explored hyperparameter configurations (neurons, activation function, and kernel initializer) for optimizing the network using GridSearchCV.

Hyperparameter	Value 1	Value 2	Value 3	Value 4
Neurons for Layer 1	64	128	256	512
Neurons for Layer 2	32	64	128	256
Activation Function for Layers 1 and 2	ReLU	tanh	–	–
Kernel_initializer for Layers 1 and 2	GlorotUniform	HeNormal	–	–

a patient, Recall is particularly critical, as it quantifies the ability to correctly identify individuals with liver disease, minimizing false negatives. Since missing a sick patient can have severe implications, Recall takes priority in our evaluation (Gupta et al., 2021).

Moreover, the model architecture for 'QML-Liver' is defined as follows. A hybrid quantum–classical approach is employed, where classical layers extract relevant features and quantum layers leverage the unique properties of quantum computation to capture complex patterns in the data.

4.3.1. Hyperparameters tuning

Deep learning is a highly effective ML approach, partly due to the large number of hyperparameters, such as the number of hidden layers and nodes, that can be tuned to enhance model performance (Jiang and Chuhan, 2022). Hyperparameter tuning aims to find the optimal combination of these values to produce the best predictive model, but the high dimensionality of the search space often poses a significant computational challenge (Shen, 2018).

Grid search is a parameter tuning technique that systematically constructs and evaluates a model for each possible combination of algorithm parameters defined within a grid (Ranjan et al., 2019). In this study, we use *GridSearchCV*, implemented in *Python* via the *scikit-learn* library, to identify optimal hyperparameters, complementing it with manual adjustments for further optimization. GridSearchCV exhaustively searches over a predefined set of hyperparameter values by evaluating model performance with cross-validation. The procedure of GridSearchCV can be summarized as follows:

Algorithm 1 GridSearchCV procedure

Require: Parameter grid $\mathcal{P}$, estimator f, number of folds K, evaluation metric $\mathcal{M}$
for each configuration $p \in \mathcal{P}$ **do**
  Initialize average score $avg\_score \leftarrow 0$
  for each fold $i = 1, \ldots, K$ **do**
    Split dataset into training set $D^{(i)}_{\text{train}}$ and validation set $D^{(i)}_{\text{val}}$
    Train estimator f with configuration p on $D^{(i)}_{\text{train}}$
    Evaluate f on $D^{(i)}_{\text{val}}$ using $\mathcal{M}$ and obtain score s_i
    $avg\_score \leftarrow avg\_score + s_i$
  end for
  Compute mean performance: $avg\_score \leftarrow avg\_score / K$
  Store result pair (p, avg_score)
end for
Select best configuration $\hat{p} = \arg\max_p avg\_score$
return Best hyperparameter configuration $\hat{p}$

We conducted multiple GridSearchCV runs, each time assigning a specific set of values to the hyperparameters. To define the search ranges, we adopted an empirical approach based on preliminary experiments, manual exploration, and an assessment of computational resources and time constraints. In each GridSearchCV execution, we randomly selected a subset of values from the predefined ranges for each hyperparameter (see Tables 3 and 4), considering the maximum number of configurations that could be evaluated within a reasonable timeframe. This approach allowed us to efficiently explore the hyperparameter space while avoiding computationally prohibitive searches. We have chosen *RMSprop* (Root Mean Square Propagation) and *Adam* (Adaptive Moment Estimation) as optimizers, both of which are commonly used in deep learning. RMSprop adjusts the learning rate by dividing the gradient by a moving average of its squared values, helping to stabilize training (GeeksforGeeks, 2025). Its update rule is (PyTorch, 2025):

$$s_t = \beta s_{t-1} + (1-\beta) g_t^2, \quad \theta_t = \theta_{t-1} - \eta \frac{g_t}{\sqrt{s_t} + \epsilon} \tag{1}$$

where θ_t is the parameter at step t, $g_t = \nabla_\theta \mathcal{L}(\theta_{t-1})$ is the gradient of the loss, s_t is the exponentially weighted moving average of the squared gradients, η is the learning rate, β is the decay factor of the moving average (typically 0.9), and ϵ is a small constant to prevent division by zero (typically 10^{-8}).

Adam combines momentum and adaptive learning rate adjustment by using moving averages of the first and second moments of the gradients (Kingma and Ba, 2017). Its update rule is:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \quad v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2, \quad \hat{m}_t = \frac{m_t}{1-\beta_1^t},$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}, \quad \theta_t = \theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{2}$$

where m_t is the first moment (mean) of the gradients, v_t is the second moment (mean of squared gradients), $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected estimates of m_t and v_t, β_1 and β_2 are decay rates for the moving averages (typically 0.9 and 0.999), and θ_t, g_t, η, and ϵ are as defined above.

Regarding the number of hidden layers, we evaluated configurations ranging from 1 to 4 layers, as increasing the depth further was computationally impractical. Additionally, we tested five different values for the number of neurons in these layers to explore their impact on model performance. We also incorporated dropout, a regularization technique where neurons are randomly dropped during training. This helps reduce computational cost and mitigates overfitting by preventing co-adaptation of neurons (Srivastava et al., 2014). The dropout rate was tested with multiple values to assess its effectiveness. Finally, we evaluated two loss functions: *Binary Crossentropy* and *Binary Focal Crossentropy*. Binary Crossentropy (BCE) measures the dissimilarity between predicted probabilities and true labels in binary classification, penalizing incorrect predictions with high confidence (Mao et al.,

2023). It is defined as:

$$\mathrm{BCE}(y,\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\Big[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\Big] \quad (3)$$

where N is the total number of samples in the dataset, $y_i \in \{0,1\}$ is the true label of the ith sample, and $\hat{y}_i \in [0,1]$ is the predicted probability for the positive class of the ith sample.

Binary Focal Crossentropy is a variant of BCE that reduces the influence of well-classified samples and emphasizes harder ones, making it useful for imbalanced datasets (Lin et al., 2017). Its formulation is:

$$\mathrm{BFCE}(y,\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\Big[\alpha(1-\hat{y}_i)^{\gamma} y_i \log(\hat{y}_i) + (1-\alpha)\hat{y}_i^{\gamma}(1-y_i)\log(1-\hat{y}_i)\Big] \quad (4)$$

where $\alpha \in [0,1]$ balances the relative importance of positive and negative classes, $\gamma \geq 0$ is the focusing parameter that reduces the weight of well-classified examples, and y_i, $\hat{y}_i$ and N are as defined above.

The GridSearchCV suggested that the best parameters were: Adam optimizer, one hidden layer, a dropout rate of 0.3, 32 hidden neurons, and the Binary Focal Crossentropy loss function, achieving an accuracy of 73%. However, while GridSearchCV suggested using a single hidden layer, further experiments demonstrated that adding a second hidden layer improved the model's capacity to learn more complex representations, resulting to a 5% increase in accuracy. It is important to mention that in this study we also explored architectures with more layers. However, since the performance metrics only improved slightly (around 2% in accuracy), we decided to adopt a single architecture with two hidden layers, each followed by a dropout layer. Therefore, we continued our investigation with two hidden layers, the Adam optimizer, the same dropout rate, and the Binary Focal Crossentropy loss function.

Next, we tested different numbers of hidden neurons per layer and experimented with different kernel initializers (see Table 4).

The choice of kernel initializer plays a crucial role in the training stability and convergence speed of neural networks. In this study, we explored two well-established initialization techniques: *GlorotUniform* and *HeNormal*. The GlorotUniform initializer, also known as *Xavier* initialization, aims to maintain a stable variance of activations across layers, making it particularly suitable for activation functions like *'tanh'* (Glorot and Bengio, 2010). On the other hand, the HeNormal initializer or *MSRA* initialization is optimized for *ReLU* functions by scaling the weight distribution based on the number of input neurons, helping to prevent vanishing or exploding gradients (He et al., 2015).

GridSearchCV recommended using ReLU functions in both layers, with 64 neurons in the first layer and 128 in the second. Additionally, it suggested using GlorotUniform initialization for the first layer and HeNormal for the second. However, after several tests, we chose 256 neurons for the first layer and 128 for the second using the GlorotUniform initializer in both layers.

Quantum Layer

Now that the classical part of the hybrid network has been optimized, it is time to focus on the quantum layer. The position of the quantum layer within the architecture was manually tested at different locations. Additionally, experiments were conducted using 2, 3, 4, 5, and 10 qubits. However, due to performance considerations (with accuracy decreasing by 4% and recall by 8%) and the limited availability of qubits, two qubits was ultimately selected for the final implementation.

The proposed quantum layer is implemented as a *QNode* using *Keras* and operates on a small number of qubits. We define this QNode using `default.qubit`, the default qubit-based simulator in *PennyLane*, and leverage operations from the templates module. For our experiments, we use `n_layers = 4`, and define the device as `dev = qml.device("default.qubit", wires=n_qubits, seed=42)`

Quantum circuits are typically composed of three main blocks:

1. Data encoding (Encoding/Feature Map): In this stage, classical data are mapped to the initial state of the quantum register, which increases the dimensionality of the Hilbert space and facilitates the representation of complex relationships among features (Lloyd et al., 2020; Lee and Banerjee, 2023). Common encoding methods include Pauli rotations, amplitude preparation, or angle encoding. In our model, `qml.AngleEmbedding` is used to encode all classical features into *n_qubits* qubits. After preliminary tests with different rotation options (`RX`, `RY`, `RZ`), we observed that `RX` provided greater stability during training and better parameter convergence.

```
# Angle embedding
qml.AngleEmbedding(inputs, wires=range(n_qubits))
```

2. Trainable layers and entanglement: The next block corresponds to the construction of trainable quantum layers along with the entanglement of the gates they include. Following the embedding, `qml.BasicEntanglerLayers` (Xanadu Quantum Technologies Inc., 2025) are applied, which consist of single-parameter rotations on each qubit (default `RX`) followed by a closed chain of CNOT gates connecting all qubits in a ring. The number of layers L is determined by the first dimension of the trainable weight tensor of shape (L, n_qubits). In our case, 4 layers with `RX` rotations are used.

```
# Trainable layers
weight_shapes = {"weights": (n_layers, n_qubits)}
qml.BasicEntanglerLayers(weights, wires=range(n_qubits))
```

This component enables the circuit to capture complex non-linear patterns in the data, making it particularly effective for detecting intricate medical features and improving sensitivity to minority classes in imbalanced datasets (Devadas and Sowmya, 2025; Kwon et al., 2025; Bai and Hu, 2024).

3. Measurement and readout: Finally, the circuit measures the expectation value of the Pauli-Z operator on each qubit, converting quantum states into classical values usable for evaluation or classification.

```
# Measurement
return [qml.expval(qml.PauliZ(w)) for w in range(n_qubits)]
```

This design ensures a minimal quantum resource requirement, making it well-suited for NISQ-era applications. Therefore, the proposed sequential model ultimately consists of the following layers:

- Input layer with one neuron for each feature of the dataset.
- First dense layer with 256 neurons and *ReLU* activation with a weight initializer *GlorotUniform* to improve gradient flow.
- Dropout layer with a rate of 0.3 to help prevent overfitting.
- Second dense layer with 128 neurons, *ReLU* activation with a weight initializer *GlorotUniform*.
- Dropout layer with a rate of 0.3.
- Third dense layer with two neurons.
- Quantum layer with only two qubits.
- Output layer with one neuron and *Sigmoid* activation for binary classification.

The choice of a simple model is motivated by the need for efficient execution on near-term quantum devices, where circuit depth and qubit count are constrained. By keeping the quantum component minimal, the model remains practical for NISQ quantum hardware era while

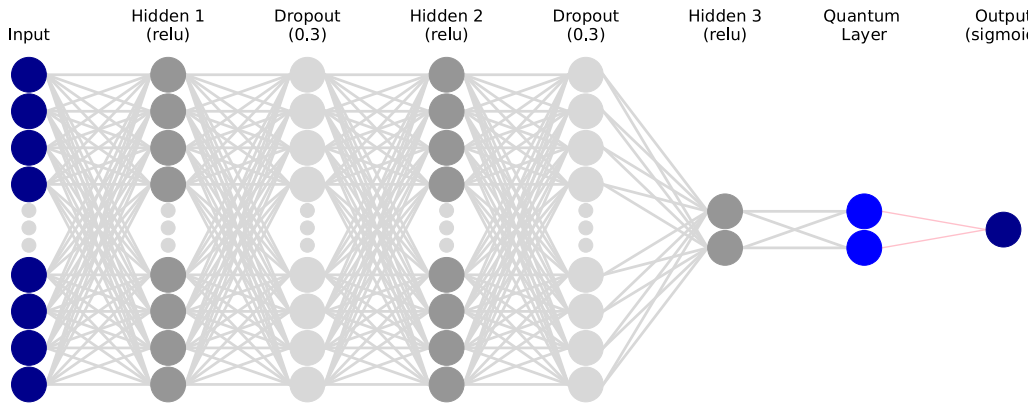

Fig. 3. Structure of the hybrid 'QML-Liver' model, illustrating classical layers with activation functions, dropout layer, and the quantum layer.

Table 5
Evaluated parameter settings for the proposed model, including learning rate, batch size, epochs, and loss-related factors.

Parameter	Value 1	Value 2	Value 3	Value 4
Learning Rate	0.0001	0.001	0.01	–
Batch Size	16	32	64	–
Epochs	20	50	100	–
Gamma (γ)	0.5	1.0	2.0	–
Alpha (α)	0.25	0.3	0.5	0.75
Smoothing Factor	0.3	0.4	0.5	0.7

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_2 (Dense)             (None, 256)               2816

 dropout_2 (Dropout)         (None, 256)               0

 dense_3 (Dense)             (None, 128)               32896

 dropout_3 (Dropout)         (None, 128)               0

 dense_4 (Dense)             (None, 2)                 258

 keras_layer (KerasLayer)    (None, 2)                 8

 dense_5 (Dense)             (None, 1)                 3

=================================================================
Total params: 35981 (140.55 KB)
Trainable params: 35981 (140.55 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig. 4. Layer-by-layer architecture of the hybrid 'QML-Liver' model, as obtained from the Keras `model.summary()` representation.

still allowing for an exploration of potential quantum advantages. The structure of the hybrid quantum–classical model is illustrated in Fig. 3.

In addition, Fig. 4 shows the architecture of the model as obtained from the Keras `model.summary()` function, providing a layer-by-layer representation of the hybrid network.

Class Weight Initialization Additionally, class weight initialization has been applied to prevent extreme weight variations that could negatively impact model training. First, the original class weights are computed using the `compute_class_weight` function with the 'balanced' option, ensuring proper weighting for imbalanced classes. Then, a smoothing factor α is defined to adjust the calculated weights. Finally, the smoothing formula is applied to the computed weights:

$$\text{adjusted\_weight}_i = 1 + \alpha \times (\text{weight}_i - 1) \tag{5}$$

where:

- adjusted_weight_i is the adjusted weight for class i, which will be used during training to compensate for class imbalance.
- weight_i is the original class weight for class i, computed by the `compute_class_weight` function to account for the class distribution.
- α is the smoothing factor which controls how much the class weights are adjusted. A higher value of α increases the effect of the smoothing on the class weights.

Training Parameters The workflow to train the model was as follows: first, hyperparameter combinations were evaluated using *Stratified Cross-Validation* with 5 folds to identify the best configuration. Next, the model was retrained with the selected hyperparameters, again using 5-fold cross-validation, to assess generalization performance across folds. Finally, the model was trained on the entire training dataset with the best hyperparameters, and its performance was evaluated on the held-out test set to report the final metrics presented in Section 5.

To configure the model, it was necessary to configure the number of epochs, batch size, and learning rate. Additionally, *Early Stopping* was applied to monitor the validation loss (`val_loss`), with a patience of 10 epochs, meaning training would stop if no improvement was observed. We continued with the *Adam* optimizer, and the loss function selected was *Binary Focal Crossentropy*. In this loss function, the parameters gamma (γ) and alpha (α) were also configured to balance the influence of different classes. α is a weighting factor for class 1, while the weight for class 0 is $1 - \alpha$. γ is a focusing parameter that adjusts the focal factor (Lin et al., 2017).

We performed an extensive exploration of hyperparameter combinations to optimize the model's performance (see Table 5). The tested configurations varied across key aspects, including learning rate, batch size, number of epochs, class weight smoothing factors, and loss function parameters such as α and γ. A total of 1296 unique combinations were evaluated, resulting from the Cartesian product of the selected values, i.e., $3 \times 3 \times 3 \times 3 \times 4 \times 4 = 1296$, chosen based on preliminary experiments and prior experience.

The impact of these combinations on key performance metrics, such as accuracy and recall, was analyzed in detail to optimize the model's performance.

In Figs. 5, 6, and 7, we compare Test Accuracy and Test Recall across different hyperparameter combinations. Each point in the plots

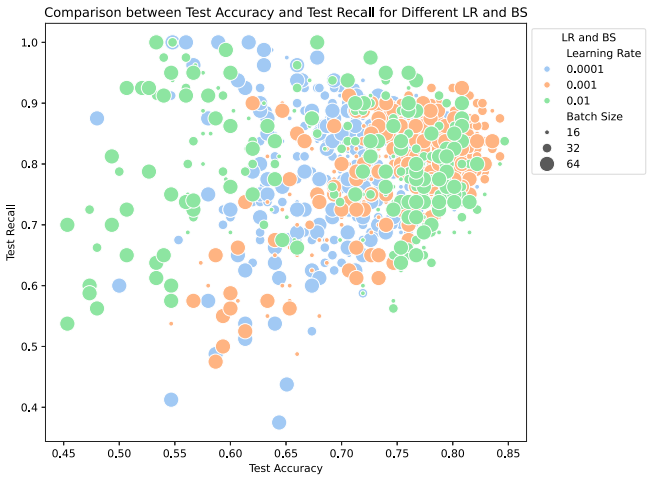

Fig. 5. Test Accuracy vs. Test Recall for different Learning Rate (LR) and Batch Size (BS) configurations. Each point represents a model, where the color encodes the LR and the size reflects the BS. The optimal trade-off between accuracy and recall is observed at LR = 0.001 and BS = 32.

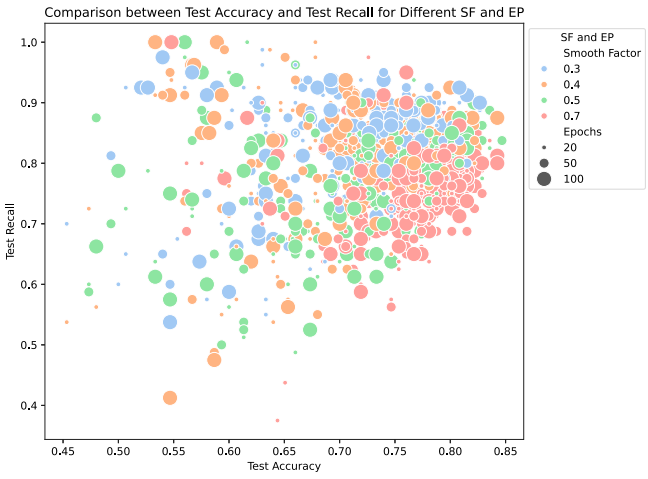

Fig. 6. Test Accuracy vs. Test Recall for different Epochs (EP) and Smooth Factor (SF) values used in class weight initialization. Each point represents a model, where the color encodes the SF and the size reflects the EP. The best accuracy-recall balance is found at EP = 100 and SF = 0.7.

represents a model with a specific configuration, where the color encodes one hyperparameter and the size reflects another. Our goal is to identify the optimal balance between accuracy and recall.

In Fig. 5, we compare Test Accuracy and Test Recall for various combinations of Learning Rate (LR) and Batch Size (BS). Each point represents a model with a specific configuration, where the color indicates the LR and the size reflects the BS. It can be observed that models with an LR of 0.001 (orange color) tend to cluster towards the right side of the plot, indicating higher accuracy values. The Batch Size of 32 appears more frequently in this region, while Batch Sizes of 16 and 64 are more scattered across the plot. This combination reflects a model that performs well in both correctly identifying positive instances (recall) and overall classification accuracy.

In Fig. 6, we compare Test Accuracy and Test Recall for various combinations of Epochs (EP) and Smooth Factor for the class weight initialization (SF). Each point represents a model with a specific configuration, where the color indicates the SF and the size reflects the EP. It can be observed that models with a Smooth Factor of 0.7 (light red color) tend to cluster towards the right side of the plot, indicating higher accuracy values. Similarly, models with an EP of 100 appear more frequently in this region, while other values of EP and SF are more scattered across the plot.

In Fig. 7, we compare Test Accuracy and Test Recall for various combinations of Gamma (γ) and Alpha (α) for the class weight initialization. Each point represents a model with a specific configuration, where the color indicates the γ and the size reflects the α. It can be observed that models with a γ of 1.0 (orange color) tend to cluster towards the right side of the plot, indicating higher accuracy values. The α of 0.3 is not the farthest to the right, but it appears most frequently in this region, while other values of γ and α are more scattered across the plot.

Based on the results of these experiments, the configurations that achieved the best performance were selected for the final model (see Table 6).

The architecture of the proposed model, its training procedure, and both the original and preprocessed datasets are publicly available at https://github.com/LauraMDonaire/QML-Liver.

5. Results and discussion

In this section, we present the comparative analysis of our proposed method against the state-of-the-art models, both classical and quantum.

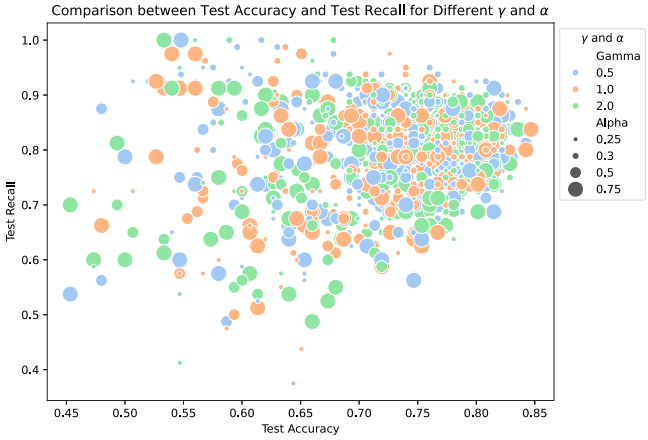

Fig. 7. Test Accuracy vs. Test Recall for different Gamma (γ) and Alpha (α) values in the loss function. Each point represents a model, where the color encodes γ and the size reflects α. The optimal accuracy-recall trade-off is observed at $\gamma = 1.0$ and $\alpha = 0.3$.

We will evaluate the performance of our approach across various metrics and compare it with existing models to highlight its strengths and limitations

In Table 1, it can be seen that the Stacking model from Alyasin and Ata (2024) is the most competitive classical model, as it combines RF, DT, XGB, and ExtraTrees classifiers, using the latter as a meta-classifier. While our approach does not outperform this model in most evaluated metrics, it achieves a higher recall. Additionally, the Stacking model incurs a significantly higher computational cost, as it integrates multiple high-cost classifiers, making it considerably more resource-intensive than our method.

Regarding the quantum approaches (see Table 2), the two best methods were those from Safriandono et al. (2024): the XGB_QFE Tomek model, which employs QFE and the Tomek-Link technique to remove hard-to-classify instances and improve dataset balance, and the LR_QFE model. Our goal is to analyze which parameters our proposal improves upon compared to the state-of-the-art. The results of this analysis are shown in Table 7.

To evaluate the performance of our proposed model, Fig. 8 presents the confusion matrix. The results indicate that our model correctly classifies 48 instances of the negative class and 75 of the positive

Table 6
Summary of the proposed final model's parameters.

Variable	Value
Optimizer	Adam
Learning Rate	0.001
Loss Function	Binary Focal Crossentropy (gamma = 1.0, alpha = 0.3)
Activation Function - Layer 1 and 2	ReLU
Dropout Rate	0.3
Batch Size	32
Epochs	100
Class Weight Initialization	Smooth factor = 0.7
Input Layer Neurons	Number of features (input dataset size)
First Dense Layer Neurons	256
Second Dense Layer Neurons	128
Third Dense Layer Neurons	2
Quantum Layer	2 qubits
Output Layer Neurons	1
Early Stopping	Monitors 'val_loss' with patience of 10 epochs
Cross-Validation	Stratified K-Fold (5 folds)

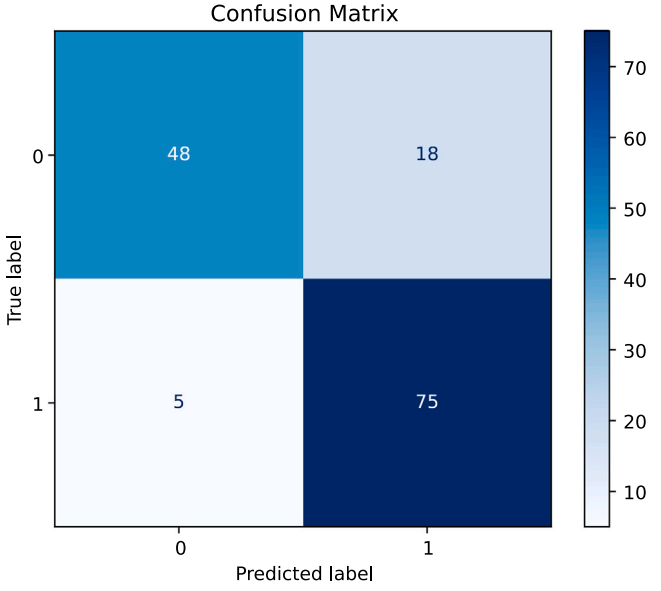

Fig. 8. Confusion matrix of the proposed model, illustrating the number of correctly and incorrectly classified instances for both classes being '0' for healthy patients and '1' for patients with liver diseases.

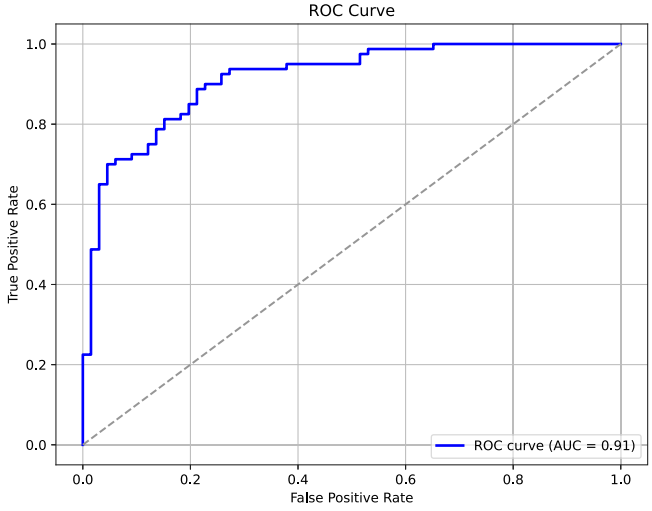

Fig. 9. ROC curve of the proposed model, showing the trade-off between true positive rate and false positive rate. The AUC indicates the model's discriminative ability.

class, with 18 and 5 misclassifications, respectively. From a medical diagnostics perspective, this outcome is favorable, as minimizing false negatives (5) is critical to avoid missing sick patients. Although false positives (18) may lead to additional testing for healthy individuals, they are generally considered less harmful than overlooking a true case. Therefore, the distribution of errors suggests that our model prioritizes sensitivity in detecting positive cases, which is a desirable quality in medical diagnostics.

The discriminative performance of the proposed model was further evaluated using the Receiver Operating Characteristic (ROC) curve. Fig. 9 shows the ROC curve, with the area under the curve (AUC) indicating the model's overall ability to distinguish between healthy and diseased patients. In our case, the model achieved an AUC of 91%, demonstrating strong discriminative performance. Higher AUC values reflect better separation between the two classes.

In terms of accuracy, our model QML-Liver achieves 84%, surpassing the XGB_QFE Tomek model (81%) and the LR_QFE model (74%), and outperforming previous quantum proposals such as VQC (74%) and QKNN (69%). Regarding precision, our model attains 81%, compared to 77% for the XGB_QFE Tomek and 75% for LR_QFE. In recall, QML-Liver reaches 94%, slightly below the LR_QFE model (99%) but higher than the rest of the alternatives. Additionally, our approach achieves an F1-Score of 87%, surpassing all models.

We also explored two alternative versions of QML-Liver to evaluate the impact of qubit count and data balancing. The 10-qubit version did not improve performance, achieving 80% accuracy and reducing the F1-Score to 73%, which indicates that simply increasing quantum resources does not necessarily translate into better results. The 'QML-Liver + SMOTE' version, improved precision to 85% and specificity to 83%, thereby reducing false positives. However, recall decreased substantially to 76%, limiting its ability to identify the positive class, and overall accuracy dropped to 79%. During cross-validation, this configuration also showed higher variance, suggesting a tendency to overfit to the synthetic data (Specificity: Std = 0.2254). This highlights the inherent trade-offs of balancing: while SMOTE can reduce false positives, it may introduce bias and reduce the model's ability to generalize. From a clinical perspective in liver disease detection, high recall is generally prioritized over high precision, because minimizing false negatives-patients who are sick but not detected-is critical for patient safety. Therefore, although SMOTE improves precision, the significant drop in recall indicates that this approach could be counterproductive in a real-world clinical setting, as it could increase the risk of missing true positive cases.

A key advantage of our model is its efficiency: it requires only two qubits and has a delay of 20, whereas the competing quantum

Table 7
Comparison of our model with state-of-the-art quantum approaches in terms of the number of qubits and delay, dataset balancing, accuracy (Acc), precision (Pr), recall (Rec), F1-Score (F1), specificity (Spe), and ROC AUC after training on the entire training set (RAC).

Author	Model	Nº Qubits	Delay	Balancing	Acc	Pr	Rec	F1	Spec	RAC
Raubitzek and Mallinger (2023)	VQC	10	11*	No	74	–	–	–	–	–
Raubitzek et al. (2024)	Catboost	N/A	N/A	No	73	65	63	64	–	–
Safriandono et al. (2024)	XGB_QFE Tomek	10	40	Yes	81	77	90	83	72	–
	LR_QFE	10	40	No	74	75	**99**	85	**99**	–
Bhaskaran and Prasanna (2024)	QKNN(K = 5)	10	23	No	69	–	–	–	–	–
Our proposals	QML-Liver	**2**	**20**	No	**84**	81	94	**87**	73	**91**
	QML-Liver	10	48	No	80	79	86	73	73	90
	QML-Liver + SMOTE	**2**	**20**	Yes	79	**85**	76	80	83	89

Table 8
For reproducibility reasons, we present the deleted records from the original dataset, which were removed from the test set due to consistent misclassification across 10 executions. These cases exhibited ambiguous patterns that could negatively impact model evaluation.

Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	A/G	Selector
18	0	1.8	0.7	178	35	36	6.8	3.6	1.10	1
17	0	0.9	0.2	224	36	45	6.9	4.2	1.55	1
24	0	1.0	0.2	189	52	31	8.0	4.8	1.50	1
60	0	2.2	1.0	271	45	52	6.1	2.9	0.90	0
60	0	0.8	0.2	215	24	17	6.3	3.0	0.90	0
38	1	2.6	1.2	410	59	57	5.6	3.0	0.80	0
35	0	2.0	1.1	226	33	135	6.0	2.7	0.80	0
11	0	0.7	0.1	592	26	29	7.1	4.2	1.40	0
65	0	0.7	0.2	265	30	28	5.2	1.8	0.52	0
36	0	5.3	2.3	145	32	92	5.1	2.6	1.00	0
48	0	0.7	0.2	208	15	30	4.6	2.1	0.80	0
65	0	1.4	0.6	260	28	24	5.2	2.2	0.70	0
62	0	0.6	0.1	160	42	110	4.9	2.6	1.10	0
65	0	0.8	0.2	201	18	22	5.4	2.9	1.10	0
17	1	0.7	0.2	145	18	36	7.2	3.9	1.18	0
62	0	0.7	0.2	162	12	17	8.2	3.2	0.60	0
65	0	1.9	0.8	170	36	43	3.8	1.4	0.58	0
23	1	2.3	0.8	509	28	44	6.9	2.9	0.70	0

models employ ten qubits with higher delays (up to 40). This makes our approach the only viable quantum solution under current NISQ hardware constraints.

As shown in Table 7, our proposed model, 'QML-Liver', demonstrates competitive and consistent performance against state-of-the-art quantum baselines.

6. Conclusion and future work

Liver disease represents a serious global health issue, underscoring the need for accurate and efficient diagnostic solutions-an area where QML shows great promise in improving disease detection. In this study, we developed a novel methodology for constructing a hybrid classifier for liver disease detection, integrating both classical and QML techniques. We began by building on these insights, we introduced 'QML-Liver', a sequential hybrid quantum model that seamlessly combined classical and quantum layers to enhance classification performance.

To ensure data quality, we implemented robust preprocessing techniques, including dummy encoding, data splitting and data standardization. After optimizing the model architecture, 'QML-Liver' demonstrated competitive performance, achieving 84% accuracy, 81% precision, 94% recall, 87% F1-Score, 73% Specificity and ROC AUC of 91%. A comparative analysis against state-of-the-art quantum models revealed that our approach not only matched or surpassed existing methods in key performance metrics but also significantly improved computational efficiency by reducing the number of qubits to just two making it a more practical solution within the constraints of NISQ devices. Besides the efficiency in quantum resource usage, the advantage of the hybrid approach lies in the ability of the quantum layer to enrich data representation. By encoding classical features into a higher-dimensional Hilbert space and leveraging quantum entanglement, the model can capture non-linear correlations that are difficult to learn with classical dense layers alone. This additional expressiveness helps the model find a better trade-off between precision and recall, especially when dealing with imbalanced classes. In practice, purely quantum models still face noise and qubit limitations, while purely classical networks often need to grow much deeper and more complex to reach similar performance. The hybrid design strikes a balance by combining the stability and generalization of classical layers with the richer representational capacity offered by quantum processing. These findings highlighted the potential of 'QML-Liver' for medical diagnostics, particularly in resource-constrained quantum environments. Moreover, they underscore its promise as a scalable and clinically relevant tool for decision support in liver disease detection.

Finally, to contextualize our approach, we conducted a comprehensive review of state-of-the-art classical and quantum models applied to the ILPD, evaluating their strengths and limitations.

As future work, we planned to explore alternative quantum encoding strategies and embedding techniques. Additionally, we intended to incorporate advanced quantum algorithms to further enhance the model's efficiency and real-world applicability.

CRediT authorship contribution statement

Laura María Donaire: Writing – review & editing, Writing – original draft, Project administration, Investigation, Funding acquisition. **Gloria Ortega:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Project administration. **Francisco Orts:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration. **Ester Martín Garzón:** Writing – review & editing, Validation, Funding acquisition. **Ernestas Filatovas:** Writing – review & editing, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been supported by the projects: PID2021-123278OB-I00, PDC2022-133370-I00 funded by MCIN/AEI/10.13039/5011000 11033, "ERDF A way of making Europe", by the 2024-2025 Research and Transfer Plan of the University of Almería, and S-A-UEI-23-11 funded by the Research Council of Lithuania under the Program "University Excellence Initiatives" of the Ministry of Education, Science and Sports of the Republic of Lithuania (Measure No. 12-001-01-01-01 "Improving the Research and Study Environment").

Appendix. Removed rows from the test dataset

See Table 8.

Data availability

The supplementary data associated with this article are available in the repository at: https://github.com/LauraMDonaire/QML-Liver.

References

Acar, E., Yılmaz, I., 2021. COVID-19 detection on IBM quantum computer with classical-quantum transfer learning. Turk. J. Electr. Eng. Comput. Sci. 29, 46–61.
Aishwarya, S., Abeer, V., Sathish, B.B., et al., 2020. Quantum computational techniques for prediction of cognitive state of human mind from EEG signals. J. Quantum Comput. 2 (4), 157–170.
Alami, M. El, Innan, N., Shafique, M., Bennai, M., 2025. Comparative performance analysis of quantum machine learning architectures for credit card fraud detection. http://dx.doi.org/10.48550/arXiv.2412.19441.
Alyasin, E.I., Ata, O., 2024. Enhancing the diagnosis of liver disease: Combining machine learning with the Indian liver patient dataset. In: Proceedings of Fifth Doctoral Symposium on Computational Intelligence DoSCI 2024, vol. 3.
Azam, Md., Rahman, A., Iqbal, S., et al., 2020. Prediction of liver diseases by using few machine learning based approaches. Aust. J. Eng. Innov. Technol. 2, 85–90.
Bai, Q., Hu, X., 2024. Superposition-enhanced quantum neural network for multi-class image classification. Chinese J. Phys. 89, 378–389.
Beer, K., Bondarenko, D., T. Farrelly, Terry, Osborne, T.J., Salzmann, R., Scheiermann, D., Wolf, R., 2020. Training deep quantum neural networks. Nat. Commun. 11 (1), 808.
Bhaskaran, P., Prasanna, S., 2024. An accuracy analysis of classical and quantum-enhanced K-nearest neighbor algorithm using canberra distance metric. Knowl. Inf. Syst..
Biamonte, J., Wittek, P., Pancotti, N., et al., 2017. Quantum machine learning. Nature 549 (7671), 195–202.
Cerezo, M., Arrasmith, A., Babbush, R., et al., 2021. Variational quantum algorithms. Nat. Rev. Phys. 3, 625–644.
Chen, S.Y.-C., Yoo, S., 2021. Federated quantum machine learning. Entropy 23 (4), 460.
Combarro, E.F., Gonzalez-Castillo, S., 2023. A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-on Approach to Modern Quantum Algorithms.
Cong, I., Choi, S., Lukin, M.D., 2019. Quantum convolutional neural networks. Nat. Phys. 15 (12), 1273–1278.
Dasari, K., Dongari, S.P., Chirra, A.R., et al., 2023. Demystifying quantum blockchain for healthcare. In: 2023 International Conference on Computational Science and Computational Intelligence. CSCI, IEEE, pp. 1456–1460.
Devadas, R.M., Sowmya, T., 2025. Quantum machine learning: A comprehensive review of integrating AI with quantum computing for computational advancements. MethodsX 14, 103318.
Dritsas, E., Trigka, M., 2023. Supervised machine learning models for liver disease risk prediction. Computers 12 (1), 19.
Dunjko, V., Briegel, H.J., 2018. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. Rep. Progr. Phys. 81 (7), 074001.
Dutt, V., Chandrasekaran, S., García-Díaz, V., 2020. Quantum neural networks for disease treatment identification. Eur. J. Mol. Clin. Med. 7 (11), 57–67.
Elsayed, S., Ismail, M.M., Abdel-Gawad, A.F., et al., 2024. Adaptive analysis of machine learning algorithms on medical datasets for disease prediction. Inf. Sci. Appl. 3. Feature-engine, Missing data imputation.
Gajendran, G., Varadharajan, R., 2020. Classification of Indian liver patients data set using MAMFFN. In: AIP Conf. Proc., vol. 2277, (1), 120001.
Ganguly, S., 2021. Quantum Machine Learning: An Applied Approach: The Theory and Application of Quantum Machine Learning in Science and Industry. Apress Media LLC, p. 551.
GeeksforGeeks, 2025. RMSProp optimizer in deep learning. (Accessed 22 March 2025).
Geetha, C., Arunachalam, A., 2021. Evaluation based approaches for liver disease prediction using machine learning algorithms. In: 2021 International Conference on Computer Communication and Informatics. ICCCI, pp. 1–4.
Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: Teh, Yee Whye, Titterington, Mike (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. In: Proceedings of Machine Learning Research, vol. 9, PMLR, Chia Laguna Resort, Sardinia, Italy, pp. 249–256.
Griol-Barres, I., Milla, S., Cebrián, A., et al., 2021. Variational quantum circuits for machine learning. An application for the detection of weak signals. Appl. Sci. 11.
Gupta, A., Anand, A., Hasija, Y., 2021. Recall-based machine learning approach for early detection of cervical cancer. In: 2021 6th International Conference for Convergence in Technology. I2CT, pp. 1–5.
Gupta, K., Jiwani, N., Afreen, N., et al., 2022. Liver disease prediction using machine learning classification techniques. In: Proceedings of the International Conference on Communication Systems and Network Technologies. CSNT, IEEE, pp. 221–226.
Havlíček, V., Córcoles, A.D., Temme, K., et al., 2019. Supervised learning with quantum-enhanced feature spaces. Nature 567, 209–212.
He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification.
Houssein, E., Abohashima, Z., Elhoseny, M., Mohamed, W.M., 2022. Hybrid quantum-classical convolutional neural network model for COVID-19 prediction using chest X-ray images. J. Comput. Des. Eng. 9 (2), 343–363.
Jiang, X., Chuhan, X., 2022. Deep learning and machine learning with grid search to predict later occurrence of breast cancer metastasis using clinical data. J. Clin. Med. 11 (19), 5772.
Joel, L.O., Doorsamy, W., Paul, B.S., 2024. On the performance of imputation techniques for missing values on healthcare datasets. arXiv preprint arXiv:2403.14687.
Kawase, Y., 2024. Distributed quantum neural networks via partitioned features encoding. Quantum Mach. Intell. 6 (15).
Keçeci, M., 2025. Accuracy, noise, and scalability in quantum computation: Strategies for the NISQ era and beyond. Open Sci. Artic. (OSAs).
Kingma, D.P., Ba, J., 2017. Adam: A method for stochastic optimization.
Kumar, S., Rani, P., 2024. Liver disease prediction using Bayesian optimized classification algorithms. In: 2024 2nd World Conference on Communication & Computing. WCONF, pp. 1–6.
Kumar, P., Thakur, R.S., 2021. Liver disorder detection using variable - neighbor weighted fuzzy K nearest neighbor approach. Multimedia Tools Appl. 80, 16515–16535.
Kwon, S., Huh, J., Kwon, S.J., h. Choi, S., Kwon, O., 2025. Leveraging quantum machine learning to address class imbalance: A novel approach for enhanced predictive accuracy. Symmetry 17 (2), 186.
Lee, H., Banerjee, A., 2023. Quantum embedding framework of industrial data for quantum deep learning. In: 2023 Winter Simulation Conference. WSC, IEEE, pp. 2956–2965.
Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. In: 2017 IEEE International Conference on Computer Vision. ICCV, pp. 2999–3007.
Lloyd, S., Schuld, M., Ijaz, A., Izaac, J., Killoran, N., 2020. Quantum embeddings for machine learning.
Maheshwari, D., Ullah, U., Marulanda, P. A. Osorio, et al., 2023. Quantum machine learning applied to electronic healthcare records for ischemic heart disease classification. Human-Centric Comput. Inf. Sci..
Mao, A., Mohri, M., Zhong, Y., 2023. Cross-entropy loss functions: Theoretical analysis and applications. In: Proceedings of the 40th International Conference on Machine Learning. In: Proceedings of Machine Learning Research, vol. 202, PMLR, pp. 23803–23828.
Mermin, N., 2007. Quantum Computer Science: An Introduction. Cambridge University Press.
Moradi, S., Brandner, C., Spielvogel, C., et al., 2022. Clinical data classification with noisy intermediate scale quantum computers. Sci. Rep. 12 (1), 1851.
Mutlu, E.N., Devim, A., Hameed, A.A., et al., 2022. Deep learning for liver disease prediction. In: Pattern Recognition and Artificial Intelligence. MedPRAI 2021. Communications in Computer and Information Science, vol. 1543, pp. 1–13.
Nahar, N., Ara, F., 2018. Liver disease prediction by using different decision tree techniques. Int. J. Data Min. Knowl. Manag. Process. 8, 01–09.
Nielsen, M.A., Chuang, I.L., 2010. Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press.
Nigatu, S., Alla, P.R., Ravikumar, R.N., et al., 2023. A comparative study on liver disease prediction using supervised learning algorithms with hyperparameter tuning. In: Proceedings of the 2023 International Conference on Advancement in Computation & Computer Technologies. InCACCT, IEEE, pp. 353–357.

Perelshtein, M., et al., 2022. Practical application-specific advantage through hybrid quantum computing. arXiv preprint arXiv:2205.04858.
Preskill, J., 2018. Quantum computing in the NISQ era and beyond. Quantum 2, 79.
PyTorch, 2025. RMSprop. (Accessed 03 September 2025).
Qi, H., Xiao, S., Liu, Z., Gong, C., et al., 2024. A high-efficiency variational quantum classifier for high-dimensional data. J. Supercomput. 81 (1), 154.
Raj, H., G., N., Kodipalli, A., Rao, T., 2024. Prediction of chronic liver disease using machine learning algorithms and interpretation with SHAP kernels. In: 2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems. ICITEICS, pp. 1–6.
Ramana, Bendi, Surendra, M, Babu, Prasad, Bala Venkateswarlu, Nagasuri, 2012. A critical comparative study of liver patients from USA and INDIA: An exploratory analysis. Int. J. Comput. Sci. 9.
Ranga, D., Rana, A., Prajapat, S., et al., 2024. Quantum machine learning: Exploring the role of data encoding techniques, challenges, and future directions. Mathematics 12 (21).
Ranjan, G.S.K., Verma, K., Radhika, S., 2019. K-nearest neighbors and grid search CV based real time fault monitoring system for industries. In: 2019 IEEE 5th International Conference for Convergence in Technology. I2CT, pp. 1–5.
Rasool, R. Ur, Ahmad, H.F., Rafique, W., Qayyum, A., Qadir, J., Anwar, Z., 2023. Quantum computing for healthcare: A review. Futur. Internet 15 (3).
Raubitzek, S., Mallinger, K., 2023. On the applicability of quantum machine learning. Entropy 25 (7).
Raubitzek, S., Schrittwieser, S., Schatten, A., et al., 2024. Quantum inspired kernel matrices: Exploring symmetry in machine learning. Phys. Lett. A 525.
Safriandono, A.N., Setiadi, D.R.I.M., Dahlan, A., et al., 2024. Analyzing quantum feature engineering and balancing strategies effect on liver disease classification. J. Futur. Artif. Intell. Technol. 1 (1).
Schuld, M., Killoran, N., 2019. Quantum machine learning in feature Hilbert spaces. Phys. Rev. Lett. 122.
Schuld, M., Sinayskiy, I., Petruccione, F., 2014. An introduction to quantum machine learning. Contemp. Phys. 56 (2), 172–185.
Shen, H., 2018. Towards a mathematical understanding of the difficulty in learning with feedforward neural networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 811–820.
Sierra-Sosa, D., Arcila-Moreno, J.D., Garcia-Zapirain, B., et al., 2021. Diabetes type 2: Poincaré data preprocessing for quantum machine learning. Comput. Mater. Contin. 67 (2).
Sinno, Salvatore, Bertl, Markus, Sahoo, Arati, Bhalgamiya, Bhavika, Groß, Thomas, Chancellor, Nicholas, 2025. Implementing large quantum Boltzmann machines as generative AI models for dataset balancing.
Skolik, A., Jerbi, S., Dunjko, V., 2022. Quantum agents in the gym: A variational quantum algorithm for deep Q-learning. Quantum 6, 720.
Sokoliuk, A., Kondratenko, G., Sidenko, I., et al., 2020. Machine learning algorithms for binary classification of liver disease. In: 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology. PIC S&T, pp. 417–421.
Sontakke, S., Lohokare, J., Dani, R., 2017. Diagnosis of liver diseases using machine learning. In: 2017 International Conference on Emerging Trends & Innovation in ICT. ICEI, pp. 129–133.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.
Tscharke, K., Issel, S., Debus, P., 2024. QUACK: Quantum aligned centroid kernel. In: 2024 IEEE International Conference on Quantum Computing and Engineering. QCE, IEEE, pp. 1425–1435.
Tychola, K., Kalampokas, T., Papakostas, G.A., 2023. Quantum machine learning-An overview. Electronics 12 (11).
Ullah, U., Garcia-Zapirain, B., 2024. Quantum machine learning revolution in healthcare: A systematic review of emerging perspectives and applications. IEEE Access 12, 11423–11450.
Wang, X., Du, Y., Luo, Y., Tao, D., 2021. Towards understanding the power of quantum kernels in the NISQ era. Quantum 5, 531.
Wei, L., Liu, H., Xu, J., et al., 2023. Quantum machine learning in medical image analysis: A survey. Neurocomputing 525, 42–53.
Williams, H., 2025. Handling missing values in clinical data | data science for health informatists. (Accessed 04 September 2025).
Xanadu Quantum Technologies Inc., 2025. qml.BasicEntanglerLayers — PennyLane 0.42.3 documentation. https://docs.pennylane.ai/en/stable/code/api/pennylane.BasicEntanglerLayers.html. (Accessed 06 September 2025).
Ying, M., 2010. Quantum computation, quantum theory and AI. Artificial Intelligence 174 (2), 162–176, Special Review Issue.